Advances in Intelligent Systems and Computing 560

Dimitris Alimisis
Michele Moro
Emanuele Menegatti *Editors*

Educational Robotics in the Makers Era

Springer

Advances in Intelligent Systems and Computing

Volume 560

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series “Advances in Intelligent Systems and Computing” contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within “Advances in Intelligent Systems and Computing” are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Dimitris Alimisis · Michele Moro
Emanuele Menegatti
Editors

Educational Robotics in the Makers Era

Springer

Editors
Dimitris Alimisis
European Lab for Educational Technology
Sparta
Greece

Michele Moro
Department of Information Engineering
Università degli Studi di Padova
Padua
Italy

Emanuele Menegatti
Department of Information Engineering
Università degli Studi di Padova
Padua
Italy

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-55552-2 ISBN 978-3-319-55553-9 (eBook)
DOI 10.1007/978-3-319-55553-9

Library of Congress Control Number: 2017934066

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The volume “Educational Robotics in the Makers’ Era” contains papers presented at the International Conference “Educational Robotics 2016 - EDUROBOTICS 2016” held on November 25, 2016 in Athens. The conference was organized by EDUMOTIVA (Greece, www.edumotiva.eu), the Department of Information Engineering, University of Padua (Italy, http://www.dei.unipd.it/en) and the INNOVATHENS – Hub of Innovation & Entrepreneurship Technopolis City of Athens (Greece, http://www.innovathens.gr). The conference was organized in the framework of the European Robotics Week 2016.

The conference history dates back to 2008 when the first International Workshop entitled “Teaching Robotics and Teaching With Robotics - TRTWR” was organized in Venice in the context of the TERECoP project (www.terecop.eu). The success of that workshop resulted in a series of TRTWR workshops held in Darmstadt (2010), in Riva Del Garda (2012) and in Padua (2014). Publications from those workshops have included open online proceedings and two special issues in the journals: Themes in Science & Technology Education, 2013 and Robotics & Autonomous Systems Journal, 2016, Elsevier.

The interest in the continuation of the international workshops and the previous successful editions witness the continuously growing interest in educational robotics worldwide and have enabled the building of a community of researchers and educators with interest in this field across EU and beyond. In 2016, encouraged by the success of the TRTWR series of workshops we decided to upgrade the workshops series to an International Conference with the (simpler) title EDUROBOTICS 2016 standing for Educational Robotics International Conference 2016.

The former TRTWR workshops have built on pedagogies inspired from constructivism and constructionism (Piaget, Papert). The EDUROBOTICS 2016 conference continued on the same pedagogical path and explored how educational robotics can support the development of STEAM (Science, Technology, Engineering, Arts and Math) education and the 21st century skills: creativity, critical thinking, team working, and problem solving.

The central theme this year was “Educational Robotics in the Makers’ Era”. The Maker Movement has emerged recently in education with the great promise to democratize access to opportunities for learning by making and the 21st century digital making technologies. Though educational robotics preceded the maker movement years ago, they share common roots in Papert’s constructionism and similar vision for an education that will enable learners to make their own (robotic or not) artifacts using 21st century technologies. Hence, it is worthy for educational robotics community to explore further connections with digital fabrication, DIY electronics and other making technologies and position itself in the broader maker movement.

The Program Committee received 28 submissions (24 full papers, 4 short papers) coming from eight different EU countries, USA, Canada, Russia, Israel, and Pakistan. Each submission was reviewed by at least 2, and on average 2.9, Program Committee members. Finally, the Committee decided to accept 14 full papers (acceptance rate 58%) and 8 papers for poster presentation. The program also included three invited talks by Meurig Beynon (University of Warwick, UK), Ilkka Jormanainen (University of Eastern Finland), and Alfredo Pina (Public University of Navarra, Spain).

The content of the book is organized in four sections.

1st section: Theory and practice in educational robotics (including the invited papers).

2nd section: Educational robotics projects in school and higher education.

3rd section: Methodologies in educational robotics.

4th section: Educational robotics and programming.

5th section: Short papers reporting good practices or work in progress presented in the conference as posters.

We thank the conference participants, academics, researchers, and educators from all the levels of education (primary, secondary and tertiary), and the young researchers and PhD and postgraduate students for their active participation and great contribution to the success of the conference and for authoring this book. Special thanks go to our Program Committee members who have reviewed the papers and provided important help to authors to improve their manuscripts.

Finally, this book is dedicated to the memory of Seymour Papert who passed away in 2016; the man who inspired the educational robotics community for years and whose theoretical work is behind educational robotics and the maker movement.

The co-editors

January 2017

Dimitris Alimisis
Michele Moro
Emanuele Menegatti

Organization

Program Committee

Dimitris Alimisis	EDUMOTIVA (European Lab for Educational Technology)
Richard Balogh	Slovak University of Technology in Bratislava
Ansgar Bredenfeld	Dr. Bredenfeld UG
Dave Catlin	Valiant Technology
Stavros Demetriadis	Aristotle University of Thessaloniki
Amy Eguchi	Bloomfield College
Nikleia Eteokleous	Frederick University Cyprus
Alexander Hofmann	University of Applied Sciences FH Technikum Vienna
Ilkka Jormanainen	University of Eastern Finland
Ken Kahn	Oxford University
Chronis Kynigos	University of Athens
Lara Lammer	Technical University Vienna
Wilfried Lepuschitz	Practical Robotics Institute Austria
Sophia Manika	INNOVATHENS – Technopolis City of Athens
Emanuele Menegatti	University of Padua
Emanuele Micheli	School of Robotics, Genoa
Tassos Mikropoulos	University of Ioannina
Stefano Monfalcon	Town Museum of Rovereto
Michele Moro	University of Padua
David Obdrzalek	Charles University in Prague
Pavel Petrovic	Comenius University in Bratislava
Alfredo Pina	Public University of Navarra
Sandra Schoen	Salzburg Research
Igor Verner	Technion – Israel Institute of Technology
Baran Çürüklü	Mälardalen University

Additional Reviewers

Mirgita Frasheri
Sabrina Rubenzer
Lan-Ahn Trinh

Contents

Theory and Practice in Educational Robotics (Invited Papers)

Mindstorms Revisited: Making New Construals of Seymour Papert's Legacy

Meurig Beynon(✉)

Computer Science, University of Warwick, Coventry, CV4 7AL, UK
w.m.beynon@warwick.ac.uk

Abstract. Seymour Papert's *Mindstorms* is a seminal text in educational technology. Its subtitle: *Children, Computers and Powerful Ideas* reflects Papert's broad visionary ambition, yet the cover of its second edition is headlined: *ALL ABOUT LOGO - HOW IT WAS INVENTED AND HOW IT WORKS*. Notwithstanding the enormous practical impact of LOGO on educational applications of computers, we should not forget Papert's declaration regarding the computer-inspired revolution in learning he envisioned: "I do not present LOGO environments as my proposal for this … [they are] too primitive … too limited by the technology of the 1970s …". This paper revisits the challenge implicit in this declaration, appraising the extent to which today's technological advances can realise Papert's vision, and proposing an alternative model for his central conception of "an object-to-think-with, that will contribute to the essentially social process of constructing the education of the future".

Keywords: Seymour Papert · Mindstorms · Constructionism · Making construals · LOGO programming · Collaborative learning

1 Introduction

Seymour Papert's *Mindstorms* [13], first published in 1980, made a seminal contribution to the field of educational technology. It stimulated research into the potential role of computer in learning that has embraced programming, microworlds and educational robotics. The spirit of *Mindstorms* is well-represented in the principal themes of the Edurobotics conference: how, by building on Piaget's pedagogical theory of constructivism and Papert's principles of constructionism, contemporary technologies such as digital fabrication and robotics can promote skills in creative thinking, collaboration, and problem solving in Science, Technology, Engineering, Arts and Math (STEAM).

Mindstorms is a visionary book. In the Introduction, Papert envisaged how 'ubiquitous computing' and 'an increasing disillusion with traditional education' can 'come together in a way that would be good for children, for parents and for learning … through the construction of educationally powerful computational environments that will provide alternatives to traditional classrooms and traditional instruction'. It might be argued that these prophesies are now being realised in a culture in which 'apps' and 'MOOCs' abound. In a recent blog entitled "How to Teach Computational Thinking" [22], Stephen Wolfram makes the case for the Wolfram Language as a transformative

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_1

tool that makes computational thinking readily accessible and in the process opens up new teaching strategies across the STEAM disciplines. In a similar spirit, the *Maker Movement Manifesto* [6], promotes 'making' as an activity accessible to all, observing that: "The tools of making have never been cheaper, easier to use, or more powerful." … "It may take some practice to get good at some kinds of making, but technology has begun to make creating easy enough that everyone can make". The implicit message is that modern technologies and practices are bringing Papert's vision into reality.

This paper reflects scepticism about such claims and questions whether – despite the practical progress that has been made towards exploiting computers in education in recent years – Papert's agenda in *Mindstorms* has yet been properly addressed. In truth, Papert's vision in *Mindstorms* went beyond realising new educational practices based on new technologies. His primary goal was a deeper understanding of how building with computers might contribute to learning that was based on key ideas set out by Piaget [15]. When Papert asserts [13: p. 11] that he sees the Turtle as "a valuable educational object [whose] principal role here is to serve as a model for other objects, yet to be invented" and that his "interest is in the process of invention of 'objects-to-think-with', objects in which there is an intersection of cultural presence, embedded knowledge, and the possibility for personal identification", he is not merely making a disclaimer regarding his advocacy of LOGO. He is posing a challenging fundamental question about the relationship between computer-based activities and learning. 'Inventing an object-to-think-with' is not the same thing as 'writing a computer program' or 'fabricating a digital object'. Papert is not primarily concerned with making it easy to frame a computation or create a digital artifact; his interest is in the way in which learning is associated with these and other processes of construction. He proposes to exploit the computer in devising objects-to-think-with as a means to address this meta-agenda. When discussing "LOGO's Roots" in Chapter 7 of *Mindstorms*, Papert's focus is not on the qualities of LOGO as a programming language, but on how far it provokes learners to reflect on the basis for their knowledge and so enables the computer to liberate 'epistemological aspects of Piaget's thought' [13: p. 156].

The three main sections of this paper examine Papert's agenda in more detail with reference to motifs that run through *Mindstorms*. The first reviews Papert's proposals for educational use of computers from a 'computational thinking' [20] perspective. The second discusses Papert's pedagogical perspective in conjunction with reflections by the psychologist Charles Crook on the role of computers in collaborative learning and identifies perceptions of the challenges involved that they hold in common. The third relates *Mindstorms* to an approach, developed by a group of researchers working under the guidance of the author over many years, that aspires to a broader vision for computing than computational thinking affords. Its key concept, that of 'making construals', is well-matched to the core challenges identified by Papert and Crook.

2 *Mindstorms* from a Computational Thinking Perspective

One of the most startling sentences in *Mindstorms* [13: p. 12] reads: "Readers who have never seen an interactive computer display might find it hard to imagine where this can lead." It reminds us what foresight Papert showed in relation to the technologies of the 1970s. It also highlights a fundamental shift in perspective on computing that is easily overlooked in hindsight.

The computational foundation for computing was established long before it was appropriate to view the computer and its associated technologies as a coherent source of experience. Computation in its strict mathematical sense is an abstract concept whose relationship to experience is indirect. Most of the algorithmic activity that goes on inside the computers around us cannot be directly experienced – only a minute part of their state-changing activity is manifest at the surface through interface devices. What is so radical and far-sighted in Papert's treatment of the computer in *Mindstorms* is that it is focused on the direct experience that computing technology enables. This emphasis on the 'computer' as a concrete specific physical artifact rather than on 'computation' is flagged explicitly in its subtitle. It is a perspective that sets our experience of the computer in the same context as our everyday experience of any other object. In this respect, it differs from addressing the computer as a technical device, as an engineer might, as Papert's deprecation of BASIC as a programming language for "the computerists" [13: p. 35] reflects.

For Papert, the semantic frame of interest is much broader than a conventional computational model can support. His stance on knowledge encompasses the process by which consensus is reached from personal understanding. By way of illustration, a paper on epistemological pluralism, co-authored by Sherry Turkle [17], describes how learners aspire to interact with programming languages and robots in ways far outside the scope of what their designers intended: "Lisa, 18, a first-year Harvard student in an introductory programming course … wants to manipulate computer language the way she works with words as she writes a poem." … "[Alex, 9] turns the Lego wheels on their sides to make flat "shoes" for his robot and harnesses one of the motor's most concrete features: the fact that it vibrates" … to make it "travel". Turkle and Papert allude to the idea that "Computers provide a context for the development of concrete thinking" but that "The practice of computing provides support for a pluralism that is denied by its social construction". Equally relevant is the fact that the traditional conceptual framework for computing is ill-suited to thinking of the computer in concrete terms as 'a source of experience'.

In *Mindstorms* and *The Children's Machine* [14], Papert identifies many of the considerations beyond the scope of traditional models of the computer that might enable us to see computers "as providing contexts for the development of concrete thinking". When he contends that "[computers] should serve children as instruments to work and think with" [14: p. 168], he proposes an engagement between a person and a computer resembling that between a scientist and musician and their instrument in which behaviours are crafted responsively moment-by-moment. When referring to "externalising intuitive expectations" [13: p. 145], to "thinking [that] is always vaguely right and vaguely wrong at the same time" [14: p. 167], and tolerance towards "false theories" [13: p. 132], he invokes contexts in which pragmatic human judgement has an essential role. Though such interactions and interpretations play a significant part in many practical applications of computing in the modern world, it is to say the least problematic to accommodate them within the rule-based system of thought that is so prominent in computer science.

Since the publication of *Mindstorms*, there have been many attempts to bridge the gap between the computational foundation of computing and the computer as a source of concrete experience. For obvious reasons, the thrust in software development practices has been towards this objective. LOGO itself has had myriad incarnations [1] that reflect such trends in software development, including the development of object-oriented and multi-agent variants such as Imagine LOGO and NetLogo. Three contemporary developments are illustrative of different approaches that have taken inspiration from *Mindstorms*: the SCRATCH environment [30], the Wolfram Language [22] and Bret Victor's *Learnable Programming* [19].

In their different ways, SCRATCH, the Wolfram Language and Learnable Programming are concerned with making connections between 'program code' and the experience of the programmer/learner. Figure 1 depicts the generic framework within which the most elementary applications of all three may be conceived. Associating 'computation' with 'experience' is an idea that would be out of place in a formal approach to giving semantics to interaction with the computer. The 'dependency' between the program code and the display canvas refers to the pragmatic connection that the learner perceives between 'changes to the code' and associated 'changes to the output'. The presumption is that this connection, which is to be established primarily by modifying the program code and observing the effect, can be crafted in such a way that it can be reliably learnt.

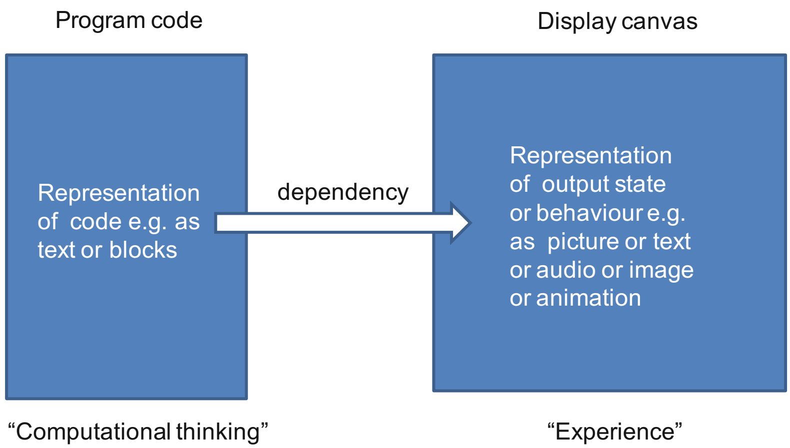


Fig. 1. From computation to experience

In SCRATCH [30], the program code is represented visually by blocks that can be snapped together to form stacks of instructions for execution. The simplest blocks represent primitive instructions of the kind that are familiar from LOGO, but much more sophisticated instructions, such as might control a video for instance, are also available. The behaviour associated with the program code is animated on the display canvas.

In the Wolfram Language [22], the program code takes the form of functional expressions using a range of powerful preprogrammed operators and the display canvas takes the form of a traditional output window in which the values of these expressions (which may have rich spatial and temporal characteristics) are displayed. A functional expression illustrated in [22] simultaneously displays maps of the vicinity of the Eiffel Tower at various different scales, for example.

Victor's proposals for Learnable Programming [19] highlight the subtlety of the considerations that determine how effectively the correspondence in Fig. 1 can be applied in learning. Victor's work gives key insights into the way in which this correspondence functions. He shows how, even in crafting behaviours, the correspondence in Fig. 1 has to be first apprehended as a connection 'in immediate experience'. He also highlights the fact that this correspondence can be invoked in two contexts that would be deemed quite distinct when considering program code in a conventional perspective: in 'designing the program' and in 'simulating its execution'. He also shows how the correspondence in Fig. 1 is intimately tied up with the intended informal meanings of variables. On this basis, Victor deprecates environments for learning programming (such as [10]) in which the learner is encouraged to experiment with unlabelled variables in order to identify their significance.

Excellent illustrations of these principles can be found in Victor's talk on "Inventing on Principle" [18]. In one demonstration, Victor shows how an image of a tree with blossom can be manipulated through live editing of its script. In another, he shows how the motion of a cartoon figure can be specified using a similar technique. In both cases, the counterpart of the 'program code' in Fig. 1 is a JavaScript program whose 'display canvas' realisation takes the form of an associated webpage.

SCRATCH and the Wolfram Language can both be seen as illustrating the power that new technology can potentially bring to Papert's vision. The world-wide dissemination of the SCRATCH culture might be taken as vindication of the idea that learners can use the computer to share and express their ideas as envisaged in *Mindstorms*. Wolfram's striking illustrative examples of interdisciplinary applications of computational thinking demonstrate Papert's power principle [13: p. 54]: "[the computer] must empower the learner to perform personally meaningful projects that could not be done without it". Wolfram's strategy of supplying the learner with a catalogue containing several thousand meaningful functions whose significance is to some degree implicit in their name (cf. [21]) brings to mind Papert's notion of learning to program as like learning a natural language. But Wolfram's critique of LOGO and SCRATCH ("while the turtle (or cat) is quite cute (and an impressive idea for the 1960s), it seems disappointingly narrow from the point of view of today's understanding and experience of computation'' [22]), is two-edged. On the one hand, it reflects a valid contemporary concern about the expressive limitations of SCRATCH. On the other, it suggests a shallow appreciation of the deeper motivations behind the invention of the Turtle as a pedagogical device. And though the Wolfram Language includes pre-constructed functions of unprecedented expressive power, it is unclear to what extent it brings new insight to the processes by which such functions are constructed – a central epistemological focus of attention in *Mindstorms*. On this basis, it seems unlikely that either of these approaches to programming is delivering all that Papert would expect of objects-to-think-with.

Victor's experimental work goes much further in attempting to link programming to experiential roots and in the process discloses challenging hitherto unexplored issues.

His discussions highlight the potential for developing the embryonic framework depicted in Fig. 1 to advance Papert's core agenda significantly. Victor is discriminating in his invocation of empirical processes in programming, insisting that the context for the correspondence in Fig. 1 is critical. Since he considers this correspondence as it relates to correlating program code with state rather than simply with behaviour, he enhances the scope for expressing both declarative and procedural knowledge. This has important implications for conveying designs as well as executable programs. It also helps to refine concepts that Papert introduced casually without adequate explanation. An experienced teacher familiar with the form that debugging typically takes in the initial stages of learning will have reservations about Papert's observation that "Experience with computer programming leads children more effectively than any other activity to 'believe in' debugging" [13: p. 114]. Victor is able to make a distinction between 'debugging' as it often presents itself to the novice in conventional programming context as an uncomfortable encounter with meaningless state and 'debugging' as taking a meaningful step towards new understanding of the domain or application. Where Papert conceives associating LOGO commands with physical actions as 'learning the language of the Turtle', the introduction of declarative elements gives scope for more versatile and expressive connection with natural language. Thinking of programming in the setting of Fig. 1 also helps to appreciate why it may be necessary to look beyond 'program code' as a metaphor when developing objects-to-think-with. Figure 1 can hardly be applied to Lisa and Alex's unconventional perceptions of what the computer might be able to deliver [17], for instance. The notion of "thinking [that] is always right and vaguely wrong at the same time" [14: p. 167] as it might apply to 'writing a poem' implies an ambiguity of interpretation that is outside the scope of program code. Likewise, it is hardly conceivable that a Lego robot with vibrating flat wheels would follow a trajectory that could be reliably associated with a specific piece of program code. The extent to which Papert's vision transcended a 'computational thinking' paradigm is the theme of the next section.

3 *Mindstorms* from a Pedagogical Perspective

For Papert, the fundamental motivation behind using the computer to create 'objects-to-think-with' is a better understanding of learning. He attributes his theory of learning to aspects of Piaget's work that have been underplayed. Papert identifies this as a 'knowledge-based' rather than a deductive theory [13: p. 166] that draws upon Piaget's epistemological studies of how children develop knowledge.

In the Preface to *Mindstorms* and his discussion of LOGO's roots in Piaget and AI [13: Chap. 7], Papert describes two ways in which the computer can help to build on Piaget's work. Referring to his childhood experience of playing with gear mechanisms, and the positive impact this subsequently had on his interest in algebra, Papert sees computer technology as a way of implementing similar models that promote learning. In this context, Papert stresses the 'affective' component, complementary to the cognitive aspects studied by Piaget, which this can bring to a child's assimilation of knowledge. They may develop an enthusiasm for geometric exploration from working with

LOGO, for instance. A second way in which computer technology can be applied is in attempting to construct "machines to perform functions that would be considered intelligent if performed by people" [13: p. 157] in the spirit of AI.

In distinguishing his approach from a deductive theory, Papert refers to his work with Marvin Minsky on *The Society of Mind* [12]: "the subjective experience of knowledge is more similar to the chaos and controversy of competing agents than to the certitude and orderliness of p's implying q's" [13: p. 172]. Though Papert refers to the possibility of making computational models that can to some degree emulate a child's subjective understanding (as in his discussion of how a child may come to understand that the same quantity of liquid may be held in containers of different height [13: p. 167]), such implementation is to be viewed as a way of obliging us to think deeply about the processes that inform human intelligence and thereby gaining insight. In a similar spirit, Papert is not concerned with using the computer simply as a way of illustrating an established theory, observing that Piaget's work calls into question the idea that teaching a child the "correct" theory is superior as a learning strategy [13: p. 133]. Papert's lack of interest in starting from the kind of comprehensive understanding of the agency in a situation that is presumed when framing a computational process is what distinguishes his perspective from that of a professional programmer. It is the pragmatic exploratory activities that may lead the learner to such an understanding that motivated Papert, and, where the computer plays a role in developing an object-to-think-with, the development is – at least in aspiration – more closely aligned with *bricolage* [14: pp. 143–146]. That is to say, its guiding principle is the crafting of experiences by whatever means come to hand.

Making connections between different aspects of their personal experience is central to Piaget's and Papert's notion of how children learn. Most importantly, these are of their essence personal connections: cf. Papert's account of "encouraging [Debbie] to make connections between different elements of what she already knows … they have to be *her* connections" [14: p. 165]. Though *Mindstorms* makes no explicit reference to William James's theory of knowledge [9], here, and elsewhere, there is synergy with the Jamesian notion that all knowing is rooted in personal connections in experience that are themselves experienced. It is to associations of this nature that Papert alludes when he asserts that "learning to control the Turtle is like learning to speak a language" [13: p. 56] and to the way in which the Logo environment entices learners into "imagining themselves inside the system" [14: p. 199]. A related sentiment is evident in Papert's appeal for "syntonicity" in learning activities [13: p. 63]: performing tasks that resonate with other aspects of experience and being emotionally in harmony with one's environment. And, as Papert remarks, the most effective learners are those who exhibit "a tendency to see things in terms of relationships rather than properties" [14: p. 199].

Papert's perspective on learning, as set out in *Mindstorms*, is complemented by the pedagogy of 'constructionism' that he later elaborated in *The Children's Machine* [14: Chap. 7]. Constructionism is the core theme of a series of international conferences to which many of Papert's students and co-researchers are major contributors. The nature and scope of Papert's vision for constructionism, which remains controversial, was the theme of a core discussion at the most recent Constructionism conference in Bangkok, Thailand [29]. Relevant questions concern: How effective a role has Logo played in education? What is the relationship between constructionism and other disciplines: to what extent is it associated with learning mathematics, computer programming or computational thinking? Is it appropriate to interpret constructionism so broadly that it encompasses craft-based activities and music-making where the computer's role is no longer prominent or may even be absent? To what degree is the impact of constructionist ideas on education being inhibited by innate societal pressures (cf. Papert's allusion to "a permanent dilemma faced by anyone who wishes to produce radical innovation in education" [13: p. 140])?

There are many indications in *Mindstorms* about what Papert considered crucial to the full realisation of his vision for learning. In reviewing these, it is helpful to bear in mind that Papert was primarily interested in how the computer could transform the experience of the learner, and was not narrowly committed to any particular technical approach to computing *per se*. In this respect, his perspective is similar to that of Charles Crook, a psychologist whose primary interest is likewise in the impact of computers on the learning experience. In *Computers and the Collaborative Experience of Learning* [2], Crook questions whether constructionism has delivered to its promises (e.g. "The Logo experience reveals that even the most engaging and ingenious computer environment can fail to support pupils' learning" [2: p. 110]) but also stresses several themes that are represented in *Mindstorms*. These include four key interrelated ideas to be amplified in the discussion which follows:

- the critical importance of being able to exploit the computer as a means to create common knowledge,
- the great yet-to-be-realised potential of the computer in this respect,
- the recognition that thinking of 'programming the computer' is not an appropriate way to conceive this role, and
- the vital need to develop a richer conceptual framework in which to address such concerns.

Creating Common Knowledge: The emphasis that Crook gives to intersubjectivity and its role in instruction [2: p. 101] may initially seem to be at odds with Papert's stance. The high profile that Papert gives to personal experience, and to learning as something that is ultimately a matter for the individual, is potentially misleading. When considered alongside his forceful criticism of school practices, it may be construed as suggesting that children learn best independently without the need for instruction and perhaps even without reference to their broader social context. In fact, in his account of constructionism [14: Chap. 7], Papert emphasises the public nature of construction and its role in externalising the learner's thinking and promoting engagement within the wider social context. A concept such as "they have to be *her* connections" should not be interpreted as denying a role for 'instruction' but as echoing Piaget's dictum (as cited in [2: p. 17]): "each time we prematurely teach a child something [s]he would have discovered for himself, the child is kept from inventing it and consequently from understanding it completely" [15].

Potential for the Computer: That Papert is profoundly concerned with the potential role that the computer can play in moving from the subjective to the objective realms of experience is most evident in relation to learning mathematics: "[The computer] is unique in providing us with the means for addressing what Piaget and many others see as the obstacle which is overcome in the passage from child to adult thinking. I believe that it can allow us to shift the boundary separating concrete and formal. Knowledge that was accessible only through formal processes can now be approached concretely. And the real magic comes from the fact that this knowledge includes those elements one needs to become a formal thinker." [13: p. 21] His aspirations for programming as a way of promoting epistemological reflection are broader than this – they relate to the way in which interaction with computers promotes communication and negotiation of meaning more generally. And whilst Crook looks critically at the ways in which computing technology is being deployed in education, he also suggests "that it has a special potential for resourcing the social construction of shared knowledge" [2: p. 189].

Beyond Conventional Programming: In his analysis of interactions with computers, Crook identifies loss of context as especially problematic for the case of activities supported by computers [2: p. 105]. His concern is one facet of the way in which programming practices encourage the learner to abstract elements from the situation to which their programs refer and so invite the pedagogist to 'isolate the pupil-technology component of the interaction' [2: p. 107]. Crook also identifies the rule-based nature of programming activity as potentially diminishing the quality of human interactions in the educational context: "… the meaning of some teaching utterance is rarely to be located in, or made manifest through, its simple surface features – as if such meaning were something to be generated by a rule-bound system of the sort that computer-programmers would seek to construct" [2: p. 119]. It is quite apparent that Papert [14: p. 171] is open to much broader interpretations of programming than a computer scientist might endorse. Programming is typically promoted as a discipline that teaches children the need for absolute precision in thought and expression, since there can be no negotiation of meaning with a computer. Yet, in [14: p. 171], Papert poses the question: "is there such a thing as 'the concept of programming', or is programming something that can be constructed in radically different ways?" and subsequently goes on to propose a notion of programming, quite alien to traditional computer science, that is "inherently biased towards evaluation not by 'is it right?' but 'where can it go from here?'" [14: p. 173].

A Richer Conceptual Framework: For Papert, the computer is the basis for means of communication that can extend, and in certain contexts perhaps supersede, language. Citing Timothy Gallwey's popular book *Inner Tennis* [3] as a source of inspiration, Papert observes: "people need more structured ways to think and talk about the learning of skills. Contemporary language is not sufficiently rich in this domain." [13: p. 98] In analysing interaction with computers in a collaborative learning context, Crook identifies ways in which the presence of computing artifacts can reduce the need for explicit verbal communication. This leads him to observe that studying discourse in order to clarify that shared knowledge is in place is problematic [2: p. 181], highlighting the need "to develop a conceptual vocabulary for talking about cognition as distributed, or shared achievement" [2: p. 138].

4 Making Construals

The need to give a good account of the connection between the formal and the experiential aspects of computing, as highlighted by Papert in *Mindstorms*, is evident in many applications. Educational robotics epitomises the challenge of reconciling a computer science perspective, rooted in computational thinking, with an engineering perspective, rooted in physically-based experimental practices. The same concern arises in general in software development, especially where the context for the development is novel and entails close integration with physical devices and systems. Blending the formal abstract framework in which computer programs are conceived and the pragmatic empirical setting of engineering raises fundamental questions about the role of mathematics and logic in relation to experience (see for example the eminent software consultant Michael Jackson's analysis of what we can expect of program verification [7]). These concerns have been the principal motivation for the Empirical Modelling research programme [26, 27] over the last thirty years.

Papert, following Piaget, is primarily concerned with the process of construction by which a child's subjective empirical understanding of the world comes to be connected with objective understandings that underlie mathematics, science and natural language. The notion of construction also has broader relevance to areas such as software development, where the perspectives of many different kinds of agent, both human and automated, have to be analysed and orchestrated. Empirical Modelling proposes principles and practical techniques that can underpin a conceptual framework for 'constructivist computing' [27:#100]. Central to this is an experientially-guided approach to creating 'objects-to-think-with' that is characterised as 'making construals'. Disseminating the concept and culture of making construals is currently the theme of the EU Erasmus+ CONSTRUIT! project [24].

A full discussion of Empirical Modelling (EM) and making construals is far beyond the scope of this paper. More background on EM in relation to software practices may be accessed from an online repository of EM papers [27:#114,#121]. EM publications on educational technology include doctoral theses by Roe [16] and Harfield [5]. Introductions to the concept of making construals [27:#128] and to the online environment that has been developed for this purpose in the CONSTRUIT! project [27:#134] are also available. This section will outline how making construals relates to Papert's agenda in *Mindstorms*. Illustrative examples relating to this theme will be accessible online via an associated webpage [25].

Informally, 'making a construal' means figuring out how you think something works. The activity plays a fundamental role in everyday life, especially when we encounter unfamiliar situations, as in trying to make sense of a new place, culture or language. The personal and provisional nature of our construals is characteristic, as is the fact that they are associated with a specific moment and context and are typically liable to change. Papert's reference to 'a notion of programming' that is "inherently biased towards evaluation not by 'is it right?' but 'where can it go from here?'" [14: p. 173] hints at a similar frame of reference, where attention is focused on conceiving the next step in the current situation.

Physical media often play a part in making construals. We may refer to a map, or draw up our own informal sketch map to explain where we think we are. In interpreting a construal, or assessing whether it is an appropriate construal, we may interact within the external situation and with the construal we have made of it, whether in our imagination or in the physical world. The physical artifact we create in this context might qualify as what Papert identifies as 'an object-to-think-with'; we can identify such an artifact with a construal by taking account of the thought we invest in interacting with it. The appropriateness of a construal is to be gauged in a pragmatic way – we are concerned with whether it is good enough to serve a purpose (cf. Papert's reference to "thinking [that] is always right and vaguely wrong at the same time" [14: p. 167]) rather than in some sense 'absolutely correct'. We are quite accustomed to accepting that our construals are personal and subjective and that this is particularly significant when communicating with novices and children (cf. Papert's concern for appreciating "the non-obviousness of what we consider to be obvious" [13: p. 41]). This principle is illustrated in a Shopping construal [25] which integrates adult and child perspectives on a shopping scenario.

Making construals is well matched to Papert's need for creating "a dynamic model of how intellectual structures come into being and change" [13: p. 166]. The adopted term 'construal' was previously introduced by David Gooding to describe the way in which the celebrated experimental physicist Michael Faraday first made sense of electromagnetic phenomena (cf. [4, 27:#114]). Such an application of making construals resonates with Papert's motivation for introducing the concept of a microworld in which the laws of physics could be freely postulated [13: p. 138]: "The propositional content of science is certainly very important, but constitutes only a part of a physicist's body of knowledge. It is not the part that developed first historically, it is not a part that can be understood first in the learning process, and it is, of course, not the part I am proposing here as a model for reflection about our own thinking." A construal of the most basic concepts of linear algebra [25] serves as a simple illustration of Papert's constructivist vision: it models the transition from concrete empirical observation of the canvas to the abstract formal structure of a 2-dimensional linear space.

In his discussion of collaborative learning [2: p. 188], Crook identifies the need "to understand more about how structural details of educational software support or constrain the possibility of collaborative work". Crook's motivating concern is how the use of the computer in an educational setting can foster intersubjectivity, an objective that is pivotal in Piaget's constructivist outlook on child development and to Papert's goals for constructionism. From an EM perspective, the aspirations for constructionism are not well-served by educational software that is conceived in conventional programming terms. For instance (cf. [27:#132,#111,#080]), the idea of construction demands a blending of the roles of the designer, teacher and learner of educational software that orthodox programming principles obstruct, but for which making construals is far better suited.

In practice, the influence of Papert's concern for "programming inherently biased towards evaluation not by 'is it right?' but 'where can it go from here?'" [14: p. 173] is seen in the style of programming that has emerged in educational technology (cf. Sect. 2 above). Figure 1 reflects the way in which the direct but informal connection

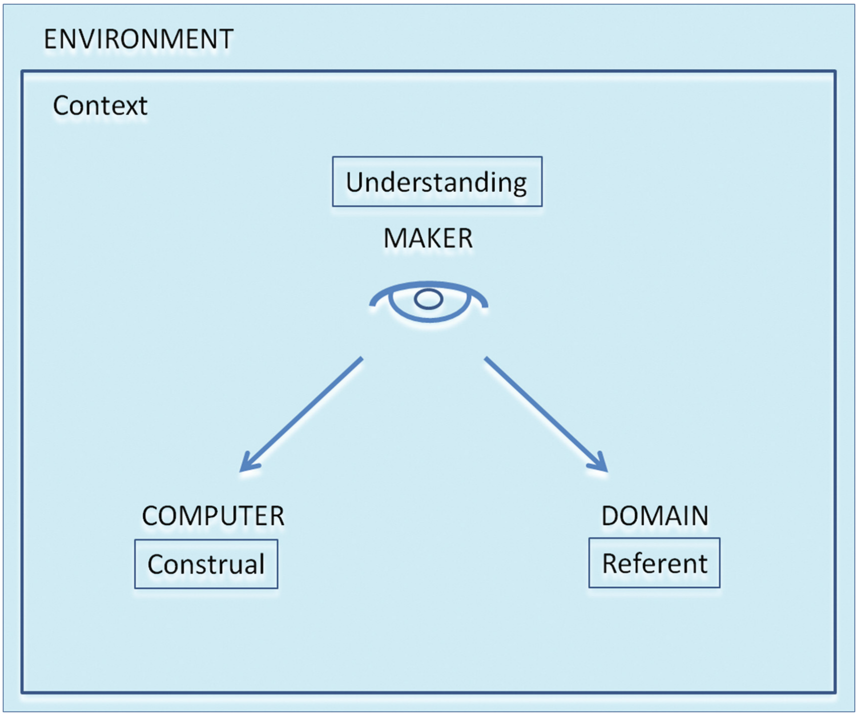


Fig. 2. Making a construal

between suitably presented forms of program code and what the learner experiences takes precedence over the formal analysis of sequences of instructions. The program code is presented not simply as a closed representation of 'the correct behaviour' but as an invitation to 'what if?' experiment. The same mental model also applies directly to educational software based on spreadsheet and dynamic geometry principles [28, 31].

Figure 2 depicts the generalisation of Fig. 1 that is associated with making construals. Note that making a construal does not necessarily have to involve a computer; it relies on an mode of interpretation and three basic concepts that are suggested naturally by Papert's question *'where can it go from here?'*. The interpretation of Fig. 2 relates to a specific moment in the experience of the maker (what Papert is informally alluding to as 'here'). In this moment, in the spirit of James's radical empiricism, the maker seeks to experience a connection between the two aspects of experience that are being offered, on the one hand, by the construal and, on the other, by its referent. This connection is robust if the construal and its referent can be correlated in such a way that there is a close correspondence between what the maker encounters in the referent and in the construal in three respects:

– the agency that is perceived to be at work ("**agents**"),
– the entities this agency can affect ("**observables**")
– how changes to these entities are deemed to be concomitant ("**dependencies**").

A good correlation guarantees that the construal can serve in the role of an object-to-think-with in relation to the referent, and that the agency attributed to the maker in the construal gives a good indication of "where it can go from here".

The building of a construal proceeds in parallel with developing understanding of the nature of its referent that is reflected in the identification and refinement of ever richer observables, dependency and agency. The most significant feature that distinguishes Fig. 1 from Fig. 2 is the reference to a 'Context' for the making that also evolves in parallel. In programming, the aim is to establish a context in which the observables, dependency and agency considered can be abstracted and viewed in computational terms – a familiar process of simplification through model-building that is characteristic of theoretical science as and when robust construals exist. When making a construal, the context in Fig. 2 is typically more fluid and volatile, in keeping with the underlying spirit of uncertainty, exploration and experiment.

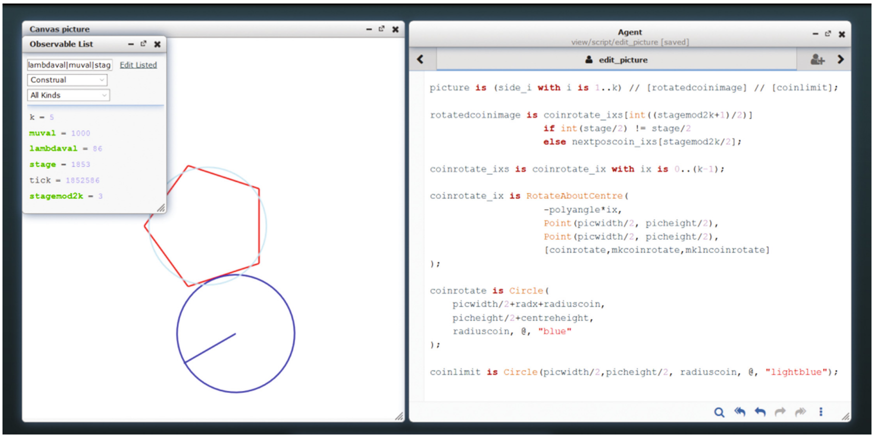


Fig. 3. A construal for Papert's penny rolling puzzle (see [25] and [13: p. 150])

In using the computer to make a construal, the current configuration of observables and dependencies is framed as an acyclic network of definitions (or "script"). A typical agent action involves assigning a new value, or giving a new definition, to an observable. Figure 3, a screenshot from the current online environment, is a sketchy indication of how a construal is made: in this case, the subject of the construal is the object-to-think-with that Papert associates with a penny rolling problem posed by Martin Gardner [13: p. 150]. Papert conceives this object-to-think-with, in which an approximating regular k-polygon of the same circumference is substituted for the stationary penny, to account for the fact that when one penny is rolled around another it undergoes *two* complete revolutions. The extract from the script includes an observable rotatecoinimage to represent the rolling coin. This is based on observing the coin in two different phases: as it rolls along a side of the pentagon, and as it turns at a corner. The observable coinrotatecoin_ixs is a list of all k possible side rolling configurations of the coin, as derived from the template observable coinrotate. The way in which the configuration of the rolling coin is correlated with passing time (as recorded in the tick observable) can be inferred by inspecting key observables that are displayed in the Observable List at the top left. They include the observable stagemod2k which determines which phase of the coin motion currently applies. In keeping with the intended agency that Papert invokes in interpreting his object-to-think-with, the number of sides of the polygon can be freely changed dynamically (even whilst the animation is in process) so that, as k increases, it converges to a circle of the same radius as the rolling penny (as depicted in Fig. 3).

In his *Talks To Teachers* [8], William James emphasises the distinction between understanding the basic psychological principles of learning and the art of teaching. He is at pains to point out that in order to be an effective teacher, it is neither necessary nor sufficient to be familiar with the underlying principles of learning. In a similar way, making construals is presented as a way of understanding what is involved in developing educational software that is to be distinguished from the practical skills needed to create a specific educational application in the most direct and effective manner. The purpose of making construals in the framework of Fig. 2 is not to displace other simpler approaches to programming specific educational applications but to clarify the fundamental principles on which these operate and in the process bring coherence to the semantic and epistemological principles involved. The most significant potential impact of introducing this new perspective is to give clear expression to the great varieties and subtleties of meaning that inhabit the space in which construction occurs. In practical terms, this can add new depth to Papert's analogy between learning to 'instruct' the computer and learning to speak a language, providing a shared interactive artifact that can enrich the conversations between the human agents who contribute to design and problem-solving in many different roles. This is a function that cannot in general be served by conventional programs.

A key point is that making construals is a way of exploiting the computer that transcends a pure computational framework. In *Mindstorms*, Papert advocates formulating computational models not necessarily because they are the most appropriate models, but because they oblige us to reflect deeply upon our understanding and the roots of our knowledge (cf. McCarty's account of computer-based modelling in the humanities [11]). He also recognises the potential for learning through interacting with concrete physical objects, both as a precursor to understanding abstract structures (as in the 'gears' of his childhood), and as a way of giving concrete expression to abstract concepts. Learning of the former kind features in 'computer science unplugged' [23], where activities that do not involve the computer are developed to teach computational thinking. Learning of the latter kind, where abstract principles are given concrete practical expression, features in educational robotics. In both these settings, the primary emphasis is on the quality of the interactive experience of concrete artifacts that computing technology enables. It is in just such contexts that the notion of making construals is most appropriately invoked. This is illustrated by a construal of giving change at [25] and by Arduino-based construals in which observables have direct physical counterparts [25].

5 Conclusion

This paper pays tribute to Papert by making connections between many of his key insights concerning 'the art of intellectual model-building' and 'making construals'. It also raises some challenging questions.

In interpreting 'constructionism', is has been natural to focus on the idea that the learner takes responsibility for constructing 'objects-to-think-with', and to infer from this that learners must acquire the requisite computing skills. This paper consolidates on previous critiques of 'constructionist' practices [27:#132,#111,#080] to argue that making construals is better suited to this role than conventional programming. It would be facile to suggest that making construals is an easy skill to acquire, however, as a closer inspection of the construal depicted in Fig. 3 indicates. Indeed, not only would it be challenging for a casual learner to make such a construal, but it is questionable whether the educational value of Papert's object-to-think-with, as set out in [13: p. 150], is sufficiently enhanced to justify the technical investment in its construction. The merits of using the computer to make such a construal relate to a much broader role and agenda.

In understanding how best to use the computer to support learning (cf. Sect. 3 above), Papert and Crook's concern is to exploit the computer to its full potential as a way of capturing and communicating common knowledge within a well-conceived supportive conceptual framework in a collaborative learning context. If we liken the environment for making construals to a musical instrument, to expect learners to build construals is perhaps only as realistic as expecting the audience at a concert to be able to compose a violin concerto, play the soloist's part and conduct the orchestra. In exploring this metaphor, the expression 'making a construal' has a helpful ambiguity. To make a construal is not necessarily to create a physical digital artifact – it is to apprehend a connection in your experience. This is something we are capable of doing however limited our technical competence, as is suggested by the choice of the term 'maker' in Fig. 2 to refer to the role of every human agent involved.

The goal of making construals is no more and no less than trying to clarify and communicate working understandings. It is a venture in collaborative learning for which there can be no guarantee of success. We should not judge the quality of a construal by how easily it can be constructed by a learner, though we are concerned with how readily the learner can make insightful connections with other experience of the learning domain. To revisit the analogy with music-making, the quality of a musical composition such as a concerto is evaluated in terms of the degree of satisfaction it gives to all who participate in its rehearsal and – the audience members, the soloist, the instrumentalists, the conductor, the critics. A successful musical composition is appreciated by all participants in different ways and degrees according to their level of familiarity and expertise. Appropriating Papert's own word, such success is achieved through 'syntonicity' – each person bringing different agency and observation to bear in such a way that they are able to make and share exceedingly rich connections.

This paper itself makes its own contribution to a wider tribute to Papert in just such a way by highlighting the diverse participants in educational technology who have been able to elaborate on, and celebrate, the products of his vision and imagination.

Acknowledgments. I am indebted to all the members of the CONSTRUIT! project team and to Dimitris Alimisis in particular for inviting me to prepare this keynote paper. The CONSTRUIT! project has been funded with support from the European Commission under the Erasmus + programme (2014-1-UK01-KA200-001818). This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

References

1. Boytchev, P.: Logo Tree Project. http://elica.net/download/papers/LogoTreeProject.pdf
2. Crook, C.: Computers and the Collaborative Experience of Learning, International Library of Psychology. Routledge, London and New York (1996)
3. Gallwey, W.T.: Inner Tennis: Playing the Game. Random House, New York (1976)
4. Gooding, D.: Some historical encouragement for TTC: alchemy, the calculus and electromagnetism. Paper presented at Thinking Through Computing, 2–3 November 2007. go.warwick.ac.uk/em/thinkcomp07/gooding2.pdf
5. Harfield, A.: Empirical modelling as a new paradigm for educational technology. Ph.D. thesis, University of Warwick, UK (2008). go.warwick.ac.uk/em/publications/phd/ant/
6. Hatch, M.: The Maker Movement Manifesto. McGraw-Hill Education (2014). http://cds.cern.ch/record/1643837. Accessed 8 July 2015
7. Jackson, M.: What can we expect from program verification? IEEE Comput. **39**(10), 65–71 (2006)
8. James, W.: Talks To Teachers on Psychology: And to Students on Some of Life's Ideals. Henry Holt and Company, NY (1899)
9. James, W.: Essays in Radical Empiricism. Bison Books, London (1996). (first published in 1912)
10. Khan Academy. https://www.khanacademy.org/computing/computer-programming
11. McCarty, W.: Humanities Computing. Palgrave Macmillan, Basingstoke, UK (2005)
12. Minsky, M.: The Society of Mind. Simon & Schuster, NY (1985)
13. Papert, S.: Mindstorms: Children, Computers and Powerful Ideas, 2nd edn. Harvester Wheatsheaf, UK (1993)
14. Papert, S.: The Children's Machine: Rethinking School in the Age of the Computer. Harvester Wheatsheaf, UK (1993)
15. Piaget, J.: Piaget's theory. In: Mussen, P. (ed.) Carmichael's Manual of Child Psychology, p. 715. Wiley, NY (1970)
16. Roe, C.: Computers for learning: an empirical modelling perspective. Ph.D. thesis, University of Warwick, UK (2003). go.warwick.ac.uk/em/publications/phd/croe/
17. Turkle, S., Papert, S.: Epistemological pluralism and the revaluation of the concrete. In: Harel, I., Papert, S. (eds.) Constructionism, pp. 161–191. Ablex Publishing Corporation (1991)
18. Victor, B.: Inventing on principle. Invited Talk at Canadian University Software Engineering Conference (CUSEC) (2012). http://worrydream.com/InventingOnPrinciple
19. Victor, B.: Learnable programming. http://worrydream.com/LearnableProgramming/
20. Wing, J.: Computational Thinking. CACM Viewpoint, pp. 33–35, March 2006
21. Wolfram, S.: The Poetry of Function Naming, Blog post, 18 October 2010. http://blog.stephenwolfram.com/2010/10/the-poetry-of-function-naming/
22. Wolfram, S.: How to Teach Computational Thinking, Blog post, 7 September 2016. http://blog.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/
23. Computer Science Unplugged. http://csunplugged.org/

24. CONSTRUIT!: construit.org
25. Edurobotics resources (2016). go.warwick.ac.uk/em/construit/year3/events/edurobotics
26. EM website. go.warwick.ac.uk/em/
27. EM papers. go.warwick.ac.uk/em/publications/papers/. (each paper is indexed by a 3 digit code #XYZ to be appended to the url)
28. GeoGebra. https://www.geogebra.org
29. Sipitakiat, A., Tutiyaphuengprasert, N. (eds.) Proc. Constructionism, Bangkok, Thailand (2016). http://e-school.kmutt.ac.th/constructionism2016/
30. Scratch - Imagine, Program, Share. https://scratch.mit.edu/
31. Spreadsheets in Education (eJSiE) electronic journal, Bond University School of IT, Australia. http://epublications.bond.edu.au/ejsie/1

Primary Level Young Makers Programming & Making Electronics with Snap4Arduino

Alfredo Pina(✉) and Iñaki Ciriza

Math & Computer Engineering Department, Public University of Navarra, Pamplona, Spain
pina@unavarra.es, zirizabeloki@gmail.com

Abstract. The main goal of this study is to show that young makers (8–12 years old) are able to start programming & making electronics with Snap4Arduino, and that this is a way to have (already trained) Makers for secondary level education. We have performed a feasibility study of the Snap4Arduino as an educational tool for elementary school students. The study has been conducted as part of a certified course carried out by the Pamplona's Planetarium where different skills such as learning to learn or sense of initiative and entrepreneurship were evaluated through a PrBL project based learning methodology. Cooperative learning was achieved focusing on teamwork attitude and behavior.

Keywords: Snap4Arduino · Young makers · Primary education

1 Introduction

It is worthwhile to start saying that "times have changed". Today's Young people are growing in the full digital period. His development goes with the technology, which allows them to adapt easily to any advance. That is why Education has to move along continuing the progress, and has to reinvent itself every day. We have to focus on those Youngers, as they have been living surrounded by technology. At present society calls them Digital Natives. The term "Digital Native" coined by Mark Prensky appoints to this age group (mainly young people) that has grown in a digital and technological framework (computers, Internet, Smartphones, MP3, etc…) and that uses those devices and media in a completely natural way.

1.1 Methodological Issues

Some teachers use from time to time ICT in class. Others have replaced the tools they used to use by technology that does the same kind of activity, and therefore the learning outcomes do not change. This is why the use of ICT has to go hand by hand with a change in the applied methodologies.

Sousa [1] carried out several studies where he verified that knowledge acquired or learnt after 24 h for one student is of 5% for lectures, 50% for group discussions, 75% for practical experiences and 90% for teaching to peers.

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_2

Following this matter, two of the most extended methodologies are Problem Based Learning (PBL) and Project Based learning (PrBL). Both have the same approach, both are authentic, both use constructivist learning approximations, both are student centered and include the role of the teacher as a guide. Such strategies (PBL & PrBL) are normally used in technology for education lectures and the main aim is to involve students in research processes and in technological problem solving.

Mettas & Constantinou [2] argue that both are used in a combined form and that are complementary. Blumenfeld et al. [3] think that PrBL learning with projects where the students have to deal with finding solutions to non-trivial problems, making and refining questions, discussing ideas, making predictions, designing plans and/or experiments, gathering and analyzing data, drawing conclusions, disseminating their thoughts and results to peers and creating new ideas or improving products and processes.

We need to change methodologies and at the same time, we need to implement new educational strategies that promote the effective development of the key skills & competencies of the XXI century; one good way to do that is introducing/using computer programming in schools.

1.2 Technological Issues: From Scratch to Snap4Arduino

Previous reflections and bringing programming into the scholar curriculum made popular the use of educational visual programming platforms like Scratch.

Scratch is a free programming software available for several devices, either offline (to be downloaded and installed) or online (through a browser). Such software has been developed by a group of researchers of the Lifelong Kindergarten Group at the media Lab of the Massachusetts Institute of Technology (MIT), supervised by Dr. Resnick. For Resnick [4] one of the best advantages of Scratch related to other similar platforms is his attractiveness and accessibility, which allows anyone starting programming for the first time.

Scratch is an educational tool, but at the same time has a huge ludic potential. As says Prof. Maria Moriana Coronel about one experience with Scratch with Primary school pupils (http://programamos.es/una-clase-de-programacion-con-scratch-para-ninos-de-1o-de-primaria/):

"Pupils we are working with are very Young and have no experience in designing and programming with Scratch, and to understand the difference between playing and learning is not so easy. We observe that the experiences they have with digital artifacts are in general very passive, they receive information or follow instructions to play. That is why the first time with Scratch they think that is a game and are trying to know how to play, what are the rules, what the cat can do or where the button to start playing is. When they discover that they can create their own games the interest, emotion and curiosity arise: Scratch is a game that makes games!"

For Salido [5] a game is a vehicle to allow pupils to learn in a natural and funny way, at the same time requires following specific rules, and enhances social communication. Six years old kids imitate acts, guided by an adult or peers, and some years after they manage to identify a situation and to assume the rules to follow. To assume and reinforce the use of rules reflects an important level of self-assertion that shows clearly the build up of a social integration process.

It is clear that the game and the emotions have a direct relation that we have to exploit. Since Goleman [6] talked to us about Emotional intelligence, we have to know and to manage emotions personal and socially, because it is closely bound to motivation. A person is emotionally intelligent in the measure he can improve his own motivation.

Another tool that has started to be used at educational level is Arduino. It was created with the goal of providing an easy to use and cheap platform to teach electronics at the design high school of Ivrea (Italy). The developers & partners are the Italian Massimo Banzi and Gianluca Martino, the Americans Tom Igoe and David Mellis and the Spanish David Cuartielles, and they decided to make their design open using a Creative Commons license. They protect their authorship and the brand name, but the hardware, software and Doc are open and free, and therefore it is possible to download the circuits' design files or the software source files and the users can modify and share their stuff.

In fact the limitation with Arduino is our mind, as the simplicity of this platform allows to non-expert people in electronics to use and control in their designs electronic devices, in an affordable way.

Moreover, in the last years, a new tool, which is a kind of union between Scratch and Arduino, has been created and it is called Snap4Arduino. The roots of Snap4Arduino can be found in BYOB (Build Your Own Blocks), which is very similar to Scratch but with more possibilities as a programming language. Then BYOB has evolved to a new version called SNAP that finally has been modified in order to be able to work and interact with Arduino (mostly with all the different types of Arduino hardware). Snap4Arduino is constantly being revised and developed at the Citilab of Cornella (near Barcelona, Spain) by the Eduteca research group. The main partners are Victor Casado, Jordi Delgado, José García, Joan Güell and Bernat Romagosa while Ernesto Laval, JensMönig, Frank Hunleth and Mareen Przybylla are the main collaborators.

To date we did not find any paper explaining experiences in the use of Arduino at the primary school.

1.3 Educative Issues and Main Goals of This Paper

In the last 4 years, we have managed to create a sustainable network of schools and a significant group of persons working in a coordinated way with robotics, in primary and secondary education levels. The Educational authorities support our work, we have established also a link between the university, the schools and the Planetarium of Pamplona, in order to work as well in the school, and out of the school (Planetarium plays the role of Science and Tech Museum). The work we present here has been done within this context.

We believe that it is possible to work with appropriate materials and methodologies in several different scenarios. In all cases it should be possible to work either standard curricula topics or key competencies (or both). The target competencies in our case are:

- Linguistic communication
- Mathematical competence and basic competences in Science and Technology
- Digital competence
- Learning to learn

- Social and Civic competences
- Sense of Initiative and Entrepreneurship

Also the curriculum in the case of Primary level (years 4 and 5) in Navarra region has included, a few years ago, a specific section referring to programming within the Math's curriculum: "Use of tools and programming languages for problem solving".

The general goal of this paper is to point out the usefulness and benefits of Snap4Arduino as a didactical proposal in Primary school level. As there are neither scientific evidences nor papers about Snap4Arduino at this educational level, we have collected objective and practical data on this matter and we have set up a certified course in the Planetarium of Pamplona for kids on those ages.

We have as well some other specific goals.

First, we want to analyse if Project based learning is a valid approach and gives good results for teaching Snap4Arduino to Primary school kids. PrBL methodology is very commonly applied in primary school pupils. Moreover, Snap4Arduinos is a tool that allows a wide range of activities of investigation. We feel natural to combine them and is one of our hypothesis.

Secondly, we would like to assess Snap4Arduino as a tool to be used in a scholar context. As we have already said, in the Math's curriculum for primary level (in "processes, methods and attitudes" within the Math's section) it says that the contents should include "Integration of Information Technologies and the use Languages and tools for programming in the learning processes". Therefore, we have to estimate if Snap4Arduino could be a tool to be used for achieving such purposes.

Thirdly, we need to check if pupils can manage basic electronics. The basic functions of Snap4Arduino are also in Scratch (and therefore already used in primary level), but the advanced ones need to design basic electronic circuits and to understand basic electronic concepts. Electronics is not included in the primary curriculum, so the fact that the pupils should be able to manage such part is an open issue

In the rest of the paper, we present the details of the course (materials, methodology and teaching details), then we present and discuss the results of the pilot course. The paper ends with some conclusions and future ideas.

2 The Course

The course had a duration of 12 h divided into 4 sessions (3 h each, 10 h30–13 h30 with a small break). We made it during 4 consecutive Saturdays (April–May 2016) in the Robotics room of the Planetarium of Pamplona. Diana Gonzalez responsible for digital & technological training at Planetarium was helping and observing during all the sessions and was in charge of the camera for pictures and videos.

The main goal of the course was to check the viability of Snap4Arduino as a didactical proposal in primary education.

2.1 Participants, Materials and Methods

The course was covered with 18 participants out of 20 available; 14 boys and 4 girls, within a range of age between 8 and 12 years. Most of them had already some basic knowledge on Scratch (they had already participated in one Scratch course in the planetarium), which was very helpful for the progression of the sessions. Due to that and because we had this information some days before the course, we decided to focus on the advanced part of Snap4Arduino which interacts with the Arduino hardware (Arduino UNO).

The material used has been:

- Computers: one per group
- Arduino UNO board
- Arduino IDE and Standard Firmdata
- Kit Arduino
- Every Project written in a paper
- Document with image rights
- Surveys
- Pictures & Videos camera

The general methodological approach was Project Based Learning. We believe that self-learning through a continuous investigation or research process is the best way to acquire knowledge. The teacher has to provide the necessary tools to achieve this aim. The knowledge acquisition is much more powerful if the student reaches the conclusion on his own (instead of just being given it to him by the teacher).

At the beginning of each session, they have been provided with a project to develop it. The third first ones were more concrete or closed projects, as the main goal was to learn basic knowledge and skills with Snap4Arduino. The last project was completely open and they had then the possibility to apply any of the acquired knowledge and even to make new things. This last project is in fact the core of the PrBL methodology. To carry out the last project they had to investigate, to explore, to "touch", to try, to fail and to try again, etc… no doubt, it was a successful experience for them and for us.

They were divided into 5 groups of 3–4 pupils each: by doing that we assured that every group had a computer, an Arduino board and one Arduino kit. Moreover, teamwork was one of the competencies we wanted to promote, in order to check how they can help each other and the whole group capacity for programming. Alonso et al. [7] state that 1×1, one computer for every pupil, is not necessarily better than one computer for a group of students as there is no a direct relation between the quantity of available technology and their effect on the teaching learning processes.

The division criterion for the groups was the age, provided than in every group there was at least one pupil with Scratch previous knowledge. Doing that we wanted to check if applying such methodology and teamwork it was possible to achieve a cooperative learning.

2.2 Course Structure: Sessions with Different Projects and Different/Common Goals

Every session was divided into seven parts: specific objectives of the session, planning the activity, description of the project, specific material to be used, expected outcomes, activity development, results and discussion of the current session.

The specific objectives for the whole the course are:

1. To learn the basics of Snap4Arduino
2. To construct basic electronic assemblies
3. To use basic functions to activate/deactivate digital outputs for Arduino UNO board
4. To use basic functions to activate/deactivate analogic outputs for Arduino UNO board
5. To use DC motors and to understand their up and running
6. To design small projects using all the previous learnt knowledge

Session 1 (objectives 1, 2 and 3): The traffic lights. Simulating traffic lights using several costumes in Snap4Arduino. Practical application, electronic assembly and programming the Arduino UNO board to design and operate a traffic lights model (see Fig. 1).

Session 2 (objectives 1, 2 and 4): The colors lamp. Mounting a lamp with an RGB led and applying different signals to every lamp pin depending on the position of every potentiometer in order to get the mix of the colors of the RGB led.

Sessions 3 and 4 (objectives 1, 2, 5 and 6): The car and our project.

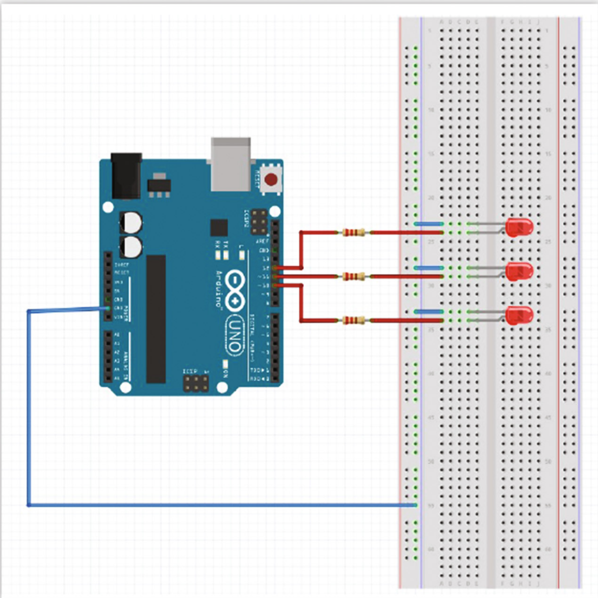


Fig. 1. The traffic lights board assembly (session 1)

The car: mounting and programming a car with DC motors with reduction in order to obtain forward, backward, left and right movements using an H bridge and programming everything in Anap4Arduino (see Fig. 2).

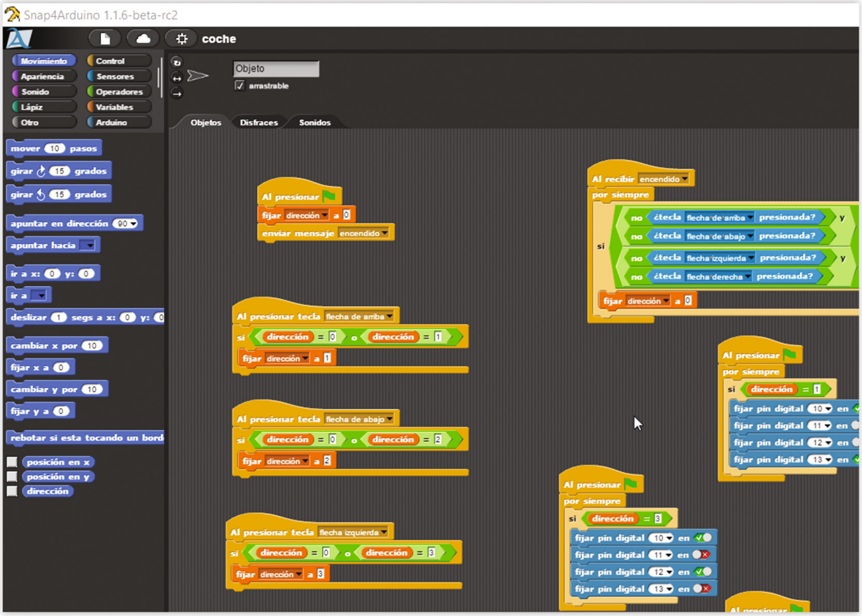


Fig. 2. Visual interface for Snap4Arduino (very similar to Scratch)

Our project: Open project using some of the already seen activities.

3 Results and General Discussion

3.1 Results and Teaching Within the Course

The obtained results in terms of the teaching experience, the solved problems and the expected outcomes were really challenging. Table 1 shows the main outcomes.

The main idea at the beginning was to leave every group to work "ad libitum", but we had some problems with team working, detected at the first session and therefore we had to apply a new strategy. We defined several different roles (demanding different skills and tasks) and everyone switched roles (see Fig. 4):

- moderator for the group management,
- mouse operator to control the mouse
- the assembler: responsible for the electronic board assembly

Table 1. Expected objectives for every session

Sessions	Expected objectives/outcomes
Traffic Lights (Digital Outputs)	Fully accomplished for all the groups
RGB Lamp (Analogic Outputs)	Electronics problems when assembling the board for the youngest group and programming difficulties for all the groups (division & potentiometer)
The car (DC Motors)	Difficulties to understand the electronic concepts (H Bridge) for all the groups. After assembling the board a detailed explanation was needed on the whiteboard (see Fig. 3)
Open Project	Challenging results and projects: • Spaceship lighting, with several color leds (UFO like) • Space invaders with red and green leds (blinking or continuous beam depending on the game) • A Story where a character has got a lantern (yellow led) • Doll jumping on a trampoline and lighting different color leds depending on the jump direction • Door with lock that opens/closes when the user plays the right/wrong piano song (servo motor for the lock, paper for the door and Snap4Arduino for the piano song)

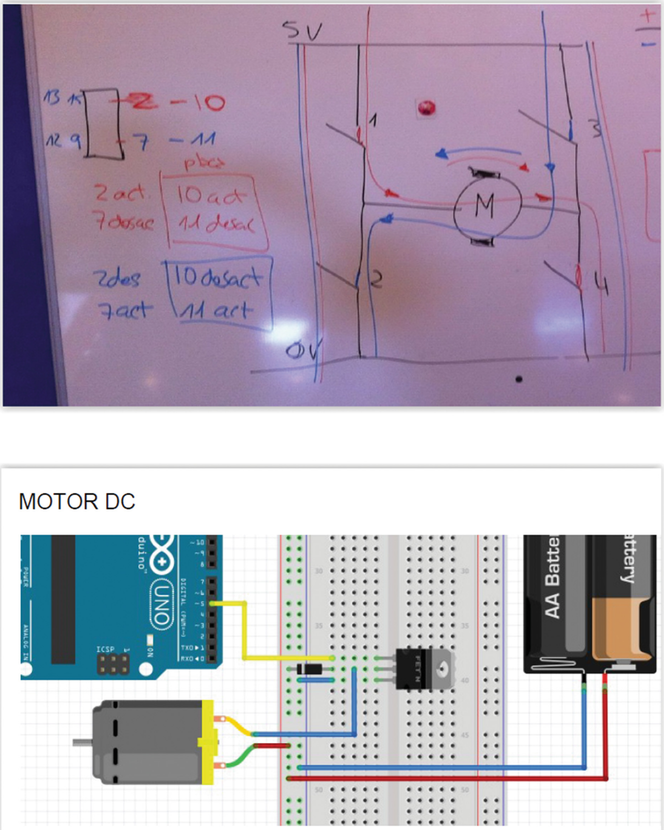


Fig. 3. Difficulties to understand electronic concepts (H Bridge) in session 3 for all the groups: needed explanations on the whiteboard

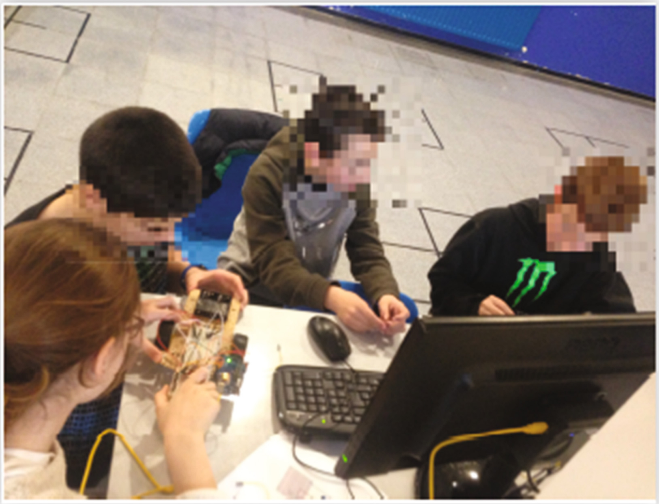


Fig. 4. Cooperative teamwork with different roles and skills

3.2 Importance of the Previous Background

As we have said before, primary students are really digital natives and they have huge skills in digital competency, as they have been all their life interacting with computers and technology. Nevertheless computers and digital artifacts are for them stuff for playing and do not see them as possible tools to learn or doing other things.

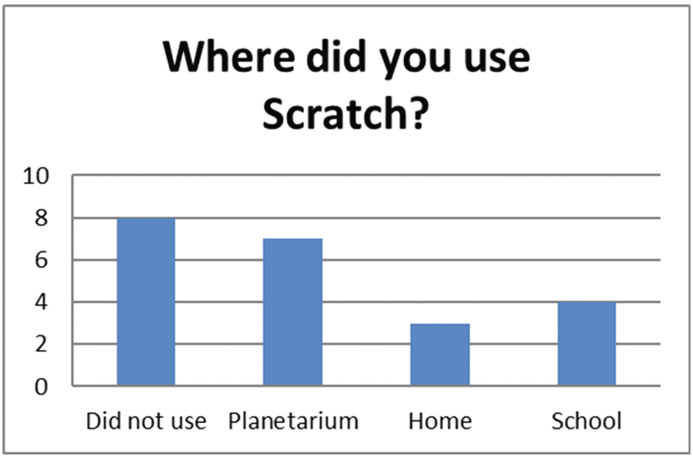


Fig. 5. Previous background in Scratch

The first day only 16 pupils came to the session. Eight of them had already used Scratch. Most of them in a course at the Planetarium but even four have used it at school and a few at home.

Working groups where organized to include at least one person with Scratch background for several reasons. First of all for achieving the objective of being able to work with Snap4Arduino: the language and interface of Scratch is nearly the same and therefore not every group was starting from scratch. Secondly, the fact that one of the pupils had to explain things to their mates was a wished didactical situation, where everybody could learn, specially the one who had to explain things to the others.

Most of the pupils had already had positive previous experiences in computer programming; the results of the pre-survey show this fact (Figs. 5 & 6).

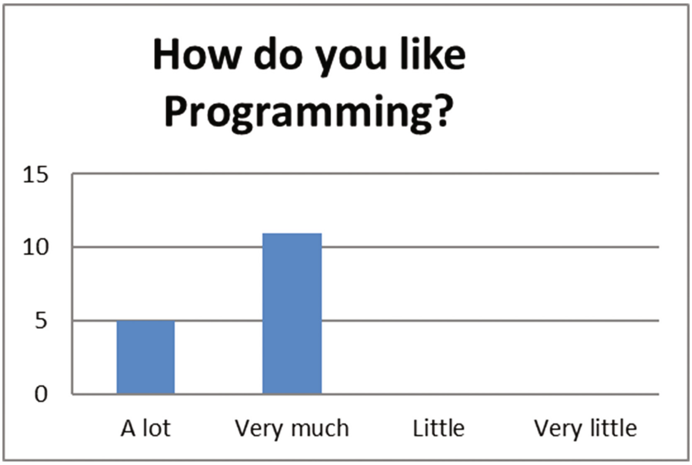


Fig. 6. Previous background in programming

3.3 Methodological Issues: Pupils Programming with a PBL Approach

In Fig. 7 we can see that most of the pupils were expressing positive feelings when they were programming all together in every group. The four pupils that were no expressing any feeling is because it was the first time they were programming.

Analyzing the teaching and learning of the 2 first days, and more specifically the sentiments and attitudes the pupils were expressing, we felt that the results could improve; i.e. more pupils will express more positive feelings when programming, which in fact is what happened during the last 2 sessions.

Having chosen PrBL as the methodology was a success. It has been possible and very pleasant to work with and we have managed to make a constructivist learning with Snap4arduino.

The four projects were designed ad hoc for that purpose. They had to “try and experiment” and “investigate” to achieve the projects. As we have already said, the first projects were closed ones (we were proposing what to do and the final goal). However, the last one was completely open as they have to decide what to do and what was the

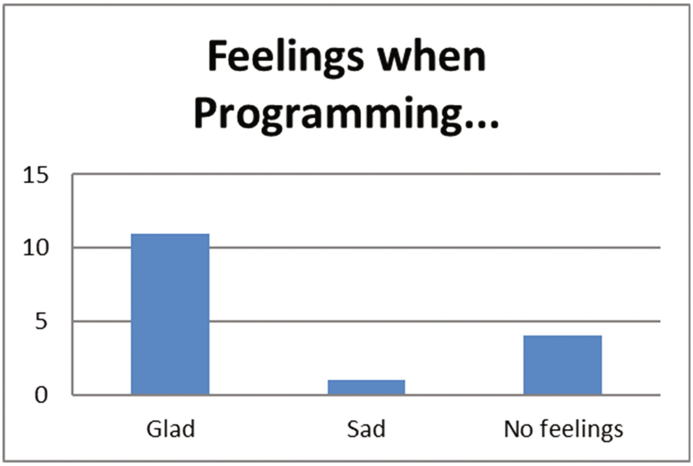


Fig. 7. Their feelings when programming during the course…

aim of the project. The methodology and the way the four projects were designed has been central to get the successful results we had. Pupils associated the projects more to a game than to an ordinary class; therefore, they could express fun, joy or amusement feelings. This can be seen on the results of the survey of the last day (Fig. 8).

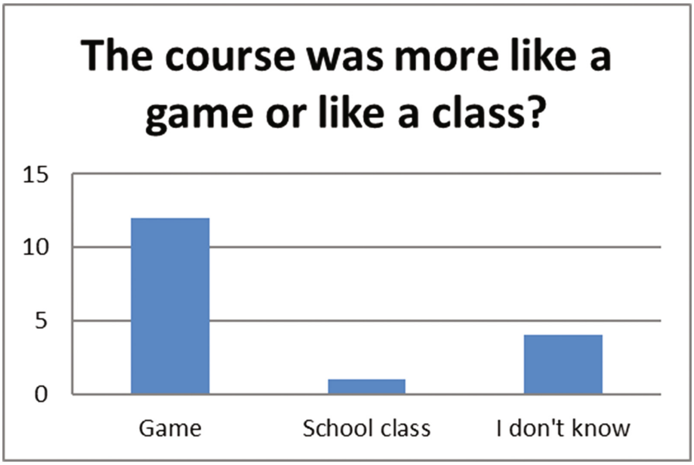


Fig. 8. Learning and playing

One of the main advantages of working through projects is that you can apply them to a wide range of ages and topics. It is true that depending on the age group or the previous knowledge, you may have limitations, but you can apply the projects in many possible different scenarios.

3.4 Curriculum and Key Competencies

It has been shown that in this case the electronics concepts and board assembly were important difficulties for the pupils; even it was the most difficult aspect during the course. Primary education curriculum does not include any electronic contents, but we have observed nevertheless that they are able to assimilate basic concepts in this area. Even that the electronic circuit assembly on the board was a heavy task, they managed to make it and, much better, they were able to re-use the projects with other practical ideas a few hours later. In our opinion, this shows that it is possible to introduce basic electronics assembly at the end of the primary level education. The key point is to start with basic things and then to go deeper, with more complicate concepts or assemblies.

We think that Snap4Arduino is a tool that provides many possibilities when working with Arduino hardware and we believe that including also basic electronics in the curriculum (not only computer programming) could be a great idea.

Seeing the results of the projects after four days, we can say that the “joint tool” Snap4Arduino/Arduino is a valid and reasonable tool that could be introduced into the classroom. All the pupils, at the end of the course, agree that programming with electronics is a way of learning.

The other point we would like to outline is the one related to the key competencies that every kid has to acquire during the primary education level (before to get into the secondary level).

Any reader of the paper can see that during the course we have been working mostly all the competencies (except the cultural expression and awareness one). In fact, depending on the project we propose to carry out to the pupils, we may be working some key competencies. Combining appropriate projects, we should/could be able to work all the key competencies. When designing projects, we argue that we do not have to think only about the contents we want to teach, but also what competencies we want to work as well; an adequate progression in every key competency will be very important for their future labor and social life. For example, in our case and during the course we had to introduce different roles for every partner of the group because the pupils needed to work more the civic and social competency.

3.5 Pupils’ Course Satisfaction

The last point we would like to note is the global satisfaction after the course. They expressed a relevant satisfaction and most of them where ready to apply for another course if possible. See Figs. 9 and 10.

Not only pupils were satisfied, also some parents were delighted with the course and they were asking for more information as they were interested in introducing such kind of courses at schools (as extra scholar activity). The planetarium of Pamplona was also very satisfied with the course and they have organized a second course with the same contents and methodology during summer 2016.

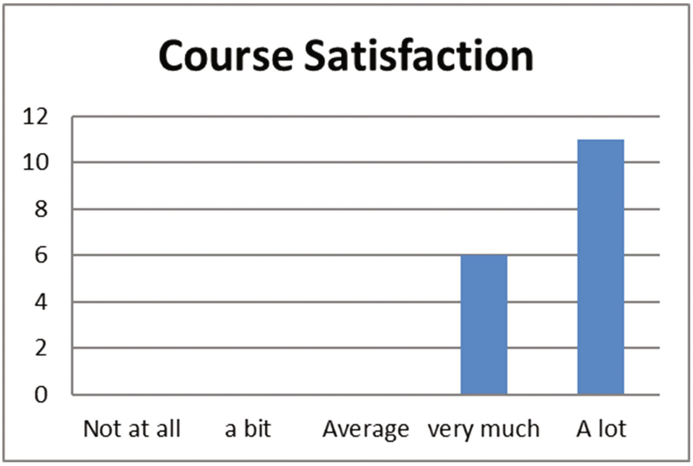


Fig. 9. Positive feedback from the pupils.

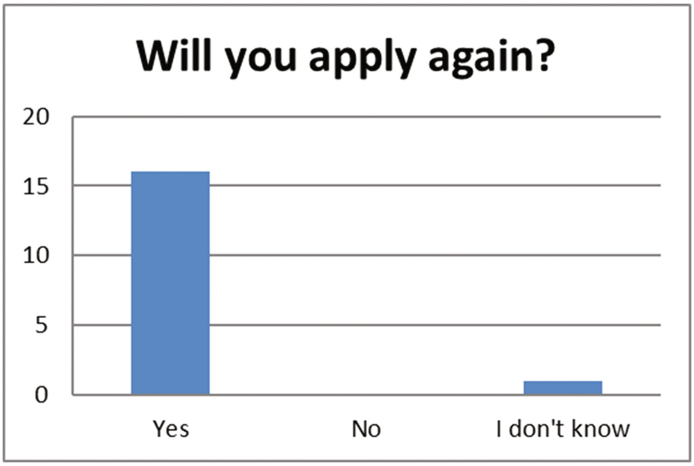


Fig. 10. Ready to continue….

4 Conclusions

The overall satisfaction score was of 8,3 (out of 10) and 100% of the students concluded that "programming is a valid learning method". We agree with them.

After the study, we conclude that Snap4Arduino is an applicable tool for an elementary school's syllabus. It achieves all the contents, evaluation criteria and learning standards established in the education curriculum of the region of Navarra. Moreover, it is a versatile tool that can be applied to all elementary school levels.

We consider that PrBL is the right methodology to work with Snap4Arduino as it allows that every pupil can discover all his potential through investigating or inquiring and applying trial and error method, which provides a significant learning.

Snap4Arduino unlike Scratch gives the opportunity to bring ideas to the real life using programming and basic electronic assemblies.

Acknowledgments. We would like to thank the Planetarium of Pamplona and in particular Diana Gonzalez for giving us the opportunity of implementing the course and for all their support.

References

1. Sousa, D.A.: How the Brain Learns. The National Association of Secondary School Principals, Reston (1995)
2. Mettas, A.C., Constantinou, C.C.: The technology fair: a project-based learning approach for enhancing problem solving skills and interest in design and technology education. Int. J. Technol. Des. Educ. **18**, 79–100 (2007)
3. Blumenfeld, P.C., Soloway, E., Marx, R.W., Krajcik, J.S., Guzdial, M., Palincsar, A.: Motivating Project-based learning: Sustaining the doing, supporting the learning. Educ. Psychol. **26**, 369–398 (1991). (3 y 4)
4. Resnick, M.: Learn to code, code to learn: how programming prepares kids for more than math (2013). https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn. Accessed 21 Feb 2017
5. Salido, M.R.: El juego de las siete notas. Exedra: revista científica **2**, 289–308 (2011)
6. Goleman, D.: Inteligencia Emocional. Kaidós, Barcelona (1996)
7. Alonso, C., Guitert, M., Area, M., Romeu, T.: Tendencias emergentes en Educación con TIC. Espiral, Barcelona (2012)

Theater Meets Robot – Toward Inclusive STEAM Education

Calkin Suero Montero(✉) and Ilkka Jormanainen(✉)

School of Computing, University of Eastern Finland, Joensuu, Finland
{calkins,ilkka.jormanainen}@uef.fi

Abstract. Science, technology, engineering and math education has been expanded by the inclusion of the arts as an integral area. As such, STEAM education is posed to be gender inclusive and accessible for all levels of technical abilities. In this paper, we present our work on the inclusion of the arts into robotics education through the *theater robotics* concept. A description of the processes involved to develop a multidisciplinary theater robotics project are presented here alongside highlighted benefits that can be expected in terms of fostering students' 21st century skills. The lessons learned from our theater robotics ventures will be leveraged in the Horizon 2020 eCraft2Learn research and innovation project, where we expect to achieve a smooth integration of technical and humanities subjects and skills, paving the way toward a motivating and inclusive education for all students in different STEAM contexts.

Keywords: Theater · Robotics education · Arts · STEAM

1 Introduction

The term STEM had its beginning in the 1990's with the National Science Foundation of the United States introducing the acronym as a shorthand for science, technology, engineering and math [1]. Despite the debate regarding the scope of the meaning of the STEM acronym [1], it has become a term widely used when discussing education policy in schools for improving natural science and technology competitiveness [2]. A recent development toward competitiveness improvement has seen the inclusion of arts and design into the traditionally hard sciences involved with STEM education [3]. As such, STEM has become STEAM to cover interdisciplinary education in the fields of natural sciences, technology, engineering, arts and math, with several initiatives deployed world-wide.

STEAM education is to play an important role when proposing solutions to overcome the well-researched and identified gender issues pronounced in many Western societies related to involve, motivate and retain girls and women in the fields of science, technology and math, including socio-cultural expectations, lack of female role models, relevancy of curricula, pedagogy and so forth [4]. Furthermore, the inclusion of arts into the hard sciences curricula is intended to foster creativity and economic growth [5]. Recently, STEAM education is actively reinforced, albeit informally, within *makerspaces* or *fab labs* [6] where individuals or groups of individuals build (and market)

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_3

products that have been recreated and assembled using virtually any raw material including recycled electronic, plastic, silicon, etc. However, problems of inclusivity are often observed in such spaces as indicated by a survey carried out by Intel [7] in USA, China and Mexico regarding girls and women involvement in the *Maker Movement* highlighting that *"gender norms, bias and stereotypes … negatively affect their involvement in making and their access to maker spaces"*.

In an attempt to create a more welcoming and inclusive formal and informal STEAM education several strategies have been put forward, including the devising projects that are collaborative in nature and meaningful not only for boys but also for girls and women, and the inclusion of makerspaces in libraries and schools [7]. To this end, our work contributes towards establishing a more inclusive STEAM education by proposing *theater robotics* as a concept that calls for the design, construct, and replay of a narrative or a theater chapter through educational robots. The aim is to attract, motivate and engage students of all level of technical ability through hands on, collaborative projects. This involves a multidisciplinary oriented work in line with the Finnish education phenomenon-based learning principles rooted in Constructivism [8, 9].

2 Arts in Technology - A Brief Review

Creative thinking has been at the central stage in education strategies of countries worldwide to foster 21st century skilful individuals [10]. Already from the beginning of the 1990s arts were known to enhance creativity, imagination and foster critical thinking [11]. It is therefore of little surprise to find many attempts to integrate arts with the hard sciences from educational policy making to academic research [3, 12]. In the USA for instance, there has been a major push to foment the arts within the traditional STEM education. For example, with the commercialization of LiliPad Arduino [13, 14] several eTextile applications to include arts into formal and informal STEM education have been proposed primarily targeting the engagement and encouragement of girls in science and technology fields [15, 16]. Benefits have been suggested for broadening of students' skills in several cognitive and aesthetic areas through STEAM education [15].

In Europe, the renewed school curricula across the continent also emphasises interdisciplinary pedagogical approaches and projects. Robotics as a part of the field of computing and engineering may be easier to integrate with basic skills in physics, electronics, programming and mechanics. A more difficult integration, however, may come when the above mentioned skills need to be combined with history, literature, religion or artistic expression of any kind. Nevertheless, research has reported successful integration of social sciences, humanities and arts with technical skills. For example, Lau et al. [17] describe a study where programming was taught through fashion and design in a course of wearable computing. Similarly, Kim et al. [18] report the development of the Artbotics curriculum combining arts and computing subjects, and Brunvand and Stout [19] describe a cross-disciplinary course for art, computer science and engineering students together.

Moreover, experiences in sub-Saharan Africa for enriching learning environments, games, and animations with cultural heritage and storytelling traditions (see for example [20, 21]) have shown that digital tools can be applied effectively in diverse contexts to bridge inter-disciplinary borders in school and after-school settings. However, as Salgian et al. [22] conclude, there are few reported efforts on combining engineering with fundamentally different disciplines such as humanities or social sciences. Our work contributes with the integration of theater as an artistic expression into robotics with the aim of creating STEAM projects that are challenging, interesting as well as inclusive for all students.

3 Theater Robotics: Context – Story – Crafting – Telling

As indicated in the previous chapter, arts and robotics has been combined in STEAM education in many different ways to foster integration of multiple subjects into a multi-disciplinary study project. However, depending on the technology and nature of the project, integration might still be superficial, where the focus of the project is on learning a particular technology instead of higher level goal to foster innovation of novel products and services through design and making. Based on examples found from literature, as well as our experiences on organizing children's technology clubs, teachers' supplementary education, and other technology-oriented activities, we have formulated the concept of *theater robotics* to frame interdisciplinary technology-oriented projects that naturally combine several school subjects even over distant disciplines, such as technology, arts, and history.

Theater, or theatre, has been defined as a collaborative art form where real or imaginative events are presented to an audience[1]. We argue that the collaborative aspect of a theatrical performance makes it a very suitable art form to enhance and develop learners' 21st century skills where collaboration takes fundamental importance [23]. Furthermore, the idea of integrating theater into robotics education has the potential to bring forward the development of problem solving, communication and critical thinking skills in the students through the processes of *ideation, planning, creation* and *sharing* of a theatrical play with the *programming* of robots as actors.

Projects in the theater robotics framework start by finding a story, fairytale, historical event, or other similar event to express as a theater play (**context**). The students first retrieve and comprehend information about the context. This can involve information searching from Internet, books, or even interviews with elderly people. After the students have comprehended the context, they write a script (**story**) for the theater play to be implemented. Both of the abovementioned steps are geared towards literature, history, communication, media literature, and other similar subjects and skills. When the story is ready (or in parallel while developing it), students start **crafting** actors for the theater play. This can be done by using a well-known robotics platform such as Lego Mindstorms (Fig. 1), or by building robots from the scratch with Arduino or similar controlling technology. A 3D printer and similar assets are very useful in this phase, albeit their usually rather complicated user interfaces and workflows.

[1] See https://global.britannica.com/art/theatre-art; https://en.wikipedia.org/wiki/Theatre.

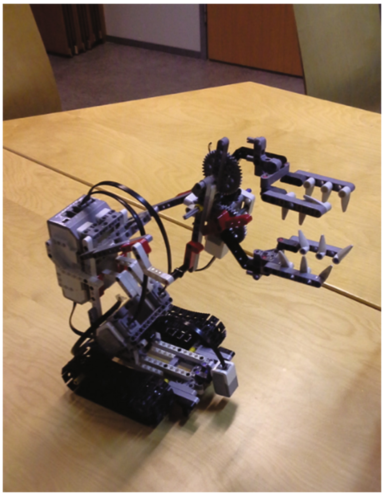


Fig. 1. A character of the *Big Bad Wolf* built with Lego Mindstorms for the Little Red Riding Hood story.

The next step in the crafting process is programming the robot actors. Depending on the platform, this phase requires a certain level of knowledge in programming. Lego Mindstroms graphical programming environment, as well as C-based language for Arduino boards follow the procedural programming concepts such as sequential execution order, loops, conditions, subroutines, and so on. In one of the previous projects [24] we have been conceptualizing also programming approaches that could be more natural in the context of theater robotics. Behavior-based programming approach with a specifically built user interfaces and a seamless collaboration over a network between project sub-groups would provide an environment where students could work as script writers instead of programmers (contextualization). This kind of setting requires an accurate indoor positioning system for the theater robotics framework. We have presented earlier [24] that a system based on machine vision and low-cost infrared beams attached to robots and the theater stage could provide a cheap and robust yet accurate enough positioning for several robot actors even in a stage as big as 10–15 m^2 (Fig. 2).

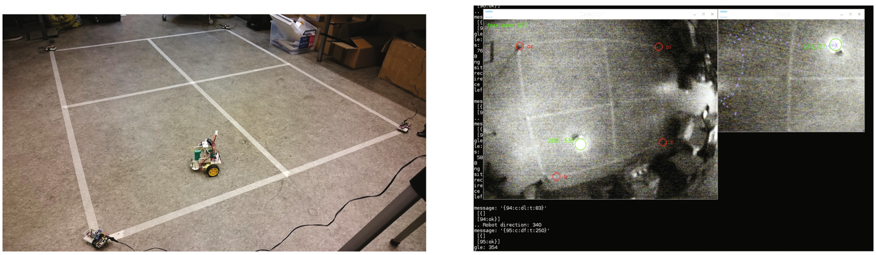


Fig. 2. A prototype of the theater robotics stage and machine vision view to the stage for accurate detection of actors' positions [24].

The final step in the theater robotics process is performing (**telling**) the story to an audience. If the previous step involved heavily subjects such as mathematics, physics, computing, and handicrafts, this phase has emphasis on creative arts, performing, presentation skills, and even sport if the students involve themselves in the theater play. Table 1 presents the essential phases of the theater robotics processes with the corresponding forecasted skills and subjects to learn.

Table 1. Description of a theater robotics project processes

Process	Description	Academic subject	Highlighted benefit
Context *(ideation)*	Students choose a story or a theater play to implement	History, literature, religion	Students gain cognitive skills in a humanity subject
Story *(planning)*	Information collection and story scripting	Literature, articulacy, information retrieval and media literacy	IT skills include use of search engines and online resources
Crafting *(creation & programming)*	Robot design and building	Arts and design, IT, engineering, technology and handicraft, mathematics, physics	The students articulate the integration of arts with technical skills
	Defining scripts for the robot actors	Computer science, logic	IT skills focus to computer science core skills, such as programming fundamentals
Telling *(sharing)*	Theater performance in front of an audience	Arts, sport, social skills, technology	Students learn feedback from their creation. This is expected to motivate their creativity and engagement in future projects

4 Ongoing Work and Future Perspectives

Although our work on theater robotics is an ongoing venture, we have already projected the added benefit from the students' perspective that could be expected through the deployment of such an arts and technology integration. Of particular interest is the development and implementation of a suitable pedagogy that allows for the combination of the fundamentals of building and making within a technical problem solving challenge with the ideation and planning of a collaborative perspective within a humanity subject.

The processes that take place during a theater robotics project (Table 1) show how such a combination of technical and humanities subjects can be achieved in a smooth manner. However, care should be also taken to train the teachers and/or instructors to successfully carry out such a multidisciplinary project in a way that is interesting and motivating for the students. This represents a challenge that needs to be addressed. Furthermore, a holistic evaluation of carrying out a *theater robotics* project in terms of qualitative and quantitative measures is part of the future perspectives of our work.

To address the above-mentioned needs for development, we leveraged the core lessons learnt from conceptualizing theater robotics concept in our forthcoming *eCraft2Learn* project funded from EU Horizon 2020 research and innovation program (see http://cordis.europa.eu/project/rcn/206165_en.html). In the project, network of 12 partners from academia and industry will seek robust and versatile solutions and develop needed pedagogical methods and learning environment assets to support a smooth integration of technical and humanities subjects and skills paving the way toward a motivating and inclusive STEAM education for all students.

References

1. Sanders, M.E.: Stem, stem education, stemmania. Technol. Teach. **68**(4), 20–26. December 2008–January 2009
2. USA National Science Board, Revisiting the STEM Workforce. https://www.nsf.gov/pubs/2015/nsb201510/nsb201510.pdf
3. STEAM - Not STEM Whitepaper. http://steam-notstem.com/articles/whitepaper/
4. Blickenstaff, J.C.: Women and science careers: leaky pipeline or gender filter? Gend. Educ. **17**(4), 369–386 (2005)
5. H. RES. 51. House of Representative Joint Resolution, USA (2013). https://www.gpo.gov/fdsys/pkg/BILLS-113hres51ih/pdf/BILLS-113hres51ih.pdf
6. Peppler, K., Bender, S.: Maker movement spreads innovation one project at a time. Phi Delta Kappan **95**(3), 22–27 (2013)
7. Wittemyer, R.: MakeHers: Engaging Girls and Women in Technology through Making, Creating and Inventing, Intel Survey (2014). http://www.intel.com/content/www/us/en/technology-in-education/making-her-future.html
8. Meriläinen, M., Piispanen, M.: Learning as a phenomenon – manuscript of phenomenon based learning. In: Proceedings of the EDULEARN12, pp. 5447–5454 (2012)
9. Silander, P.: Phenomenon-based learning, teaching by topics (2015). http://www.phenomenaleducation.info/home.html
10. Shaheen, R.: Creativity and education. Creative Educ. **1**, 166–169 (2010). doi:10.4236/ce.2010.13026
11. Art enhances creativity for all at MIT. MIT Tech Talk **37**(22) (1993). http://news.mit.edu/1993/arts-vest-0210
12. Land, M.H.: Full STEAM ahead: the benefits of integrating the arts into STEM. Procedia Comput. Sci. **20**, 547–552 (2013)
13. Buechley, L., Eisenberg, M.: The LilyPad Arduino: toward wearable engineering for everyone. IEEE Pervasive Comput. **7**(2), 12–15 (2008)
14. Buechley, L., Eisenberg, M., Catchen, J., Crockett, A.: 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, pp. 423–432. ACM (2008)
15. Peppler, K.A.: STEAM-Powered computing education: using e-textiles to integrate the arts and STEM. IEEE Comput. **46**(9), 38–43 (2013)

16. Kafai, Y.B., Lee, E., Searle, K., Fields, D., Kaplan, E., Lui, D.: A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. ACM Trans. Comput. Educ. (TOCE) **14**(1), 1 (2014)
17. Lau, W.Y., Ngai, G., Chan, S.C.F, Cheung, J.C.Y.: Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE 2009). ACM, New York (2009)
18. Kim, J.H., Coluntino, D., Martin, F.G., Silka, L., Yanco, H.A.: Artbotics: community-based collaborative art and technology education. In: ACM SIGGRAPH 2007 Educators Program (SIGGRAPH 2007). ACM, New York (2007)
19. Brunvand, E., Stout, P.: Kinetic art and embedded systems: a natural collaboration. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, pp. 323–328. ACM (2011)
20. Bada, J.K., Suhonen, J.: NetAIDS: digital learning environment for HIV/AIDS education in Ugandan schools. J. Health Inf. Developing Countries **5**(2), 247–258 (2011)
21. Duveskog, M., Sutinen, E.: Enriching student HIV awareness by digital storytelling. J. Educ. Multimedia Hypermedia **22**(4), 383–406 (2013)
22. Salgian, A., Nakra, T.M., Ault, C., Wang, Y.: Conducting robots – bridging the gap between science, technology and the arts in the undergraduate curriculum. In: Integrated STEM Education Conference (ISEC), pp. 1–4. IEEE (2013)
23. Rotherham, A.J., Willingham, D.T.: "21st-Century" skills: not new, but a worthy challenge. Am. Educ. **34**(1), 17–20 (2010)
24. Laamanen, M., Jormanainen, I., Sutinen, E.: Theater robotics for human technology education. In: Proceedings of the 15th Koli Calling Conference on Computing Education Research, pp. 127–131. ACM, New York (2015)

Educational Robotics Projects in School and Higher Education

A Training Course in Educational Robotics for Learning Support Teachers

Francesca Agatolio[1], Monica Pivetti[2], Silvia Di Battista[2], Emanuele Menegatti[1], and Michele Moro[1](✉)

[1] Dipartimento di Ingegneria dell'Informazione, University of Padova, Padova, Italy
{francesca.agatolio,emanuele.menegatti,michele.moro}@dei.unipd.it
[2] Dipartimento di Scienze Psicologiche, della Salute e del Territorio, University G. d'Annunzio of Chieti-Pescara, Chieti, Italy
{m.pivetti,s.dibattista}@unich.it

Abstract. This paper discusses the new implementation of a strengthened introductory training course in Educational Robotics for pre-service and in-service learning support teachers. By means of a final written questionnaire we compare the results of the course in 2015 with this year course, when the number of hours were doubled. This year participants expressed a higher appreciation and a better attitude towards robotics. Teachers agreed on the conviction that robotics can enhance students' motivation to learning and that educational robotics sustains a new point of view on science for teachers. Regarding the implementation in class, approximately two third of the participants declare they had already an idea on how to integrate robotics in curricula. More specifically, participants named ASD (Autism Spectrum Disorders), ADHD (Attention deficit hyperactivity disorder), learning disabilities, mild mental retardation as aspects that can be effectively addressed by ER.

Keywords: Educational robotics · Learning support teacher · Special needs education · Inclusive education · Teacher training · Course evaluation

1 Introduction

After several years of studies, experimentations, a variety of proposed approaches, an increasing need to reform the educational system, particularly at the European level, the use of Information and communication technologies (ICTs) in education is no longer under discussion. But a profound rethinking of the role of this latter in the 21st century era is the mainstream for any serious reforming attempt [1–3]. Nonetheless this generally accepted conviction includes some critical aspects when special needs education (SNE) is concerned (in our context we consider in this category of students not only those with severe disabilities but also with mild forms of dislexia, dysgraphia, dyscalculia and ADHD, who are mixed with not affected students in ordinary classes; this is the usual case at least in Italy where special schools are very rare).

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_4

The first question is about the applicability of ICT for SNE: are ICTs suitable also for special needs students and which are the benefits in introducing alternative technological tools in these cases [4,5]. The second question is about the awareness that actually ICTs are driving forces of a new way of spreading knowledge and promote good practices in teaching/learning especially regarding SN students [6,7].

About the first question, literature highlights some of the the potentialities of using ICT for SNE [8–11]:

– improving collaborative and exploratory learning allowing students to interact with the material;
– helping in communication;
– improving the involvement of the students and their achievement in some of the major subjects;
– furthering an inclusive education.

With these premises, if curricular teachers must be usually trained to effectively use ICT besides the sole introduction of productivity tools in their usual teaching process, there are even stronger reasons and needs to suitably train teachers who are being specialized to give support to SN students to assure the providing of an inclusive education (i.e. diverse needs without making differences).

Software interfaces and languages are aspects to be carefully considered as potential barriers for SN students: this could give the impression that choosing specific software would be advisable or even compulsory: this is actually not always the case because not compliant with an idea of real inclusion that should involve all the students in the class. Similar considerations are valid for hardware and devices [12]. Because our focus is on Educational Robotics (ER), a discipline which impacts both with software and hardware issues, this point is of crucial importance.

In the last two years, we have been organizing a training course dedicated especially to learning support teachers (LSTs) with the specific aim at introducing ER as an affordable option for their future "special" work. LSTs strictly collaborate with the curricular teacher(s), therefore our main goal was, and still is, to provide them a basic competence to promote through them the diffusion of ER as a powerful ICT tool with a special attention to inclusion. The first year experience is described in detail in [13] and that first experience is the basis of the relevant improvements we introduced this year and that are the main subject of this paper.

Summarizing, the paper describes the structure of the training course, how special needs were dealt with through the proposed activities and what changes we observed in the attitude of the trainees towards robotics after the course, thanks to a comparative evaluation with the results emerged last year. Section 2 shows motivations and challenges, Sect. 3 is a detailed description of the proposed activities, Sect. 4 presents the results of the evaluation; finally, Sect. 5 exposes our conclusions.

2 Robotics in a Classroom with Special Needs

2.1 Motivations and Challenges

Working with students with special needs and promoting an inclusive education aimed at reducing gaps between students and, in any case, at providing equal opportunities of a harmonic personal development, are some of the most challenging tasks that a teacher has to face. Among ICT tools, ER has proved as one of the most promising for supporting teachers in this job [14,15]. Besides the abovementioned benefits, common to other ICT tools, ER is particularly well-suited to create the conditions for an inclusive learning environment. Among the others, we could emphasize these relevant aspects [16–19]:

– improve social interactions and cooperative learning experiences;
– provide an inclusive and flexible learning environment, suited to bring out diverse types of skills (for instance giving relevance to the pupil's role as a constructor rather than a programmer);
– promote a self-built learning, consistent with constructivist and constructionist theories;
– transmit some complex and abstract concepts through practical situations and hand-on experiences.

These qualities, which has been often validated in usual classes, express their full potential when the teaching/learning process faces SN issues, provided the collaboration between the curricular and LS teachers adapts the proposed curricula to such needs.

Unfortunately, even if robotics has been proved to be such a useful teaching tool, it remains a subject not yet widely known and seldom adopted in schools. In part this is due to the teachers' lack of expertise in ICT and to the misconception that such activities are too complex for their personal competences [20]. In [6] the European Agency for Development in Special Needs Education, highlighting the necessity to train teachers in the use of ICT, states it is not reasonable to expect that teachers can effectively integrate these type of tools with traditional learning approaches if they don't receive any initial support from other specialists. Moreover, the European Agency underlines that the full potential of ICT in the learning process is achieved proportionally to their degree of conviction that these tools are really useful.

2.2 Keypoints of the Training Course

The proposed training course is part of a one-year course provided by the School of Human and Social Sciences of the University of Padova and offered to temporarily employed and not yet employed teachers who want to get a specific qualification as LSTs [21]. This full year training aims at presenting a wide spectrum of SN-oriented methodologies and tools including Special pedagogy, Integrated models, Developmental psychology, School legislation, Labs on Motion and Dance, just to mention some. Our course is essentially an introductory lab about ER. Trainees last year reported essentially a positive feedback to our course but one limitation, highlighted by most of them, was the amount of time we dedicated to every class (just one session of 4 h). For this reason we were asked this year to double the number of sessions per group, enlarging the total number of hours to eight. We had a total of about 200 participants: 196 of them, accepted to provide evaluation data at the end of their couple of sessions.

The first positive effect of enlarging the amount of hours was to encourage a more active participation of each trainee, but more important it gave us room for presenting a larger variety of exemplary curricula and for asking homogeneous groups of trainees to design and discuss in a plenary form a draft of didactical unit in order to deepen SN issues. In other words, we wanted to motivate the trainees to show interest in integrating robotics in a class with the presence of SN students, for example with Autism Spectrum Disorders (ASD), attention deficit disorders (ADHD) and mild mental retardation.

In the case of a pupil with physical disabilities, it is imperative that his/her difficulties cannot prevent him/her from working together with peers in project-based learning activities. The aim is to realize a good balance between the role of the disabled student and the other members of the group. In addition, in this case the wide and diverse set of skills required for ER activities allows on the one hand to make the disabled student able to provide a fruitful contribution to the group, on the other hand to compel the group to help the disabled member, creating a positive synergy. Finally, it is very important to make the trainees aware of the possibility to easily integrate any ER project designed for disabled students into a project suitable for the entire class, while promoting the collaboration between the LST and the other teachers.

The course evaluation was done through the distribution, at the end of the group of 8 h, of a short personal questionnaire. The proposed questions were very similar to the ones on which the questionnaire prepared for the previous year was based, so making it possible a sensible and useful comparison.

3 Description of the Activities

The 8 h course was designed in order to address these main aspects:

- methodology;
- engagement and familiarization;
- exemplary experiences;
- designing of didactical units.

ER is very often associated with the constructionist methodology and we briefly introduced the strongest motivations to adopt this methodology for a fruitful introduction of robots in class [22,23]. We also suggested some propedeutic readings dealing with this methodological approach [22,24,25]. But, due to the specific focus of the course, we added some cues to tackle the effects of the presence in a group of a SN student. The general principle is not to modify the essence of the experience, but to adapt the role of this student in the group regarding his/her limitations. For example, in case of a physical disability, the student could be asked to specifically participate to the design process, to the definition of the objectives, to the evaluation through visual observations. A student with ADHD could be more profitably engaged in manual constructions. Exploiting the potential inherent to the team synergy proves as the best way to promote a really inclusive ER.

Focusing on the initial engagement of the trainees it is crucial to allure the trainees with a “first taste” of ER. Most of the trainees, who come from very different disciplines and levels, are very skeptical (especially the ones teaching non-scientific disciplines) and they perceive the use of robots in the class as something strange and new, not clearly motivated and difficult to implement. It is not advisable to present the full set of commands beforehand, but the trainees should gradually be exposed to the most useful and frequent ones through the suggested experiences.

For this reason, we always start with a simple and immediately rewarding experience, the “line follower” (in its simplest form) which allows to introduce the most important programming blocks. This implementation requires just 4 commands (one loop, one switch, and two alternative “move steering” blocks). The possibility for the trainees to implement in a few minutes a correct solution and to observe the “intelligent” behavior of the robot, only tuning the two involved parameters (speed and steering factor), usually produces a strong emotional engagement and a positive effect on the degree of acceptability of the overall proposal. The other offered experiences were:

– straight motion: make the robot move for a certain distance and stop for a while, and then repeat the same actions for 4–5 times; this example offers the opportunity to reason about the relationship between rotational and straight line motions;
– polygon: make the robot “paint” a regular polygon (equilateral triangle, square, pentagon, hexagon etc.); this highlights the problem to precisely control the turning of the robot;
– obstacle avoidance (Fig. 1): by using the ultrasonic sensor, it is possible to sense the presence of an obstacle on the way and to overtake it trying to realign the robot on the same original straight line;
– stop and go: again the ultrasonic sensor makes the robot stop when the obstacle is close enough, and move forward again when the obstacle is moved beyond a certain (greater) distance; if you put more robots one behind the other, all of them programmed for this same experience, the complex of robots seems to move as a multistage vehicle.

In spite of the limited available time (8 h), we successfully showed that the ER can be used to deal with multidisciplinary themes in an amusing and team-oriented perspective.

Part of the practical lab was designed according to the school level. More specifically, with training kindergarten and primary school teachers, we suggested, and also showed through a few simple examples, the use of authoring environments like Scratch to provide a first robotic-like experience to very young

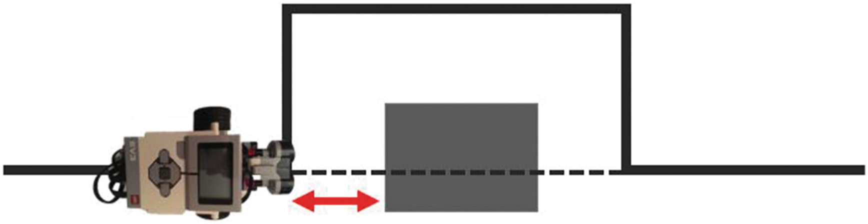


Fig. 1. Obstacle avoidance experience

kids. We have also to consider that very simple floor turtle robots like bee/blue-bot were specifically designed for these school levels. Using effectively such robots means first of all to integrate the role of the machine within a multidisciplinary 'story-told' scenario, possibly related to real life, in order to stimulate discussions, reflections, research and teamwork [26].

The last part of the course for each class was dedicated to briefly develop a multidisciplinary didactical unit. We divided the class into 5–6 groups of 4–6 people each, asking every group to find a main theme and to design the unit around this theme, imagining that in the class group of students a SN student is present. The development of a didactical unit is the moment when making the trainees more deeply aware of the inclusiveness of ER through a synthesis of the presented ideas. The points we suggested to develop were:

1. possible focused discipline;
2. theme;
3. preparatory activities (to set up the scenario);
4. role of the robot;
5. didactical objectives and expected new skills;
6. learning support issues.

Each group had the possibility to briefly present their developed unity, emphasizing what they thought to be critical aspects and new potentialities. From these presentations it came out a solicitation for further deepening the relationship between ER and SN and how the wide spectrum of applicability of ER assures to design diverse and "personalized" scenarios. Moreover it was argued that pupils with an apparent cognitive disease may reveal unexpected capabilities in terms of problem solving, personal initiative and constructive manipulation abilities. Another aspect which should never be minimized is the promotion of interactions and social skills, even in the case of severe diseases. For example, pupils with autistic spectrum disorders (ASD) may manifest a higher degree of concentration, communication and social skills during robotic-enhanced activities [27,28].

4 Evaluation of the Training Course

4.1 Instruments and Procedures

The same short questionnaire of the last year [13] was administered to the participants of this year at the end of the classes. No questionnaire was administered before attending the class, as the course was so short that we expected participants would have remembered their own answers for the first questionnaire when filling in the second one (i.e. "recency effect"). The questionnaire was anonymous and it was divided into 4 sections (Table 1).

Table 1. Description of the questionnaire

What we evaluated	How we evaluated it	Examples of items/questions
Teachers' perception of robotics	Section 1: a semantic differential including 12 bipolar pairs of adjectives to measure the participants' perception of robotics; the respondent was asked to choose where her/his position lies, on a 5-point scale between two bipolar adjectives [29];	bad-good, difficult-easy, passive-active, cold-warm.
Teachers' attitude toward ER	Section 2: 16 items on attitudes toward robotics in education were rated on a 5-point Likert scale [30,31]	I think that robotics can be a valid didactical tool; I think that robotics can foster autonomy in learning process
Course evaluation	Section 3: The positive and critical issues of the course were assessed via 5 open-ended questions	In your opinion, what are the most interesting aspects of the course? (e.g. the contents, the approach, the learning environment etc.); Do you have any idea about how to use robotics in class?

The last section contains socio-demographic items such as gender, age group and school level of the institutes where participants were currently teaching, role held by the teacher and time spent on that role.

4.2 Participants

The study involved 196 participants, 154 females (females = 83.7%; missing = 12). The prevalence of women reflects the usual predominance of female teachers, particularly in kindergarten and primary school in the Italian education

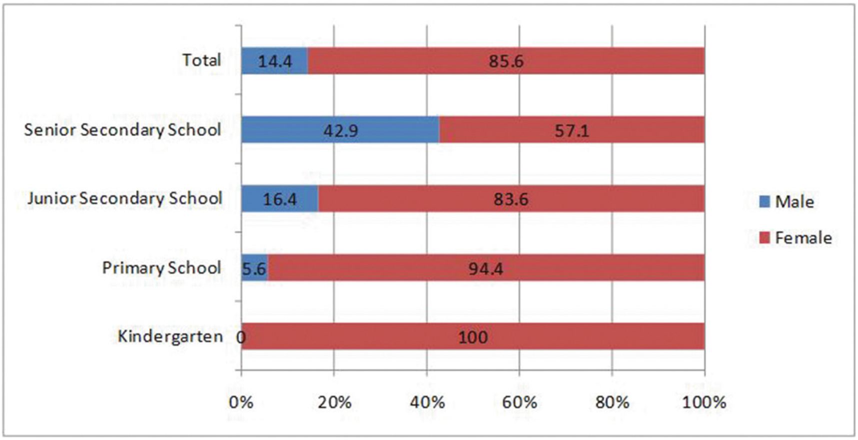


Fig. 2. Teachers by gender and school level

system. Almost 39.1% is less than 35 years old, 48.4% is from 36 up to 45, and 12.5% is more than 46. Twenty-one participants (10.7%) teach at kindergarten, 72 (36.7%) in a primary school, 58 (29.4%) in a junior high school, and 33 (16.8%) in a high school one (missing = 12) (Fig. 2). Eighty-seven of them are LSTs (44.2%) and the remaining are teachers with other specializations (e.g.: music, mathematics, foreign languages, literature, and others; missing = 12). Almost 50.3% of them has been teaching for at least 9 years (ranging from 1 up to 22 years; M = 8.45; Mdn = 9; SD = 4.12; missing = 9). The research was compliant with the Code of Ethics of the Italian Psychology Association (AIP, 2012).

4.3 Data Analysis

First, we present the data on the course evaluation and then the data on the evaluation of attitude towards robotics and attitude towards ER.

Course Evaluation. The answers to the open-ended questions on the positive and negative aspects of the course were content analysed. Participants' answers were coded according to the meaning into superordinate categories by trained judges (SDB, MP). Most frequent categories are reported in the following. As for the positive aspects, participants named:

1. the didactic approach characterised by practice: participants appreciated the chance to develop practical skills, to practice over a concrete exercise, to use robots and software by themselves; this is the most frequent category as 104 excerpts were coded into this category; e.g. *"Innovative lesson content and*

the possibility to practice immediately after the theoretical part" (participant #14, woman, junior secondary school teacher, less than 35 years old);
2. the innovative course content: participants named the theoretical content of the course dealing with ER and underlined the novelty of the course; this is the second most frequent category with 62 excerpts; e.g. *"... the introduction of a discipline that I have never considered before, especially with its practical side during classes"* (participant #15, woman, kindergarten teacher, less than 35 years old);
3. the participants' involvement into the course: as this course has a practical orientation, participants pointed out that it was intellectually and emotionally stimulating, and attending the course was fun; this category was named 27 times; e.g. *"A different way to approach the discipline, that makes it more stimulating"* (participant #28, woman, primary school teacher, less than 35 years old);
4. team working and cooperation: participants appreciated the possibility to work in groups and to exploit a cooperative learning approach among teachers; e.g. *"Team working has allowed the co-construction of the knowledge, by trying what could be introduced to school students"* (participant #29, senior secondary school teacher, woman, between 46 and 55 years old).

Regarding the negative aspects of the course, participants reported:

1. lack of time: participants complained about the small amount of hours allotted to this course; they needed more time to practice; this is the most frequent category with 66 excerpts; e.g. *"(the course is) too short. Given the modernity of these technologies, lesson hours should be more."* (participant #128, woman, junior secondary school teacher, 36 and 45 years old);
2. the complexity of the course contents: participants pointed out that the course was difficult in terms of proposed activities and concepts; this is the second most frequent category with 28 excerpts; e.g. *"completely new concepts, based on non-consolidated knowledge, to be learned in a small amount of time"* (participant #126, woman, junior secondary school teacher, between 36 and 45 years old);
3. lesson rooms: participants claimed that the lesson rooms were not fully equipped for ER lessons; participants named this category 25 times; e.g. *"The room was too big; we had few robots at disposal"* (participant #110, woman, junior secondary school teacher; between 46 and 55 years old).

Participants suggested some improvements in the same areas they identified as problematic. For instance, participants suggested to increase lesson hours to practice more with robots, and to locate this course at the beginning of school year in September, to better plan their laboratory activities. When answering the question *"Would you recommend this course to a colleague of yours?"*, 73.8% of participants replied YES (adding answers 4 and 5), showing an overall general positive evaluation of the course. When comparing the data of 2015 and 2016, we observed an improvement in the evaluation of the course (Fig. 3). Independent-sample t-test showed that 2016 participants had a more positive

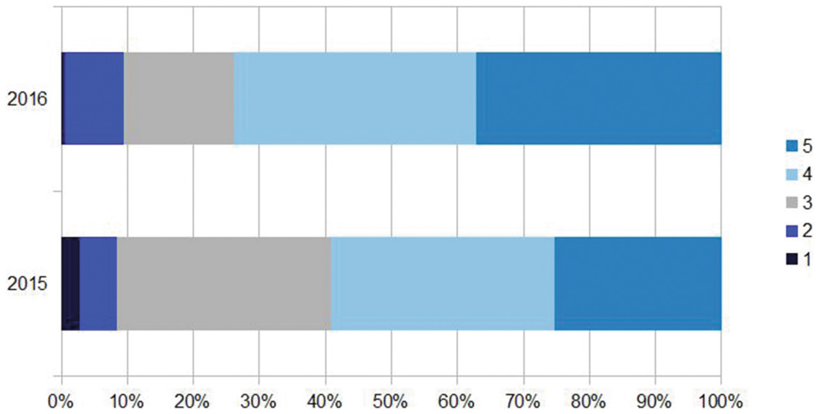


Fig. 3. Distribution of the answers to the question "Would you recommend this course to a colleague of yours?" Participants were asked to respond on a 5-points likert scale (1 = no, 5 = yes).

evaluation of the course than 2015 participants ($t(260) = -2.04$; $p = .04$; $M_{2015} = 3.73$; $M_{2016} = 4.01$). Participants appreciated more the 2016 course, where lesson hours increased from four to eight.

Attitude Towards Robotics: Semantic Differential. The semantic differential consists of a set of couple of opposite adjectives, and the respondent was asked to mark a position closer to the adjective that he/she perceives as more suitable to describe the stimulus word. One-sample T-test was performed on each couple of adjectives to test if the mean was significantly different than the average point of the response scale (= 3) (Fig. 4). After attending the course, participants considered robotics as good ($M = 4.06$; $t(190) = 19.18$, $p < .001$), beautiful ($M = 3.97$; $t(187) = 16.06$, $p < .001$), attractive ($M = 3.79$; $t(184) = 12.85$, $p < .001$), pleasant ($M = 3.91$; $t(191) = 12.55$, $p < .001$) and strong ($M = 3.63$; $t(185) = 10.26$, $p < .001$). Moreover, participants perceived robotics as warm ($M = 3.32$; $t(187) = 4.66$, $p < .001$), fast ($M = 3.48$; $t(189) = 7.61$, $p < .001$), active ($M = 4.11$; $t(190) = 18.36$, $p < .001$) and trustworthy ($M = 3.88$; $t(189) = 14.83$, $p < .001$). Eventually, robotics was perceived as difficult ($M = 2.86$; $t(188) = -2.06$, $p = .04$). No significant differences according to socio-demographic items were observed.

Attitude Towards Educational Robotics. One-sample t-test revealed that teachers agreed with the statement that (1) robotics can enhance students' motivation to learning ($M = 4.13$; $t(194) = 19.60$; $p < .001$), (2) group works with robots can improve students' social competences ($M = 4.07$; $t(194) = 16.39$; $p < .001$), (3) educational robotics sustains a new point of view on science for teachers ($M = 3.74$; $t(193) = 10.34$; $p < .001$) Fig. 5 shows the percentages of

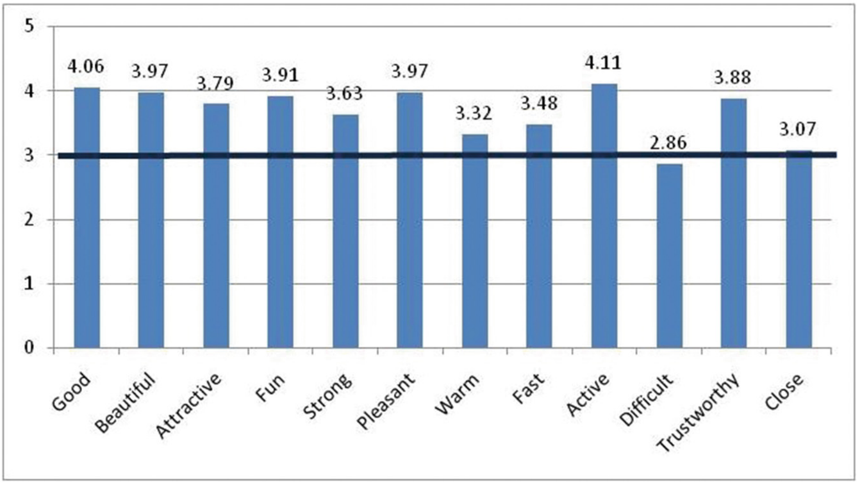


Fig. 4. Means for each couple of adjectives

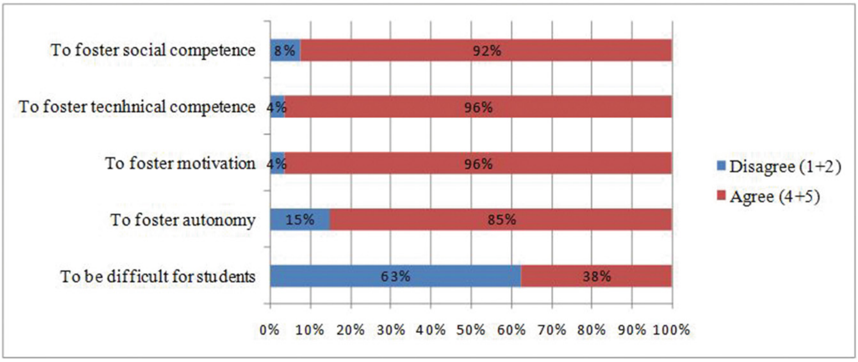


Fig. 5. Percentages of responses to the attitude items

responses for the agreement (answers = 4 and 5) and disagreement (answers = 1 and 2) to some items on attitude towards robotics. Response 3 (i.e. middle point of the response scale) was deleted from the analysis to make the data more readable. The majority of participants showed a positive attitude toward robotics, after attending the class. Moreover, a minority of teachers (38%) agreed that educational robotics could be difficult for students.

Moreover, a t-test for independent sample was run to test the influence of teachers' gender and of years of teaching experience on the attitude toward educational robotics. As for gender, male teachers agreed to the statement that ER is difficult for teachers more than female teachers ($M_M = 3.88$ vs $M_F = 3.25$; t(115) = 1.88; p = .004). Male teachers agreed to the statement that ER can distract attention from lesson themes more than female teachers ($M_M = 2.53$ vs

M_F = 1.77; t(153) = 3.09; p = .002). As for teaching experience, experienced teachers (teaching for more than 11 years; n = 43) agreed that ER can be a valid didactic tool more than less experienced teachers (teaching for less than 5 years; n = 51) (M_E = 3.91 vs M_{LE} = 3.56; t(106) = -1.986; p = .05). No other significant differences according to socio-demographic items were observed.

When comparing 2015 and 2016 answers, 2016 participants were less optimistic than 2015 ones as the possibility that robotics could foster students' autonomy (t(386) = 2.364; p = .002; M_{2015} = 4.33; M_{2016} = 4.13). As stated before, one possible explanation for this results refers to a less ideal view on robotics that participants could build after the 2016 longer course.

How to Use Robotics in Class. One open-ended question asked participants whether they had already an idea about how to use robotics in class, in which subject, for which pupils and to reach which aims. Approximately 69.9% of participants declared they had already an idea on how to integrate robotics in class (significantly more than the previous year (61%)). As for disabilities, participants named Autism Spectrum Disorders (ASD), attention deficit disorders (ADHD), learning disabilities, mild mental retardation, among others as needs to be properly addressed with robotics in class. For instance, two kindergarten teachers recognized that robotics could be used to improve learning simple didactical unit on math. As for disabilities, both teachers mentioned Autism Spectrum Disorders (ASD).

"I would use it with children with ASD, for the learning of simple didactical units on maths" (participant #7, women, less than 35 years, LST, 3-year experienced).

As for primary school teachers, they mentioned geography, geometry, maths and science as possible subjects where to employ robotics. Twelve of them (22.2%) mentioned ASD, ADHD and learning disabilities:

"... activity to be implemented with pupils with deficit of attention, with Down syndrome" (participant #28, between 46 and 55 years old, 10-year experienced).

As for junior secondary school teachers, twelve of them (28.6%) mentioned subjects such as technology, maths and informatics:

"Technology subject, for all the pupils, for pupils with ADHD, to work on attention, socialization and motivation" (participant #98, women, between 36 and 45 years old, LST, 10-year experienced).

As for secondary junior school, seven teachers (28%) mentioned maths, informatics, and geography:

"As I am a maths teacher, I would plan some multidisciplinary activities on informatics and maths, involving mild mentally retarded pupils, to learn having fun and to support their autonomy" (participant #177, women, up to 35 years, 7-year experienced).

5 Evaluation Summary and Conclusions

The outcoming results showed a general improved acceptability of ER, both as a feeling and as an analytic judgment. Not surprisingly ER was still perceived as difficult for the trainees but this is something that can be smoothed with a real activity in class and implementing a constructionist teaching/learning process during it. They also transferred this perception to the students' side according to their general experience. Nonetheless most trainees agreed on considering ER a powerful tool for promoting all the relevant skills showed in Fig. 5 also for SN students. Our personal conviction is that the new structure of the course this year amended some limitations observed during the last year course and more specifically increased the trainees' perception of being able to make a good implementation of robotics in future classes.

Acknowledgement. We thank the director of the Specialization course for teaching support of disabilities for the opportunity of the ER lab. This work was partly supported by the project: ERASM: Educational robotics as a validated mindtool: methodology, platforms, and an experimental protocol, code: CPDA145094, funded by the University of Padova.

References

1. Oliver, R.: The role of ICT in higher education for the 21st century: ICT as a change agent for education. In: Proceedings of the Higher Education for the 21st Century Conference, Curtin (2003)
2. Voogt, J.: Consequences of ICT for aims, contents, processes, and environments of learning. In: van den Akker, J., et al. (eds.) Curriculum Landscapes and Trends, pp. 217–236. Springer, Netherlands (2004)
3. Avvisati, F., Hennessy, S., Kozma, R.B., Vincent-Lancrin, S.: Review of the Italian strategy for Digital Schools. OECD Education, Working Papers 90. OECD Publishing (2013)
4. Istenic Starcic, A., Bagon, S.: ICT supported learning for inclusion of people with special needs: review of seven educational technology journals, 1970–2011. Br. J. Educ. Technol. **45**(2), 202–230 (2014)
5. Brodin, J., Lindstrand, P.: What about ICT in special education? Special educators evaluate information and communication technology as a learning tool. Eur. J. Spec. Needs Educ. **18**(1), 71–87 (2003)
6. Watkins, A.: Information and communication technology (ICT) in special needs education (SNE). European Agency for Development in Special Needs Education (2001)
7. Course, Specialized Training: ICTs in Education for People with Special Needs. UNESCO Institute for Information Technologies in Education (IITE) (2006)
8. Williams, P., Jamali, H.R., David, N.: Using ICT with people with special education needs: what the literature tells us. Aslib Proc. **58**(4), 330–345 (2006). Emerald Group Publishing Limited
9. Florian, L., Hegarty, J.: ICT and Special Educational Needs: A Tool for Inclusion. McGraw-Hill Education, UK (2004)
10. Valtonen, T., Hacklin, S., Kontkanen, S., Hartikainen-Ahia, A., Kärkkäinen, S., Kukkonen, J.: Pre-service teachers' experiences of using social software applications for collaborative inquiry. Comput. Educ. **69**, 85–95 (2013)
11. Singleton, C.: Using computer-based assessment to identify learning problems. In: ICT and Special Educational Needs: A Tool for Inclusion, pp. 46–63 (2004)

12. Bittencourt, I.I., Baranauskas, M.C., Dermeval, D., Braga, J., Pereira, R.: Designing for different users and multiple devices: a roadmap towards inclusive environments. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013, Part I. LNCS, vol. 8117, pp. 605–622. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40483-2_43
13. Di Battista, S., Menegatti, E., Moro, M., Pivetti, M.: Introducing educational robotics through a short lab in the training of future support teachers. In: 6th International Conference on Robotics in Education, RiE 2015. Yverdon-les-Bains, Switzerland (2015)
14. Eguchi, A.: Educational robotics for promoting 21st century skills. J. Autom. Mob. Robot. Intell. Syst. **8**(1), 5–11 (2014)
15. Karna-Lin, E., Pihlainen-Bednarik, K., Sutinen, E., Virnes, M.: Can robots teach? Preliminary results on educational robotics in special education. In: Sixth IEEE International Conference on Advanced Learning Technologies (ICALT 2006), pp. 319–321. IEEE (2006)
16. Virnes, M., Sutinen, E., Krn-Lin, E.: How children's individual needs challenge the design of educational robotics. In: Proceedings of the 7th International Conference on Interaction Design and Children, pp. 274–281. ACM (2008)
17. Alimisis, D., Arlegui, J., Fava, N., Frangou, S., Ionita, S., Menegatti, E., Monfalcon, S., Moro, M., Papanikolaou, K., Pina, A.: Introducing robotics to teachers and schools: experiences from the TERECoP project. In: Clayson, J., Kalas, I. (eds.) Proceedings for Constructionism, Paris, France, pp. 16–20 (2010)
18. Mikropoulos, T.A., Bellou, I.: Educational robotics as mindtools. Themes Sci. Technol. Educ. **6**(1), 5–14 (2013)
19. Demo, G.B., Moro, M., Pina, A., Arlegui, A.: In and out of the school activities implementing IBSE and constructionist learning methodologies by means of robotics. In: Robots in K-12 Education: A New Technology for Learning, pp. 66–92 (2012)
20. Tondeur, J., van Braak, J., Sang, G., Voogt, J., Fisser, P., Ottenbreit-Leftwich, A.: Preparing pre-service teachers to integrate technology in education: a synthesis of qualitative evidence. Comput. Educ. **59**(1), 134–144 (2012)
21. http://www.unipd.it/corsi/formazione-iniziale-insegnanti/corsi-specializzazione-sostegno-didattico-alunni-disabilita. (in Italian)
22. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013)
23. Urschitz, T., Moro, M.: Low threshold: how inclusive is Constructionism? In: Futschek, G., Kynigos, C. (eds.) Proceedings of the International Conference on Constructionism 2014, Wien, pp. 299–308 (2014)
24. Moro, M., Menegatti, E., Sella, F., Perona, M.: Imparare con la robotica. Applicazione di problem solving. Edizioni Centro Studi Erickson, Trento (2011)
25. Alimisis, D., et al.: Teacher education on robotics-enhanced constructivist pedagogical method. School of pedagogical and technological education, Athens (2009)
26. Cacco, L., Moro, M.: When a bee meets a sunflower. In: Proceedings of 4th International Workshop Teaching Robotics Teaching with Robotics and 5th International Conference on Robotics in Education, Padova, Italy, pp. 68–75 (2014)
27. Silva, S., Soares, F., Pereira, A.P., Costa, S., Moreira, F.: Robotic tool to improve skills in children with ASD: a preliminary study. Int. J. Life Sci. Med. Res. **3**(4), 162 (2013)
28. Albo-Canals, J., Heerink, M., Diaz, M., Padillo, V., Maristany, M., Barco, A., et al.: Comparing two LEGO robotics-based interventions for social skills training with children with ASD. In: 2013 IEEE RO-MAN, pp. 638–643. IEEE (2013)

29. Osgood, C.E., Suci, G.J., Tannenbaum, P.H.: The Measurement of Meaning. University of Illinois Press, Urbana (1957)
30. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012)
31. Mama, M., Hennessy, S.: Developing a typology of teacher beliefs and practices concerning classroom use of ICT. Comput. Educ. **68**, 380–387 (2013)

A Didactical Model for Educational Robotics Activities: A Study on Improving Skills Through Strong or Minimal Guidance

Soumela Atmatzidou(✉) and Stavros Demetriadis

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{atmatzid,sdemetri}@csd.auth.gr

Abstract. This paper introduces a proposed didactical model for organizing educational robotics activities, addressed to primary and secondary school, and called "CPG+" after Collaboration, Problem, Game - competition, while "+" stands for supplementary teachers' supportive interventions such as promoting students' problem solving and computational thinking skills. Then, a study conducted in an elementary school is presented, which was based on the CPG+ model and investigates the development of computational thinking and problem solving skills, focusing on the role of guidance (strong vs. minimal). For developing computational thinking skills we focused on the following basic concepts: abstraction, generalization, algorithm, decomposition, modularity, and debugging. The results show that: (a) educational robotics activities fringed by the didactical model CPG+ can be a vehicle for the development of high order skills, (b) although providing written answers is tiring and boring for students, it is an important learning tool.

Keywords: Computational thinking · Problem solving · Guidance · Prompting · Educational robotics

1 Introduction

Educational Robotics (ER) is an excellent teaching and learning tool with the potential to support the educational process at all levels. Over the last years, researchers have turned their attention on the role of ER in the development of students' cognitive and social skills [1]. Especially increased is the interest for the contribution of robotics in the development of problem solving and computational thinking skills [2, 3], which have been recognized as fundamental for all students, and as highly important for controlling and managing cognitive activities in all disciplines [4]. However, certain researchers point out that ER alone cannot affect and act directly on students' learning. Therefore, they highlight that an appropriate educational philosophy, learning environment and teaching methodology is needed for the successful incorporation of educational robotics in school [5–7]. An important issue pertaining to the teaching methodology and attracting the attention of researchers is the impact of prompting strategies on the students' effective engagement in the deeper processing of the learning material. It is also noted that more research is needed regarding

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_5

the benefits of ER in the development of specific skills, such as computational thinking and problem solving [6].

This work introduces a didactical model that aims to create an effective, motivating, collaborative and entertaining learning environment for robotics activities, and also focuses on students' development skills. This model is called "CPG+" after Collaboration, Problem, and Game - competition, while "+" stands for the supplementary teachers' supportive interventions for the development of higher order skills. Next, we describe the implementation of ER activities in an Elementary school, the focus of which is on promoting students' problem solving (based on Polya's steps) and computational thinking skills (based on a previous study of ours [8]). Further investigating the prompting strategies, we divide students in two groups both of which accomplish the activities with a step-by-step guidance of worksheets: (a) one group (''strong guidance) is prompted to provide written answers, while (b) the other group ("minimal guidance") receives no prompting to answer in writing.

In the following, we present: (a) the theoretical background of our work, (b) a proposed didactical model for organizing educational robotics activities, addressed to elementary and secondary school, and (c) a pilot study conducted in an elementary school that focuses on the development of computational thinking and problem solving skills, and contrasts the impact of two different intervention/prompting strategies. The results provide encouraging evidence regarding the CPG+ model in the ER activities and argue for the positive impact on the development of students' problem solving and computational thinking skills.

2 Theoretical Background

Educational Robotics (ER) is considered to be a powerful, flexible and innovative learning tool offering opportunities for the construction and control of a robot [5]. ER activities motivate and encourage students to solve authentic problems that are meaningful to them, providing them with the opportunity to directly and visually see the outcome of their solution. Researchers argue that a guided instructional approach with robots facilitates teamwork, develops conceptual understanding, enhances critical thinking, and promotes higher-order learning in the domains of science [9, 10]. Many studies also indicate that robotics can be used as a tool that offers opportunities for students to engage in and develop problem solving and computational thinking skills (e.g. [11, 12]).

The relevant literature provides various ER scenaria, some of which are technocentric, while others focus on creation and exploration, aiming to promote science and programming. One methodological model of development of interdisciplinary projects in ER has been proposed by the TERECoP (Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods) research team and consists of the following development stages: engagement, exploration, investigation, creation and composition, and finally evaluation. This model establishes an environment which promotes self-learning, mutual support of the members of the class, free expression and creativity [13]. A different didactic approach is presented by Gaudiello et al. [14], whereby students acquire new knowledge, and enhance new problem-solving skills with the guidance of the nine-phase framed procedure: assessment, familiarization, building, bridge, alphabetization, explorative, working, goal, report.

Despite the fact that ER enjoys a rich theoretical background, little research has been conducted so far on defining either the appropriate pedagogical methods relating to learning or the educational approaches that need to be used in order that ER be properly integrated in the classroom. Therefore, further research is necessary if we are to establish good teaching practices, techniques and methods which will allow the students to develop specific skills in ER activities [1, 15, 16].

2.1 Computational Thinking

Computational thinking (CT) is a fundamental skill which promotes new ways of thinking to the students across all disciplines of science. Wing (2006, p. 33) described CT as a way of "solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science" [4]. The concepts that are widely accepted as concepts of CT skills are: computation, communication, coordination, recollection, automation, evaluation, design, algorithm building, conditional logic, debugging, simulation, working effectively in teams and analyzing problems [4, 17]. Bottino and Chioccariello [18] advocate that "learners should practice computational thinking in playful contexts where they can develop personal projects, for example building, share and discuss their construction with others". Many studies have focused on the environment of robots as an appropriate tool for the development of CT. In recent years, researchers have been investigating the role of educational robotics in fostering the development of students' CT skills [11, 19–21]. Some of them relate that children, when programming robots, learn and apply core concepts such as abstraction, automation, analysis, decomposition, generalization, modularization, and iterative design [11, 19, 20]. More specifically, studies demonstrate that children from 4 to 6 years old, when building and programming simple robotics projects, understood basic programming and became acquainted with concepts such as sequencing and choosing the correct instructions [19, 20]. Other studies, engaging children aged from 10 to 17 and exploring the development of CT skills in robotics activities, reported that there is over all a very positive effect [21, 22], and also that robotics is a helpful tool for young students, "*facilitating a more abstract understanding*" [23]. Finally, a study conducted by Penmetcha [24] investigated the relationship between robotics and learning about programming and algorithmic thinking in university students, and showed that robotics can be used to teach concepts such as designing, programming and testing on a more abstract level.

Despite these efforts, more in-depth analysis is needed if we wish to study: the development of CT skills [8], and their integration into the curriculum and classroom environment [25]; the strategies needed for assessing the development of CT; and also the role of ER as a tool in developing CT.

2.2 Strong vs. Minimal Teacher Guidance

The role and impact of teacher guidance during learning activities has received considerable attention in recent years in the literature. On one side are researchers who believe that students learn best when they discover or construct their knowledge by themselves, as well as those who mention that the process of solving challenging and authentic problems allows for a flexible adaptation of the guidance, without the need for more explicit guidance [26].

On the other side, there are researchers who argue that students learn best when provided with full, explicit instructional guidance [27], and those who claim that prompting students to provide written answers "triggers additional cognitive activity that results to better learning outcomes" [28, 29]. These differences create the need for further investigation that would determine the appropriate guidance for learners.

2.3 Research Questions

Considering the above, we focus on the following research questions:

1. When educational robotics activities are based on didactical model CPG+, do they support Computational Thinking and Problem Solving (PS) skills efficiently?
2. Do the minimal and strong guidance have the same impact on the CT and PS skills in ER activities?

3 Proposed Didactical Model CPG+

The proposed didactical model CPG+ is based on two axes: (a) the pedagogical character of the activity, in which case it relates to the interaction between students as well as between students and educational material, and (b) the temporal organization of the activity.

3.1 Pedagogical Character of the ER Seminar

An effective activity of educational robotics focuses not only on the deeper understanding of the learning object, but on the development of general horizontal skills as well. The model suggests that ER activities have to be based on Collaboration, Problem and Game-competition as well as on appropriate supportive interventions by the teacher with the aim of developing the students' higher order skills. In particular:

- **Collaboration:** Students work in small groups (2–4 members) assuming procedural or cognitive roles and implementing appropriate collaboration scripts. Role assigning offers students the opportunity to interact as peers and equals, and creates favourable learning conditions. Moreover, the implementation of appropriate collaboration scripts (e.g. jigsaw, send a problem, think aloud pair problem solving (TAPPS) etc.) involves in the process all students equally, favours the effective communication between the members of the team, offers them the opportunity to present their ideas and arguments as well as to develop learning and collaboration skills, while creating at the same time conditions of assuming personal responsibility in the context of the team effort.
- **Problem:** The activity is based on realistic-authentic problems of advancing difficulty which attract the students' attention. The students combine their pre-existing knowledge with new knowledge in order to solve the problem and choose appropriate strategies which focus on analysis, synthesis, evaluation, reflection etc. Incorporating these strategies helps students improve their skills in problem solving and develop their critical thought.

- **Game-Competition:** The activity involves a game perspective in order to further attract the students' attention and willingness to get involved and learn. In order for the students to see the ER activity as a pleasant game, activities of a playful character are chosen (e.g. let's make a doll which cries when we put her down and stops crying when we take her in our arms, the bumper cars in a theme-park, etc.) and there is no reference to "lessons", or "exercises" and "problems", but to "training" and "activities" respectively. The game atmosphere creates motivation for participation and socialization in students, cultivates the appropriate conditions for interaction and therefore activates cognitive processes. Moreover, the prospect of the teams' participating in a final challenge creates a sense of competition between the teams. Competition motivates the teams to set goals and implement strategies in order to achieve victory, helps maintain a strong interest for the team effort, while giving at the same time the opportunity for application of the principles of fair play. Finally,
- **"+":** The symbol refers to the development of skills such as: problem solving, metacognition, computational thinking, collaboration etc. The teacher supports the development of these skills with appropriate interventions during the activity. One way to achieve this is the three-step technique, whereby:
  (a) **Modelling of the skill** based on the relevant literature, we have to determine what the student should be able to do while developing this skill as well as the steps comprising this particular skill. For example, in the cases of abstraction: the student has to be able to distinguish the same underlying programming pattern and the common behaviour in problems with apparently different characteristics and to tell the important from the redundant information.
  (b) **Analytical guidance** for the development of the skill in the initial lessons, planning and assigning roles to the students, which alternate from one activity to the next. The worksheets provide detailed guidance, defining the steps and the appropriate prompts to be followed for the development of this particular skill (skill scriptlet).
  (c) **Fade out** of strict guidance, supporting the skill with role assignment, providing no detailed guidelines in the worksheet, and setting reflection questions which incite students to think about the application of the skill. During the process, the teacher observes – checks whether the students internalize the role/skill and, if students have difficulty responding, (s)he intervenes supportively and offers the appropriate feedback.

Particularly important is the role of the trainer-teacher who directs the learning process, creates motivation and guides the students so that they can acquire skills and enhance their knowledge. (S)he provides immediate feedback, discusses with the students and encourages them to work and express their views and questions. (S)he intervenes as a facilitator and advisor, regulating the process with suggestions and comments, such as: (a) Why is this happening? or Why is this not happening? (b) Explain how …? (c) What will happen if …? (d) How is … influenced …? (e) What is the basic idea …?

At the same time, the trainer-teacher underlines the importance of teamwork and collaboration, reinforces healthy competitiveness between students, approaches the problems arising between the members of the teams and uses them in order to give feedback to the class, thus facilitating the collaboration between students. Finally, (s)he ensures that they understand that eventually everybody is a winner when participating actively in the aforementioned educational process.

3.2 Temporal Organization of the ER Seminar

An ER seminar is organized into the phases presented in Table 1:

Table 1. The organization of the ER seminar.

Phase	Session/Duration	Activity
Introductory	1 session 2 h	Acquaintance with the robots Introduction to the programming environment Organization of the class
Training	7–12 sessions 2 h per session	With the robots, the students implement authentic problems of graduated difficulty
Challenge	1 session 3-hour competition	Each group gets prepared for the challenge The challenge starts

In the Introductory phase, the robot and its programming environment are introduced. Then, the class is divided into small groups (2–4 students) and the students are informed about the roles they are to assume in each activity; the roles are to alternate so that the students become familiar with all the roles. Then, the goal of winning in a final challenge-test ("competition") is set and we emphasize that in order that victory be achieved the students have to acquire sufficient knowledge of robot programming.

In the Training phase worksheets are given which follow appropriate protocols of questions and actions and guide the students through the implementation of authentic problems of advancing difficulty. The worksheets also define: (a) the script of collaboration by ascribing roles to the students (Analyst, Algorithm Builder, Programmer, Debugger -Evaluator), (b) the way in which they are to play their role, and (c) the interaction with their collaborators within their group as well as in the other groups. During the sessions, and as the students begin to enhance their knowledge in programming and to develop skills, guidance is gradually faded out so that the students be able to act autonomously.

In the Challenge phase the groups are given a demanding (yet within their reach) problem, and clear rules are set for grading and appointing the winner. Upon completion we note what happened and why, analyzing the strengths and weaknesses in the solutions given to the test and in the strategies adopted by each group.

4 Study

The study is based on CPG+ model and focuses on students' development Computational thinking and Problem solving skills.

The ER seminars took place in Elementary School in Thessaloniki. In total, 77 students participated in the study (45 boys and 32 girls of 6th grade). The study was conducted in 9 sessions that lasted two hours each. The students worked in groups consisting of 3 or 2 members. For the purpose of this study we used the Lego Mindstorms NXT 2.0 educational tool.

Specifically, in the seminars were engaged the students in two conditions:

- Minimal guidance (MG) 39 students, 23 boys and 16 girls
- Strong guidance (SG) 38 students, 22 boys and 16 girls

4.1 Learning Design - Implementation – Procedure

This study is based on a model for the development of CT skills which we presented in a previous study [8]. That model focused on specific CT concepts: abstraction, generalization, algorithm, decomposition, modularity. Now, however, debugging is being added as an important concept of CT. Regarding the guidance leading to the development of PS skills, we were based on Polya's [30] methodology: by means of prompts, we guided students to understand the problem, and to plan, implement and evaluate the solution.

The students in both groups work collaboratively in groups of 3 or 4 members assuming a number of roles such as analyst, programmer and debugger, which are alternated in each activity. Students are guided by the instructions of the worksheets to accomplish authentic complex problems, in order to develop basic skills of computational thinking and problem solving. The specific differences in guidance between the two groups are as follows: (a) The group with minimal guidance follows a low prompting approach. Students are prompted during the activities to follow step-by-step the instructions in the worksheet, to verbalize their reflections orally, and to analyze and discuss their thinking about the solution process within the group; (b) The strong guidance group follows a high prompting approach. Students are prompted to follow step-by-step the instructions, to discuss their thinking on the solution process in the group and to submit their answers in writing. In both groups the instructor as facilitator and consultant provides support in the form of hints, prompts, feedback, etc. In the following, we refer to group that follows minimal guidance as "MG" and to group that follows strong guidance as "SG".

The implemented ER activities were based on CPG+ model. At the beginning, we started with an introduction to Robots and to the Lego Educational programming environments NXT-G. In every session there was given a worksheet that guided students to accomplish their activities and also to understand and assimilate the CT concepts and PS strategies. At the end of the 3rd session, we handed out an individual quiz-activity in order to investigate the students' understanding of the CT and PS skills. In the next six sessions the activities integrated CT and PS skills in complex authentic problems of graduated difficulty, such as a car that follows the rules of traffic, an alarm, a recycler, etc. After the 9th session, one more individual quiz-activity and a students' opinion questionnaire follows to record the students' views. There followed a challenge whereby all the groups were required to implement an activity in which the winning group would be the one with the best performance. Finally, with the Think-aloud protocol each student described the solution process in a given task and there followed a semi-structured interview in which they gave their opinion regarding all the activities.

4.2 Data Collection

The evaluation tools in the present study were:

- Two Quiz-activities (CT-3 & CT-9): The first Quiz was handed out after the 3rd session (CT-3), when students acquired an initial familiarization with the CT concepts, and the second Quiz after the 9th (CT-9). Students were asked to solve problems and cite how they employed CT concepts in the solution of problems (e.g. identify the concept of abstraction between various problems, propose a more general solution, etc.). The assessment of students' responses was based on a graded criterion instrument (rubric) using a 4-point Likert scale (1 = 'unsatisfactory', 2 = 'quite satisfactory', 3 = 'satisfactory', 4 = 'excellent').
- Think-aloud protocol (CT-TA): After the training, students individually were given a certain robot programming task and were asked to describe aloud the solution process that they would follow to implement it, and also were prompted to reflect on CT concepts that they employed in their solution. The assessment was based also on the same graded criterion instrument (rubric) as in the Quiz-activities measure.
- Interview: After the think-aloud activity, semi-structured interviews were conducted, with a fairly open framework, to elicit students' subjective views and opinions regarding the following key aspects of the activity: (a) understanding CT concepts and developing relevant skills, (b) improving problem solving skills, (c) benefits or possible drawbacks in-group collaboration, (d) the contribution of the guidance in the educational process, and (e) likes and dislikes relevant to the robotics activity. We used content analysis on collected data to identify the views that dominated the statements of the students.
- Observation: Systematic monitoring of the students' work was applied by taking notes on structured observation sheets.

4.3 Results

Reflecting on the initial state, the Independent Samples test was applied on students' CT-3 and PS-3 scores in Minimal Guidance (MG) and Strong Guidance (SG) groups and presented in Table 2, below:

Table 2. Comparing CT-3 and PS-3 scores between MG and SG groups

Skills	MG *(N = 39)*		SG *(N = 38)*		t-test
	M	*(SD)*	M	*(SD)*	
CT-3	2.04	*(.38)*	2.10	*(.42)*	t(75) = .626, p = .533
PS-3	2.41	*(.37)*	2.53	*(.50)*	t(69) = 1.201, p = .234

Regarding CT. The results regarding CT are as follows:

(a) Table 3 presents statistical controls applied on students' CT-3 and CT-9 scores in MG and SG groups.

Table 3. Comparing CT-9 and CT-3 scores between MG and SG groups

Level	CT-3	CT-9	Paired t-test	ANCOVA
	M *(SD)*	M *(SD)*	CT-9 compared to CT-3 (same student group)	CT-9 compared to CT-3 across student groups
MG	2.04 *(.38)*	2.41 *(.38)*	$t(38) = -8.888, p < .001$*	$F(1,75) = 12.856$ $p = .001, \eta^2 = .146$
SG	2.10 *(.42)*	2.72 *(.38)*	$t(37) = -11.239, p < .001$*	
Total	2.07 *(.40)*	2.56 *(.40)*	$t(76) = -13.327, p < .001$*	

*Significant difference at the 0.05 level

(b) Table 4 presents statistical controls applied on students' CT-3 and CT-9 scores analysed in each of the six dimensions of the CT model.

Table 4. Comparing CT-9 and CT-3 scores analytically for the six CT dimensions

Skills	Level/N	CT-3	CT-9	Paired t-test	ANCOVA
		M *(SD)*	M *(SD)*	CT-9 compared to CT-3 (same group)	Comparing CT-9 and CT-3 across student groups (CT-3 as covariate)
Abstraction	MG 39	1.95 (.63)	2.41 (.82)	$t(38) = -4.558$ $p < .001$*	$F(1,74) = 4.330$ $p = .041$ $\eta^2 = .055$
	SG 38	2.07 (.74)	2.79 (.72)	$t(37) = -5.500, p < .001$*	
Generalization	MG 39	1.86 (.55)	2.28 (.60)	$t(38) = -5.500, p < .001$*	$F(1,74) = 5.585$ $p = .021$ $\eta^2 = .070$
	SG 38	1.99 (.67)	2.65 (.66)	$t(37) = -5{,}504, p < .001$*	
Decomposition	MG 39	1.84 (.71)	2.06 (.50)	$t(38) = -2.055, p = .047$*	$F(1,74) = 8.771$ $p = .004$ $\eta^2 = .106$
	SG 38	1.73 (.66)	2.35 (.37)	$t(37) = -4.552, p < .001$*	
Modularity	MG 39	1.96 (.67)	2.49 (.79)	$t(38) = -5.144, p < .001$*	$F(1,74) = 0.036$ $p = .850$ $\eta^2 = .000$
	SG 38	2.17 (.69)	2.66 (.69)	$t(37) = -5.217, p < .001$*	
Algorithm	MG 39	2.36 (.56)	2.66 (.42)	$t(38) = -4.026, p < .001$*	$F(1,74) = 30.514$ $p < .001$* $\eta^2 = .292$
	SG 38	2.47 (.49)	3.16 (.37)	$t(37) = -7.342, p < .001$	
Debugging	MG 39	2.29 (.73)	2.58 (.73)	$t(38) = -2.176, p = .036$	$F(1,74) = 1.364$ $p = .247$ $\eta^2 = .018$
	SG 38	2.17 (.56)	2.69 (.70)	$t(37) = -5.008, p < .001$	
Total	MG 39	2.04 (.38)	2.41 (.38)	$t(38) = -8.888, p < .001$*	$F(1,74) = 18.404$ $p < .001$* $\eta^2 = .199$
	SG 38	2.10 (.42)	2.72 (.38)	$t(37) = -1.239, p < .001$	

*Significant difference at the 0.05 level

(c) Table 5 presents statistical control applied on the students' CT-TA scores (both total and analytical scores for each of the six CT dimensions).

Table 5. Comparing CT-TA scores between groups

CT skills	MG	SG	Independent t-test
	M *(SD)*	M *(SD)*	
Abstraction	2.76 (.94)	2.88 (.79)	$t(75) = .632$, $p = .529$
Generalisation	2.36 (.83)	2.73 (.65)	$t(75) = 2.161$, $p = .034$*
Decomposition	2.28 (.51)	2.61 (.45)	$t(75) = 2.937$, $p = .004$*
Algorithm	2.88 (.70)	3.31 (.51)	$t(75) = 3.088$, $p = .003$*
Modularity	2.59 (.79)	2.92 (.73)	$t(75) = 1.915$, $p = .059$
Debugging	2.69 (.94)	2.76 (.65)	$t(68) = .386$, $p = .701$
Total CT-TA	2.59 (.57)	2.87 (.44)	$t(75) = 2.360$, $p = .021$*

*Significant difference at the 0.05 level

(d) Table 6 presents statistical controls applied on students' CT-TA and CT-3 scores in groups.

Table 6. Comparing CT-TA and CT-3 scores between groups

Level/N	CT-3	CT- TA	Paired t-test	ANCOVA
	M *(SD)*	M *(SD)*	CT-TA compared to CT-3 (same student group)	Comparing CT-TA across student groups (CT-3 as covariate)
MG 39	2.04 *(.38)*	2.59 (.57)	$t(38) = -8.093$, $p < .001$*	$F(1,74) = 6.994$ $p = .010$ $\eta^2 = .086$
SG 38	2.10 *(.42)*	2.87 (.44)	$t(37) = -15.363$, $p < .001$*	
Total 77	2.07 *(.40)*	2.73 (.53)	$t(76) = -14.983$, $p < .001$*	

*Significant difference at the 0.05 level

Table 7. Comparing PS-9 and PS-3 scores between groups

Skills	Level/N	PS-3	PS-9	Paired t-test	ANCOVA
		M *(SD)*	M *(SD)*	PS-9 compared to PS-3 (same group)	Comparing PS-9 across student groups (PS-3 as covariate)
PS	MG 39	2.41 (.37)	2.69 (.41)	$t(38) = -3.833$, $p < .001$*	$F(1,74) = 23.971$ $p < .001$* $\eta^2 = .245$
	SG 38	2.53 (.50)	3.18 (.44)	$t(37) = -7.142$, $p < .001$*	
	Total 77	2.46 (.44)	2.93 (.49)	$t(76) = -7.515$, $p < .001$*	

*Significant difference at the 0.05 level

Regarding PS. The results regarding PS are as follows:

(a) Table 7 presents statistical controls applied on students' PS-3 and PS-9 scores in MG and SG groups.
(b) Table 8 presents statistical control applied on the students' PS-TA scores comparing: (i) between groups, (ii) to PS-3 in the same student group (Paired t-test) and (iii) across student groups with PS-3 as covariate.

Table 8. Comparing PS-TA between groups and PS-3 scores

Level/N	PS-3	PS-TA	Independent t-test	Paired t-test	ANCOVA
	M *(SD)*	M *(SD)*		PS-TA compared to PS-3	Comparing PS-TA across student groups
MG 39	2.41 (.37)	2.86 (.38)	$t(75) = 4.915$ $p < .001$*	$t(38) = -5.730$, $p < .001$*	$F(1,74) = 21.972$ $p < .001$* $\eta^2 = .229$
SG 38	2.53 (.50)	3.36 (.50)		$t(37) = -8.641$, $p < .001$*	
Total 77	2.46 (.44)	3.10 (.51)		$t(76) = -9.784$, $p < .001$*	

*Significant difference at the 0.05 level

4.4 Discussion

The present study investigates the effect of ER activities that are based on the proposed didactical model CPG+, as well as the appropriate guidance for the improvement of computational thinking and problem solving skills.

A first observation is that both groups were comparable regarding students' skills in CT and PS at the end of the 3rd session, since their answers in the Q1 test were practically identical, as the t-test results showed in Table 2. Despite this, at the end of the robotics activities, as the results showed above, all the students improved significantly their CT and PS skills, independently of the guidance condition in the activities and the different modality instruments in assessment (written and orally).

Computational Thinking skills. Reflecting on the results of the two Quiz-activities (CT-3 & CT-9) and comparing CT-9 and CT-3 scores, we observe that both group MG and group SG exhibit a statistically significant improvement in CT skills. Focusing further on Table 4, we see that comparing CT-9 measures (with CT-3 as covariant) significant differences between SG and MG group are identified in most cases. Additionally, we observe that as the training proceeds, students in the SG group show a statistically important improvement in comparison with students in the MG as regards the concepts: abstraction, generalization, algorithm and decomposition of CT skills. However, modularity and debugging skills are further developed (from CT-3 to CT-9) at the same level in both groups. As regards the debugging skills, we believe that the explanation is that the robotics activities allow students to practice debugging systematically independently of their response to the instructions - written or oral - and this is reflected in their scores. Finally, as regards algorithm skills, we identify the maximum statistically significant difference from CT-3 to CT-9 between the two groups favouring students in the SG group. This, we believe, is due to the fact that the students in SG group, following the worksheet guidelines and expressing their thinking about algorithmic steps in writing, are more familiarized with these skills.

Moving on to the CT-TA scores (Table 5), and comparing them with the results of the CT-9 (Table 4), when evaluating students' CT skills in the description of a problem's solution with Think-Aloud protocol, we observe that students reach higher scores in all CT skills. One possible explanation is that the think-aloud protocol allows the students to thoroughly express their more complex thinking required to describe a problem solution and to reflect on CT concepts relevant to their solution, and another is that the Think-aloud protocol took place after the CT-9 measurement and the final challenge, so the students had familiarized to a greater degree and were better able to respond to these challenges.

Next, we focus on Table 5 that presents analytically all the components of CT-TA scores for both groups. We see that the previously discussed results for CT-9 between the two groups appear again for all the components of CT, but not for the abstraction. In particular, the SG group improve significantly their CT-TA score for generalisation, decomposition and algorithm, but no significant differences appear in the scores for abstraction, debugging and modularity (strong tendency for SG, $p = .059$). We observe that the aforementioned change in abstraction is explained by the higher score in CT-TA ($M = 2.76$, $SD = .94$) than in CT-9 score ($M = 2.41$, $SD = .82$) for MG group, a fact that is observed when the modality of the assessment instrument changes. For this reason we believe that this is mainly due to the familiarization of the MG group with oral answers. Additionally, comparing CT-TA measures with CT-3 as covariate (ANCOVA in Table 4), we identify a statistically significant difference between the two groups favouring students in the SG group.

Finally, by reflecting on the students' subjective impressions in the semi-structured interviews, regarding the CT skills, we observe that students reported a satisfactory understanding and assimilation of these skills as the training proceeded. The key findings can be summarised as follows: (a) Students emphasized that although initially they had no prior knowledge of these concepts, during the activities they acquired a familiarity with them; (b) As regards abstraction, they experienced difficulties in identifying and describing the common concept/behaviour of the various scenarios, but despite this, after the initial training they were able to easily identify the common programming concepts; (c) As regards generalization, they regarded it as the most difficult concept. However, as the activities progressed students were able to offer more comprehensive solutions, but not just as easily for each activity; (d) As regards decomposition, most of the students divided the problems easily into smaller ones, when required to do so by the complexity of the problem. However, they declared that in some cases they would prefer not to implement the stage of decomposition, because they considered that they could address the problem as a whole and because they would like to proceed rapidly with the resolution process, in order to see as soon as possible their robots in action; (e) With regards to the algorithm design, they stated that it took them some time to become familiar with the detailed step-by-step description of the algorithm, and also they emphasized that the indicative steps of the worksheet instructions were very helpful; (f)

Concerning modularity, most said that they found it interesting and useful to create their own programming command ("myblocks"); (g) with regards to debugging, they said that initially they asked teachers for help to identify errors, however, with their guidance they soon managed to locate and to solve the errors by themselves, something that gave them great pleasure.

Problem Solving skills. Reflecting further on the data of Tables 7 and 8, which present the results of students' scores in problem solving (from measures Quiz-activities (PS-3, PS-9) and Think-Aloud (PS-TA)), we observe that all the students developed significantly the PS skills at the end of the sessions, with the SG students achieving significantly higher level of those skills compared to the MG group. Additionally, students reported that the step-by-step instructions on the problem-solving process in the worksheets were very helpful, and also they reported that some steps came to mind when they were solving mathematical problems.

Robotics activities based on proposal model CPG+. Reflecting further on students' semi-structured interviews and researchers' observations we report the following:

(a) Students said that initially they were not accustomed to working in teams assuming specific roles, so it took time and the teacher's support to pursue their roles. In the progress of the sessions, they liked to "play" the roles and considered them useful for supporting collaboration.
(b) The authentic problems, that are meaningful and interesting to them, excite and encourage students to participate in activities, for example: the alarm activity, the colliding cars at amusement parks etc.
(c) Robotics is an entertaining and creative activity for students. The spirit of competition in the final challenge offers a productive motive for an efficient method of instruction and also for acquisition of knowledge. We must emphasize the importance of the coaches' role in cultivating the spirit of fair play in their students' groups.
(d) Guidance with detailed instructions aiming at the accomplishment of the activities is necessary for effective learning. This was perceived in the MG group where, although there were detailed instructions, as students were not required to respond in writing, they discussed the resolution stage superficially and quickly moved to the next step. Soon however, the need arose to turn back and review the gap steps in order to be able to complete their activity.

5 Conclusions

Overall, this study showed that: (a) The robotics activities based on CPG+ create an entertaining and creative educational framework, which can develop effectively students' skills, such as problem solving and computational thinking; (b) The SG group students outperformed students in the MG group in improving skills such as Computational and Problem solving. Given that both groups had the same set of instructions and guidelines, it is clear that this difference occurred because of the requirement for written reviews in the SG group. A future study could look at a new instructional design that will reduce the workload of writing without diminishing the learning benefits.

References

1. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013)
2. Grover, S.: Robotics and engineering for middle and high school students to develop computational thinking. (650), pp. 1–15 (2011)
3. Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., Korb, J.T.: Introducing computational thinking in education courses. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education-SIGCSE0000 2011, pp. 465–470 (2011)
4. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
5. Alimisis, D., (ed.): Teacher education on robotics-enhanced constructivist pedagogical methods. School of Pedagogical and Technological Education (ASPETE) (2009)
6. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2011)
7. Matari, M.J.: Robotics Education for All Ages, Robotics (2004)
8. Atmatzidou, S., Demetriadis, S.: Advancing students' computational thinking skills through educational robotics: a study on age and gender relevant differences. Robot. Auton. Syst. **75**, 661–670 (2016)
9. Blanchard, S., Freiman, V., Lirrete-Pitre, N.: Strategies used by elementary school children solving robotics-based complex tasks: innovative potential of technology. Procedia – Soc. Behav. Sci. **2**(2), 2851–2857 (2010)
10. Chambers, J.M., Carbonaro, M., Rex, M., Grove, S.: Scaffolding knowledge construction through robotic technology: a middle school case study. Electron. J. Integr. Technol. Educ. **6**, 55–70 (2007)
11. Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., et al.: Computational thinking for youth in practice. ACM Inroads **2**(1), 32–37 (2011)
12. Repenning, A., Webb, D., Ioannidou, A.: Scalable game design and the development of a checklist for getting computational thinking into public schools. In: Proceedings of the 41st ACM Technical Symposium on Computer Science Education - SIGCSE 2010, p. 265 (2010). doi:10.1145/1734263.1734357
13. Frangou, S., Papanikolaou, K., Aravecchia, L., Montel, L., Ionita, S., Arlegui, J., Pina, A., Menegatti, E., Moro, M., Fava, N., Monfalcon, S., Pagello, I.: Representative examples of implementing educational robotics in school based on the constructivist approach. In: Workshop Proceedings of SIMPAR 2008, pp. 54–65 (2008)
14. Gaudiello, I., Zibetti, E., Carrignon, S.: Representations to go: learning robotics, learning by robotics. In: Workshop Proceedings of Interational Conference on Simulation, Modeling and Programming for Autonomous Robots, SIMPAR, pp. 484–493 (2010)
15. Eteokleous-Grigoriou, N., Psomas, C.: Integrating robotics as an interdisciplinary educational tool in primary education. In: Society for Information Technology & Teacher Education International Conference, vol. 2013(1), pp. 3877–3881 (2013)
16. Williams, D., Ma, Y., Prejean, L.: A preliminary study exploring the use of fictional narrative in robotics activities. J. Comput. Math. Sci. Teach. **29**(1), 51–71 (2010). Chesapeake, VA: AACE
17. Settle, A., Perkovic, L.: Computational thinking across the curriculum: a conceptual framework. Technical reports (2010)

18. Bottino, R., Chioccariello, A.: Computational thinking: videogames, educational robotics, and other powerful ideas to think with. In: KEYCIT: Key Competencies in Informatics and ICT, vol. 7, p. 301 (2015)
19. Kazakoff, E.R., Sullivan, A., Bers, M.U.: The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Child. Educ. J. **41**, 245–255 (2013). doi:10.1007/s10643-012-0554-5
20. Bers, M.U.: The TangibleK robotics program: applied computational thinking for young children. Early Childhood Res. Pract. **12**, 1–20 (2010). http://ecrp.uiuc.edu/v12n2/bers.html. Accessed Sep. 2014
21. Grover, S., Pea, R.: Computational thinking in K–12: a review of the state of the field. Educ. Researcher **42**, 38–43 (2013). doi:10.3102/0013189X12463051
22. Eguchi, A.: Learning experience through RoboCupJunior: promoting engineering and computational thinking skills through robotics competition. In: K-12 Computer Science Computational Thinking Initiatives, Indianapolis, Indiana, pp. 1–18 (2014). https://peer.asee.org/20743
23. Touretzky, D.S., Marghitu, D., Ludi, S., Bernstein, D., Ni, L.: Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In: Proceeding of 44th ACM Technical Symposium Computer Science Education - SIGCSE 2013. ACM Press, New York, p. 609 (2013). doi:10.1145/2445196.2445374
24. Penmetcha, M.: Exploring the effectiveness of robotics as a vehicle for computational thinking. Purdue University (2012). http://search.proquest.com/docview/1221535411. Accessed Mar. 2014
25. Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. Procedia – Soc. Behav. Sci. **47**, 1991–1999 (2012)
26. Schmidt, H.G., Loyens, S.M., Van Gog, T., Paas, F.: Problem-based learning is compatible with human cognitive architecture: commentary on Kirschner, Sweller, and Clark (2006). Educ. Psychol. **42**(2), 91–97 (2007)
27. Clark, R.E., Kirschner, P.A., Sweller, J.: Putting students on the path to learning: the case for fully guided instruction. Am. Educ. **36**(1), 6–11 (2012)
28. Menary, R.: Writing as thinking. Lang. Sci. **29**(5), 621–632 (2007)
29. Papadopoulos, P., Demetriadis, S., Stamelos, I., Tsoukalas, I.: The value of writing-to-learn when using question prompts to support web-based learning in ill-structured domains. Educ. Technol. Res. Dev. **59**(1), 71–90 (2011)
30. Polya, G.: How to Solve it: A New Aspect of Mathematical Model. Princeton University Press, Princeton (1945)

The Effectiveness of Integrating Educational Robotic Activities into Higher Education Computer Science Curricula: A Case Study in a Developing Country

Ernest B.B. Gyebi(✉), Marc Hanheide, and Grzegorz Cielniak

School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK
{ergyebi,mhanheide,gcielniak}@lincoln.ac.uk

Abstract. In this paper, we present a case study to investigate the effects of educational robotics on a formal undergraduate Computer Science education in a developing country. The key contributions of this paper include a longitudinal study design, spanning the whole duration of one taught course, and its focus on continually assessing the effectiveness and the impact of robotic-based exercises. The study assessed the students' motivation, engagement and level of understanding in learning general computer programming. The survey results indicate that there are benefits which can be gained from such activities and educational robotics is a promising tool in developing engaging study curricula. We hope that our experience from this study together with the free materials and data available for download will be beneficial to other practitioners working with educational robotics in different parts of the world.

Keywords: Educational robotics · Effectiveness of teaching methods · Education in developing countries · Tertiary education

1 Introduction

There is a general belief that in learning, robots can be an effective means to facilitate more engagement, higher motivation, and the development of practical skill sets, beyond the focus of robotics itself. For instance, the effectiveness of robotics as a subject to convey a larger skill sets to students has been analyzed and found to be promising [1]. The positive effect, to a large extent is gained from the "embodiment" and physical presence of robots, which make the outcomes of programming very vivid and immediately accessible, providing a continual formative assessment of learning progress and encouragement to students. Following these ideas, selected institutions in developing countries have recently made some efforts to break the traditionally very theoretical and presentation-based lecturing and introduced robotic activities to improve the quality of teaching and learning [2]. However, most of the studies in this area so far involved recreational and extracurricular activities, which do not provide

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_6

an environment for assessing its effectiveness for teaching formal courses. In the study presented in this paper, we set out to embed constructionist learning by means of robot programming into an existing formal classroom curriculum in a developing country to test its effectiveness and impact. Particularly, in this paper, we present a study to investigate the effects of educational robotics on an undergraduate Computer Science education which was carried out in a public university in Ghana. In this work, robotics itself is not among the key learning objectives, rather is it employed as an educational tool set to help conveying key concepts of computing in the first year of an undergraduate degree programme.

The key contributions of this paper are the longitudinal design of the study, spanning the whole duration of one taught course, and its focus on continually assessing the effectiveness and the impact of robotic-based exercises as opposed to paper-based exercises. The study assessed the students' motivation, engagement and level of understanding in learning general computer programming. In particular, we set out to answer the following three research questions:

(i) *Does learning with educational robots have impact on students' understanding of programming concepts?*
(ii) *Does interacting with educational robots have impact on students' motivation and engagement in learning programming?*
(iii) *Is the use of robotic set-up as a tool for conducting educational activities effective?*

To answer these questions and determine the impact of educational robotics on students' learning, a 10-week full semester study involving educational activities was conducted at the Department of Computer Science, University of Ghana. The activities were embedded into one of the relevant courses on computer architecture and principles of programming. Students' responses to their interactions with the learning activities as well as examination scores from five pre-intervention years and the intervention year were analyzed for trends in students' performances.

2 Related Work

Effective teaching and learning in higher education is one of the biggest challenges faced by many developing countries. For example, the curricula of most institutions in these countries are usually designed to encourage the traditional lecture-based method of teaching without instructional strategies that facilitate the inclusion of all students [3]. Lecturers tend to use the old-fashioned style of teaching whereby emphasis is placed on information transmission rather than on students active involvement. This invariably reduces students engagement and participation in the lessons. Also due to the introduction of measures like affirmative action aimed at improving the representation of disadvantaged groups on the campuses, which is perhaps peculiar to developing countries [4], categories of students enter the universities with gaps in the knowledge and skills required for studying particularly in key areas such as mathematics and science [5]. Consequently, many of these students may be under-prepared to undertake university tasks which usually present challenges for students to overcome. Therefore selected institutions in developing countries have recently made efforts to introduce effective ways of improving the quality of teaching and learning in order to support the students.

One of the practices that has been identified particularly, in computer science, and is promoted by many of the western higher education institutions [6] to effectively improve teaching and learning involves hands-on exercises and collaborative learning. Some educators such as T.M. Akey also argue that students learn more and retain more information when they actively participate in the learning process and when they can relate to what is being taught [7]. Therefore efforts to improve the quality of teaching and learning by the adoption of practices that facilitate more engagement, higher motivation and active participation of students are a promising direction.

Educational theorists such as S. Papert [8] believe that robotic activities have tremendous potential to improve classroom teaching. They argue that students can gain a sense of power over technology by creating an environment where they can program computers and robots. Other studies have also identified the concrete nature of robots as being one of their important advantages when used as learning tool. For example, students can understand abstract concepts and gain a more functional level of understanding when they learn with robots [9]. However, it is important to emphasize that the robot is just another tool, and it is the educational theory that will determine the learning impact coming from robotic applications. Alignment with theories of learning, proper educational philosophy, well designed curricula and supportive learning environments are some of the important elements that can make any educational innovation, including robotics, successful [10].

Activities with educational robotics can serve learning objectives from a wide range of disciplines from technology and design to mathematics and science education. They are hands-on activities with important experimentation features [11]. From this point of view educational robotics creates an active, cooperative learning environment which emphasizes on students' participation. So incorporating robotic technologies into tertiary curriculum can enrich teaching practices with great impact in addressing teaching objectives from different disciplines with an innovative approach. This fact is backed up by research which suggests that robots tie into a variety of disciplines [12]. A robot is made of component parts of motors, sensors and software. Each of these parts depends on different fields of knowledge such as engineering, electronics, and computer science. This interdisciplinary nature of robots means that when students learn to engineer robots they will inevitably learn about the many other disciplines that robotics utilize [13].

Accordingly, educational robots are being used to teach various subjects at the university level. At Carnegie Mellon University, USA, an open source robot application development framework called Tekkotsu has been designed specifically for education [14]. This application development framework is based on

C++ which can be used for teaching maths topics such as vectors, matrices and linear algebra. Educational robotics has also been successfully used to teach Physics for undergraduate students in Brazil [15]. Students developed prototype robots that they used to learn electricity and electronics, with emphasis on building electrical circuits. In their study at the Department of Computer Science, University of Waterloo, B.W. Becker [16] also used 'karel' the robot for teaching Java programming. They designed an Introductory to Computer Science course that used the robot to teach object oriented approach to programming.

Despite these encouraging reports, the majority of considered studies are short-term, conducted for a selected subject and group of students only. In our previous work [17] we presented initial results from implementing small scale pilot studies on introducing educational robotics into higher education activities in developing countries. The current work takes these efforts one step further and reports on fully integrated educational robotic activities into undergraduate Computer Science curricula in an attempt to address some of the challenges of higher education in developing countries.

3 Methodology

The purpose of this study was to assess the impact of educational robotics on a formal undergraduate Computer Science education in developing countries. Our goal was to determine the effectiveness of educational robotics activities on students' motivation, engagement and level of understanding in learning computer programming. In this section, we present the methodological considerations and procedures adopted for the study, which include the course overview, learning activities, study design, surveys and participants.

3.1 Course Overview

Our study considered a core Introduction to Computer Science course teaching basics of computer architectures and programming principles to all first year students at the Department of Computer Science, University of Ghana. The course consists of lectures and laboratory workshops, and its syllabus includes introduction to programming languages, data representation, logic operations but also program structure and flow control (see Sect. 3.2). In a typical setting, the lab workshops involve paper-based activities designed to enable students practice the topics learned in lectures. The structure of the paper-based activities is based on collaborative learning, an instructional method in which students work together in small groups toward a common goal [18]. Team work is emphasized and the students are assessed as a group. Therefore each student has a shared responsibility for meeting the course challenges such as submitting exercises for marking on time. The designed learning activities are usually in a form of hands-on exercises which often require computers in the lab. Students do the exercises on paper and submit their work to teaching assistants who also provide in-class help in case of difficulties.

(a) Thymio II educational robot.

(b) Our students taking part in robotic activities in the lab.

Fig. 1. Lab activities using educational robots.

Our focus was to embed robotic activities alongside the paper-based activities in order to make direct comparisons between the two. Based on our earlier survey [19] which looked at the existing suitable robotic platforms for teaching Computer Science at African institutions, we have selected an affordable robot Thymio II to conduct the robot-based activities. Thymio II (see Fig. 1(a)) is a simple two-wheel differential platform equipped with an array of LEDs and a number of simple sensors including odometry, proximity and temperature sensors, a microphone, etc. The robot comes with a dedicated programming environment called Aseba, which is an open-source scripting language, and a set of on-line tutorials and additional materials that can be used when designing own teaching activities. We build on top of that setup by introducing a set of workshops thematically covering the five computer science topics described in the following sections. We share the developed materials including workshop descriptions, survey questionnaires and the course module in an open-source fashion on our website http://lncn.eu/gyebi16edu so these can be re-used by other educators teaching similar topics.

3.2 Learning Activities

Our designed learning activities comprise of five workshops, all geared towards assisting students to learn the basic concepts in programming introduced during lectures. The course description and specific learning outcomes that students were expected to achieve at the end of each workshop are presented in Table 1, which was adopted from existing course content and learning outcomes for the core introduction to Computer Science course. We have made it available on our website. We used an iterative, incremental development approach involving intermediate constructs, each adding more capabilities to make the tasks cognitively appropriate to the developmental levels of our students. Students were gradually introduced to the learning activities by performing the easier tasks before progressing to more difficult tasks which required application of skills

Table 1. Introduction to Computer Science - course description and associated learning outcomes.

Topic	Learning outcomes
Introduction to programming languages, modern standards and applications	Identify the various types of programming languages and their use
Data representation and number systems	Understand representations for different data types, perform conversions between different number systems
Digital logic and Boolean algebra	Implement logic operations using the basic types of logic gates
Basic data types, operators and program structure	Demonstrate the ability to use variables, operators and functions
Program control structures, branching and loops	Apply control structures to write simple programs

and knowledge acquired from previous activities. To enable a direct comparison between paper- and robot-based activities, we created a set of similar tasks based on the same design goal for each of the five workshops, which we now describe in detail.

Introduction to Programming Languages: The learning activities introduced students to the main types of programming languages and how they are used to write a sequence of instructions that a computer can execute. Since most of the students were learning programming for the first time, the challenge was how to assist them in building simple programs. Therefore, the paper-based activities concentrated on getting familiar with the programming environment through building provided examples in Visual Basic and teaching how to make simple code modifications, commenting and debugging. In the first week of robot-based activities, the students were introduced to the robot, its components and Aseba programming environment. Similarly to the paper-based activities, the students were provided with simple demo programs which illustrated individual parts of the setup and basics of program execution. These activities were mainly based on the existing tutorials for the Thymio robot.

Data Representation: This workshop was designed to teach students practical aspects of number systems and data representation. Paper-based tasks required the students to exercise number system conversions from decimal to binary using the repeated division method. Difficulty of the task depended on the value of the converted numbers. Further tasks included also manual conversions into one's and two's complement representations. For the robot-based activities, we adopted the robot's 8 LEDs as a way of visualizing binary numbers and using each LED to represent a single bit. This way the students could visually see the results of counting in unary and binary systems, conversions from decimal to binary and different complement representations. The conversion code was provided and students were only required to make small changes to the code (e.g. changing variable values). Conversions from binary to decimal number system proved to be more difficult to implement as there were no obvious ways of visualizing the decimal results either on the robot or in software and therefore these activities were not included.

Logic Operations: The learning activities were designed to teach fundamentals of Boolean algebra, basic logic gates and a way of combining them into more complex systems. The paper-based activities included a range of standard exercises involving drawing gates and circuits and deriving truth tables. For the robot-based activities, we used proximity sensors as inputs to logic circuits so the robot movement was depended on logical combination of these inputs. This enabled learning about fundamental logic operations, universal logic gates but also Boolean functions combining multiple inputs through different operators. This final task required students to combine the provided code for individual logic gates into a combinatorial circuit and so teaching them how to reuse and restructure the existing code.

Programming Fundamentals: This workshop introduced students to the programming fundamentals including data types, variables, operators and functions. The paper-based tasks required association of different data types with variables, changing mathematical expressions into simple programming statements and also using programming operators to represent given statements. For the robot-based activities, the concept of variables was re-iterated by using timers and speed variables with basic movement behaviors. The students could observe differences in robot behavior depending on different values of these variables. The analysis of the provided code examples enabled students to learn about different types of operators. Further tasks required combining simple behaviors implemented as functions into more complex behaviors such as driving in a square. Due to certain limitations of the Aseba language, function parameters are not easily implemented and therefore were not exercised in the proposed activities.

Control Structures: The final set of workshop activities was designed to introduce program control structures such as conditional statements and loops. For the paper-based activities, programming tasks involved writing simple algorithms that would satisfy certain conditions by using IF-ELSE statements and tasks involving drawing of flow charts and implementing loops for algorithms calculating average values. Robot-based activities involved simple obstacle avoidance behaviors using conditional statements controlling the robot movement depending on the state of proximity sensors or controlling the direction of movement depending on manual key presses. The loops and conditional statements were employed to check the state of an array of 7 proximity sensors which then defined different robot movement.

3.3 Study Design

In contrast to our earlier study [17] which was conducted as a relatively short pilot to identify the main implementation issues, the proposed activities were incorporated into a regular program of study and formed part of the learning activities for the first year undergraduate curricula. The experimental design used one group made up of all first year Computer Science students, but the learning activity was divided into two parts (referred to as paper-based and robot-based) to allow for direct comparison of the experimental results. The ethical policy of the University of Ghana required that students in a particular year group learn the same course topics and content in all formal courses in order to give the best and similar experience to all students. Also there were resource constraints such as inadequate labs which posed organizational problems of systematic issues with timetabling of the sessions, preventing us from splitting the class into two distinct sets in a formal setting. As a result, we could not implement an experiment with a typical control group in our situation.

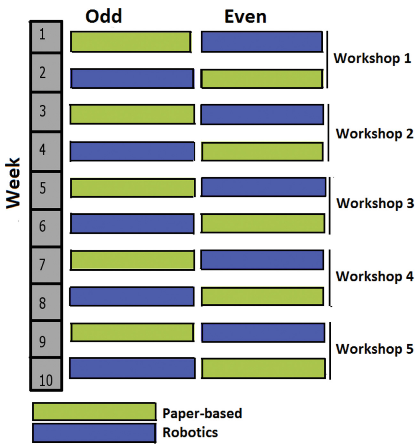


Fig. 2. Weekly schedule for workshop activities.

Each of the five workshop activities consisted of paper-based and robot-based tasks, which required 2 sessions to complete. The students were divided into 10 groups and those in odd numbered groups (i.e., 1, 3, 5, 7 and 9) participated in the paper-based activities while the remaining even numbered groups (2, 4, 6, 8 and 10) practiced the robotic activities, initially. The groups were swapped for the odd and even numbered groups to participate in the robot-based and paper-based activities respectively in the following workshop session to complete the first workshop activity. This was repeated for all five sessions and therefore all students participated in all workshop activities. Figure 2 depicts the workshop schedule in a graphical format.

3.4 Student Surveys and Performance

Student surveys which included an initial survey, workshop evaluations and a final survey were conducted through a set of questionnaires to determine the student's motivation, engagement and level of understanding in both the paper-based and robotic activities. The purpose of the initial survey was to gain a better understanding of student background and knowledge before the study.

The final survey provided means to evaluate the overall workshop experience and knowledge gained, whereas workshop evaluations helped assess the effectiveness of each lab session. The specific questions for respective surveys were grouped into following categories:

- *Initial Survey* asked for student background such as prior education, gender, selected study course but also prior learning experience in programming and using robots.
- *Workshop Evaluation* was issued to students after each session for both paper-based and robot-based activities. To encourage regular participation and honest answers, we made this part completely anonymous and consisting only of three questions on workshop difficulty, relevance to concepts learnt from lectures and re-use of concepts learnt in previous sessions.
- *Final Survey* included overall ratings for both paper-based and robot-based activities with regard to student motivation and engagement, intuitiveness and understanding. To gain the learning experience, the students were also asked about topics learnt and if they perceived them as beneficial. Finally, the survey asked for rating of organizational aspects of the workshops including their organization, delivery, quality of feedback received and quality of the overall experience. These questions overlap with typical questions asked in other places such as the National Student Survey in UK. In addition, the final survey encouraged free comments both positive and critical to capture other issues not covered by the presented questions.

The method for assessing students' performance on the course was based on their scores in the workshop activities and a final exam. The questions for the final exam, which was taken at the end of the semester covered the workshop activities and lecture sessions. The final exam carried a total of 60% of the overall examination mark. Students' engagement and understanding of the learning activities were assessed by their performances in the hands-on lab exercises. The exercises were marked at the end of each workshop and carried a total of 40% of the overall examination mark. This is the model used for assessing workshop activities at the University of Ghana and we decided to maintain it to enable us make a fair comparison between students' performances in the pre-intervention years where no robots were used and the intervention year where robots were introduced to conduct the learning activities.

3.5 Participants

The study was conducted for the first year Computer Science students at the Department of Computer Science, University of Ghana, during their first semester of study in Winter 2015. From a total of 202 students who enrolled on the course, 194 participated in at least one workshop session while 166 participated in all the workshops. 178 students participated in the initial survey and 166 participated in the final survey. The students were divided into 10 groups (roughly 20 students per each group). Because of the large size of the first year cohort and a limited number of robots (we only had 10 Thymio II robots available), the students were divided into pairs with one robot allocated to each pair. Although a limiting factor, this enabled us also to assess students abilities working in teams and observe their interactions during the learning activities.

Based on initial survey, we were able to assess the students' background. The majority of students declared their prior educational background in Science (77%) with the remainder roughly split equally between Arts and Business. 19% of the students declared Computer Science as their major study subject and similar figures were reported both for minors in Maths and Stats (24%) and Psychology (32%). The reminder (13%) accounted for minors in Maths and Economics. 22% of participants were female students which is a slightly higher figure than in the majority of the developed countries. Since this was the first year of study, it was not surprising that the majority of student (63%) had no prior contact with programming or very basic understanding (25%). Similarly, the students prior contact with robots was either through leisure activities (9%) or basic programming (9%) but the majority have not experienced any contact with such technology (82%).

4 Results

The experimental data for the surveys and exam were statistically analyzed using SPSS for Windows version 21 to determine the effect the robotic activities had on students' learning. This was done by comparing the individual workshop survey results for the paper-based and robot-based activities, analysis of reports from the final survey and comparing the exam results from the pre-intervention and intervention years. In all comparison results, we report a significance value p and assume that cases where $p < .05$ indicate statistically significant differences. We have highlighted all such entries in the tables to make visual comparisons easier. In addition, we also report a Pearson's correlation coefficient r for paired final survey data on aspects of student learning to indicate the strength and direction of the linear relationship between two metrics ranging from -1 (perfect negative effect) to $+1$ (perfect positive effect) with 0 indicating no effect.

4.1 Students' Self-assessment of the Activities

An independent-samples t-test was conducted on students responses concerning the workshop activities to compare the means scores between the paper- and robotic-based activities. Table 2 presents detailed results from direct comparisons of each workshop activity. The table reports the workshop number, type of activity (paper- vs robot-based) and the student mean responses to three questions on workshop difficulty, relevance to concepts learnt in lectures and re-use of concepts learnt in previous sessions.

Since each group of students exercised the same topic twice, either as paper- or robot-based activity, we have also made a clear distinction between groups which practiced the activities in the first or second week for a particular topic.

Table 2. Student assessment of individual workshop activities. 'W' denotes workshop number, 1^{st} paper refers to paper-based activities taken in the first week and 2^{nd} paper in the second week for a particular topic. The robot-based activities are indicated in a similar fashion.

W	Activity	Concepts employed		Concepts re-used		Difficulty	
		$\mu \pm \sigma$	p	$\mu \pm \sigma$	p	$\mu \pm \sigma$	p
1	1^{st} paper	2.9 ± 1.2	.57	$\mathbf{2.8 \pm 1.2}$	**.00**	$2.4 \pm .8$	.06
	1^{st} robot	2.7 ± 1.4		$\mathbf{1.9 \pm 1.2}$		$2.0 \pm .7$	
	2^{nd} paper	3.3 ± 1.2	.66	3.0 ± 1.3	.56	2.6 ± 1.2	.25
	2^{nd} robot	3.1 ± 1.1		2.8 ± 1.1		2.1 ± 1.2	
2	1^{st} paper	$4.2 \pm .8$	.25	3.9 ± 1.2	.82	$2.2 \pm .7$	.05
	1^{st} robot	3.8 ± 1.2		3.8 ± 1.3		2.9 ± 1.0	
	2^{nd} paper	4.2 ± 1.0	.46	3.9 ± 1.1	.48	$\mathbf{1.9 \pm 1.0}$	**.00**
	2^{nd} robot	4.0 ± 1.0		4.0 ± 1.0		$\mathbf{2.6 \pm 1.0}$	
3	1^{st} paper	4.0 ± 1.1	.22	$\mathbf{3.2 \pm 1.3}$	**.00**	$2.9 \pm .9$	.84
	1^{st} robot	$4.2 \pm .9$		$\mathbf{4.0 \pm 1.0}$		2.9 ± 1.0	
	2^{nd} paper	4.1 ± 1.0	.48	3.9 ± 1.1	.68	$\mathbf{2.4 \pm 1.0}$	**.00**
	2^{nd} robot	4.0 ± 1.1		4.0 ± 1.1		$\mathbf{3.0 \pm 1.2}$	
4	1^{st} paper	3.1 ± 1.1	.25	$\mathbf{3.2 \pm 1.2}$	**.01**	$3.6 \pm .8$	.10
	1^{st} robot	3.3 ± 1.1		$\mathbf{3.9 \pm 1.1}$		3.1 ± 1.0	
	2^{nd} paper	3.7 ± 1.1	.08	3.8 ± 1.0	.67	$3.2 \pm .9$	.57
	2^{nd} robot	3.4 ± 1.2		3.8 ± 1.0		3.1 ± 1.0	
5	1^{st} paper	$\mathbf{3.3 \pm 1.3}$	**.02**	$\mathbf{3.1 \pm 1.1}$	**.00**	$\mathbf{3.5 \pm 1.1}$	**.01**
	1^{st} robot	$\mathbf{3.9 \pm 1.1}$		$\mathbf{4.0 \pm 1.1}$		$\mathbf{2.9 \pm .9}$	
	2^{nd} paper	$3.9 \pm .8$	.13	$3.8 \pm .8$	.36	$\mathbf{3.4 \pm 1.0}$	**.00**
	2^{nd} robot	3.7 ± 1.1		3.6 ± 1.0		$\mathbf{2.7 \pm 1.0}$	

Thanks to that we could see if some familiarity with the topic has any influence on the subjective perception of activities.

There were no major differences between paper- and robot-based activities when employing concepts learnt during lectures, with an exception of Workshop 5 on Control Structures, where robotic activities were perceived as more relevant to the lecture content. Overall, it can be seen that in this category activities prepared for Workshop 2 and 3 (Data Representation and Digital Logic) were considered as particularly useful indicated by higher mean scores and that students saw less relevance of lecture material corresponding to Workshop 1 (Introduction to Programming).

In gradual development of skills, indicated by a 'concepts re-used' question, the robot-based activities were perceived as more beneficial (see scores for Workshops 3, 4 and 5). The scores for Workshop 1 favor paper-based activities but that question was probably less relevant just at the beginning of the course

Table 3. Students' self-assessment for their learning. Sample size = 166.

Metric	Activity	$\mu \pm \sigma$	p	r
Motivating	Paper-based	$\mathbf{3.7 \pm .8}$	**.00**	**.38**
	Robot-based	$\mathbf{4.2 \pm .8}$		
Engaging	Paper-based	$\mathbf{3.9 \pm .8}$	**.00**	**.33**
	Robot-based	$\mathbf{4.2 \pm .7}$		
Understanding	Paper-based	$4.1 \pm .8$	.75	.53
	Robot-based	$4.1 \pm .7$		
Difficulty	Paper-based	$\mathbf{3.1 \pm .9}$	**.02**	**.23**
	Robot-based	$\mathbf{2.8 \pm 1.0}$		
Intuitive	Paper-based	$3.6 \pm .8$	.83	.59
	Robot-based	$3.6 \pm .8$		

and perhaps the students were not exactly sure what that particular question refereed to in that week.

There seems to be an increasing trend in the perceived difficulty of both types of workshop activities indicated by increasing mean scores for the difficulty question. Robotic tasks in Workshop 5 were perceived easier than paper-based assignments. There is an indication that paper-based activities for Workshop 2 and 3 might have been easier but since this is only reported by groups taking robotic tasks in the second week, these results are not entirely convincing.

Table 3 presents results obtained from the final survey on different aspects of learning for both paper-based and robot-based activities. In this overall comparison, the robot-based activities were more positively scored in categories of motivation and engagement, and also were perceived as easier as indicated by the mean scores. There was no difference in responses to questions on understanding and intuitiveness of both types of activities though.

4.2 Students' Exam Performance

We have also compared the overall student exam results in the intervention year, where the robotic activities were introduced, to 5 pre-intervention years. The results presented in Table 4 indicate potential differences in Year 3 and 5 but overall there are no clear indications that the robotic activities impacted the marks achieved by students. It is worth to note that the course was taught by different lecturers in these years so the marks might be more correlated to the lecturing style rather than the type of activities. The natural variations in overall student academic abilities each year might also have influenced these results.

5 Discussion

In this work, we set out a goal of assessing the effectiveness and impact of robotic activities introduced into a regular higher education computer science curricula

Table 4. Comparison of pre-intervention and intervention years Exam results. *Year 6 is the intervention year.

Year	Sample size	$\mu \pm \sigma$	Comparison with Year 6
			p
1	182	62.9 ± 11.0	.11
2	269	62.8 ± 11.8	.08
3	308	$\mathbf{62.5 \pm 11.6}$	**.04**
4	373	65.2 ± 10.0	.65
5	209	$\mathbf{70.1 \pm 9.8}$	**.00**
1–5 combined	1341	64.6 ± 11.2	.84
6	**166**	$\mathbf{64.8 \pm 11.3}$	

in a developing country. The presented results are promising especially in areas of student motivation and engagement. This was also evident in provided free comments which include terms like practical, engaging, interactive, etc. The results from evaluation of individual workshops indicate that robotic activities seem to also have a positive effect on gradual development of skills and seeing links between different topics of the course.

In our case, it was more difficult to see the benefits of using educational robots for teaching specific theory topics and technical skills. This is a complex problem in which all components of the course delivery are playing a vital part, including lecture content, lecturing style but also quality and execution of the prepared practical activities. One particular problem we encountered is related to the ordering of topics in this particular course. As the robot exercises required learning some programming concepts from the very start, this introduced another level of complexity for students when they were just learning fundamentals of computer architectures. It would be interesting to see if a course which would start with programming principles first would be more effectively delivered. The analysis of student performance in pre- and intervention years suggest that using overall student marks from the course as an effectiveness indicator is of very limited use and methodological developments in evaluating similar activities should focus on other methods (e.g. short tests after each activity).

Another purpose of our study was to evaluate the effectiveness of using robotic set-up as a tool for conducting educational activities. The results of the study indicate that our robotic set-up was generally effective. For example, students' attendance to the lab sessions improved as the result of the introduction of the robot-based activities. Also students were able to perform different educational tasks with the robots which made their learning meaningful. This was noticeable from some of the comments students made concerning the robot-based activities. Statements like "it was very understandable and interesting", "it was interactive and helpful", "it opened my mind to different ways of approaching a problem" were made in the self-assessment report. The selected hardware platform, Thymio II robot proved to be a reliable choice. Its rich functionality enabled us to introduce a wide range of activities and programming concepts. There were some limitations in illustrating specific technical topics (e.g. binary to decimal conversions) due to platform limitations but nevertheless from the point of view of an educator this was a great platform to work with. The provided teaching materials and example projects helped our students to understand some of the programming concepts.

The main problem we encountered in delivering these activities were outside of typical spectrum of educational challenges. Frequent power outages disrupted the scheduled sessions and might have had a negative impact on student perception of the course. Such infrastructure problems are not only limited to a single developing country but more prevalent across this part of the world. One way of solving this problem might be to consider environments relying on mobile devices only, which require less power resources and can be powered from batteries. Also due to the limited number of robots we used for the study, students could only practice their coding skills in the labs where the robots were kept. Students complained about this situation and expressed the desire to own the robots so that they can work on their assignments at the halls of residence as well. Some of these problems might be alleviated by introduction of simulators which could be used outside of the scheduled activities.

6 Conclusions

In this work, we reported on our experience in integrating educational robotics activities into higher education curricula in developing countries. In particular, we investigated methodology for assessing the effectiveness and impact of such activities on student engagement and learning. In future work, we would like to look closer at potential learning gains when using educational robots for teaching specific topics in Computer Science. We would also like to investigate how the effectiveness of such activities changes in longer term. Our results indicate some positive benefits gained from such activities and indicate that educational robotics is a promising tool in developing engaging study curricula. We hope that our experience from this study together with the provided materials resulting from this study will be beneficial to other educators working with educational robotics in different parts of the world.

References

1. Cielniak, G., Bellotto, N., Duckett, T.: Integrating mobile robotics and vision with undergraduate computer science. IEEE Trans. Educ. **56**(1), 48–53 (2013)
2. Mills-Tettey, G.A., Dias, M.B., Browning, B., Amanquah, N.: Teaching technical creativity through Robotics: A case study in Ghana. In: International Conference on AI Workshop on AI in ICT for Development (2007)
3. Gyimah, E.K.: Teachers use of instructional strategies in primary schools in Ghana: implication to inclusive education. Educ. Res. J. **1**(3), 46–52 (2011)

4. Onsongo, J.: Affirmative action, gender equity and university admissions-Kenya, Uganda and Tanzania. Lond. Rev. Educ. **7**(1), 71–81 (2009)
5. Paras, J.: Crisis in mathematics education. student failure: challenges and possibilities. S. Afr. J. High. Educ. **15**(3), 66–73 (2001)
6. Yerion, K.A., Rinehart, J.A.: Guidelines for collaborative learning in computer science. ACM SIGCSE Bull. **27**(4), 29–34 (1995)
7. Akey, T.M.: School Context, Student Attitudes and Behavior, and Academic Achievement: An Exploratory Analysis. MDRC (2006)
8. Papert, S.: The children's machine. Technol. Rev. Manchester NH **96**, 28–28 (1993)
9. Nourbakhsh, I.R., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., Wilkinson, K.: The robotic autonomy mobile robotics course: robot design, curriculum design and educational assessment. Auton. Robots **18**(1), 103–127 (2005)
10. Alimisis, D.: Robotics in education & education in robotics: shifting focus from technology to pedagogy. In: Proceedings of the 3rd International Conference on Robotics in Education, pp. 7–14 (2012)
11. Frangou, S., Papanikolaou, K., Aravecchia, L., Montel, L., Ionita, S., Arlegui, J., Monfalcon, S.: Representative examples of implementing educational robotics in school based on the constructivist approach. In: SIMPAR Workshop on Teaching with Robotics: Didactic Approaches and Experiences, Venice, Italy (2008)
12. Barker, B.S., Ansorge, J.: Robotics as means to increase achievement scores in an informal learning environment. J. Res. Technol. Educ. **39**(3), 229–243 (2007)
13. Rogers, C., Portsmore, M.: Bringing engineering to elementary school. J. STEM Educ. Innovations Res. **5**(3/4), 17 (2004)
14. Touretzky, D.S.: Robotics for computer scientists: whats the big idea? Comput. Sci. Educ. **23**(4), 349–367 (2013)
15. Souza, M.A., Duarte, J.R.: Low-cost educational robotics applied to physics teaching in Brazil. Phys. Educ. **50**(4), 482 (2015)
16. Becker, B.W.: Teaching CS1 with karel the robot in Java. In: ACM SIGCSE Bulletin, vol. 33(1), pp. 50–54. ACM (2001)
17. Gyebi, E., Hanheide, M., Cielniak, G.: Educational robotics for teaching computer science in Africa-pilot study. In: International Workshop on Educational Robots (2015)
18. McNamee, L., Roberts, T., Williams, S.: Online collaborative learning in higher education (2001)
19. Gyebi, E., Hanheide, M., Cielniak, G.: Affordable mobile robotic platforms for teaching computer science at African universities. In: International Conference on Robotics in Education (2015)

Educational Robotics and STEM Education in Primary Education: A Pilot Study Using the H&S Electronic Systems Platform

Maria Stergiopoulou(✉), Anthi Karatrantou, and Chris Panagiotakopoulos

Department of Primary Education, University of Patras, University Campus, 26504 Patras, Greece
pde6804@upnet.gr, {akarat,cpanag}@upatras.gr

Abstract. In this paper an attempt is made to utilize educational robotics applications in Primary Education in order to teach basic principles of Automatic Control Systems and Programming. For this purpose, the robotic package H&S Electronic Systems was used in the frame of the STEM education approach. According to the latter, emphasis is given to the connection of the four subjects, Science, Technology, Engineering and Mathematics (STEM). Educational robotics can be proven to be an important tool to achieve these goals, but also to develop students' motivation to participate in an active way in learning. Within this work the students are asked to work in groups to design, develop and implement their programs to control the behavior of their robotic constructions, following specially designed worksheets. This work, finally, aims to investigate and highlight educational benefits emerging from the data analysis of students' work.

Keywords: Educational robotics · STEM education · Programming fundamentals · H&S electronic systems

1 Introduction

Robotics is defined as the branch of science that studies those machines that can replace humans in performing a task, which combines physical activity with the decision-making process [1]. However, robotics since the late 80 s, starting outside Greece, has been used in education both as a teaching subject, but also, as an auxiliary tool in teaching various concepts of subjects such as Mathematics, Sciences, Engineering as well as Technology and Computer science [2].

Apart from teaching certain concepts, however, robotics is used to achieve skills such as problem solving, specific and abstract reasoning, critical thinking and effective cooperation [3]. For a robotic system to be functional, two types of activities are required: the construction and the programming. In this way, cognitive processes are developed and skills such as computational thinking are emerging supporting students to solve authentic problems in the school environment as well as in real life [2, 4]. Problem solving is a demanding process, which invites the student to deal with logic,

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_7

semantics and sometimes with abstract thinking in order to conceive and solve the problem. Moreover, the construction process of the robotic system and its programming contribute to the socialization of the student through cooperation for the implementation of the activities [2, 3].

In this paper an attempt was made for the use of educational robotics applications in Primary Education, based on the theoretical framework of STEM education. As a means for accomplishing this purpose, the hardware and software environment of the Basic Robotic System platform of the company H & S Electronic Systems was used. For this purpose, educational activities were designed to support teaching of basic principles and structures of Programming and Automatic Control Systems in which students are supposed to work in groups.

2 Educational Robotics and STEM Education

2.1 Educational Robotics

Educational robotics is the process during which, students having at their disposal small robotic systems, they should assemble and program them to perform certain behavior for educational purposes. Therefore, educational robotics from a Pedagogical perspective is considered to be grounded in the theories of classic constructivism [5] and in particular that of the constructionism [6]. The learning environment provides activities imbedded in problem solving procedures and, thus, learners build more effective knowledge as they are involved actively in the design and construction (manual and digital) of real objects that have meaning for them in a more natural way.

Educational robotics combines game with learning and, thus, learning becomes a fun activity that is easier, more interesting and is achieved more efficiently [7]. Mainly in Primary Education, the aspect of game that robotic constructions involve is an important and positive factor of individual action and motivation [8].

What is more, the development of research interest is promoted and children are enabled to act as scientists – inventors and discover their own original ideas and solutions thus enhancing their self – efficacy [9]. By solving authentic problems students are actively involved and support their exploratory attitude having incentives to study Science and Technology.

Students are, also, involved in situations that require from them to use and apply their knowledge from Mathematics, Science, Technology, Engineering [10] gaining the opportunity to develop a strong conceptual basis for the reconstruction of their knowledge at a later time [7].

Educational robotics allows students to express themselves freely and promotes the development of creativity and imagination, as it invites students to envision what they will make and what goal they want to achieve through the programming of their constructions [11]. In addition, the construction of their robotics is a problem – solving situation that provides instant feedback and promotes a multidisciplinary and interdisciplinary approach [12].

At the same time, educational robotics activities are based on cooperation and interaction of individuals and groups, promoting thinking through conflicts in cognitive and social level as children are required to explain ideas, opinions and thoughts and justify their answers [13]. Additionally, students need to analyze, plan and implement their work which constitute high level mental skills. Such activities with younger ages (Primary school) pose some rate of failure, since, according to Piaget, they are in the Concrete Operational Stage and are just moving into the Formal Operational Stage (11, 12 years old). However, dealing with problems for their level, which require concrete thinking, help children familiarize and deal with abstract concepts (e.g. speed, time, variable), having as a result their better preparation for the next cognitive stage [14].

What is more, educational robotics contributes in learning Programming and helps students familiarize with the basic principles of Programming [15]. Programming robotic constructions creates a completely new learning environment which is highly motivating, favors the test – error strategy, which is a strategy familiar to Primary school students, and highlights accepted approaches and solutions, since a behavior can be explained in many ways [3]. Furthermore, it supports metacognitive learning processes, since students need to think on the way they thought and acted [16].

2.2 STEM Education

The term STEM Education and the need for such a method arose for the first time in America. In particular, low performance of American schoolchildren in major subjects such as Mathematics, Engineering, Sciences and Technology pointed out the need for a restructuring way of teaching that focuses on these subjects. The National Science Foundation (NSF) in 2001 launched the term STEM (Science, Technology, Engineering, Mathematics) education for this educational approach [17].

The objectives of STEM education include the development of the scientific interest of children and their capability to solve authentic problems as well. With the development of such skills it is expected that the function of the whole world would be seen through the connection of all four fields. Specifically, it is aimed scientific knowledge to be used for understanding the natural world and decisions that affect it to be made. Moreover, regarding Technology, students to be able to use new technological tools and understand how Technology as a whole affects and modifies the surrounding world. Regarding Engineering, they are expected to understand its significance in real world and to link it with technology, since it is the means for it to become “alive” [18]. Finally, regarding the mathematical field, it is aimed students’ abilities, such as analysis, documentation, solution of problems, etc. to be improved, so that they become able to cope with problematic situations that arise in their daily lives [19].

STEM education refers to an integrated and scientific approach according to which the four fields form a single and indivisible whole where each member interacts and affects one another. The special features of this approach require the use of alternative modes of teaching. Such methods involve traditional teaching by providing guidance from the teacher, use of the project method, laboratory practices and technological tools [20].

Environments that focus on STEM education are characterized by strong leadership, professionalism of teachers, strong links between society and parents, student-oriented methods and the use of educational scenarios for teaching [21]. Additionally, collaborative

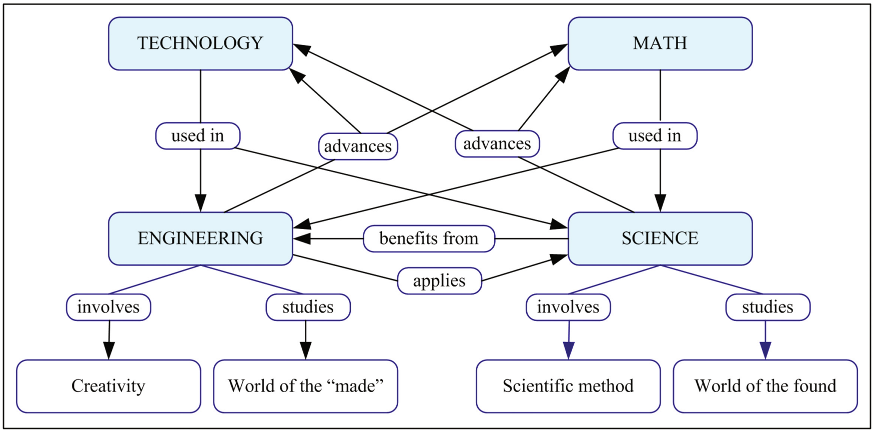


Fig. 1. An Interdisciplinary Approach. (Source: The Thornburg Center for Space Exploration [17])

learning, finding multiple solutions to a problem, failure as part of the learning process, are integral parts of what we call STEM Education [17] (Fig. 1).

For the success of STEM education programs, however, certain parameters have to be taken in concern. These include the existence of a demanding content, a research learning environment, initially defined educational outcomes, clear objectives and commitment and support from society [22].

Programs that have been implemented successfully and meet the above descriptions are: 'Math of the Box' (https://www.youtube.com/watch?v=CqnkgEkChKc), 'Project Lead The Way' (https://www.pltw.org), 'Seeds of Science/Roots of Reading' (http://scienceandliteracy.org), etc.

2.3 Robotics and STEM Education

Robots nowadays have the opportunity to become the next step in education. This is because their innovative character, the hands-on experience they offer and the enthusiasm they cause, make children more receptive to learning stimuli. Apart from this, however, robots have a lot to offer and become a significant educational tool in the STEM approach [9].

Beyond the direct relationship that seems to be more obvious between robotics and Technology and Engineering, as robots are a technological tool and a product of Engineering and Technology, the correlation to the other two areas of Science and Mathematics might not be crystal clear.

Studies in schools have reported that robots helped significantly for skills with fractions and proportions to be improved as well as learning of decimal and ratios to be more interesting and successful. What is more, a large-scale study in Peru showed improvements in problem solving skills and skills of quantity and amount recognitions [23].

Regarding Science, activities that used robots and were oriented in principles of classical physics proved beneficial for students as for example LEGO robots. It is noted to have been used in order the relationship between distance, time and speed to be examined and discussed through the construction, programming and design of motion of the robots [23]. The motivation of the students, their attitude towards learning and their engagement was claimed to be significantly improved [24].

Moreover, content knowledge of the students was significantly strengthened through activities that focused on concepts like friction, forces, weight, the diameter of the wheels and were associated with the basic laws of Newton by utilizing robotic constructions [23]. Other examples in the same direction refer to a solar car, a rubber catapult and a wind-turbine which contributed in teaching the basic forms of energy and energy conversion techniques [25].

Studies that were developed for the analysis of activities related to Physics which required theoretical and practical understanding for their implementation showed that students improved both in their written ability to explain science and in the interpretation of graphs relating to speed, time, acceleration and deceleration by means of robots [23].

In conclusion, the use of robotic constructions has a lot to offer as it has already been mentioned above and to positively contribute to the increase of students' motivation, their engagement in learning, their creativity and their positive attitude towards education and its subjects through problem – solving situations. Thus, the importance of using robots as tools for both the introduction to the robotics field as well as teaching concepts of Programming and STEM approach is indisputable.

3 Platform of H&S Electronic Systems

3.1 Description

The platform of H&S Electronic Systems was developed and manufactured by the Czech professors Jiří Hrbáček and Jan Sýkora (from Brno University – Czech Republic). The systems are designed for educational activities or for leisure activities of pupils of Primary school up to High school level, but also for utilization in industry. The company and its products are not only about electronic systems, but their components, automatic systems, digital technology, a variety of electronic and electrical engineering for use in robotics as well [26].

The basic system (Basic kit) of the company consists of: two reflective plates of short-range infrared sensors and two of longer range. It also includes a board with switches for the robot's manual control, a dual H-bridge (bridge plate), two modules with four and eight LED lights respectively, and, most importantly, the main processor board PICAXE20M2 incorporating a USB host port. To continue, a series of cables and plastic clips, a kit for mounting, two wheels and two motors with a reduction gear and rubber ring are included in the package. Finally, a battery case is available (for four AA batteries), a connecting cable for the computer, and bolts, nuts, a small wrench and a screwdriver for assembling the system to the chassis [26].

With these components, someone can create a simple robotic system. The construction of it can be accomplished with information, images and a step by step guide which are available in the Czech website: http://www.hses.cz in the 'Navody' section.

The robot can be controlled by the control board's buttons or by running the programs that the user has already passed into its memory using the "PICAXE Programming Editor" programming environment. For programming the robot a computer connection is required as well as the installation of the driver for the identification of the device (Fig. 2).

Fig. 2. Basic Kit (Source: http://www.hses.cz/aktualni-nabidka-hs-electronic-systems.html) (left) and the robotic construction used in the study (right)

The use of robotic systems may have a variety of educational purposes. A main example refers to the introduction of children to Programming fundamentals. Applications of these systems can be watched in the websites: http://www.hses.cz/soutez2013.html, http://www.hses.cz/reference.html.

3.2 Main Advantages

It is widely known that many educational institutions at all levels use a variety of robotic packages such as Lego Mindstorms, Boe-Bot, ActivityBot or other platforms like Arduino, for teaching basic automatic control principles and fundamentals of Programming. The reason why this robotic system was chosen for our work is because, among other things, presents a number of advantages.

Firstly, it is a new proposal and a new educational tool. Most educational systems are familiar with the use of Lego Mindstorms or Arduino platform and follow a prescribed course, the usual applications and delivery methods. With the use of a new tool, it can be shown whether there is a real understanding of robotic systems and their programming as well as potential errors that may arise from standardized measures [27].

Secondly, it is a platform that focuses on low-level achievements and on familiarizing beginners with programming environments with the use of simple methods. There are two modes for programming the platform. The first one is by creating a flowchart. It is user friendly and easily manageable which makes it suitable for age groups of Primary school. The second one is by creating a code in a programming language based in BASIC, the eminently programming language for beginners' introduction to the structures and logic of Programming [27] (Fig. 3).

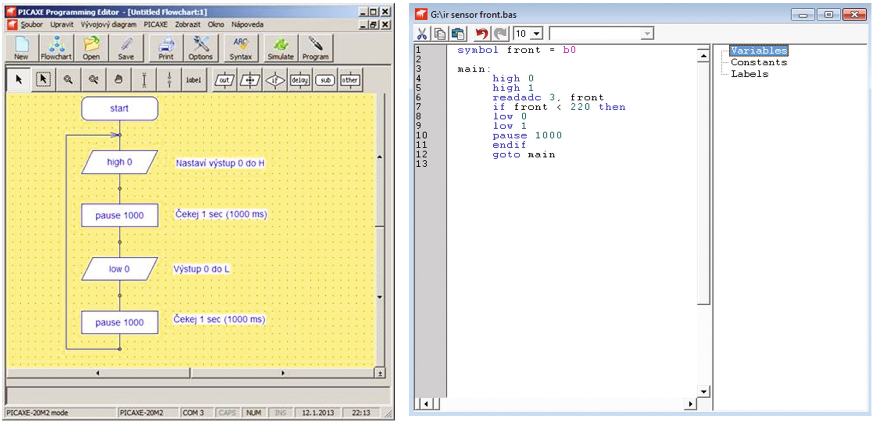


Fig. 3. Program as a flowchart (left) and program in code (right) in Picaxe Programming Editor (Source: http://www.hses.cz/p/prvni-program-3398.html)

Thirdly, it is a very low cost platform. It is considered to be more economical than others and the PICAXE processor can be used as a separate structural component [26, 27].

One of the drawbacks of this robotic system, however, is its external appearance. As mentioned in the description, it consists of a chassis which is the basis on which the other components are mounted. When the robot is finished, it does not look like a humanoid machine, as it happens with Lego Mindstorms for example, and its wires that are placed on it can be seen. Nevertheless, the disadvantage is easily converted into an advantage when it comes to student creativity. It can be challenging for children to build and paint covers for the robot through which it can be transformed into child–friendly objects that they have imagined (a toy, a car, a turtle etc.). Examples can be seen in the website: http://www.hses.cz/kurzy-na-zs.html.

4 Methodology

The educational activity under discussion took place in two public Primary schools of Patras. A total of twelve (12) sixth grade students participated voluntarily in the study. Eight of them were girls (8) and four (4) were boys. This age group was chosen because students at this age have coped with the widest range of subjects and have been taught the necessary concepts that are needed for the completion of the activities described in this paper. The students were divided randomly into groups of three persons (4 groups) and the educational activity took place during two two-hour sessions in the Computers Laboratory of the schools.

The 1st session started with the distribution of a Diagnostic Questionnaire to the students and a discussion on the topic of robotic systems. Subsequently, the first and the second worksheet were given to the students to work. The 2nd two-hour session included work on the third and fourth worksheet and the answering of the evaluation questionnaires by the students. The four worksheets were of increasing difficulty and guided the students to construct and develop more complex constructions and programs.

During the process three methods of data collection were used:

a. monitoring (groups and members) and personal notes of the researchers,
b. recording of groups' discussions,
c. completion of short questionnaires before (Diagnostic Questionnaire) and after the activity (Assessment Questionnaire). The questionnaires were based on similar works [28].

The Diagnostic Questionnaire consisted of eight (8) closed-type questions and students had to give anonymous answers about their age, gender, their previous programming experience, their interest in learning more in Programming and what they mainly used the computer for. Monitoring and personal notes of the researchers as well as the recording of students' group discussions were used to analyze the way students were thinking, discussing, working together, expressing questions/doubts/ideas, and connecting knowledge from different subjects while trying to complete the worksheets each time. The Assessment Questionnaire was used to evaluate the overall procedure (the worksheets, the robotic kit and the programming environment, the use of math and science, the learning outcomes) and consisted of ten (10) closed- type and eight (8) open-ended questions.

Two teachers - researchers were observing the discussions, activities and reactions among students. They kept notes and made interventions when students needed help, adopting a supportive and facilitative role of students' work and learning.

4.1 The Educational Activity

The purpose of the educational activity was to introduce the students to basic concepts of Automatic control systems and concepts, principles and structures of Programming by utilizing the robotics construction presented, as well as to make use of students' knowledge of subjects such as Science and Mathematics. After the end of the activities students were aimed to *be familiar with basic electronic circuit elements, understand a program written in a programming language (that of the environment Picaxe Programming Editor in our case), understand the connection between the subjects of Engineering, Science, Mathematics and Technology, be aware that a robotic system needs to be assembled (hardware) and programed (software), describe the structure and function of a program for controlling the robot, choose the appropriate programming structures to control a robot, recall and make use of knowledge from other subjects in order to design the program for their robot.*

At the same time it was aimed for the students to develop a positive attitude towards innovation and the use of Automatic Control Systems and robotic constructions, become interested in Programming and its applications and to work effectively in teams as well. Special emphasis was given to achieve a cooperative spirit, build trust between students and their teachers, and highlight the need for the development of critical thinking and the continuous discovery of knowledge.

During the 1st session a familiarization phase took place. The Diagnostic Questionnaires were distributed and an introductory discussion about robotics and Automatic control systems as well as a short presentation of the system in use started the session. Subsequently, students started to interact with the robotic construction following the steps of Worksheet 1 and the programming environment Picaxe Programming Editor. Students were asked to make changes to commands in an already prepared program in order for the robot to turn on the LED/LEDs suitably. The programming structures of sequence of commands and looping (with no condition) were introduced to the students.

In Worksheet 2 a new feature was added, the robot's direct movement. For this, changes in the circuit (replacing the wires of the lamps in the power positions with those of the wheels) were necessary. Children examined the circuits of the robotic construction and in order to understand how it works they used knowledge coming from Science and Engineering realizing the importance of the installation of all the appropriate materials on the system (Fig. 4).

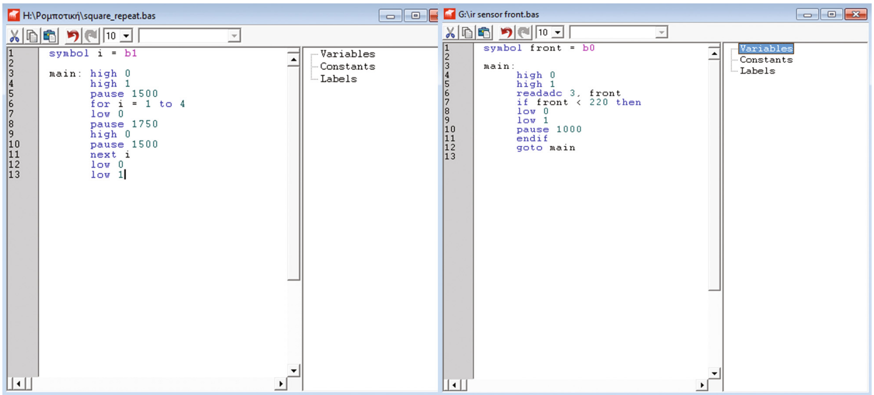


Fig. 4. Program of the 3rd (left) and one of the programs of the 4th (right) worksheet in code

During the 2nd session students worked on Worksheet 3 and Worksheet 4 and the process was closed with the final Assessment Questionnaire. Worksheet 3 focused on robot's movement as well as its ability to turn with a further purpose of creating a square. Students working on this worksheet used mathematical knowledge to solve the problem, and were introduced to the repeat structure (loop) for a certain number of times (FOR loop structure - for creating a square the repetition of certain commands for 4 times was needed) and to the VARIABLE, an important concept in Programming.

In Worksheet 4 a new element in the circuit was introduced, the infrared sensor. Students examined the construction and studied the already written program for one of the sensors placed in the front of the construction. They were watching the behavior of the robot which stopped moving when an obstacle was in from of it closer than 22 cm and they were trying to understand the algorithm the program was based on as well as the suitable commands and programming structures. The program made use of a variable (for the registration of the distance of the robot from the obstacle) and the control structure (selection structure) "IF…THEN…". Subsequently, students had to change the program in order to make use of the second sensor placed in the back (rear) of the robot and program it to stop when an obstacle was present 22 cm behind it and to turn right when an obstacle was present 22 cm in front of it. Knowledge from Physics concerning the infrared radiation, the flow of electricity, etc. was used.

The four worksheets consisted of activities of graded difficulty that required from the students to think about the problem each time, to design the solution (making an algorithm), to work on programs and to modify them suitably, introducing them to more complex structures each time.

For our programs we used the code mode of the PICAXE programming environment, as it was considered easy to learn and to use by the students. Additionally, it was considered more suitable in saving time making short-length programs by using simple commands.

5 Findings

Students' work during the educational activity and their answers to the distributed questionnaires offered important educational information. Regarding the Diagnostic Questionnaire, the majority of the students had "much" to "very much" previously used the computer (12/12), most of them found it quite easy to use (6/12) and all of them would be interested to learn programming (12/12) (Table 1).

Table 1. Results of the Diagnostic Questionnaire (closed-type questions)

	Not at all	Little	Quite	Much	Very much	Sum
Previous use of the computer	0	0	0	4	8	12
Ease of use	0	0	6	1	5	12
Programming experience	0	3	5	2	2	12
Difficulty of programming	0	4	6	0	1	11
Future interest in programming	0	0	0	5	7	12

The majority of students considered "little" to "quite" difficult making a program (10/12). Finally, all students had previous programming experience from "little" to "very much" (12/12) basically with Scratch.

The last question they had to answer (closed-type) was about the reason why they usually used the computer and had to choose between "for fun", "for information', "in order to contact", "to make applications through a programming environment", placing three of their choices in order, according the use they most made. The vast majority of the students mainly use computers "for fun" (8/12), while only three students answered mainly "for information" and only one's first answer was "to make applications through a programming environment".

Analyzing and combining data collected during the whole procedure (personal notes of the researchers, recording groups' discussions) as well as the completion of the worksheets and questionnaires by the students, offered important information about the achievement of the study aims and is presented below.

The discussion that followed the Diagnostic Questionnaire helped students to realize that robotic systems consist of assembled material (hardware) and a program (software). Moreover, that a computer and an appropriate programming environment are necessary for this to be accomplished. Finally, that knowledge from subjects like Engineering, Science, Mathematics and Technology is crucial for these kind of activities. Children seemed to understand the logic and to follow the conversation in both schools and phrases like: *"has tools, wires", "needs to be assembled", "has knowledge that we have put", "to give it a command", "it is connected to the computer", "we need knowledge from Physics, Engineering, Technology for the assembly"* etc.) were noted.

In the first question of the 1st worksheet about what the commands of the program should be in order to light up the LED, children did not answer successfully and focused mainly in the theoretical description of the system. Specifically, three out of four groups (3/4) responded similarly that the system should be connected to the PC with the appropriate wires to launch the board. Only one team answered successfully describing step by step the commands.

However, after discussing it they managed to answer correctly, orally, about the needed program. In the remaining questions of the worksheet ("how can we change the time the LED is on?", "how can we make the LED light on and off continuously", "how can we make use of more LEDs?") three out of four groups answered successfully. The answer of one group on the last question about the use of more LEDs was remarkable, as they actually designed the structure of the program.

According to the 2nd worksheet, students were asked about what the commands should be in order for the robot to move forward (activity 1) to move forward for specific time (activity 2), to move forward for 2 s stop for 3 s and continue moving for another 5 s (activity 3). At the 1st school students worked practically only on the first activity of the worksheet. The time was not sufficient for the last two activities, so a discussion took place while the robot was "acting". In contrast, at the 2nd school all activities were completed. One group completed correctly the program structure. It is worth mentioning that the same group, improvised by creating its own program that combined the use of four power positions, two positions for LED lights and two for the wheels.

In the 3rd worksheet (second two-hour session) the students had to work focused on the movement of the robot as well as its ability to turn with a further purpose of creating a square. Finally, they had to modify the final program for the formation of two squares, for four or for continuous formation of squares.

The groups' answers about the commands for the robot's turn were not successful and two out of four noted the command "turn" which does not exist in Picaxe Programming Editor, but in Scratch environment. However, after discussion and practical trials the students were able to use the new concepts and structures for the square formation quite satisfactorily. In this way, they were introduced to the repeat structure (loop) for a certain number of times (FOR loop structure) and to the concept of VARIABLE.

In the 4th worksheet the infrared sensors were used. Children were, firstly, asked to think of the appropriate commands in order to immobilize the wheels when the robot perceived an obstacle in front of it in range less than 22 cm. Secondly, they had to make use of the second sensor which, when it perceived an obstacle with the same conditions behind it, the robot should also stop.

Thirdly, they were asked to modify the program so that instead of the robot getting immobilized, when it perceived an obstacle in front of it, to turn. The 4th worksheet was considered the most demanding. Despite the difficulties, however, they seemed to adequately follow the logic, discovering the command “IF…THEN…”, and using it in an appropriate way to complete the activities of the worksheet (Table 2).

Table 2. Results of the Assessment Questionnaire (closed-type questions)

Questions	1[a]	2[a]	3[a]	4[a]	5[a]	Sum
Did you enjoy the activities?	0	0	1	2	9	12
How easy were they?	0	0	7	4	1	12
Would you like to concern yourself with such activities in the future?	0	1	1	1	8	11
Was Picaxe Programing Editor understandable?	0	0	2	5	5	12
Are you more interested in programing now?	0	0	1	4	6	11
Would you like to make your own programs with Picaxe Programming Editor?	0	0	1	7	4	12
Did your knowledge from Science and Mathematics help you do the activities?	0	0	0	6	6	12
Did technology (the robotic system) help you understand concepts of Science and Mathematics?	0	0	3	8	1	12

[a] *1 = Not at all, 2 = Little, 3 = Quite, 4 = Much, 5 = Very much.*

At the end, an evaluation questionnaire was given to every group member. The majority of students enjoyed the activities “much” to “very much” (11/12). When asked whether their knowledge from Science and Mathematics contributed in their work, all the answers (12/12) were positive ranging “much” to “very much”, although in the question whether the robotic system contributed to the understanding of Science and Mathematics concepts the majority ranged from “quite” to “much” (11/12) with only a single child feeling that it contributes “very much”.

The open-ended questions were seeking information about what the students learned during the educational activity and at what extent the objectives of the study were achieved. Regarding the most difficult part of the activity, nine (9) out of twelve (12) children mentioned, as expected, the last worksheet. The majority of the participants (10/12) seemed to understand the role of the repetition structure and to realize how “IF” works (9/12), although some of them connected it directly with the function of the sensors used. They were, also, able to recognize repetition (8/12) and selection (9/12) structures in our programs (Indicative answers: *"When we want our commands to be repeated", "When we want the robot to do something for ever", "When we want to show something hypothetical", "when we want to show that what we say, to be done", "for sensors").*

The next question required the recognition of three elements on the circuit of the robotic construction and an explanation of their use. The majority of the students (9/12) we can say that answered satisfactorily recognizing and describing the function of the LEDs, the cables, the sensors, the batteries and the processor. Furthermore, regarding the contribution of Science and Mathematics as well as Technology in the activities, a variety of answers were given like: *"Knowledge from the chapter of "Light" helped me", "the energy, the cables", "the opened and closed circuit", "electricity" and "geometry and multiplication", "the distance and angles", "numbers", "addition and subtraction".*

6 Discussion and Conclusions

The educational activity described aimed to introduce students to Programming fundamentals and basic principles of Automatic control systems and to investigate whether they can understand the connection between Science, Technology, Engineering and Mathematics while dealing with such activities.

Data collected by monitoring students' work and recording group discussions as well as personal notes of the researchers and the completed questionnaires offered important information about the achievement of the study aims.

Special effort was made to ensure that the educational activity met with all the features and suggestions, from a pedagogical aspect, of the STEM approach. Worksheets with pre-defined objectives were used containing activities with graded difficulty, meaning that each activity added something more or required the combination of individual parts and, thus, was characterized more complex than the previous one. Students had the opportunity to work in a laboratory experimental learning environment with hands-on programming activities that required from them to solve problems in a constructionism framework.

During and at the end of the activity most children were able to identify key parts of the circuit of the robotic construction (e.g. LEDs, wires, sensors) and to effectively describe their function based on knowledge from Physics as well as to make use of mathematical knowledge and combine them with algorithmic thinking in order to make a program for the effective control of their robotic construction.

They were introduced and used appropriately basic programming structures (sequence, repetition, selection) and concepts as the VARIABLE while designing and developing a program.

According to the students' answers to the questions of the questionnaires as well as to their answers during discussion, they seemed to be familiar with the connection among Engineering, Science, Mathematics and Technology. All students thought that Mathematics and Science and their previous knowledge from these subjects contributed to the implementation of the activities. In addition, everyone considered themselves been helped by the robotic system in understanding concepts of these subjects, although on a smaller scale. Consequently, it seems that they understand the value and contribution of each subject to the other and recognize the existence of links between them, even though the degree of this association varies.

Moreover, all children answered positively in the questionnaires regarding the increase of their interest in Programming, their involvement in such kind of activities and their desire to create their own programs. We can say, therefore, that there are great chances of forming a positive attitude towards Programming and its applications.

They also had the opportunity to develop a positive attitude towards innovation and the use of technology by working with simple automatic control systems and robotic constructions.

During the whole intervention, students, according to Papert's ideas, had new tools and technologies at their service, they expressed and experimented with ideas that they formed through different media, they dealt with "dive – into" situations [29]. By doing all these, they gained a better understanding about several subjects and apart from the digital construction they went through (the programs they made), children could be supported to have constructed part of their knowledge. At the same time, it is also supported that they achieved a cooperative spirit, they developed critical thinking and trust between students and their teachers was built.

In conclusion, the results of this study due to the small number of students participated cannot be generalized. However, they can be considered indicative and may be used for the design and the implementation of educational activities, which make use of educational robotics applications in a STEM education environment.

References

1. Doulgeri, Z.: Robotics: Kinematics, dynamics and control of articulated arms. Kritiki, Athens (2007). (In Greek)
2. Karatrantou, A., Panagiotakopoulos, T.C.: Educational robotics and teaching introductory programming within an interdisciplinary framework. In: Jimoyiannis, A. (ed.) Research on e-Learning and ICT in Education. Springer, New York (2011)
3. Alimisis, D. (ed.): Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods, School of Pedagogical and Technological Education (2009)
4. Karatrantou, A., Panagiotakopoulos, T.C.: Algorithm, pseudo-code and Lego Mindstorms programming. In: Proceedings of International Conference on Simulation and Programming for Autonomous Robots/Teaching with Robotics: Didactic Approaches and Experiences, Venice, Italy, pp. 70–79 (2008)
5. Piaget, J.: The Principles of Genetic Epistemology. Basic Books, New York (1972)
6. Papert, S.: The Children's Machine. Basic Books, New York (1993)
7. Eguchi, A.: Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, Padova, Italy, 18 July 2014, pp. 27–34 (2014)
8. Panagiotakopoulos, T.C.: Applying a conceptual mini game for supporting simple mathematical calculation skills: students' perceptions and considerations. World J. Educ. **1**, 3–14 (2011)

9. Barker, S.B., Nugent, G., Grandgenett, N., Adamchuk, I.V.: Robots in K-12 Education: A New Technology for Learning. Information Science Reference. IGI Global, United States of America (2012)
10. Alimisis, D., Karatrantou, A., Tachos, N.: Technical school students design and develop robotic gear-based constructions for the transmission of motion. In: Eurologo Conference "Digital Tools for Life Long Learning", Warsaw 27-31/08/2005
11. Mubin, O., Stevens, J.C., Shahid, S., Mahmud, A., Dong, J.: A review of the applicability of robots in education. Technol. Educ. Learn. (2013). doi:10.2316/Journal.209.2013.1.209-0015
12. Arlegui, J., Menegatti, E., Moro, M., Pina, A.: Robotics, Computer Science curricula and Interdisciplinary activities. In: Workshop Proceedings of SIMPAR International Conference on Simulation, Modeling and Programming for Autonomous Robots, Venice, Italy, pp. 10–21 (2008)
13. Kiriakidi, F., Taraperas, C., Karatrantou, A., Panagiotakopoulos, T.C.: Educational robotics in support of teamwork activities in primary education. In: 6th National Congress of Computer Science (2012) (In Greek)
14. Feldman, R.S.: Developmental Psychology, Development Across the Life Span, vol. 1. Gutenberg, Athens (2009) (In Greek)
15. Papadanelis, G., Karatrantou, A., Panagiotakopoulos, T.C.: Utilization of lego mindstorms NXT for teaching programming: the concept of the variable. In: Proceedings of the 6th Panhellenic Conference "Didactics of Computer Science" (2012) (In Greek)
16. Alimisis, D., Arlegui, J., Fava, N., Frangou, S., Ionita, S., Menegatti, E., Monfalcon, S., Moro, M., Papanikolaou, K., Pina, A.: Introducing robotics to teachers and schools: experiences from the TERECop project, Constructionism 2010, Paris, France, 16–20 August 2010
17. Hanover Research: Best Practices in Elementary STEM Programs, March 2012. http://school.elps.k12.mi.us/ad_hoc_mms/committee_recommendation/4.pdf. Accessed
18. Thornburg, D.D.: Why STEM Topics are Interrelated: The Importance of Interdisciplinary Studies in K-12 Education. p. 3. Thornburg Center for Space Exploration (2008)
19. National Governors Association Center for Best Practices: Building a Science, Technology, Engineering and Math Agenda (2007). http://www.nga.org/files/live/sites/NGA/files/pdf/0702INNOVATIONSTEM.PDF. Accessed
20. Means, B., Confrey, J., House, A., Bhanot, R.: STEM High schools: specialized science technology engineering and mathematics secondary schools in the U.S. Bill and Melinda Gates Foundation Report. SRI International. (2008)
21. National Research Council: Successful K-12 STEM Education: Identifying Effective Approaches in Science, Technology, Engineering, and Mathematics. Committee on Highly Successful Science Programs for K-12 Science Education. Board on Science Education and Board on Testing and Assessment, Division of Behavioral and Social Sciences and Education. The National Academies Press, Washington, DC (2011)
22. Bayer Corporation: Planting the Seeds for a Diverse U.S. STEM Pipeline: A Compendium of Best Practice K-12 STEM Education Programs (2010)
23. Karim, M., Lemaignan, S., Mondada, F.: A review: can robots reshape K-12 STEM education? In: Proceedings of the 2015 IEEE International Workshop on Advanced Robotics and its Social impacts (2015)
24. Robinson, M.: Robotics-driven activities: Can they improve middle school science learning? Bull. Sci. Technol. Soc. **25**(1), 73–84 (2005)
25. Saygin, C., Yuen, T., Shipley, H., Wan, H., Akopian, D.: Design, development, and implementation of educational robotics activities for k-12 students. In: Proceedings of American Society for Engineering Education Annual Conference & Exposition (2012)
26. H&S Electronic Systems. http://www.hses.cz

27. Černý, M.: Arduino versus PICAXE (2014). http://robodoupe.cz/2014/arduino-versus-picaxe/. Accessed
28. Michalopoulos, P., Bania, S.: Use of the training package robotic Arduino construction with Scratch (S4A) in the educational process. Graduate Thesis. School of Pedagogical and Technological Education (2015) (In Greek)
29. Ackermann, E.: Piaget's constructivism, Papert's constructionism: What's the difference? In Future of Learning Group – MIT Media Lab (2001)

Duckietown: An Innovative Way to Teach Autonomy

Jacopo Tani(✉), Liam Paull, Maria T. Zuber, Daniela Rus, Jonathan How, John Leonard, and Andrea Censi

Massachusetts Institute of Technology, Cambridge, MA 02139, USA
jtani@mit.edu
http://duckietown.mit.edu

Abstract. Teaching robotics is challenging because it is a multidisciplinary, rapidly evolving and experimental discipline that integrates cutting-edge hardware and software. This paper describes the course design and first implementation of *Duckietown*, a vehicle autonomy class that experiments with teaching innovations in addition to leveraging modern educational theory for improving student learning. We provide a robot to every student, thanks to a minimalist platform design, to maximize active learning; and introduce a role-play aspect to increase team spirit, by modeling the entire class as a fictional start-up (Duckietown Engineering Co.). The course formulation leverages backward design by formalizing intended learning outcomes (ILOs) enabling students to appreciate the challenges of: (a) heterogeneous disciplines converging in the design of a minimal self-driving car, (b) integrating subsystems to create complex system behaviors, and (c) allocating constrained computational resources. Students learn how to assemble, program, test and operate a self-driving car (*Duckiebot*) in a model urban environment (*Duckietown*), as well as how to implement and document new features in the system. Traditional course assessment tools are complemented by a full scale demonstration to the general public. The "duckie" theme was chosen to give a gender-neutral, friendly identity to the robots so as to improve student involvement and outreach possibilities. All of the teaching materials and code is released online in the hope that other institutions will adopt the platform and continue to evolve and improve it, so to keep pace with the fast evolution of the field.

Keywords: Duckietown · Autonomous vehicles · Educational robotics · Active learning · Constructive alignment · Backwards design

1 Introduction

Autonomous cars are a quickly maturing robotics application being researched and developed by all major automotive companies. The need for autonomy engineers drives academia to develop training methods that effectively meet the teaching challenges of this developing field.

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_8

Teaching autonomy is challenging due to factors involving *what* is taught, *how* it is taught and with what *resources*. Robotics is an interdisciplinary field: a student approaching the design, construction and operation of a robot will require theoretical and practical knowledge of mathematics, physics, controls, computer vision, as well as mechanical, software and electrical engineering. Additional challenges arise from the intimate blending of hardware and software required to make a robot work. Sensors and actuators are constantly improving and algorithmic approaches to solving specific tasks, such as path planning, localization, image processing, and many others are active areas of research and are continually evolving.

The strong experimental component of robotics invalidates the traditional "frontal" approach to teaching where students are passively fed notions. Modern educational theory has identified alternative strategies that increase the students' long-term knowledge retention and transfer to practical problems [1–4]. *Backwards design* [5], *constructive alignment* [6] and *active learning* [1], are focused on student learning rather than the instructor teaching. Learning is more effective when students actively engage in activities designed to reach the intended learning outcomes (ILOs) [1,2]. In particular, generation Y (or "Millennials") do not respond well to classical teaching methods [7]. Team-oriented and more informal approaches to learning achieve better outcomes, and students find new teaching methods that incorporate multimedia and social networking more appealing [8].

In this paper we describe *Duckietown*, our approach to address the new educational demands of autonomy. Duckietown is an engaging and affordable robotics teaching, outreach and research platform. It was taught for the first time during Spring 2016 at the Massachusetts Institute of Technology. We first introduce the Duckietown platform, comprising of *Duckiebots* and *Duckietowns* (Sect. 2), highlighting their modularity and minimality. We then detail the course design (Sect. 3) with attention to the *intended learning outcomes*, *teaching and learning activities* and *assessment tools*. We then describe an innovative way of creating a broader team spirit among instructors and students by introducing a role-playing aspect to the course (Sect. 4). Finally we describe the results of the first implementation of Duckietown as a vehicle autonomy class (Sect. 5), followed by concluding remarks (Sect. 6).

We have released all teaching and other materials as open-source on the website duckietown.mit.edu, in the hope that in the future this could become a standardized platform for high school, undergraduate, and graduate level robotics education.

2 The Duckietown Platform

The Duckietown platform [9] comprises self-driving vehicles (*Duckiebots*) and model urban environments (*Duckietowns*).

The **Duckiebot** is a minimal platform consisting of mostly off-the-shelf components. *Computation* is performed on-board on a Raspberry PI 2 (RPi2), strong of 4 900 MHz ARM cores and 1 GB of RAM. *Actuation* is provided through two

DC motor controlled wheels, included with the frame in the chassis kit. Wheel odometry is not used for estimation; therefore, the chassis is the most fungible part of the robot and can be replaced by any setup that uses two DC motors in differential drive configuration. The Adafruit DC motor board is designed to attach directly to the RPi2. *Sensing* is provided exclusively through a high definition camera. The RPi2 supports a custom-designed camera via a dedicated parallel connection. For setup, debug, and optional manual control, *communication* to the Duckiebot is managed through WiFi. For large deployments a solution to network saturation is an onboard access point on each Duckiebot. These mobile hotspots create a 5 GHz network, are powered independently, and connect to the RPi2 through ethernet. Moreover, a wireless joystick can be used for making manual control more convenient. Duckiebots can also be equipped with RGB LEDs to enable global behaviors that require inter-vehicle communication (Fig. 1).

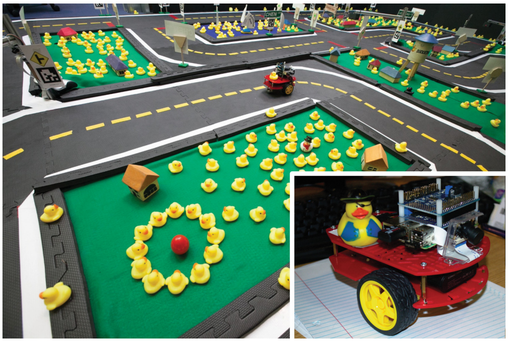


Fig. 1. *Duckiebots* are minimal autonomy platforms designed to operate in *Duckietowns* of arbitrary topology.

Duckietowns are structured urban environments in which Duckiebots are designed to operate. A Duckietown floor is made by *roads*, and above the floor are *signals* (i.e., *signs* and *traffic lights*). These layers enable:

- *modularity* that allows for construction of road topologies of variable complexity;
- stringent *appearance guidelines* that ensure consistency;
- *low-cost* components for universal adoption;
- *scalability* of functionality that provides teaching and research pathways;
- opportunities for *customization* that enhance student ownership.

Floor Layer - Roads: The road level of a Duckietown is built using interlocking foam tiles. Three tile types exist: (i) straight roads; (ii) 90° turns and (iii) intersections. Intersections must be abutted on all sides by straight roads, and may be (iii-a) T junctions (3 way) or (iii-b) crossroads (4 way).

Signal Layer - Signs: are present in three types: (i) traffic signs, (ii) street names, and (iii) localization signs. All three sign types utilize AprilTags [10] to convey information to the Duckiebots. The use of AprilTags in unison to images/text allows for a progression of functionality and programming skill sets. Such scaling in difficulty and functionality are a cornerstone of Duckietown as an educational and research tool. The appearance and location of signs is formalized [9] to ensure recognition by Duckiebots.

Signal Layer - Traffic Lights: Traffic lights are designed as a “car without a chassis” and therefore utilize the same hardware and software as the Duckiebot. They produce signals (i.e., stop and pass) encoded in the blinking frequencies of the LEDs, which are more robustly detected than color-based coding. This architecture provides the opportunity to network the traffic lights and scale the problem to include investigation of real time traffic monitoring, smart infrastructure, and traffic flow optimization.

If the rules for city construction [9] are followed, a Duckiebot will navigate Duckietowns of arbitrary topology by a sequence of: (a) navigating one or more straight tiles until a red strip appears; (b) waiting for a coordination signal; (c) executing an open-loop motion; and (d) re-localizing on a straight tile.

2.1 Why Duckies? A Friendlier Image for Autonomous Vehicles

Duckies are entirely non-functional, purely decorative, yet essential. Robots are typically thought of as dangerous, strong, fast, aggressive and unpredictable. We designed Duckiebots instead to be safe, weak and slow. Moreover, through the use of the *duckie* theme, they are perceived as curious, friendly and fun. The Duckietown in our laboratory attracts plenty of attention, even when there are no robots (Fig. 2d).

We also designed Duckiebots so not to suffer from preconceived gender stereotypes by using a *gender-neutral approach* in the choice of colors, symbols and themes. Humans are very sensitive to gender cues, even in inanimate machines like robots. Previous work has explored the role of perceived robot “gender” during human-robot interaction [11]. For example, robots are considered “male”

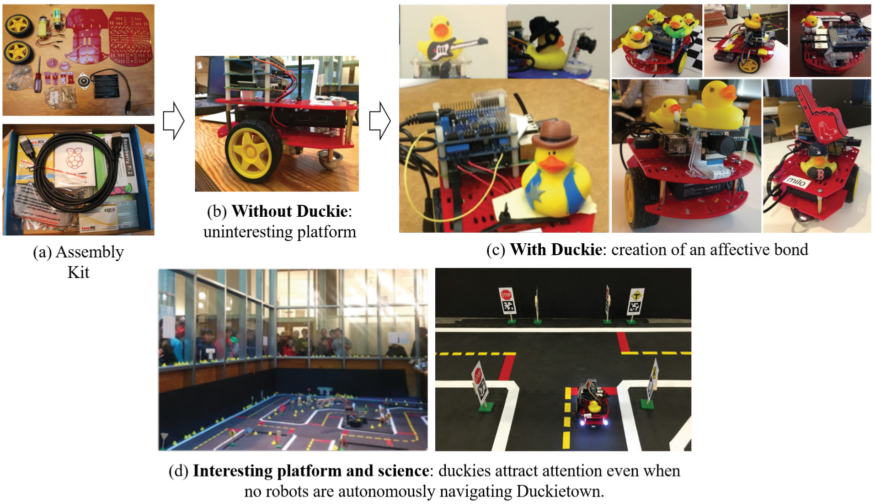


Fig. 2. Duckies make the difference between a boring, cheap platform (b) and an object with which the students can form an emotional bond (c).

if their function is dangerous. "Male" robots are less appealing to prospective female roboticists [12].

3 Course Design

In designing this course we adhere to the practice of *backwards design*, where the final educational objectives are considered first and condensed in a list of intended learning outcomes (ILO) [13]. The teaching and learning activities (TLAs) are then crafted to align with the ILOs but also be actively engaging for the students. Active learning eases understanding, long-term retention and transfer of knowledge from the classroom to the practical applications. Assessment tools were distributed throughout the course to allow for feedback exchange between students and staff in order to improve their convergence to the ILOs. The course has the final objective of making students:

- appreciate how *heterogenous disciplines* such as control theory, machine learning, computer vision, and artificial intelligence are integrated into a complex system;
- design and implement subsystems that work well together (problem of *integration*);
- analyze the constraints imposed by limited on board resources (e.g., computation, sensing, actuation) and operation environments, and design trade-offs to maximize performance while minimizing shared resources use (problem of *co-design*).

3.1 Intended Learning Outcomes

Assuming as students prerequisites: the ability to program, minimal Linux knowledge, and an undergraduate-level class in perception or control; we divide the ILOs in four categories: *operational tools*, *development methods*, *autonomy (perception, control, coordination)*, and *dissemination/documentation*.

By the end of the course, students are able to:

1. Use **operational tools** to:
   (OP-Construction) Build a Duckiebot, given the hardware and a set of instructions. Includes hardware assembly;
   (OP-Configuration) Configure the software and network as well as calibrate their actuators and sensors. Includes Linux and Robotic Operating System (ROS);
   (OP-Operation) Demonstrate correct operation of their Duckiebot.
2. Master **system development methods**, such as:
   (DEV-ROS) Develop ROS software modules and integrate them in the system;
   (DEV-Tools) Utilize *standard tools* for software development (e.g., source code repositories; branching and merging);

(DEV-Test) Utilize the best practices of system development, including *test-driven* and *data-driven* development;
(DEV-Group) Familiarize with the dynamics of open-source development, including the challenges of integrating independently-developed functionalities.

3. Implement new features in a system demonstrating **autonomy** capabilities, such as:
   (AU-Perception) Image processing, Bayesian filtering, localization;
   (AU-Control) Navigation, control, coordination;
   (AU-Behaviors) Integrate perception and control into complex behaviors.
4. Achieve effective **dissemination** by:
   (DIS-Explain) Explaining design choices and trade-offs;
   (DIS-Document) Documenting their work, by creating step-by-step instruction sets to enable future users to reproduce their results;
   (DIS-Demonstrate) Disseminating the results of their work through a demonstration to the general public.

3.2 Teaching and Learning Activities

The course structure (Table 1) is designed to gradually introduce TLAs that facilitate achievement of the ILOs. TLAs are a mix of traditional *lectures* and *active learning tasks*. Lectures (Table 2) cover topics that introduce autonomy

Table 1. Overview of the course.

Phase	Activities	Educational intent	Addressed ILOs
I: Autonomy tutorial	Course and staff introductions	High level overview	OP-Construction, Configuration
	Baseline system development	Bring all up to speed	OP-Construction, Configuration, Operation DEV-ROS, Tools
II: Team projects	Team-projects I: Features development	Creation of fundamental capabilities	OP-Configuration, Operation DEV-ROS, Tools, Test, Group DIS-Explain, Document, Demonstrate
	Team-projects II: Behaviors development	Creation of complex interactions	
III: Public demonstration	Demo: Open House	Dissemination, Assessment	OP-Configuration, Operation DIS-Demonstrate
	Wrap up of demo	Documentation formalization	DIS-Explain, Document

Table 2. Course lectures.

Phase I: Autonomy tutorial	Phase II: Team projects	Addressed ILOs
Image formation process, Camera calibration, Lane detection and filtering, Nonlinear Filtering	Target tracking and intersection estimation, Object detection, Imperfect data analysis, State estimation from motion blur	AU-Perception
Robust localization and mapping, Motion Planning	SLAM, Global localization, Navigation, Visual inertial navigation	AU-Control
	Autonomy architectures, Safety & correctness, Advanced safety and formal methods	AU-Behaviors
Software architecture (ROS)		OP-Construction DEV-ROS

and more specific content that informs students in designing features and global behaviors of Duckiebots. Staff provide hands-on mentorship in laboratory sessions. Additional active learning tasks include *group-based projects*, *presentations* to the class and *public demonstrations*.

The course is divided in **three phases** (Table 1): *Phase I*: is the *autonomy tutorial*, where TLAs are focused on classroom lectures and laboratory sessions; *Phase II*: is mostly characterized by *team-based* and *project-based* work; in *Phase III*: finally students integrate findings of the previous phase to deliver a public demonstration.

Phase I – Autonomy Tutorial. This phase provides a tutorial in autonomy and bringing all personnel (students and staff alike) up to speed on the basic operational tools necessary to contribute to the project.

Core activities such as the setup of computers and accounts for shared resources, assembly of Duckiebots and creation of ROS modules (homework) are addressed during this phase. During this phase the following TLAs take place: (a) classroom lectures; (b) laboratory sessions; and (c) homework.

During *classroom activities* the lectures (Table 2) focus on the theoretical foundations of the practical applications of vehicle autonomy. Throughout this phase, the following topics are presented: robotics architecture (ROS) [14], image formation process [15], camera calibration [16], minimal sufficient representations for visual tasks, nonlinear filtering [17], robust localization and mapping [18], and motion planning [19].

The topics treated in class are then explored through implementation in *laboratory* sessions. Each student *and* staff member (Fig. 3) is provided with a *Duckiebox* with their own robot components and instructions on know how to

Table 3. Team projects during the two iterations of Phase II and examples of future development.

Teams 1st iteration: Features development	Teams 2nd iteration: Behaviors development	Examples of future development
Illumination invariance	Parallel autonomy	Manipulation
LED detector, traffic lights	Object/Vehicle avoidance/following	Inter-bot wireless communication
Odometry calibration from sensor measurements	Traffic light coordination	Model-based control
Lane filtering and control	Stop sign coordination	Vehicle passing
Vehicle detector	Localization and mission planning	Smart infrastructure
AprilTags detection	Robust illumination invariance	Optimal co-design
Local object detection and avoidance		Mobility on demand
Bumpers and shells design and manufacturing	Bumpers and shells design and manufacturing	Safety guarantees

assemble it. *Everyone* has to be able to operate their Duckiebot and learn how to develop features on it. Students alternate between working individually and with the support of senior staff and peers to problem-solve their way through assembling, calibrating and testing their Duckiebots. While students follow instructions prepared by the mentors, they comment and modify them when better or alternative methods are found through experimentation or individual expertise. The contributions of students to the documentation remain beyond the course for the benefit of future Duckietown users through the open source paradigm.

The *Homework* consists of reproducing, customizing and implementing software modules with the aim of preparing for future system development. The first phase ends with an autonomous navigation test, where each Duckiebot drives around a loop to pass.

Phase II — Team Projects. This phase includes two independent iterations of feature development. In each iteration, students are required to form teams of 2–5. Each team has 1–2 mentors. Mentors are typically senior students and staff.

At the beginning of each iteration, students are provided with project-oriented intended outcomes with technical specifications (e.g. the maximum allowable latency is 50 ms). The specifications for different teams might be inter-functional and inter-team communication and cooperation are necessary.

Each team is then free to *self-organize* to achieve the intended result. Examples of project-driven TLAs include: (a) design meetings; (b) project-based work (individual or not: team is self-organized); (c) inter-group communication/cooperation; and (d) presentation of design document to class.

At the end of each iteration, groups document their design choices in an analysis document, write step-by-step instructions for reproducing their results, prepare a demonstration of the developed capabilities and present the project to the class.

The two group-based iterations differ in the hierarchy of the developed functions. During the first iteration, groups develop *basic features* and during the second, *global level behaviors* that involve integrating different basic features. Table 3 contains the project topics and examples of potential future development directions.

Classroom activities support this phase by focusing on more advanced material than in Phase I (Table 2). The topics introduced are: navigation [20], safety and correctness (intersections navigation) [21], advanced safety and formal methods [22], multi-vehicle coordination [23], simultaneous localization and mapping (SLAM) [24], estimation from motion blur [25], distributed systems [26], and visual inertial navigation [27].

Phase III – Public Demonstrations. In this phase groups showcase their work during a public demonstration event. Interacting with the general public is an opportunity for students to act as teachers and appreciate the relevance of their work through the general public's feedback. During this phase the following TLAs take place: (a) demo planning; (b) demo day; and (c) demo documentation.

On the day of the demo, students (and mentors) are in charge of integrating and deploying the previously developed functions and achieved behaviors to the general public. Every group demonstrates their own work by building an independent Duckietown, which they have to design and build to specification. After having prepared documentation for dissemination (demo-specific posters, informative brochures, flyers, ...) students operate their stations by continuously running the demonstration, often interactive, while explaining their work. After the event, students archive their material and provide step-by-step instructions for demo reproduction.

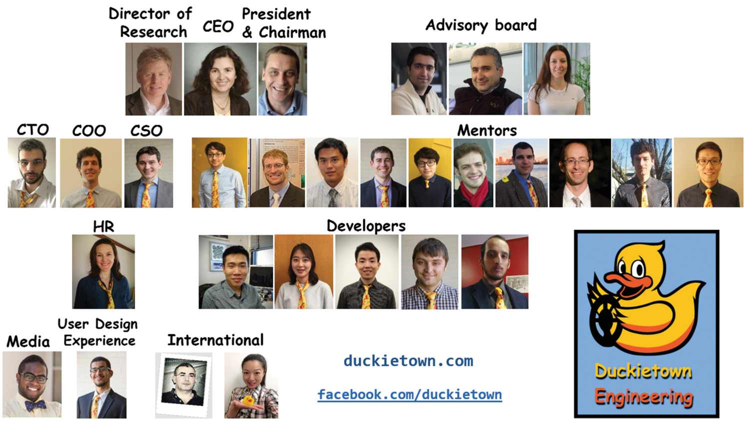
3.3 Assessment Tools

The achievement of the ILOs is assessed mainly through *peer evaluation* and there is no formal test or examination students need to take throughout the course. This works as an incentive for teamwork and general involvement allowing for a more holistic evaluation of the student's learning and contributions.

Grades are assigned through a weighted average of problem sets (30%), participation (35%) and projects (35%). Projects grades were further specified by: team-based grading (70%) and individual grading (30%).

Homework modules in Phase I are graded by the instructors. Instructors are also responsible for providing grades for students' involvement, productivity and quality of work during the project-based work of Phase II. Projects require a *demo*, a *presentation* to the class, and a *project report*. During the presentations students evaluate and leave feedback for each other through online anonymous forms.

At the end of the course, students and staff answer a questionnaire about the value of every other member, including the mentors, for each team they have participated in. The questionnaire asks for open comments and a numerical evaluation of their "open source" participation. This score takes into account diverse contributions to the project, e.g.: (a) improving and (b) documenting



Designation	Titles	Roles	People
Executive	Board Members	Invited lectures	Principal Investigators (PIs)
	Advisors	Invited lectures	PI, Postdoctoral Researchers
Staff	Chief Technical Staff	Instructors, head of daily operations	PI, Postdoctoral Researchers
	Engineers	Mentors	Postdoctoral Researchers
	Human Resources	Accounts, general assitance	Non technical
	Media	Recording of classes and events	Non technical
	Assitants	City building, inventory management	Undergraduates
	International Operations	Enabling implementations abroad	Non-MIT affiliates
Students	Engineers in Training	Students	(Under) Graduates

Fig. 3. *Duckietown Engineering Co.*: a fictitious company created to provide a broader sense of team spirit. A "Duckietie" dress code was enforced for on-duty staff.

the software provided; (c) adding functionality; (d) taking logs; (e) helping other members of the team; (f) running demos.

Interactions among students and mentors are chronicled in the open source platforms used to manage the documentation and code, facilitating objective evaluation.

4 The "Duckietown Engineering" Role-Play

Teamwork is fundamental for accomplishing complex tasks and robotics is no exception. While Phase II of the course (Sect. 3.2) enables cooperation in the class and labs, we implemented a role-play aspect to Duckietown to ensure that teamwork was an integral part of the Duckietown learning experience. When enrolling in the course, students automatically join a fictitious company, *Duckietown Engineering Co.*, with the title of "autonomy engineers in training". The *Duckietown Engineering* brand includes a logo, a website with an official roster, as well as LinkedIn and Facebook pages. The staff are the "senior management" and the supervising professors are "board members" or "advisors" (Fig. 3). This role-play serves the functions of setting a whimsical mood, while communicating a clear message: the *team* includes the *whole company*; students, staff and board directors working towards the common objective of *learning robotics*.

5 Feedback from the First Implementation (MIT, Spring 2016)

Duckietown was first offered as a graduate-level class, officially known as "MIT 2.166: *Autonomous Vehicles*". It has exceeded our expectations because of:

- the large number of student applications (67) show the appeal of concept and topic;
- the proactivity of students and staff, who over-delivered in effort and results;
- the production of over fifty functional, standardized robots;
- the detailed documentation formalizing every step for reproducing the results;
- the media coverage and public interest (53 k views on YouTube at time of writing).

5.1 Teaching Staff

This first edition of Duckietown counted 34 staff, 13 of which were post-doctoral associates (Postdocs). Postdocs were recruited both before and during the course and performed most of the teaching activities. By volunteering their time and competences they gained: (a) *teaching experience*, by lecturing the class on their field of expertise; (b) *leadership experience*, by mentoring student teams in the development of features, (c) opportunity for *scientific publications*, enabled by having access to a complex experimental setup, and (d) the opportunity to gain detailed knowledge of a *customizable* and *open source* course that can potentially be used in their future academic careers.

5.2 Student Demographics

Applications were solicited, through a questionnaire, for 12 spots. We received 67 applications from students with backgrounds mainly in mechanical engineering and computer science departments. In addition to standard questions about educational background and previous experiences, each response needed to contain two essays: "Why would you like to take this course?" and "What can you contribute to Duckietown?". Some of the responses are shown in Fig. 4 and highlight the students' enthusiasm for joining the team. Out of 27 accepted students (1 dropped the course), eight were women (30%), which is significantly higher than the proportion of female graduate students in the Electrical Engineering and Computer Science Department at MIT (20%).

- […] **In short please please please let me take this class!** Quack!
- […] **This class is really the best class I've ever seen […]** All the things about the Duckies are cool as it shows that you've had (and will have) a lot of fun creating this class. But I was even more impressed by what is actually going to be thought and developed over the span of this semester.
- "[…] **This is the most excited I've been for a class at MIT!** […]"
- "[…] **I think that this class will be very challenging and exciting. I would be also really glad to contribute to a class that will be widespread in the next terms."** […]
- "[…] **It looks really amazing!!!!** - the talk was super cool and I think this class will be awesome. […]
- "[…] Looking at the Facebook page it seems like this course has had a ton of effort and passion already poured into it. **I would love to be a part of it and to make it work!"** […]
- "[…] **This class is one rare chance** to explore - with a scientific approach - the state of the art in autonomous vehicles transversally, including hardware design, perception, planning, control. More importantly, it is a chance to hit the current technological boundaries in these fields, using a real system. The latter is very inspirational to guide my future research […]"
- "[…] I am very interested in taking this course because, while I have extensive experience in mechanics and electronics and some experience with robotics and controls, I don't have much experience with uniting these in decision-making autonomous systems. **I can tell the class will be very enjoyable given the excitement and enthusiasm displayed by the staff**."
- […] The class is founded upon the philosophy that the best way to learn something is by using it to build cool stuff. Along those lines, I think Duckietown qualifies as "cool stuff". With regards to 6.S08, and hopefully 2.166, **I'm excited to help mold a new educational avenue for MIT students.**

Fig. 4. Some extracts from the answers to the question: "*Why would you like to take this course?*"

5.3 Deliverables

The course produced results that highlight the successful achievement of the ILOs. Students and mentors defined, developed and deployed the *functionalities* of the Duckiebots. This was accomplished during the group based *projects* (Table 3) in Phase II (Sect. 3.2). For a technical description of the projects we refer the interested reader to [9]. All these contributions were merged back into the base system, providing the behaviors that are now available to the broader community. The most obvious outcome of the course was the number of "functional" Duckiebots and compliant Duckietown(s) that were produced, along with the documentation for reproducing these results.

Documentation/Reproducibility. Cooperative development of a complex system requires thorough documentation, an aspect often insufficiently stressed during engineering academic training. Students were required to document, through step-by-step instructions based on a given template, every contribution to the project. Figure 5 shows, as an example, the overview of the autonomy tutorial instructions. Each step in the process links to open source documentation containing the implementation details, common challenges and troubleshooting

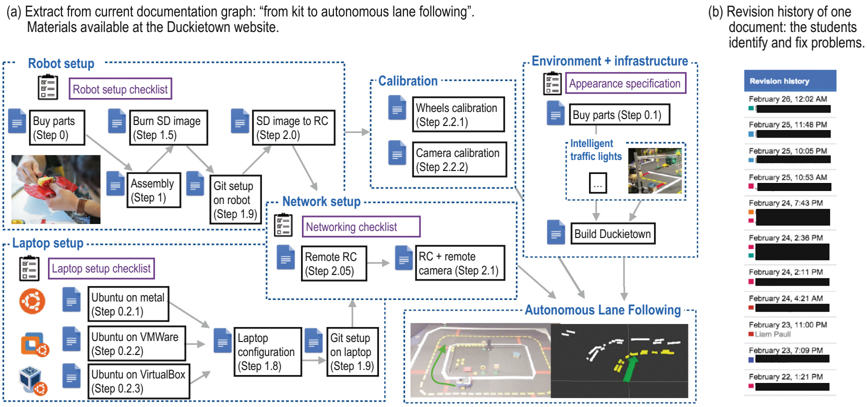


Fig. 5. (a) From kit to autonomous lane following in a dozen steps. (b) Extract of revision history for an instruction document: 12 people have contributed to this file in one week. The names of students are censored for privacy.

options. Moreover, all the documentation being openly available allows for students to dynamically correct and improve it.

Based on these documents, two high school students managed to reproduce the robot behaviors from scratch with minimal supervision.

Public Demonstration. Another measurable outcome of the course was the public demonstration, given in occasion of the MIT Open House Day. The installation area covered by Duckietown was roughly 330 m^2 (3600 sqft.) and the class constructed eight different Duckietowns (Fig. 6), deployed tens of Duckiebots and engaged over two thousand people over the course of a day. In each Duckietown area teams showcased a specific global behavior or functionality developed.

Fig. 6. Overview of the demo day Duckietown installation layout. Over one thousand duckies were taken by the public on that day.

The public walked a path through the installation being gradually introduced to increasing levels of abstraction, from the mechanical design of a Duckiebot to "parallel autonomy" [28] and intersection coordination of multiple robots. At each station students presented posters describing the underlying mechanisms, ensured the continuous and correct operation of their demonstrations and answered questions from the public, effectively disseminating their work. Mentors oversaw the students' efforts to ensure smooth operations.

Public feedback was unanimously positive, and over one thousand souvenir duckies were taken by the end of the day.

5.4 Reaching Underserved Demographics

We have performed a test by observing what names the students chose for their robots (Fig. 7). The conclusion, from the available sample, is that the Duckiebots tend to be seen as feminine by women, and as masculine by men. Our current interpretation is that this is due to a process of self-identification with the duckie and/or the robot, which is evidence that we have been successful in removing gender cues from the Duckiebots, while making them very relatable. Our empirical observations show that the duckies are perceived particularly well by women. Therefore, although Duckietown is not focused on the recruitment of women, this effort might help towards reaching underrepresented talent from that demographic.

Experiment: Gender perception of Duckiebots during Duckietown 2016

Setup: The students were asked to choose any name for their robot. We deliberately never mentioned any of our thoughts about gender and stereotypes reported above.

Overview: The data shows evidence of self-identification: if the name is not neutral, there is a strong correlation between the student gender identity and the perceived robot gender. Further data will be accumulated in future years to further test this hypothesis (we have not included names from staff operated Duckiebots).

Details: The students are eight women and 18 men. The names chosen by women are in *italic*. These are the Duckiebots with a **masculine name**: *Duckula*, Rex, Neptunus, Nikola, Ernie, Milo, Bill, Charles, Ayrton, Morty. These are the Duckiebots with a **feminine name**: *Julie*, *Lily*, Ada. These are the **neutral names**: *Oreo*, *Thing*, *Magitek*, *Cookie*, Quackmobile, Duckmobile, Amadobot, Penguin, Redrover, Lebeast, Pipquack, Setlist, Starducks, Maserati.

Fig. 7. Gender identification mini-experiment.

5.5 Student Feedback and Discussion

At the end of the course we asked students to provide feedback, on a 1–7 scale, on three queues related to the perceived quality of teaching. The results are shown in Fig. 8. We observe that most student were happy with the experience. The most critical feedback was related to the "helped me learn" queue, in particular it highlights a margin for improvement on the integration of the more theoretical aspects of the course in the hands-on activities. We believe this was a consequence of this being the pilot edition of the course. Fundamentally *nothing existed* of the Duckietown platform prior to this course and many structural "one-time" tasks had to be accomplished. For example: defining the communication protocols, the documentation templates, creating appearance specifications of cities, writing the instructions, coordinating staff and students, deciding what

Quality of Teaching	Rating Scale: 1=Strongly Disagree, 4=Neutral, 7=Strongly Agree, N/A=Not Applicable (7 is best)				
	AVG	1 2 3 4 5 6 7	RESPONSES	MEDIAN	STDEV
Stimulated interest	6.1		15	7.0	1.49
Displayed thorough knowledge of subject material	6.0		15	7.0	1.69
Helped me learn	5.7		15	7.0	1.88

Fig. 8. Student feedback was overall positive and it highlights opportunities for improvement.

software to develop and how, etc. These tasks took a toll on the other aspects of the course. In subsequent editions less time and effort will be necessary for the definition of structural features, in favor harmonization of the theoretical topics underlying vehicle autonomy and the developmental component of the course.

Nonetheless, *we were overwhelmed by the student engagement*. In Duckietown, it was normal for students to fix problems before the instructors were aware of them, extend the system without being prompted to do so, and cultivate a healthy obsession with duckies. Moreover, students have pulled together to help their peers, as was encouraged through the course forum hosted on Slack [29].

We believe that these outcomes were enabled by the "one robot per child" policy, in turn made possible by the affordable platform design. The fact that every student had their own personal hardware to work with cultivated a vastly different dynamic as compared to the case where students are only allowed to use equipment during lab hours. Students brought their Duckiebots home and even on their travels during vacation time. Moreover, students were increasingly motivated by the possibility that their contribution to a regular class could live on beyond the course itself.

We moreover noticed how students put significant effort in creatively customizing their Duckiebot, *de facto* developing an emotional bond with it. We believe this dynamic was enabled by the choice of the duckie theme (Sect. 2.1), in particular for its contrast to the stereotypes on robotics, gender neutrality, and "cuteness" factor.

6 Conclusions

Duckietown is a *research* and *outreach* platform in addition to being an *educational* one. We believe that teaching, research and outreach are complementary and enhance learning when considered together.

Duckietown creates significant research opportunities by providing a modular experimental testbed with a multitude of autonomous vehicles. *Research* opportunities attract senior staff to volunteer their time and abilities in the course, enhancing the quality of learning for students while at the same time fostering collaborations.

Outreach fights common stereotypes attracting younger generations to robotics in particular and STEM fields in general. Interaction with the general public gives students the opportunity to "teach" themselves, increasing their subject knowledge as well as the awareness of their social role as future engineers and scientists.

From a resource standpoint, we note that the implementation of Duckietown as described required the use of several spaces: a small inventory room to store all the hardware and a laboratory permanently equipped with a Duckietown. An additional room was temporarily equipped with a second Duckietown during phase II of the course, to accommodate the need of students to frequently test their work in progress. From a hardware perspective, all students had their own laptop (per institute policy), with a few additional laptops (*Duckietops*) made available in case of need. All Duckiebot hardware was kept by students throughout the course, and handed back in at the end of it.

Overall, we feel that this paradigm for robotics education is novel and successful. The feedback that we have received from the robotics community, the MIT community, and the public at large has been overwhelmingly positive. It has created enthusiasm and active participation of students and staff, attracted national and international attention from other institutions (National Chia Tung University, Taiwan; Tsinghua University, China and Rensselear Polytechnic Institute, USA have or are implementing versions of Duckietown aligned with their didactics) and media.

All materials produced during the course are freely available online, under open-source licenses. For pointers to all materials, please see http://duckietown.mit.edu.

Acknowledgments. This work was funded by the National Science foundation through award IIS #1318392 and through the National Robotics Initiative award #1405259. The work was also supported by the Toyota Research Institute and the Ford Motor Company.

References

1. Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., Wenderoth, M.P.: Active learning increases student performance in science, engineering, and mathematics. Proc. Nat. Acad. Sci. **111**(23), 8410–8415 (2014)
2. Bloom, B.S.: The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. Educ. Researcher **13**(6), 4–16 (1984)
3. Wieman, C.E.: Large-scale comparison of science teaching methods sends clear message. Proc. Nat. Acad. Sci. **111**(23), 8319–8320 (2014)
4. Halpern, D.F., Hakel, M.D.: Applying the science of learning to the university and beyond: teaching for long-term retention and transfer. Change Mag. High. Learn. **35**(4), 36–41 (2003)
5. Wiggins, G., McTighe, J.: What is backward design? Understanding by Design, pp. 7–19 (2011)
6. Biggs, J.: Aligning teaching for constructing learning. Higher Education Academy, pp. 1–4 (2003)
7. Monaco, M., Martin, M.: The millennial student: a new generation of learners. Athletic Training Educ. J. **2**, 42–46 (2007)

8. Novotney, A.: Engaging the Millennial Learner **41**(3), 60 (2010)
9. Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y.F., Choi, C., Dusek, J., Fang, Y., Okuyama, I.F., Hoehener, D., Liu, S.Y., Novitzky, M., Pazis, J., Rosman, G., Varricchio, V., Wang, H.C., Yershov, D., Zhao, H., Benjamin, M., Carr, C., Zuber, M., Karaman, S., Frazzoli, E., Del Vecchio, D., Rus, D., How, J., Leonard, J., Censi, A.: Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In: IEEE International Conference on Robotics and Automation (ICRA) (2017), Submitted. http://duckietown.mit.edu/materials.html, Accessed 15 October 2016
10. Olson, E.: Apriltag: a robust and flexible visual fiducial system. In: IEEE International Conference on Robotics and Automation (ICRA), May 2011
11. Siegel, M., Breazeal, C., Norton, M.I.: Persuasive robotics: the influence of robot gender on human behavior. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, Institute of Electrical and Electronics Engineers (IEEE), October 2009
12. Green, T.: Why are too few females in robotics? could it be the robots? (2015)
13. Greensted, C.: Intended learning outcomes. EFMD Glob. Focus **8**(1), 20–25 (2014)
14. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3, p. 5, Kobe (2009)
15. Sturm, P.: Pinhole camera model. In: Computer Vision, pp. 610–613. Springer (2014)
16. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
17. Daum, F.: Nonlinear filters: beyond the kalman filter. IEEE Aerosp. Electron. Syst. Mag. **20**(8), 57–69 (2005)
18. Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., Munich, M.E.: The vSLAM algorithm for robust localization and mapping. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 24–29. IEEE (2005)
19. Masehian, E., Sedighizadeh, D.: Classic and heuristic approaches in robot motion planning-A chronological review. World Acad. Sci. Eng. Technol. **29**(1), 101–106 (2007)
20. Crowley, J.: Navigation for an intelligent mobile robot. IEEE J. Robot. Autom. **1**(1), 31–41 (1985)
21. Hafner, M.R., Cunningham, D., Caminiti, L., Del Vecchio, D.: Cooperative collision avoidance at intersections: algorithms and experiments. IEEE Trans. Intell. Transp. Syst. **14**(3), 1162–1175 (2013)
22. Duperret, J.M., Hafner, M.R., Del Vecchio, D.: Formal design of a provably safe robotic roundabout system. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2006–2011. IEEE (2010)
23. Cao, Y., Yu, W., Ren, W., Chen, G.: An overview of recent progress in the study of distributed multi-agent coordination. IEEE Trans. Industr. Inf. **9**(1), 427–438 (2013)
24. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press, Cambridge (2005)
25. Tani, J., Mishra, S., Wen, J.T.: Motion blur-based state estimation. IEEE Trans. Control Syst. Technol. **24**(3), 1012–1019 (2016)
26. Tanenbaum, A.S., Van Steen, M.: Distributed Systems. Prentice-Hall, Upper Saddle River (2007)

27. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: a survey. J. Intell. Rob. Syst. **53**(3), 263–296 (2008)
28. Naser, F., Dorhout, D., Proulx, S., Pendleton, S.D., Andersen, H., Schwarting, W., Paull, L., Alonso-Mora, J., Jr., M.H.A., Karaman, S., Tedrake, R., Leonard, J., Rus, D.: A parallel autonomy research platform. In: IEEE Conference on Robotics and Automation (ICRA), pp. 1–8 (2017). Submitted
29. Slack: Slack (2016) Accessed 10 October 2016

Teacher Education to Analyze and Design Systems through Reverse Engineering

Igor Verner(✉) and Moshe Greenholts

Faculty of Education in Science and Technology, Technion – Israel Institute of Technology, Haifa, Israel
ttrigor@technion.ac.il, mosgrini@campus.technion.ac.il

Abstract. This paper proposes to enhance engineering design education in high schools by introducing reverse engineering (RE) and making activities. The rationale is to foster meaningful learning by engaging students in the analysis of design solutions of existing technological systems and in practical implementation of new designs in the spirit of makers. We present a pilot technology teacher education course which includes studies of engineering and pedagogical fundamentals, RE practices, development of instructional units, and participation in a maker fair. Positive reflections on the course encourage further development of the proposed approach and exploration of its possible use in high schools.

Keywords: Reverse Engineering · Technological systems · Teacher education · Engineering design · Maker movement

1 Introduction

There is a high and increasing need for technologically literate people capable to develop solutions for problems of the modern technologically oriented world. Many countries, including Israel, experience the lack of specialists in design and operation of modern technological systems and difficulties in attracting people to study these professions [1]. Educational systems make growing efforts to expand technology education (TechEd) in schools and direct it to the two main objectives: impart technical knowledge and skills for career and life, and give students the opportunity to apply their mathematics and science knowledge to real technological problems. These efforts are supplemented by informal educational programs initiated by industrial companies, military and social institutions.

Design and production of engineering systems using digital technologies are central issues of the modern TechEd. High school students acquire knowledge and basic skills in applying computer aided design (CAD) tools, constructing and programming robots, and operating computer-controlled systems. Educators face pedagogical challenges in teaching these subjects to school students that have limited background in mathematics, science, and engineering. To cope with the challenges, educators offer learning activities based on using advanced computer interfaces for visualization, analysis and control

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_9

provided by modern CAD, robotics, and manufacturing systems. The automation utilities often allow the school student to carry out the task without requiring a deep understanding of the technological processes [2].

Researchers call for the need to enhance meaningful learning in TechEd by strengthening the connection between theoretical fundamentals and real practice [3]. Many studies in this direction have been conducted in educational robotics. They showed that science, technology, engineering and mathematics concepts can be effectively learned in robotics environments [4–8]. Nowadays, numerous robotics programs are implemented in formal and informal education. They connect robotics to different disciplinary and interdisciplinary subjects.

In recent years there have been growing claims that the traditional approaches to educational robotics need to be extended [9, 10]. Rusk [11] proposed possible “alternative pathways” to increase engagement and quality of robotics education, and called for development of new pathways.

A productive way to determine such pathways, for our opinion, is by rethinking the dilemmas considered by the society 40-50 years ago when the TechEd came to swap its predecessor, the industrial arts [12, 13]. The industrial arts paradigm that highlighted product fabrication in shop classes and acquisition of daily-life skills was replaced by the TechEd paradigm which focused the studies on technological systems and processes, and gaining problem solving skills. TechEd offers experiential activities that can efficiently prepare for professional careers, but it limits students’ opportunities for the development of daily-life skills, and therefore also self-expression and self-dependency.

The phenomenon of maker movement refers to the growing number of people who are self-motivated to engage in the creative use of digital technology to produce daily life artifacts. In recent years the movement has rapidly increased, manifesting its aspirations to make, share, give, learn, tool up, play, participate, support and change [14]. Maker spaces have been developed as communities of practice, where makers learn technical skills, design and fabricate new products [15].

The high values cultivated by the maker movement evoke enthusiasm towards its potential use to enrich formal and informal technology education [9, 15]. Alimisis [10] calls to empower educational robotics by emphasizing the values of the maker movement and learning by making. We see the following potential benefits of introducing making in project based robotics education:

1. Providing students with the opportunity of self-expression to self-dependently develop ideas, design and fabricate usable products.
2. Development of the skills of choosing, learning, and self-dependent use of proper technologies and materials.
3. Creation of open multi-age affinity groups for peer collaboration in learning for making.
4. Paving a new learning-by-doing pathway focusing on technological entrepreneurship.

Lee and Fields [16] note the growing acknowledgement of the maker movement, rapid dissemination of making activities in informal education at different levels, and noticed the first attempts to integrate these activities in the school curriculum. In this context the authors raise the principle question: “What is it that people are learning when they ‘make’?” They called for evaluation of learning outcomes of the making activities.

One of the main objectives of the modern STEM (science, technology. engineering, and mathematics) education is providing meaningful learning [17]. The sense of meaning, constructed by students based on their personal experience, reflection, and conceptualization enables them to use the acquired knowledge across contexts. The value of meaningful learning in technology education and, particularly, in educational robotics has been discussed [18]. There is a need to discuss the issue with regard to learning by making. Therefore, we propose one more principle question: How to utilize “making” to enhance meaningful learning in technology education?

This paper addresses the two above questions in the context of the teacher education course that was developed and delivered at the Technion Faculty of Education in Science and Technology in collaboration with the Intel Makers Program. In the next sections we will consider the framework and outline of the course, discuss students’ projects and learning outcomes.

2 Educational Framework

The educational system in Israel offers studies of different technological disciplines, four of which are defined science-rich subjects, including mechanical engineering [19]. The technology/mechanics education track at the Technion Faculty of Education in Science and Technology is the only authorized university undergraduate program in Israel for training teachers of the subject.

Technion has recently come up with the Views program which calls upon undergraduates and graduates from all faculties to study for an additional B.Sc. in Science and Technology Education, and offers them full tuition scholarships. Many students and graduates have joined the program. As a result, the technology education track has upgraded the infrastructure and curricula to meet the associated challenges in research and education. The upgrade is based on the following principles:

- Deliver the program with involvement of employers and social partners.
- Connect educational research and teacher training.
- Provide prospective teachers with up-to-date knowledge.
- Enhance the role of teaching practice.
- Promote labor market insertion.

The faculty laboratory of technology plays key roles in the teacher training as a ground for experiential learning through design and operating instructional engineering systems and a venue for discussions and meetings. Laboratory activities with engineering systems are included in all technology education courses. Some of the courses are open for students majoring in science and mathematics education and for students from other departments. Many of the learning activities are oriented to project assignments of design and programming robotic systems and development of instructional units, using the systems as teaching aids.

In this paper we consider one of the courses called “Selected problems in teaching mechanics technology” which concentrates on topics selected by the lecturer. In the 2015-2016 course, we decided to introduce and explore two innovations. The first was to use making in order to enhance students’ knowledge of digital technologies and self-expression, and the second was to foster meaningful learning of engineering design through the reverse engineering (RE) approach. In the course we proposed and implemented a new educational strategy which integrates making with the RE approach. We explored the features of the strategy, by following up the learning activities and analyzing students’ reflections.

3 Reverse Engineering

RE is the investigation of an existing technological system for the purposes of uncovering its design solutions and developing an improved or new system [19]. The essence of reverse engineering is the analysis of the system by breaking it down into functional components and finding out how the relationships among components provide the system’s overall functioning.

Educators report on the effective courses of engineering design based on the RE approach [20–22]. They point out the advantages of the approach, particularly in educating novice engineering students:

- Hands-on dissection of the system enables students to directly observe its mechanisms and their interactions, making it easier to understand the principles of system design.
- By the suitable choice of RE tasks, the teacher can direct students to learn the systems that are of interest in the course.
- The approach emphasizes the role of analysis in engineering design.

Portsmore and Rogers [23] highlight the special importance of practices in engineering analysis, and particularly RE, for technology education in schools. They propose to enhance hands-on activities with technology that develop a qualitative sense of engineering methods by more advanced lessons in which the student will use mathematics and science to “construct conceptual prototypes for their ideas”. The systematic and documented RE process can foster development of skills that are used for analysis of technological systems. The authors argue that the skills of engineering analysis are necessary for school teachers who guide design projects, as they need to promptly find out the challenges faced by the students, evaluate the developed design solutions, and give recommendations.

Here in this paper we emphasize the potential of RE as an approach which can effectively promote meaningful learning in school education. Nowadays meaningful learning is put forward in STEM education as a main strategy of the development of knowledge and skills “for a full and productive life” [24].

The conception of meaningful learning comes from the constructivist learning theory [25]. Accordingly, people actively construct their knowledge through interaction with the environment, while meanings are created in response to events. The event is first represented in imagination as a picture, then, by reasoning, its structure and relationships are captured and brought to consciousness, and thus the event becomes meaningful for the person. Mayer [17] noted that hands-on learning activities not always lead to meaningful learning. The activities that really promote meaningful learning are that of selecting, organizing, and integrating knowledge. The author points out that learning should be evaluated not on the amount of doing, but rather on its impact on meaning making.

Kollöffel and Jong [26] suggest to promote meaningful learning in technology education in schools by offering inquiry-based assignments that involve identifying variables and relations, generating hypotheses, interpreting results, and drawing conclusions. Coming back to the above description of RE, we see that the system dissection and analysis activities, proposed by this approach, inherently include inquiry-based assignments and can promote meaningful learning.

4 Teacher Education Course

The one semester 2 h/week course "Selected problems in teaching engineering mechanics" is part of the technology teacher training program. In the course, students study issues in teaching technology of material handling, basic technical drawing, machine elements and strength of materials.

The didactic issues include guidance of experiential learning and teaching disadvantaged pupils. Individually or in pairs, they make inquiries into selected topics, develop instructional units, practice in teaching and evaluating lessons, and submit reports. The instructional unit assignments include making physical models for demonstrating the concepts studied in the unit.

In the past courses students constructed the models using LEGO parts. In the 2015-2016 academic year we decided to update the course, based on making and reverse engineering. To cope with this challenge, we created a partnership in which the first author was a course lecturer and the teaching assistant was the second author, an experienced engineer and teacher doing MSc research on didactics of the RE approach in technology education.

An important member of this partnership was the group of engineers participating in the Makers Program of Intel. The Intel company supports the maker movement internationally and Israel. The group organized by the Israel division promotes making and supports makers by providing hardware and software tools, organizing maker fairs and outreach courses. In our teacher education course the Intel group participated in the development of hands-on activities, provided Intel Arduino boards, sensors and CAD software and taught how to use them.

16 students participated in the course, half of them are experienced engineers, and three are high school teachers of technology. The course lectures (L), workshops (W), homework (H) and assignments (A) are listed in Table 1. The students in the course acquired subject matter and pedagogical knowledge, developed plans of instructional units, discussed them in class and presented to school students and to the community at the Maker Fair event that was the culmination of the course.

Table 1. Course lectures, workshops, homework and assignments

Meeting	Topics
1	L1 – Course introduction. Analysis and meaningful learning in engineering design education. H1 – Answering questions related to engineering analysis.
2	L2 – Reverse engineering approach in design education. Development of instructional units based on RE of technological systems. H2 – Answering questions related to the RE approach.
3	W1 – Product analysis: Gathering information on and primary disassembling of a technological system. A1 – Make a disassemble specification of a certain system.
4	L3 – Introduction to Arduino. W2 – Electronics prototyping.
5	W3 – Product analysis (cont.): Detailed disassemble of a technological system. A2 – Analyze the system and develop an idea of a new/improved system.
6	L4 – Introduction to Tinkercad. W4 – Product design with Tinkercad.
7	W5 – Design and analysis of the new/improved system. A3 – Develop instructions for making the new/improved system and assessment.
8	W5 – Continued. A4 – Develop an instructional unit plan.
9-10	W5 - Continued. Guidance on the development of instructional unit plans.
11-12	Student presentations of instructional unit plans and discussion.
13	Student public presentations at the Maker Faire.

Engineers from the Intel Program conducted the course meetings 4 and 6. The Fair was organized by Intel, the Israel Ministry of Education, Haifa Municipality and the Technion Faculty of Education in Science and Technology, with participation of high school students from a number of schools.

5 Instructional Units

In the assignments A1-A4 the students had to dismantle and analyze a technology system they chose, and develop an instructional unit for experiential learning of a STEM subject through the RE approach. The units developed by the students in the course are presented in Table 2 including their learning subjects, instructional objectives, and technological tools.

One of the instructional units listed in Table 2 "Development of an optical mouse for people with disabilities" is described below. Two students selected to perform reverse engineering of an optical mouse. They dismantled the mouse and analyzed its functionality, mechanism, sensors, and design (Fig. 1). Through examining the role of each of the components specified in the figure, the students developed an idea on how to redesign the mouse for other use cases. They decided to develop a tool which can serve people with disabilities and developed an original static optical mouse controlled by a special ring wore on the person's finger.

Table 2. - Learning subject details

	Learning subject	Instructional objectives	Learning tools
1	Development of a computer tactile interface.	Understand the joystick mechanism. Analyze possible applications of tactile interfaces. Develop a new interface.	Joystick, Arduino kit, LEGO parts.
2	Development of an optical mouse for people with disabilities.	Understand the optical mouse functionality, mechanism, sensors, and design. Gain basic CAD skills.	Mouse, Arduino kit, 3D printer, CAD software.
3	Development of a desktop scanner to deter unwanted scanning.	Analyze a scanner mechanism and operation. Determine mishaps of unwanted scanning and solutions.	Desktop scanner, Arduino kit.
4	Development of a mechanical mouse indicating coordinates of the cursor symbol.	Analyze mechanics and interface of a mouse. Create an interface for determining the direction of a computer mouse cursor. Gain basic CAD.	Mechanical mouse, Arduino Kit, CAD, 3D printer.
5	Development of a device for experiential learning of angular velocity	Analyze the CD ROM mechanism by dismantling to sub-components. Design and create a tool for learning angular velocity.	CD ROM, Arduino kit.
6	Development of a device to improve energy efficiency of a slide projector.	Understand physics and evaluate the projector electrical efficiency. Develop a tool to measure electrical efficiency of the projector.	Slide projector and Neon, Halogen, and PL bulbs.
7	Development of a desktop computer temperature control system.	Identify and map the temperature values of a computer. Understand the computer fan operation, mechanism, design. Develop an adaptive computer fan.	Computer fan, Arduino Kit
8	Develop a tool for experiential learning of DC Motors.	Understand the principles of motor operation in a hair cutter. Create a tool for experiential learning of DC motors.	Hair cutter, Arduino kit, LEGO parts.

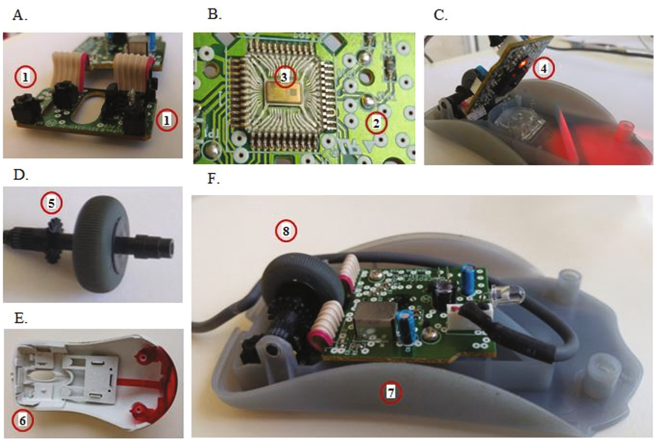


Fig. 1. Optical computer mouse disassembles and sub-assembly: 1 – click switches, 2 – board, 3 – processor, 4 – LED, 5 – encoder, 6 – cover, 7 – base, 8 – scroll wheel.

The students designed the mouse using the Creo computer aided design software (Fig. 2). Then they designed and fabricated the finger ring using Tinkercad and 3D printer (Fig. 3A). The developed prototype is presented in Fig. 3B.

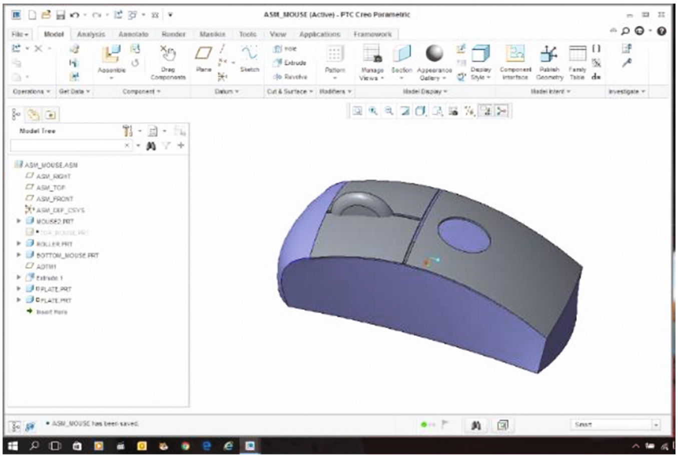


Fig. 2. Optical mouse new cover design using CREO

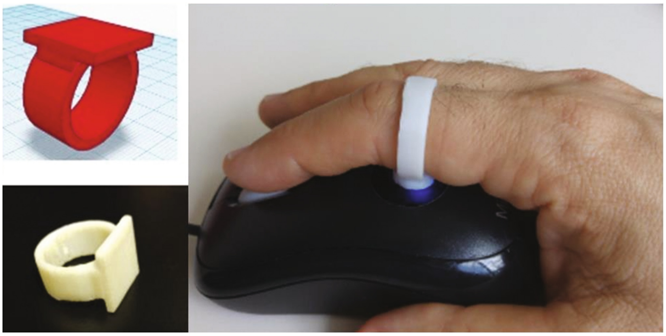


Fig. 3. A. Finger Ring; B. The prototype

6 Student Reflections

At the end of the course the students were asked to answer a questionnaire and share their reflections about the value of the reverse engineering approach implemented in the course. Free-form reflections accepted from 16 students were qualitatively analyzed and categorized. As found, all the respondents greatly appreciated the opportunity to learn about reverse engineering and its use in design education. They noted that the use of RE in design education allowed them:

- "To get inside the head of the designer and think like an engineer".
- To understand the mechanism and operation of technological systems used in everyday life.
- To learn new aspects of engineering design.

The students noticed that the practice of RE engaged them in learning new digital technology tools (Tinkercad, Arduino) and in mastering technical skills that are important for teaching technology in schools. The approach directs the student to learn solutions implemented in existing technological systems before proposing a new solution. It teaches that systematic and rigorous design procedures are required to develop efficient technological systems.

The students pointed that as different from the past courses, in which the system to be designed was specified, in this course they directed their design thinking towards implementation of the design solutions that they evaluated and learned from the selected systems. The practice in engineering analysis increased the level of learning, made the learning meaningful. The students who are school teachers expressed their intention to use the reverse engineering approach in their practice.

There were also some critical comments. Three students noticed that in the course they did not have enough time to complete design of the new system, but only to learn the approach and make the first prototype. Two other students remarked that 30 min given for presentation and discussion of their instructional unit was not enough. One of the respondents reported on difficulties in understanding the approach and the assignment which, for his opinion, was too open-ended. From personal discussions, our impression was that the course was closer and clearer to students who had experience in design of real technological systems.

7 Conclusions

We addressed the two principle questions, posed in the introduction, in the specific context of our teacher education course, in which the practice of making was based on the reverse engineering approach. With regard to the first question, in the case study we observed what students learned through this practice. We have seen that in the course assignments the students self-dependently generated ideas, designed and prototyped new technological systems, and developed instructional units. When performing the assignments, the students chose and used different technologies and materials. The course facilitated creation of a learning community in which the students worked on the assignments in pairs, while sharing the acquired knowledge with peers and getting their feedback. Students' participation in the Maker Fair gave them the opportunity to evaluate the projects presented by school students, the learning outcomes, and the potential to improve such projects by introducing the reverse engineering approach. Our observations are in line with the potential benefits of making practices discussed in the introduction.

With regard to the second question, our case study demonstrated that the reverse engineering approach can be used to enhance meaningful learning through making practices. We observed indications that RE and making together foster meaningful learning of both the subject matter and pedagogy. In this symbiosis, the RE approach directed to deep analysis of the design ideas while making provided tools and strategies for their practical implementation.

The following ideas for further development of learning by making emerged from the course:

- The need to learn from the past experience of learning technology through the industrial arts approach.
- The need to create in the course an atmosphere encouraging students to follow the values manifested by the maker movement.
- The course scope should be extended to allocate more time for studies of theoretical concepts, practical assignments, and their presentations.

The authors plan to continue the study, develop a course based on RE and making for high school students, and explore its outcomes.

References

1. Getting skills right: assessing and anticipating changing skill needs. OECD (2016). http://www.skillsforemployment.org/KSP/en/Details/?dn=WCMSTEST4_171460
2. Koch, D., Sanders, M.: The effects of solid modeling and visualization on technical problem solving. J. Technol. Educ. **22**(2), 3–21 (2011)
3. Howland, J.L., Jonassen, D.H., Marra, R.M.: Meaningful Learning with Technology. Pearson, Upper Saddle River (2012)
4. Benitti, F.B.V.: exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012)
5. Verner, I., Waks, S., Kolberg, E.: Educational robotics: an insight into systems engineering. Eur. J. Eng. Educ. **24**(2), 201–212 (1999)
6. Cuperman, D., Verner, I.: Learning through creating robotic models of biological systems. Int. J. Technol. Des. Educ. **23**(4), 849–866 (2013)
7. Verner, I., Ahlgren, D.: Fire-fighting robot contest: interdisciplinary design curricula in college and high school. ASEE J. Eng. Educ. **91**(3), 355–359 (2002)
8. Alimisis, D., Kynigos, C.: Constructionism and robotics in education. In: Alimisis, D. (ed.) Teacher Education on Robotic-Enhanced Constructivist Pedagogical Methods, pp. 11–27. ASPETE, New York (2009)
9. Blikstein, P.: Digital fabrication and ''making'' in education: the democratization of invention. In: Walter-Herrmann, J., Bóching, C. (eds.) FabLabs: Of Machines, Makers and Inventors, pp. 1–21. Transcript Publishers, Bielefeld (2013)
10. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013)
11. Rusk, N., Resnick, M., Berg, R., Pezalla-Granlund, M.: New pathways into robotics: strategies for broadening participation. J. Sci. Educ. Technol. **17**(1), 59–69 (2008)
12. Clark, S.C.: The industrial arts paradigm: adjustment, replacement, or extinction? J. Technol. Educ. **1**(1), 7–12 (1989)
13. Herschbach, D.R.: From industrial arts to technology education: the eclipse of purpose. J. Technol. Stud. **23**(2), 20–28 (1997)
14. Hatch, M.: The Maker Movement Manifesto. McGraw-Hill, New York (2014)
15. Halverson, E.R., Sheridan, K.M.: The maker movement in education. Harvard Educ. Rev. **84**(4), 495–504 (2014)

16. Lee, V., Fields, D.: A Clinical Interview for Assessing Student Learning in a University-Level Craft Technology Course. ITLS Faculty Publications. Paper 479 (2013). http://digitalcommons.usu.edu/itls_facpub/479
17. Mayer, R.: Rote versus meaningful learning. Theor. Pract. **41**(4), 226–232 (2002)
18. Somyürek, S.: An effective educational tool: construction kits for fun and meaningful learning. Int. J. Technol. Des. Educ. **25**(1), 25–41 (2015)
19. Verner, I., Betzer, N.: Machine control - a design and technology discipline in Israel's senior high schools. Int. J. Technol. Des. Educ. **11**, 263–272 (2001)
20. Rad, H.: Reverse engineering as a learning tool in design process. In: ASEE Annual Conference, Louisville, KY (2012)
21. Wood, K.L., Jensen, D., Bezdek, J., Otto, K.N.: Reverse engineering and redesign: courses to incrementally and systematically teach design. J. Eng. Educ. **90**(3), 363–374 (2001)
22. Dalrymple, O.O., Sears, D.A., Evangelou, D.: The motivational and transfer potential of disassemble/analyze/assemble activities. J. Eng. Educ. **100**(4), 741–759 (2011)
23. Brophy, S., Klein, S., Portsmore, M., Rogers, C.: Advancing engineering education in P-12 classrooms. J. Eng. Educ. **97**(3), 369–387 (2008)
24. Barron, B., Darling-Hammond, L.: Teaching for Meaningful Learning: A Review of Research on Inquiry-Based and Cooperative Learning. Book Excerpt. George Lucas Educational Foundation (2008)
25. Greeno, J.G.: On claims that answer the wrong questions. Educ. Res. **26**(1), 5–17 (1997)
26. Kollöffel, B., Jong, T.: Conceptual understanding of electrical circuits in secondary vocational engineering education: combining traditional instruction with inquiry learning in a virtual lab. J. Eng. Educ. **102**(3), 375–393 (2013)

Methodologies in Educational Robotics

29 Effective Ways You Can Use Robots in the Classroom

An Explanation of ERA Pedagogical Principle

Dave Catlin(✉)

Valiant Technology Ltd., 3 Grange Mills, Weir Road, Balham, London, SW12 0NE, UK
dave@valiant-technology.com

Abstract. Catlin and Blamires proposed the “Educational Robotic Application (ERA) Principles”. These 10 principles give researchers, designers, educators and teachers a way of evaluating and comparing educational robots and their activities. The ‘Pedagogical Principle’ was one of these ideas. It stated you could use many different developmental theories to view and describe the learning involved. It also identified 28 (now 29) different ways you could use a robot in a classroom. Catlin and Blamires did not give a satisfactory explanation of the these methods. This paper corrects this.

Keywords: ERA principles · Roamer · Educational robots · Turtles · Tagging

1 Introduction

Blamires and Catlin wrote the ten Educational Robot Application (ERA) Principles to help understand the value of robots when used in radically different ways [1]. For example, comparing how a robot aids a 4-year old understand number, with how they aid an 18-year-old grasp vector analysis. The original paper did not fully explain the ERA Pedagogical Principle. This paper will resolve this issue.

The paper starts with a brief review of ERA and a detailed summary on the Pedagogical Principle. It reviews some recent work by other researchers in this area and critiques their efforts before developing the ERA idea. This exposition shows the Pedagogical Principle has two strands: the developmental theories supporting the use of robots and 29 different characteristics of educational robot projects. The paper explains these traits and illustrates them with practical examples.

1.1 Data and Method

The data used to support this report is from the archives of Valiant Technology. Valiant gathered this over the last 33 years. It is a mixture of research projects like the UK National Turtle Project [2], doctoral and master theses [3] and [4]. It includes conference papers [5–7] journals and books [8–10] and various reviews by educators and teachers [11, 12]. It also uses data from Valiant’s customer archives. This unique data set includes communications from tens of thousands of teachers using robots to teach in regular classrooms.

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_10

The work took place with pre-K to Grade 12 children in 27 different countries, notably the UK, USA and Shanghai. All the work used either the Valiant Turtle or the Turtle-like Roamer. Most of the work took place in the school classroom, but a significant amount took place in informal learning settings like museums, after-school clubs or special learning events like the Big Bang [13]. Students varied from those needing extra tuition to gifted and talented, as well as students with special needs.

Many people consider much of this data as anecdotal and not worthy of consideration [14]. Others consider some of the references as old and out-of-date. They're wrong to do so. Persistence validates ERA and the Pedagogical Principle. A teacher reporting their experience of using a robot in the classroom has a value which increases with repetition. This is even more so when this repetition occurs over several decades, from different locations and radically differing learning environments. This paper is an analysis and summary of this data.

1.2 The Robots

The Pedagogical Principles comes from experience with Turtle Robots. The first of these is the Valiant Turtle, first produced in 1983 and controlled by the programming language Logo. The next robot is the Classic Roamer, first produced in 1989: students program this using its on-board keypad and chip based version of Logo.

The current Roamer started production in 2012. You can configure this robot in different ways, which enables it to support the educational need. For example, you can change keypads and the behavior stored in the chip. You can program this Roamer with a keypad, or from the computer using Logo, Scratch or any other programming language. Indeed, you do not need to program the robot, you can make it work using Human Computer or Human Robot Interface technologies.

You need to make a distinction between Turtle robots and construction robots. The teacher can use Turtle within a few moments of the lesson starting – you do not need to build it. Turtle robots focus on "Teaching with Robots (TWR)" and construction robots on "Teaching Robotics (TR)". However, students can change the design of the Turtle and once built, teachers can use construction robots to teach ideas. While the data comes from using Turtles, an informal review of research papers shows the Pedagogical Principles applies to both types of robot.

2 ERA Principles

The ten ERA Principles comes from a meta-analysis of the work described in the Method and confirmed by their persistence. ERA refers to physical robots (**embodiment**) that children **interact** with. The robot has an internal **intelligence** or behavior that involves students in **personalized** learning experiences. Work with robots form an **equitable** learning environment where students **engage** with **curriculum** objectives in a way that develops their sustainable learning skills. Teachers must find using the technology **practical** in a busy classroom. The Pedagogical Principles connects all of this to the science of learning and describes the characteristics of robotic activities.

3 Educational Robotics Café and Important Thoughts

A workshop called the educational robotics café took place at the at the 6th International Conference on Robotics in Education (May 2015, Yverdon-les-Bains, Switzerland) [15]. The group running the workshop aimed to take advantage of the conference expertise to find out an "overview of the educational robotics landscape".

The workshop lasted 3 h. A report on the workshop outcome stated: "…difficulty presented itself in finding common elements that made sharing or comparing possible. The approaches presented in the pre-phase weren't described in that way. It would have helped, had they been broken down into elements that many approaches or activities have in common." The Pedagogical Principle deals with this issue. It contains a list of basic "elements" that we can combine in different ways.

I disagree with one conclusion of the Café participants: "In pre-school and elementary, children use a lot of imagination, thus robots have to be contextualized (e.g. using storytelling). In junior high school, students use robots as tools to learn different concepts; in senior high school, the focus is building (mechanics and electronics) and coding "real" robots." Characterizing the use of robots in this way builds unnecessary barriers that lessen the educational potential of robots. Why do we want to limit imagination to young children or building robots to older students? While this is common in contemporary practice, counter examples show it is not a valid premise.

It is a time for vision. Trying to describe a landscape based on a survey of current practice and today's technology will not succeed. ERA defines a set of simple axiomatic statements, that will not change when innovations appear. They provide an analytical set of tools, but they also provide a guide for developers.

Catlin and Blamires did not simply review the decades of evidence available; they also looked at technological trends in subjects like AI, robotics and computing. A more subtle issue sneaks into the discussion – our current obsessions with STEM, coding and even robotics. Anyone who has survived the education sector for a few decades will know that in 5 or 10 years times we'll be chasing new educational panaceas. Initiatives, come and go, often inspired by technological advances. The ERA Principles needed to look beyond the current trends.

4 ERA Pedagogical Principle

ERA defines the Pedagogical Principle as, *"The science of learning underpins a wide range of methods available for using with appropriately designed educational robots to create effective learning scenarios."* This notion splits into theory and application. The original paper adequately explains the theory. While it recognizes the original connection to Piaget's constructivist ideas of learning, it takes a more eclectic view. Catlin and Blamires recognize all theories describe a learning process from a particular perspective. Therefore any theory is valid if it helps develop better robots or robot activities. Better means they help students to learn more, gain a deeper understanding, learn faster, remember longer and enjoy the experience.

4.1 Application

The Pedagogical Principle is applied through activities. The Principle identifies 29 traits which characterize an activity. This paper proposes these form the basis of a Pedagogical Tagging scheme.

You can use the characteristics in two ways. First to help understand the nature of an activity and second to help design tasks. It's easy enough to think of an exciting robot challenge, but not so simple if you want to meet ERA's Curriculum and Assessment Principle. This principle ties work with robots into helping teachers deliver the curriculum and assess the effectiveness of the student's work. The "Backward Assessment Model" is an important instructional design approach [16]. It starts with knowing how you plan to assess the student's work against a learning objective – what evidence do you want to gather. What activity will allow you to gather that evidence? The Pedagogical tags make it much easier to come up with good ideas when you've specific learning objectives in mind. Valiant used the tags over the last 10 years to help develop over 350 activities. They've used them for the last 5 years to tag and describe activities and correlate research in their e-Robot project [17]. This longitudinal study is helping to refine and validate the tags.

4.2 Tag Candidature

A tag is a characteristic of the activity, independent of who, what, where and how the activity is done. In deciding if something is a tag it is necessary to remember that the Pedagogical Principle is one of ten ERA Principles. So, for example, is the gender structure of a group doing an activity a tag? The answer is no because the activity and a tag are the same whichever group does it. (The gender issue is dealt with in the ERA Equity Principle). On the other hand, problem solving is a tag, because in this context it is not a variable. Should we consider "fun" a candidate? Again no. You can do an activity with a group of students who find it great fun and others who do not. This has nothing to do with activity, but a complex of personalities, transient feelings of participants and many other factors. Besides, fun is a subset of the more powerful ERA Engagement Principle. Engagement is about the student giving their attention, whether it is fun or not. While Engagement is a separate principle it is included as a tag, because some educators did activities with the sole purpose of engaging the student. This illustrates another thing about this list, it is derived "bottom up". A candidate does not have to fit into Bloom's taxonomy or any other structure. What validates a candidate is that it is useful (ERA Practical Principle) and its persistence. That is you look at the 3 decades of data the same ideas appear time and again irrespective of the circumstances. You can deduce the list is not new. By 1990 most of the tags were in use. This paper is merely defining and formalizing the list as well as acknowledging the characteristic's persistence.

4.3 The Activities

Teachers devised most of the activities used to illustrate the tags. Yes, they could have used different methods without using the robot. They choose not to. They choose to use the robot because of the reaction they get from children. One young woman reminisced about her school days, "I always knew we were in for a good day when the teacher got Roamer out." This reaction related to the ERA Principle of Engagement. It is enough to justify using a robot.

4.4 The Tags

What follows is a definition and brief explanation of the tags, illustrated with an example activity[1]. The activity title is formatted in bold.

1. Catalyst. Using a robot to make an exciting start to an activity. In most activities robots play a "starring role." However, they can make a useful cameo appearance. In the Roamer Activity: **Going to the Seaside** the students engage in a murder mystery game [18]. A salmon is trying to swim to the sea, but keeps meeting pollution points which gradually slow it down until it eventually stops dead. Working in groups the students solve the pollution problems enabling the fish to make its journey. The robot dramatically captures the students' attention by simulating the salmon's fatal journey. If the students resolve the pollution problem they get to reprogram the robot and they can watch the salmon successfully swim to the ocean.

2. Challenge. These are small-scale problem solving tasks. You can characterize challenges as closed (one solution) or open (many solutions). Programming Roamer to navigate a maze is a traditional Logo exercise. In the challenge, **Escaping Baron von Bugbyte** students programmed the Valiant Turtle to escape the evil Baron by navigating their way through a rabbit warren. It is a closed challenge – you escape or you do not. Good challenges have multi levels to them which allow you to personalize the challenge to the needs of the student. For example allow step-by-step programming or insist the students work out and program the robot in one try. Navigate the maze using left turns, or right turns only. Make the robot autonomous and use a sensor to find a way through.

3. Coding. Papert invented the Turtle robot to provide a physical representation of the effect of coding using the LOGO language. The basic idea of LOGO was for students to learn mathematics by coding a computer to solve math problems. This involved students thinking about the basic structure of mathematical problems. LOGO branched out beyond mathematics and the recent interest in coding and computational thinking has revived the ideas behind LOGO. In the activity **Mondrian** students program Roamer to draw Mondrian style paintings. This involves looking at a design style through the computational thinking patterns like procedures and repeats [19].

4. Conceptualizing. Robots help students understand ideas through concrete experiences. This is one of Papert's basic ideas. He saw a Turtle as a transitional object: something you can identify with. You can play Turtle by imagining yourself as the robot

[1] All the activities are subject to the Creative Commons Licence . The citation should clearly state "This is an adapted Roamer® Educational Robot Activity from Valiant Technology Ltd.

moving and turning: and while you do this, you're connecting to powerful mathematical and scientific ideas. Many people mistakenly think this class of application is for young children, but Papert realized that "What's good for thinking is good for thinking" – irrespective of your age or ability [20]. It is worth reminding ourselves of Einstein's many famous thought experiments where he imagines he is a light beam travelling at the speed of light. Valiant have conducted informal experiments asking hundreds of adults "What is the answer four, minus, minus three?" Only 5% have correctly answered 7 and none have answered why. Yes, they can answer something like "two minuses give a plus", but why does this rote learning rule work? Teaching student's to understand is a maxim of the science of learning [21]. Knowing the rule helps people pass exams, but it doesn't help them understand the math. Indeed, it confuses: "two minuses" is mathematical nonsense. Minus is a math operation called subtraction and negative three is the name of a number symbolized as −3. Conceptualizing **Negative Number Arithmetic**: students use the robot to move up and down a number line which includes both positive and negative numbers. Forward means addition and backwards represents subtraction. As they do this, students start to understand the ideas involved.

5. Cooperative Task. These tasks focus on students working together. Cooperation is one of ERA's Sustainable Learning Skills. These abilities form an intrinsic part of work with educational robots, for example, projects involve students doing different tasks to achieve a common goal. However, cooperation is an incidental benefit, while in a Cooperative Task it's the main aim. **Journey to the Stars** involved 5 students with behavioral problems. Getting them to work together on the smallest task was a major achievement. The group program Roamer (a spaceship) to visit various planets. Each member has a set of instruction cards and they can only achieve their goal by sharing their instructions. These students found it a struggle to agree, but gradually with some inspired guidance by the teacher they succeeded. In follow-up conversations, several students admitted they'd enjoyed the result and felt they could apply what they learned to improve their lives.

6. Creativity. Work with educational robots often encourages students to develop their creativity. We say you're creative if you produce a new perspective on the world. We often associate creativity with the arts and ignore the feats of science, mathematics and engineering. Robots belong to both arts and sciences. **In the Doghouse** students make Roamer into a robotic dog and then program it to behave like a dog [22]. If you were a painter you would create an image that when people looked at it, they would think – dog. Of course, as René Magritte's shows in his famous painting, "This is not a pipe" it is a picture that makes you think of a pipe. The robot dog is a robot that makes you think of a dog. One child made a tail out a moderately stiff rubber, they programmed the robot to wiggle so it looked like a dog wagging its tail. While many students programmed their dog to sniff around "trees", one child programmed her canine to chase its tail.

7. Deductive Thinking. Students deduce ideas from their experience of using the robot. The Total Turtle Trip theorem came from Papert's invention of Turtle Graphics - a geometry which describes a robot's movement in space. The theorem states the total amount of turning a robot does when traveling around a closed shape is always equal to

360°. So the journey around an ‘L’ shape is 5 right turns and one left turn of 90°: a net sum of 4 quarter turns in a right direction. In the **Total Turtle Trip** task the students send the robot around the perimeter of some closed shapes [23] As a deductive thinking task you expect the students to discover this Total Turtle Trip rule.

8. Demonstration. In a demonstration, students react to a robot. They cannot influence it. In **The Sunbeam Traveler**, Roamer plays the role of a sunbeam traveling from the center of the solar system and visiting each planet. You place pictures of Earth and sun on the floor of a large hall. Students take pictures of the other planets and stand where they think the sunbeam will stop. They never guess correctly, but they get an emotional charge as the robot gets near their planet before passing them by. Normally they become awestruck at the distances involved.

9. Design. This is the province of robots like Vex and Lego, but some Turtle robots, like Roamer, allow students to create designs. In the task **Hazardous Spherical Objects**, students designed and made a robot for collecting and disposing of waste objects [24]. While you can connect Roamer to kits like Lego and Fischertechnik, you can also use methods from the Maker Movement. In the early 1980s, British schools created a subject called design technology. This involved using junk materials and basic tools to make wonderfully inventive machines it embraced many of the ideas now included in the Maker Movement.

10. Engagement. You apply this to activities whose primary purpose is to engage the student. Engagement is an ERA Principle, but that refers to the general nature of robots to attract pupils. **The Chicago Story** is an example where a US teacher worked one-on-one with a problem student. His home life was a disaster and this reflected on his experience of school. The teacher introduced him to Roamer and started to explain how to program it. She asked, “How much do you want to make it turn?” He replied, “I want to make it turn all the way around,” and he tried the instruction right 4: the teacher said, “He was visibly shocked when it only turned 4°.” But the experience had him hooked. She reported that for the first time in his life he engaged with mathematics and in a 45-minute session he discovered the magic number 360. Before this, he had never used double-digit numbers.

11. Experience. Robots can help students build learning foundations based on experience. Papert said, “That experience is at the heart of Piaget’s real message, knowledge built on experience.” When 2 years old Papert became fascinated by gears, he learned how they worked and what happened to one when you turned another [25]. This helped the teenage Papert, to understand equations with two variables. Many of his friends who lacked his gear background struggled with the math. Papert’s stated an educational researcher would not have noticed this key psychological moment. The science of learning now recognizes the importance of prior knowledge. In the **Pizza Delivery Roamer**, the robot is working on a number line. The students program the robot to deliver Pizza to some of the houses. “Does the delivery order matter?” Students use the robot to explore and then explain the problem. Normally, you do this activity with students younger than 7, but they do not need to know, they’re dallying with the mathematical idea of equivalence. There's no need to tell them, but they will subconsciously store the experience for future use.

12. Experimentation. You perform an experiment on how the robot behaves and you use that data to solve a problem, puzzle or challenge. In the **Robot Rally Race**, students test how fast Roamer travels over different terrains and they use that data to calculate the fastest route around the rally course [6]. They then test the results.

13. Exploration. We use the robot to explore and discover the knowledge hidden in a Microworld. This exercise adds excitement to primary school history lessons. For example, Roamer is an **Archaeologist** and it starts to explore an Ancient Roman Site. Pupils program Roamer to explore the site. They discover artifacts and patterns that tell them whether the site was a marketplace, a barracks or a Roman bathhouse.

14. Focused Task. We use a robot to help students understand a specific learning objective. In **Understanding Fractions** we use Roamer to help students grasp key ideas about fractions. In Part 1 we ask the student to program the robot to move forward a distance 4 times. We mark the stopping points and discuss each step is a quarter of the whole distance traveled. In Part 2 the students repeat the first task but the robot moves forward a different length. In Part 3 the robot moves based on time on not distance. In Part 4 we ask the students to program Roamer to travel along a path and split it into quarters. However, the path is not straight: Roamer has to turn. Should they use time or distance? Will they get different answers? You can repeat the activity based on different fractions for example halves, fifths and thirds. Part 1 shows students an idea, part 2 consolidates it, part 3 extends it and part 4 challenges the student's understanding.

15. Games. We use the robot in a competitive challenge between teams or individual players. In the **Quizzer** game, we place Roamer in the center of an obstacle course which has two goals [6]. The teacher asks a question and the first team to answer correctly wins a randomly chosen instruction card. Rules govern collecting, playing or stealing instruction cards. Playing the instruction card means programming Roamer to move towards your opponents goal. The first team to score wins. We've found playing this game at the end of a semester an excellent way to test student's knowledge about work they've covered that term. It adds excitement and engagement to a question and answer session.

16. Group Work. Students work with a robot in a group. UK Primary Schools organize students in groups of 6. This gives you an ideal working environment for use of robots like Roamer. Students will work together, discuss ideas and inspire each other or criticize in a way teachers cannot. Like the ET Movie, in the **Little Lost Roamer** task, the robot is an alien lost on Earth. How do the children communicate with it? The students invent a body language that enables the robot to tell people whether it is happy, sad, frightened, hungry and so on. In this case working in groups is not simply an optimum learning environment, it is a critical part of the task.

17. Inductive Thinking. Students develop a theory and test it. One of Papert's core ideas is that children develop theories about the world without any help from adults –

it's simply what humans do. Of course not all their theories are correct, but, any theory, right or wrong forms part of learning. Papert calls these transitional theories. The Turtle gave students an object to think with – a tool that they could use to create and verify their ideas. In **Negative Number World** you set up Roamer to simulate arithmetic with negative numbers. The robot keypad has both positive and negative numbers and instead of forward and backward you have plus and minus keys. When the students explore the robot behaviour they form theories which they can then test.

18. Links. You can use robotic tasks to link several ideas. Roamer offers endless opportunity to present problems in exciting ways and the links approach shows students how ideas interconnect. In the **Adventures of Myrtle the Turtle**, the robot travels back in time and invents geometry [26]. The time traveler teaches a caveman to measure so he can build a bridge across a river. Then Myrtle showed the Egyptians how to draw squares and triangles, enabling them to build the Pyramids. She then taught Pythagoras about polygons and Archimedes about the circle.

19. Memorization. This theory states that robots can help students memorize facts. This is a theory I am currently testing. There are a few brain based theories on memorization. The "Method of Loci" or the "Memory Palace" methods come from around the first century BC. They're published in a book by an unknown author on rhetoric called "Rhetorica ad Herennium". It is still a technique used by 8 times World Memory Champion Dominic O'Brien, who calls it journeys. The basic idea is it easier to remember if you attach your memory to the location on a journey. This seems ideal for Roamer. The Tour Guide task involves students programming the robot to take visitors on a guided tour of facts. If the theory proves correct the students will do better at recalling the facts. It is included here as a candidate.

20. Modeling. You can use robots to model ideas. **Follow That Curve** uses graphing software like GeoGebra to create a graph, which controls the motion of the robot. Students can see what a graph means by watching it move Roamer. They can experiment and see how the robot responds to changes they make to the graph. Once again the value of the robot is giving students concrete meaning of abstract mathematical ideas. Other modelling scenarios exist which robots also help students model ideas.

21. Pacifier. Educators have used Roamer to overcome student's bad temper and resistance to learning. Roamer is both simple and neutral. In a Cranial Trauma case study, Professor Christian Sarralié reported the case of an adolescent student who had suffered brain damage in a car accident [27]. He had lost the ability to do basic math and became aggressive when teachers tried to reteach him basic arithmetic. Eventually, his teachers gave him a Roamer task, which needed the student to use simple addition. He then realized is the problem, relaxed and became teachable. Roamer has a history of use by teachers working with students whose behavior means they cannot work in a customary classroom setting. They normally work one-to-one with special tutors.

22. Presentations. Students use robots to present their work. My work using Assessment for Learning methods shows presentations create opportunities for peer assessment. In, **On the Buses**, students design bus routes and timetables, which they model with Roamer [28, 29]. They used the robot to present their solution and explain their thinking. Other students had worked on the problem asked insightful questions in what turned out to be a lively debate.

23. Problem Solving. Robots provide excellent opportunities to engage students in problem solving tasks. Educators recognize problem solving is an effective teaching strategy. Valiant use the term problem solving for complex problems and challenges for small-scale simple problems. In **Supply and Demand**, students use the robot to redistribute tea, coffee, milk and sugar between 4 cities. The problem has different levels. You can start students programming the robot to get equal amounts of each commodity in each city. You can gradually make it more complex by adding time constraints and limits to how many commodities you allow the robot to move at any one time. Finally, you ask the students to model a business whose aim is to make profit.

24. Projects. Valiant reserve the word projects for tasks that take several weeks to complete. Students work on projects as a team, but they often work independently on different parts of the project which they need to bring together. **The Oxon Hill Middle School Peace Pledge** is a Robotics Performing Arts™ project [30–33]. Students write the script for a movie, create the characters and sets for the film. They program the Roamer to act out the parts, film it and edit the movie. Although the robot is central to the project, not all the project team, such as the video editor, work with the robot.

25. Provocateur. You can use a robot to provoke students into thinking and discussing an issue. A catalyst starts an activity, but your aim with Provocateur is to get the students into a discussion. In, **"What Number is This?"** the robot moves up and down the number line telling the student the name of the number it stops on. It then starts asking the students to name the numbers. When the robot moves off the line to a number like 12 it provokes a discussion. The students know the names of numbers higher than 9 so they can answer the question. You provoke a bigger debate when Roamer moves to a negative number and asks, "What Number is This?" When Valiant tried this activity students eventually suggest "adding numbers going the other way". In other words, they invent negative numbers.

26. Puzzle. You can use educational robots to solve a puzzle. The dictionary defines a puzzle as, "a toy, problem, or other contrivance designed to amuse by presenting difficulties to be solved by ingenuity or patient effort". They're usually abstract problems, where it is normal for a challenge or a problem solving task to have a context. In the **Biggest Number** (Fig. 1) students write a program that will move the robot from the start to the finish around a 5×5 matrix [13]. Each cell of the matrix as a number and the robot picks up a mathematical operator depending on how they arrive at the cell. They can only visit a cell once. The highest score found to date is a remarkable 760,230.

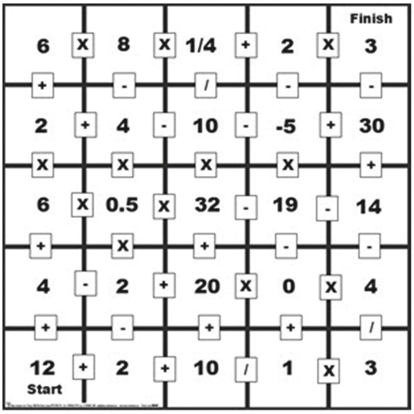


Fig. 1. **Biggest number grid**

27. Relational Artifact. People form bonds with objects and evidence shows they form special bonds with robots. The ERA paper deals with this in detail. Sherry Turkle reports experiments with students and seniors who interact with robots as though they were living creatures. Turkle cites examples of children revealing deeply held concerns that affect their lives [34, 35]. Valiant have direct experience of this. Teachers over-heard a young girl talking to Roamer and realized somebody was abusing her. You can use this in a positive way when introducing children to a robot. **Getting to Know You** introduces young children to the robot and teaches them how to look after and program it. I believe this principle has a lot more to offer in the future.

28. Topic Work[2]. When teachers organize lessons around topics, robots can contribute. **The Great Fire of London** is a topic studied in all English Primary Schools. If you search the Internet you'll find plenty of teacher made resources on this topic. They include lessons in English, history, math, art, science and so on. In the Roamer task, the robot is a fire ember jumping between the closely packed medieval buildings. This simulates how the fire spread from Pudding Lane and burned down nearly 90% of London's homes. Children program the robot to spread the fire to as many buildings as possible, while others find ways of stopping the robot arsonist destroying the city. This ties the children into thinking about how fire spreads, what materials burn and fire safety.

29. Transfer. Transfer happens when you apply knowledge gained in one context to another. This takes place in three ways. You learn something in class, and use it with Roamer. You learn something with Roamer and use it in class. You apply a Roamer lesson to a different Roamer challenge. We tag tasks that use these methods as examples of transfer. The sequence example, consolidate, extend and problem solve involves transfer. You can for example follow up Understanding Fractions with non-robotic work on fractions or you can do it the other way. In **Plow the Field** students study medieval

[2] In the original list this class of activity was called curriculum.

agriculture. They learn how Imperial Measurement (miles, furlongs, chains, yards) came from this farming activity. The students write programs using repeat commands to simulate the process. In **What's That Tune** students adapt the same mathematical and programming skills to programming the Roamer to play music.

4.5 Using the Pedagogical Tags

You need to use several tags to describe an activity. It is important to write the main tag first and then add others – usually limited to 3 or 4 tags in total. It is essential you develop consistency. Table 1 lists a few examples taken form Sect. 4.2.

Table 1. Examples showing the use of multiple Pedagogical Tags.

Activity	Main tag	Other tags
In the Doghouse	Creativity	Design, Inductive Thinking, Presentation
Biggest Number	Puzzle	Focused Task, Game, Engagement
Robotic Performing Arts™	Project	Design, Group Work, Links
Pizza Delivery Roamer	Experience	Modeling, Inductive Thinking, Challenge

5 Conclusions

This paper clarifies details of ERA's Pedagogical Principle. This has two parts the first covers developmental theories and the science of learning, and the second to a Pedagogical Tagging Scheme. This scheme identifies 29 tags which we can use to characterise educational robotic activity. The value of developmental theories lies in their ability to help design better educational robots and better activities. Better means the capacity to help students to learn more, gain a deeper understanding, learn faster, remember longer and enjoy the experience. The tagging scheme describes fundamental nature of an activity irrespective of how, where, when or who uses the activity. The tags do describe the basic nature of an activity in an abstract way, which helps researchers compare different tasks.

Acknowledgements. I would like to thank Martin Kandlhofer for reviewing the document and his helpful suggestions.

References

1. Catlin, D., Blamires, M.: The Principles of Educational Robotics Applications (ERA): a framework for understanding and developing educational robots and their activities. In: Proceedings of Constructionism Paris (2010)

2. Mills, R., Staines, J., Tabberer, R.: Turtling Without Tears: National Council for Educational Technology, A report of the DTI/MESU curriculum development project. The benefits associated with the use of floor turtles in the classroom. England (1989)
3. Holmquist, S.K.: A multi-case study of student interactions with educational robots and impact on Science, Technology, Engineering, and Math (STEM) learning atitudes. University of South Florida, Ph.D. Disssertation. Tampa Bay (2014)
4. Mitchell, D.: An Investigation in to the use of Roamer Turtle in developing the spatial awareness of visually impaired children. Ma Thesis: University of Reading. Reading (1992)
5. Cardoso, I., Cardoso, I.: Roamer Robot in Portugal. EuroLogo Warsaw (2005)
6. Catlin, D.: Roamer Olympics. http://roamerrobot.tumblr.com/post/27185616533/roamer-olympics
7. Csink, L., Farkas, K.: Lifelong playing instead of lifelong learning. In: International Workshop "Teaching Robotics, Teaching with Robotics" Hosted by SIMPAR, Darmstadt (2010)
8. Gay, P.: Roamer in space. Br. J. Vis. Impairment **8,** 103–106 (1990)
9. Betts, J.S.: I thought i saw a turtle roaming. In: Maddux, C.D., LaMont Johnson, D. (eds.) Logo: A Retrospective, pp. 183–188. Hawthorne Press Inc., Binghamton (1997)
10. Char, C., Forman, G.E.: Interactive technology and the young child. In: Wright, J.L., Shade, D.D., (eds.) Young Children Active Learners in a Technological Age, pp. 147–177. National Association for the Education of Young Children, Washington DC (1994)
11. Finley, S.: Roamer Product Review. Multi Media & Internet @ Schools, November 2005. http://www.infotoday.com
12. Roberts, R.: Project for Digital Literacy Program Evaluation: San Diego State University (2003)
13. Catlin, D.: Roamer a Big Bang Experience (2013). http://roamerrobot.tumblr.com/post/45871836724/roamer-a-big-bang-experience
14. Reyna, V.: What is scientifically based evidence: presentation given at the use of scientifically based research in education. In: Working Group Conference Department of Education, Washington DC (2002)
15. Lammer, L., Vincze, M., Kandlhofer, M., Steinbauer, G.: The educational robotics landscape: exploring common ground and contact points. In: International Conference on Robotics in Education, Vienna (2016)
16. McTighe, J., Wiggins, B.: Understanding by Design Framework, ASCD, Alexandria, Va, USA (2005)
17. Catlin, D., Blamires, M.: The e-Robot Project: A longitudinal on-line research collaboration to investigate the ERA Principles. Darmstadt: Learning. International Workshop "Teaching robotics, teaching with robotics" hosted by SIMPAR, Darmstadt (2010)
18. Catlin, D.: Going to the Seaside. GO Magazine. http://go.roamer-educational-robot.com/activities/going-to-the-seaside-roamer-murder-mystry-game
19. Clayson, J.: Visual Modeling with Logo: A Structural Approach to Seeing. MIT Press, Cambridge (1988)
20. Papert, S.: Talking Turtle. Open University Production in Association with BBC Horizon, Narrator Martin Jarvis, Academic Consultant Peter Raggatt (1983). http://research.roamer-educational-robot.com/1983/12/14/talking-turtle/
21. Bransford, J.D., Brown, A.L., Cocking, R.R. (eds.): How People Learn: Brain, Mind, Experience and School. National Academy Press, Washington DC (2000)
22. Catlin, D.: In the Dog House. Roamer Tumblr Site (2014). http://goo.gl/sOZbvo
23. Papert, S.: Mindstorms, Children Computers and Powerful Ideas., p. vii. Basic Books, New York (1980)

24. Farley-Pettmann, K.: Hazardous Spherical Objects. GO Magazine (1981). http://go.roamer-educational-robot.com/1991/05/01/hazardous-spherical-objects
25. Papert, S.: Gears of My Childhood. Roamer Research, 15 August 1980. http://research.roamer-educational-robot.com/1980/08/15/gears-of-my-childhood/
26. Gregory, C.A., Ginn, A., Catlin, D.: Adventures of Myrtle the Turtle. GO Magazine (1991). http://goo.gl/LZr1WR
27. Sarralié, C.: The Roamer: an object for readapting in the case of adolescents with a cranial trauma (2002). http://research.roamer-educational-robot.com/1998/05/18/the-roamer-an-object-for-re-adapting-in-the-case-of-adolescents-with-a-cranial-trauma
28. Catlin. D., Otenio, E.: Roamer Routemaster. GO Magazine (2015). http://go.roamer-educational-robot.com/2015/01/31/roamer-routemaster
29. Catlin, D., Otenio, E.: Educational robots and mathematical modelling. In: 6th International Conference on Robots in Education. Springer, Yverdon-les-Bains (2015)
30. Catlin, D.: Robotics Performing Arts Project: An approach to STEM through cooperation not competition. Constructionism Paris (2010)
31. Catlin, D.: Roamer movie premiers at US congress. GO magazine (2016). http://go.roamer-educational-robot.com/performing-arts-and-us-congress
32. Catlin, D.: Robotic performing arts: GO magazine (2012). http://go.roamer-educational-robot.com/2012/03/15/robotic-performing-arts
33. Catlin, D.: Robotic performing arts: an artisitc approach to STEM. Roamer Podcast (2016). http://podcast.roamer-educational-robot.com/2014/02/03/robotic-performing-arts
34. Turkle, S.: Evocative Objects: Things We Think With. MIT Press, Cambridge (2007)
35. Turkle, S., Taggart, W., Kidd, C.D., Daste, O.: Relational Artifacts with children and elders: the complexities of cybercompanionship. Connection Sci. **18**(4), 347–361 (2006)

Orbital Education Platform: Introducing Orbital Robotics to Secondary Education

Nikolena Christofi, Evangelos G. Papadopoulos(✉), and Iosif S. Paraskevas

National Technical University of Athens, 9 Iroon Polytechneiou, 15780 Athens, Greece
nikolena.christofi@gmail.com, {egpapado,isparas}@mail.ntua.gr

Abstract. In this paper, the development of an Orbital Robotics educational prototype platform is presented, consisting of a hardware-developed physical system and an accompanying set of curriculum-based lessons (IB Physics curriculum) that target upper secondary students (16–18 years old target group). The platform was the outcome of a joint project between the European Space Agency and the National Technical University of Athens. The students use the hardware interactively, consisting of a bespoke space-emulating table and small satellite mockups. The lessons are inquiry-based and are structured so that the students are actively engaged in the learning process, according to a learner-centric approach. During the lessons, this platform allows students to acquire knowledge of the dynamics of space systems, as well as of the environmental conditions and physical constraints that characterize on-orbit operations. Students are able to manipulate the space robot (satellite) mockups performing basic tasks such as docking and landing, or grasping space debris. Additionally, a smartphone application has been implemented to allow the interaction with the platform, via a dedicated User Interface (UI).

Keywords: Educational robotics · Orbital science · Physics · Secondary education · IB curriculum · ESA education

1 Introduction

The use of robotic applications in education is increasing rapidly, after the realization of the various benefits they can bring to the learning process. These include both the motivational and pedagogical value provided by a direct hands-on and interactive experience, that places the learner at the centre of an educational journey of discovery; during this journey the learner's critical and innovative thinking is stimulated, leading to a deeper understanding of physics and technological principles, [1]. A basic advantage is that it engages the students in synthesis and engineering, both of which are missing from many secondary education schools' curricula.

Currently, the approach to STEM (Science, Technology, Engineering and Mathematics) Education in Europe is still based extensively on passive learning (theoretical knowledge transmitted through lecturing); to a large extent, it still

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_11

lacks the practical, inquiry-based and learner-centered dimension required for students to become an active part of the learning process. This would lead to an efficient integration of theoretical knowledge into an experimental process of learning.

Inadequate STEM teaching in school is considered by several recent European studies as one of the key factors responsible for the relative decrease of young people's interest in STEM-related studies and professions in western countries today, [2].

Under this scope, the Educational Project *OrbiLEP: Orbital Levitation Educational Platform*, aims to provide an active-learning tool for physics and technology, specifically in the fields of orbital mechanics and space robotics. The development is taking place at the facilities of the Automation and Robotics Laboratory (ARL) of ESTEC, ESA's European Space Research and Technology Centre in the Netherlands.

The project, aimed at the upper secondary school target group (16–18 years old), includes the development of an educational space environment emulator using air levitation (planar space emulator platform) and a set of space robot (satellites) mockups, which can float on its surface (for the rest of this work the term mockup will be used for the robots). A set of lessons, which for the prototyping phase are based on the International Baccalaureate's (IB) Physics guide, accompanies the emulator. An earlier implementation of this work can be found in [3].

2 Pedagogical Targets

The challenge of the lessons is to raise student interest and engagement in STEM subjects. This is achieved by guiding the students through the application of physical principles in the space environment - a context usually fascinating to students. For example, as the teacher demonstrates a physical phenomenon, such as rolling a ball on the table which stops at some point, s/he asks the students to describe what they saw and to explain what happened. In this way, the teacher can introduce Newton's Laws and friction. At the same time and in order to highlight the difference between space and Earth dynamics, the teacher shows a video of a satellite on orbit, of landing on an asteroid, and of docking between two objects in space.

During such a lesson, the students are called to describe both what they saw and to notice the differences that exist between the bodies' behaviour on Earth and on orbit. In this way, the students start to think about what they have just experienced. Meanwhile, the teacher keeps stimulating the active learning process by searching for the rationale behind a space mission. Essentially, the methodology developed seeks to trigger all the learning routes that students with different personality types choose to follow into perceiving, acting and decision-making.

2.1 Lessons' Content

The topics addressed through the lessons are based on the IB Physics Curriculum. The first lessons address basic physics principles and build up to discuss more advanced concepts. Alongside with the physics laws, in principle, the lessons' content will raise awareness on space in general and space missions: motivation for space missions, basic mission analysis, robot interaction in free-floating environments, docking, space debris and trajectory design (in a gamificated final lesson).

3 Pedagogical Approach

The pedagogical approach followed in this work consists of deploying a set of inquiry-based lessons that gradually build up the knowledge and the skills needed to complete the final lesson, which makes use of principles introduced in the previous ones. The learning process and the lessons, whose goal is teaching physics through the students' experience of a two-dimensional representation of space dynamics, implement cognitive neuroscience research and developmental psychology known as MBE (Mind, Brain and Education) Science, [4]. Through an interactive use of the physical platform, the learning process also introduces a hands-on experience. Methodologies are being implemented that are derived from the belief that the human brain constantly searches for meaning and seeks patterns and connections [5], while the main focus remains on reaching a deep understanding of the basic school physics concepts that lie in the fundamentals of space and orbital robotics.

The lesson plan is based on the 4MAT Theory [6], a use of the extensive research on brain-based teaching methods. The implementation of the 4MAT Cycle into teaching engages the students through all the steps of the learning experience. The movement around the 4MAT cycle represents the learning process itself; it is a progression from: (a) experiencing, to (b) reflecting, to (c) conceptualising, tinkering and problem solving, to (d) integrating new knowledge with the self.

The development of the lessons takes into account the different personality types (according to the Myers Bricks' type indicator [7]); all personality types are equally respected. The 16 distinctive personality types result from the interactions among the preferences: (a) *favorite world*: whether one chooses to focus on the outer world or in his/her inner world -Extraversion (E) or Introversion (I), (b) *information*: whether one prefers to focus on the basic information he/she takes or he/she prefers to interpret and add meaning -Sensing (S) or Intuition (N), (c) *decisions*: when in dealing with the outside world, whether one prefers to get things decided or he/she prefers to stay open to new information and options -Thinking (T) or Feeling (F), (d) *structure*: in dealing with the outside world, whether one prefers to get things decided or he/she prefers to stay open to new information and options -Judging (J) or Perceiving (P) [8].

In more detail, the *sensing types*, perceive what is happening around them making logical connections in their mind, opposite to the *intuitive types* who use their intuition. Some take decisions based on their logic and others on their emotions (*thinkers* and *feelers* respectively), while for some, the decision making process is more difficult than for others (*perceivers* and *judgers* respectively). Personality types also account for the different channels one uses to express oneself: for the *extraverts*, the source of energy is external, while for *introverts* it is internal.

Overall, the lessons suggest a pedagogical approach that follows a pattern that provides the students the opportunity to acquire the knowledge in the way that is the easiest and most preferable for them, and are meant to favour the development of their communication skills, since most of the work is carried out in teams. The students are led through interactive activities into the basics of physics principles and of their specific phenomenology in the space environment. Every lesson is structured around a themed assignment, or mission, that makes use of orbital robotics and which has to be accomplished as a team. At the end of the assignment, the students have to present the process they have run through, the results of their mission, and the outcome of their teamwork in which every member had specific tasks. Then, under the guidance of their tutor, they have to engage in an open discussion to share the trade-offs made, the challenges overcome, the lessons learnt both individually and as a team, etc. Through this process, we expect the students to gain a first-hand understanding and direct experience of the curricular subjects introduced, an experience of project management and teamwork, and the development of their communication skills, self-confidence and awareness in a learning environment that respects the wide variety of personality and learning types.

4 Platform Design Objectives

Assuming lab infrastructure is not present, a central objective of this work was to create a hardware platform that is affordable for schools and suitable for classroom environments. However, this introduces many challenges to the design process. The implementation should be of a low-cost, low-weight, low-energy, portable and easily mountable system, while allowing the students to reach the learning objectives of each lesson through active participation. In more detail, the physical platform design requirements includes:

- Emulation of the behaviour of space robots in a 2-Dimensional environment: resemblance to the effects of zero-g environment on bodies on orbit, in 2D;
- encouragement student interactivity;
- small-sized, low-cost, low-weight, low-noise and safe for use in a classroom environment;
- sustainability and reusability by different student ‘generations’.

These requirements and constraints led the design decisions and trade-offs made during the prototype development. The final layout, the selection of materials, and the functionality of all components have been developed in accordance with the educational scope of the project. As the construction of a 3D emulator is out of the scope of this work, a planar space environment emulator allowing for an effective two-dimensional representation of satellite motion in space was chosen. The emulator was based on the concept of a frictionless air-hockey table, allowing the mockups to hover on top of it, in zero-gravity and with zero-friction.

The table had to provide working space for two mockups, such that a number of interesting operations could be performed on its surface. The mockups had to be lightweight and with a small footprint, and be capable of performing basic space tasks resembling to real satellites with a good degree of fidelity. A great effort has been put into reducing the minimum power and mass budget requirements of both the mockups and the table.

5 The Planar Space Emulator

5.1 Concept and High Level Requirements

From the early stages of the development, one of the main challenges was the design and construction of a system able to emulate zero-gravity environment conditions, while at the same time taking into account all project requirements and the physical limitations. The chosen design incorporates the benefits of planar simulators already developed in the USA (MIT, Stanford), Europe (U. of Padova, U. of Southampton, NTUA, etc.) and Japan (Tohoku University), scaled down to a low budget and to a simplified system that does not need compressed air. Planar emulators using air bearings are perhaps the most versatile and least expensive systems for emulating zero-g environment, and allow for repeated and thorough testing of control algorithms and verification of dynamics [9]. They require minimal preparation compared to other emulation methods and are easier to upgrade and adapt to alternative scenarios.

These systems usually use a thick and massive (e.g. granite) or fragile (e.g. glass) material as the surface on which the air bearing space emulators hover. However, these are not suitable for a classroom environment. Moreover, the direct use of air bearings would have introduced a considerable weight, cost and complexity to the mockups due to the required payload for gas or air supply. Therefore it was decided that the platform would follow the example of planar emulators using air bearings, although this time the method of hovering would be reversed: the air flow would come out from the table instead from the mockups. For this reason, the planar educational space emulator developed follows the air hockey principle consisting of an assembly made of additive manufacturing (AM) parts.

5.2 Implementation

The platform is designed as a modular structure consisting of small modules. The modules are glued together to provide a total working surface (top surface of the upper part of the assembly) with dimensions $380 \times 300\ \mathrm{mm}^2$. The module's upper part is a modular box-shaped element -more parts can be glued together in order to increase the total surface, with custom-made nozzles and holes on top, which produce the air film. Each box-shaped part has provision for attaching a fan at the lower part or a cap (sidewall) in case of no fan. The lower part includes the mounting point for the fan or cap, the mounting element between upper and lower part and the support truss. This way the structure is provided with sufficient support and distance from the ground, while using the least possible amount of material.

In Fig. 1, four modules are connected to form a larger orthogonal working surface; the modularity of the design allows for an increase of the working surface area with the addition of more modules.

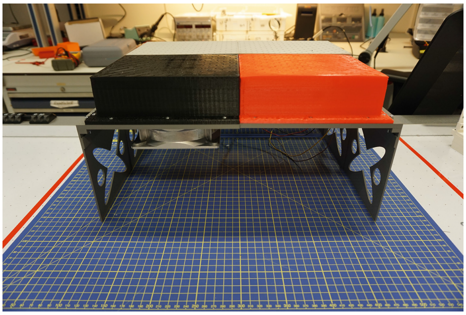


Fig. 1. A working surface (table) obtained by assembling four modules.

It is to be noted that the students are not aware of the fans-enclosed-in-nozzle mechanism used to simulate the thrusters function on space objects. This alternative in design was used in order to reduce the mass budget of the platform and its complexity. The use of small compressed gas containers (usually of Carbon Dioxide - CO_2) with tubes and valves, is another option that was taken under consideration, however overpasses the satellites mass limit set by the lifting capability of the air levitation platform.

5.3 The Space Robot 'satellite' Mockups

The mockups represent either the bodies orbiting in space (space debris) or satellites. The observation of their movement and the interaction with other objects on the planar space emulator environment are meant to allow the students to study the dynamics of orbiting bodies.

The goal is to make the mockups as small and light as possible. This will enable them to be lifted by the film of air created by the platform, and at the same time decrease the minimum required dimensions of the upper surface due to additive manufacturing limitations. The fact that no pressurized air has beem used by the platform, created various challenges to overcome during design.

In principle, there should be no contact between the bottom surface of the mockups and the upper surface of the platform. However, a small mass misalignment on the mockup could cause an instantaneous impact during its movement. Moreover, aerodynamic friction forces develop within the air gap, resulting from the motion of the mockup over the platform.

Additionally, since resemblance with real satellites is required, translational and rotational motion of the mockup must be generated by thrusters. This would have introduced the problem of using compressed air and a pneumatic system on board, increasing the mass and power budget, not to mention the complexity. Instead, small nozzle-enclosed fans were selected, which look realistic both in appearance and function. To emulate thruster/rocket propulsion, the fans operate in a pulse-like mode, significantly reducing the inertial issues that occur. In addition, their mass is minimal, making the previous assumption reasonable. The first system implementation employs four fans, as shown in Fig. 2; while more fans can be added to allow a full set of motions.

5.4 Remote Control

The fans are connected to an Arduino® system, which is an open-source physical computing platform based on a simple microcontroller board [10], controlled remotely via an android smartphone application and Bluetooth.

6 The Platform Exploitation

The platform (air levitation generator and the satellites mockups), which will be used by the students and for the needs of each lesson, can be upgraded with

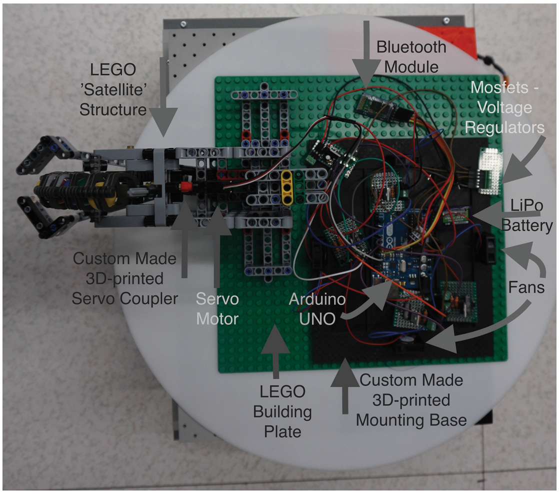


Fig. 2. The satellite mockup - top view

more LEGO® components (on the satellites mockups), during the execution of each exercise. Except from the First Lesson, which will be introductory, the platform is expected to be used in various ways. The Lesson Plan consists of 10 Lessons, which can be excecuted either individually or in a particular order. The concept is that each Lesson builds up on the theory delivered by the previous Lesson, as well as by the structure of the satellite mockups.

Firstly the satellites are used in their primary form, for demonstration purposes only (the friction and absence of friction effect on objects). Then the students are complying the 'thrusters' and observe the satellites' behaviour in a free floating environment. Later a LEGO® arm is mounted on the satellite LEGO® base, which the students are either manipulating with the use of the Android Application, in order to move it, or to shoot LEGO® balls, by assembling and attaching the LEGO® shooter on the arm. Following that, the students are called to build an identical LEGO® satellite mockup, and try to execute tasks between the two satellites, the final one being docking (a LEGO® mechanism with magnets is attached on the mockups for the needs of the docking exercise).

The two last lessons are of higher difficulty. On the 9th lesson the students attach any kind of rope to the balls of the LEGO® balls shooting mechanism, and shoot the ball inside a cavity to 'retrieve' it, thus demonstrating a way of debris removal. In the last lesson, the students manipulate the 'thrusters' through the application and have to go from point A to point B through a maze, while finding the optimal path. The team to reach the final point in the least time is the winning team.

7 Discussion and Future Work

Talking about space always fascinates young students. Through the active discussions and activities taking place in this course, students are led through an introduction to the space environment, orbits, applications of space technology and robotics, and their impact on modern society. They also learn how to work in teams while improving their communication and presentation skills. It is foreseen that a detailed manual will be produced on how the lessons can be replicated in a classroom environment.

8 Conclusion

This paper presents a proposal of an Educational Platform on Orbital Robotics. Using a set of mockups, able to float on a planar space emulator platform, the authors aim to reach OrbiLEP's educational objectives through a lesson plan which focuses on the active interaction of the students in the learning process. The students are led through a set of inquiry-based lessons, and are introduced to concepts of orbital mechanics and space dynamics. This is achieved through a 2-dimensional representation of the zero-g, zero-friction space environment, made possible through the manipulation of the mockups hovering on a space emulator. Additional fine-tuning of the design of the planar emulator platform and of the mockups is intended, as well as the study of the results of the implementation of OrbiLEP in Upper Secondary Education classrooms. Through this platform, the authors intent to increase the awareness of public in STEM.

Acknowledgment. The authors would like to thank the ESA Automation and Robotics Laboratory (ARL) for providing their facilities and assistance throughout the development of the project, and the head of the Automation and Robotics Section, Gianfranco Visentin, as well as Kletil Wormnes, ESA mechatronics engineer, Monica Talevi, Head of Coordination and Outreach Unit, Education & Knowledge Management Office, and Slawomir Zdybski, ISS and International Education Operations Coordinator, for their continued mentorship and support.

References

1. Eguchi, A.: Robotics as a learning tool for educational transformation. In: Proceeding of 4th International Workshop Teaching Robotics, Teaching with Robotics and 5th International Conference Robotics in Education, Padova, Italy, pp. 27–34 (2014)
2. Nugent, G., Barker, B., Grandgenett, N., Adamchuk, V.I.: Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes. J. Res. Technol. Educ. **42**(4), 391–408 (2010). Taylor & Francis
3. Christofi, N., Talevi, M., Holt, J., Wormnes, K., Paraskevas, I., Papadopoulos, E.: Orbital robotics: a new frontier in education. In: 6th International Conference on Robotics in Education (RiE 2015), Yverdon-les-Baines, Switzerland (2015)
4. Tokuhama-Espinosa, T.: Why mind, brain, and education science is the "New" brain-based education. New Horiz. Learn. 9(1) (2011)
5. Thinking Styles. http://www.robertjsternberg.com/thinking-styles/
6. McCarthy, B., McCarthy, D.: Teaching around the 4MAT Cycle: Designing Instruction for Diverse Learners with Diverse Learning Styles. Corwin Press, Thousand Oaks (2006)
7. Pearman, R., Albritton, S.: I'm Not Crazy, I'm Just Not You: The Real Meaning of the 16 Personality Types. Nicholas Brealey Publishing, London (2010)
8. The Myers Briggs Type Indicator (MBTI), The 16 Personality Types. http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/the-16-mbti-types.htm
9. Schwartz, J.L., Peck, M.A., Hall, C.D.: Historical review of air-bearing spacecraft simulators. J. Guidance Control Dyn. **26**(4), 513–522 (2003)
10. ARDUINO. http://www.arduino.cc

A Scenario-Based Approach for Designing Educational Robotics Activities for Co-creative Problem Solving

Vassilis Komis[1(✉)], Margarida Romero[2], and Anastasia Misirli[1]

[1] University of Patras, Patras, Greece
{komis,amisirli}@upatras.gr
[2] Université Laval, Québec, Canada
Margarida.Romero@fse.ulaval.ca

Abstract. Educational robotics are increasingly integrated from early childhood to postsecondary education. In some cases, robots are used as an extracurricular activity that is not oriented towards the learning objectives of the curriculum. In other cases, the educational robotics activities are aligned to the curriculum objectives or the development of the 21st century skills such as collaboration, problem solving, creativity, critical thinking and computational thinking. In this paper, we introduce a scenario-based approach for designing educational robotics activities aiming to support co-creative problem solving in K12 formal education contexts. After describing the scenario to support planning and orchestration of ER in K12 education, we introduce a taxonomy composed of five different ER activities according to the learners' engagement in the co-creative knowledge building and problem solving process.

Keywords: Educational robotics · Robots · Computational thinking · Constructionism · Constructivism · Knowledge building · 21st century education · Problem solving · Educational scenario · Learning activities

1 Introduction

Educational robotics (ER) engages learners in the use of robotic technologies for the development of one or more learning objectives, skills or competencies in formal or informal contexts. According to Komis and Misirli [1] ER refers to the practice of teaching in which students use robots to build knowledge for the robots themselves or with the help of robots. Other authors have defined ER as cognitive tools [2] able to sustain active learning activities [3] in which the learner is actively engaged in the knowledge building process [4].

As a research field, ER "is at the crossroads between psychology, artificial intelligence and educational sciences" [5]. Research in the field of ER has known an early development with the seminal work of Papert and his colleagues at the MIT in the 1970s [6] and is now becoming an important field of research from early childhood education to postsecondary education [1, 7, 8].

The present paper has the following structure: the first section includes a conceptual framework of educational scenarios as an appropriate epistemological and theoretical

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_12

basis for planning and organising educational robotics (ER) activities. The last section provides a discussion on the interrelation between the proposed taxonomy and competencies developed when co-creative projects are taking place.

2 From Informal Activities to Curricular-Integrated ER Activities

Educational robotics (ER) activities are very diverse, both from the point of view of the robotic tools (Bee-Bot, Mindstorms EV3, Thymio, Nao …) and from the point of view of the learning activities proposed to the learners. Some educational robotics activities are carried out in the form of non-integrated extracurricular activities disciplines [9]. These informal activities are often considered as ludic introductions to educational robotics but are not conducted with the objective of achieving one or more objectives in relation to the K12 curriculum (Fig. 1).

Fig. 1. Informal educational robotics activity in the 'Librairie du Quartier' bookshop in Québec with BeeBot and 'Vibot the robot' children's' book [10].

The informal educational robotics (ER) activities aim to engage children in a ludic introduction to robotics but does not aim to evaluate the learning performances. Despite not having well-defined learning objectives or assessment purposes, children have the opportunity to learn in an informal way the algorithmic thinking (in case of programmable robots or automatons) or even mathematics (measure, arithmetic's…). In informal ER activities, participants can learn from their failures and success in a context without academic pressure. Not having explicit learning objectives or an explicit alignment with the curricular objectives does not prevent learners to learn and develop highly valuable knowledge and skills that can later be transferred to a formal educational setting. For this reason, the extracurricular ER activities, summer camps or other informal activities inviting children to explore (if possible creatively) the design, construction or programming of robots are valuable for their education.

In other cases, ER activities are integrated in formal educational settings (often in relation with STEM education) but the level of alignment with the curriculum is variable. Among these activities, some follow a procedural step, step by step, which leaves little room for creativity. In other cases, the robotic activities are integrated in an interdisciplinary way and allow the learner to engage in a collaborative and creative approach, integrating a complex-enough challenge which requires him to engage in a problem-solving activity [8]. Some of these activities are aligned with the co-creative uses of technologies [11] in which learners are engaged in ill-defined situations that require them to use co-creative problem solving strategies [12].

In order to ensure that the ER used in the classroom can engage the learners in a curricular-integrated activity, we propose a scenario-based approach for designing ER activities in the next section.

3 A Scenario-Based Approach for Designing ER Activities

Educational scenarios aim to design a learning situation composed by different activities in order to achieve certain learning objectives through different learning strategies. Pernin and Lejeune [13] describe the learning scenario as "a description, made a priori or a posteriori, of the progress of a learning situation at a given level, or a learning unit, whose goal is to ensure the appropriation of a precise set of knowledge. A scenario describes roles, activities and also knowledge resources, tools and services necessary to the implementation of each activity". Learning scenarios include instructions for teachers, a theoretical framework for each addressed problem, the materials required for implementation, activity sheets for students, and possibly other materials (e.g., software, robotic tools, lesson plans). The learning scenario is therefore implemented as a series of teaching and learning activities. In this context, the scenario is a complete instructional intervention: it encompasses specific objectives, goals, the consideration of potential learning problems, and the implementation process, including appropriate activities and teaching strategies, assessment procedures, and so forth.

In Komis et al. [14] a framework for designing educational scenarios that incorporate digital technologies as a guide for prospective and practicing teachers was proposed. The design and implementation framework for educational scenarios integrates digital technologies into the pedagogical approach according to contemporary learning theories. It is also based on principles of science teaching, according to the main components of the TPACK (Technological Pedagogical Content Knowledge) model [15]. An educational scenario that integrates digital technologies (in our case, robotics) as an educational and cognitive tool, describes the teaching and learning activities and the tools used (abstract tools such as schemata or software and/or physical tools such as special artifacts), which constitute both the starting point for teachers and learners' activities and the framework within which they will take place. It involves the application of effective teaching and learning strategies as a means of achieving learning objectives through the use of an appropriate digital environment (educational software, robotics tools or other materials). In most cases, the scenario targets the teaching and learning of one or more main concepts in a curriculum subject area (for example programming or robotics or both). The scenario may also address concepts that belong to different subject areas, in an interdisciplinary perspective, or it may target concepts beyond the curriculum. This is the case when the use of robotics enhances concepts from STEM education in fostering art or social skills.

In Misirli and Komis [16] the educational scenario was modified to create an appropriate conceptual model for teaching, especially programming concepts through the integration of robotics. This conceptual model aims to minimize methodological faults in order to maximize the validity of research findings developing a knowledge construction and competencies adopted in the learning process. Therefore, five different types of educational activities are proposed (Fig. 2).

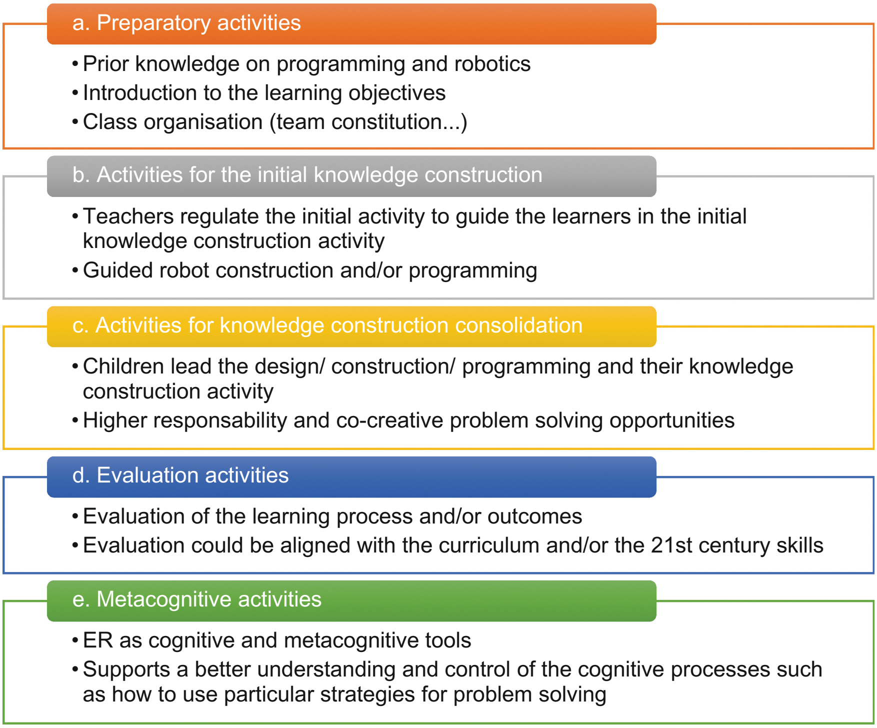


Fig. 2. Overview of the five activities constituting the ER scenario.

Here are the descriptions for the activities within the ER scenario:

a. **Preparatory activities.** These activities aim to prepare the learners for the ER activity. The preparatory activities often include a teacher introduction to the scenario objectives in whole of the class, small groups or individual organization, without manipulating a robotic tool. The teacher attempts to trace the children's prior knowledge of programming and robotics and at the same time introduces the robotic tool.

b. **Activities for the initial knowledge construction.** In this second type of activities, the teacher guides the children in the use of ER for manipulation (inquiring, exploring, discovering and cognitive conflict) through peer-group (collaborative) interaction and involvement. In some cases these activities are strongly guided in terms of construction and/or programming depending on the learner's initial knowledge of ER.
c. **Activities for the knowledge construction consolidation.** In these activities, the children have more responsibilities as they get to design, manipulate (inquiring, exploring, discovering and cognitive conflict) and interact with peers. These activities imply more co-creative problem solving opportunities because of the larger opportunities to design, construct and/or program the robot. These activities lead the children through manipulation, to a more reflective process for testing and access either on a robotic construction and its programming or only on programming of a pre-constructed robot.
d. **Evaluation activities** can be embedded through the prior knowledge construction activities or can be carried on in a separate way. Evaluation could focus on learning objectives aligned with the curriculum (e.g. force transmission) or skills such as the 21st century skills related to ER [12]: problem solving [17, 18], collaboration [19], creativity [20], critical thinking and computational thinking [21].
e. **Metacognitive activities** consider robotics as a potential cognitive or metacognitive tool that can bring the children to better understand and control their cognitive processes such as how to use particular strategies for problem solving [22]. Metacognitive activities can also be embedded through the prior knowledge construction activities or can be carried on in a separate way.

We suggest considering the five activities of this scenario for planning and organizing educational robotics (ER) activities that aim to achieve a curricular integration and ensure a progressive level of guidance towards the consolidation of the knowledge building process. The guidance is based on the scaffolding strategies of the Zone of Proximal Development (ZDP) [23]. The scaffolding is partly defined in the initial planning by the teacher but could be adapted according to the prior knowledge identification from the first activity.

4 Diversity in the ER Activities Within the Scenario

After introducing the five-activity scenario to support the planning and orchestration of ER in K12 education, we identify in this section the diversity of the ER activities that could be used in the three first activities of the scenario. Within the ER scenario, the level of engagement in the three first activities could be differentiated according to the learning objectives. The preparatory activities could be done in a more transmissive (lecture-based) style or be conducted as a debate (socio-constructivism) to raise the learners' prior knowledge and to discuss their misconceptions [24] (Fig. 3).

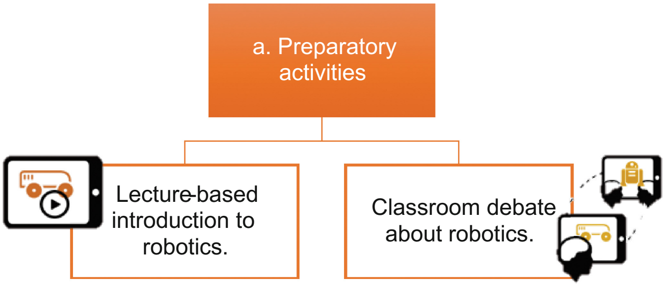


Fig. 3. Diversity of the preparatory activities according the learners' engagement.

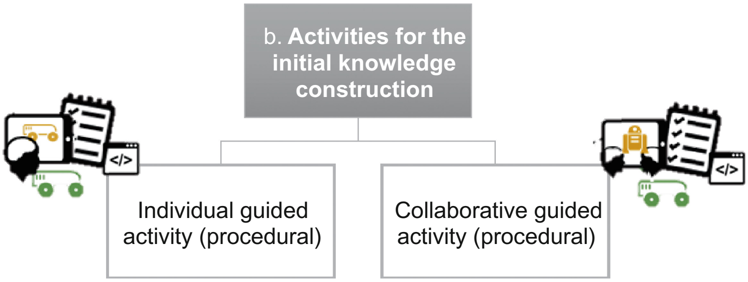


Fig. 4. Initial knowledge activity modalities: individual or collaborative.

The ER activities for the initial knowledge construction usually show a high degree of guidance, often leading to a procedural learning process in which learners follow step-by-step instructions to develop their familiarity with the ER tools and the learning objectives defined for the ER learning unit (Fig. 4).

The third activity of the scenario aims to support the highest level of the learning objectives. In some cases, the learning objectives are limited and the activities involve the learners in individual or collaborative engineering tasks in which they should solve a specific problem of construction or programming, with a limited margin for co-creating a new project. In other cases, the learners are engaged in co-creative project-oriented robotic challenges which not only allow the full potential of ER, but also an important level of complexity, creativity potential and responsibility for regulating the learning project. This makes these activities unsuitable for younger learners and those not experienced or unable to deal with uncertainty (Fig. 5).

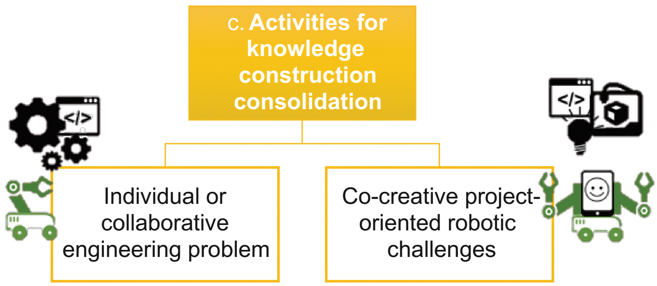


Fig. 5. Consolidation knowledge activities according to the complexity.

4.1 A Taxonomy of ER Activities According to the Learners' Engagement in the Knowledge Building Process

We introduce in this section a taxonomy of activities, based on the degree of engagement of the learner in the knowledge building process. These different typologies of activities are organized into five levels: (1) passive exposure to robotics (without manipulation); (2) discussion or debate about robotics (without manipulation) the individual procedural robotics team is in; (3) individual or collaborative step-by-step robotics (procedural); (4) engineering-oriented robotics (individual or collaborative) and (5) co-creative project-oriented robotics to solve a realistic challenge. Those levels provide content for different ages and educational contexts and may be implemented from preschool through secondary education (Fig. 6).

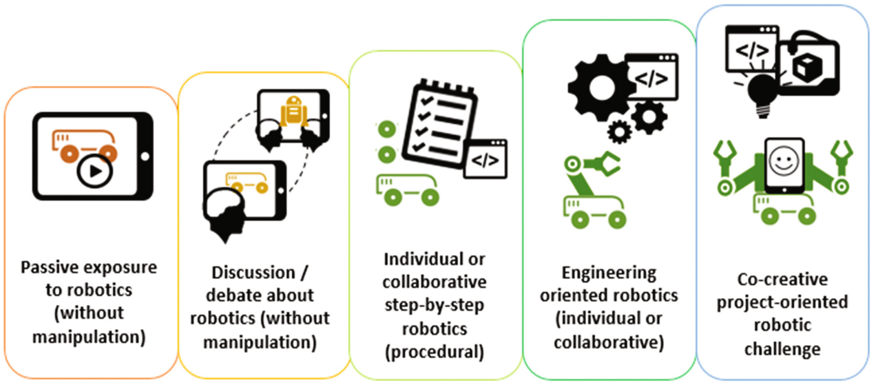


Fig. 6. Learners' engagement according to the educational robotics activities.

We describe in the next section the five levels of the learners' involvement according to the educational robotics activities and the robotic tools available to different ages and educational situations. The first two levels described in this taxonomy do not imply a creative process of solving problems or learning by constructing because they introduce robotics indirectly, without allowing participants to manipulate the robots or their interfaces. However, these two first levels may be of relevance to the development of critical and epistemological questionings about robotics before engaging the participants in joint creative problem-solving activities (the three next levels of the learner engagement). This taxonomy implies a value judgment on the relevance of educational activities because each of these different typologies of activities contributes to sustaining ideas, knowledge and specific skills, with progressive levels of difficulty as those factors are basic elements of a scenario-based learning design.

4.1.1 Level 1. Passive Exposure to Robotics (Without Manipulation)

In the most passive level, the learners listen to the teacher's lecture on robotics, sit in on a conference or presentation on robotics, read resources on robotics (books, articles…) or simply observe a robot and its different capabilities.

This level corresponds to the "Preparatory activities" and has been proven to be of great importance for every teaching planning [25]. These activities imply teaching strategies without manipulation of the robotic tool, in whole-class discussions or peer-group coordination, in which the user/child formulates initial ideas and cognitive representations through inquiries and new discoveries. Therefore, tracing children's former ideas and cognitive representations contributes to an appropriate teaching planning. The present teaching content addresses robotics without programming or engineering and construction concepts. Additional robotic tools for this type of activities are programmable robots (Bee-Bot, Blue-Bot, Pro-Bot, Roamer, Turtle Robot, Thymio, Poppy Ergo Jr, Alpha 1S, Buddy, NAO). Most of them have animistic and humanoid appearances, which help the user/child (especially the young learners) to be part of his learning process and to belong to his team as an equal member.

4.1.2 Level 2. Discussion About Robotics (Without Programming)

Participants discuss robots, their components, behavior and issues concerning their use in society. The discussion can allow collaborative construction of new meanings and could be used to develop critical thinking around robotics in education but also in other areas.

Level 2 includes the same type of educational activities as in the first level, "Preparatory activities". The slight difference between the two levels is their teaching strategies, since manipulation of the robotic tool is involved in peer-group collaboration. The robotic tools (programmable robots) provide to the user/child an immediate familiarisation with a robotic system through direct manipulations. These activities lead to discussions, communication and cognitive conflicts about the robotic tools' functions and controls. The user/child is led to predict possible operations of the robot without having to deal with programming, engineering or construction concepts.

We may call levels 1 and 2 levels of "robotic awareness," as they prepare the user/child for higher-order thinking skills of the learning process.

4.1.3 Level 3. Individual or Collaborative Procedural Robotics (Programming/No Construction)

Procedural ER activities bring the learners to follow step-by-step instructions predetermined by the teacher or by tutorials. The students must follow closely the various stages of construction, programming and execution to achieve a result that is already defined in advance and that leaves no room for change in the process or the creative solution. Although this approach allows creativity, it can support learners (individual and team) in a problem-solving process if they encounter difficulties during the various procedural stages of the activity.

This level is a core one since basic programming competencies (algorithmic thinking, problem solving and engineering) are incorporated to “Activities for subject teaching,” facilitating their construction. Moreover, a teaching plan oriented towards early and primary education may also encompass “Activities for the knowledge construction consolidation” and “Evaluation activities” due to appropriate robotic tools (Bee-Bot, Blue-Bot, Pro-Bot, Roamer, Turtle Robot, Thymio, Poppy Ergo Jr., Alpha 1S, Buddy, NAO).

Teachers should take into account that those three levels are fundamental for the development of computational thinking.

4.1.4 Level 4. Engineering-Oriented Robotics (Programming and Construction)

In engineering-oriented robotics, the learners are engaged in challenges with a certain level of problem solving. Challenges in robotics are often performed as a team and aim to mobilize the co-creative resolution of the problem raised by the challenge. The challenges therefore provide a framework for implementing coopetition game mechanics (intragroup cooperation and intergroup competition) that make challenges fun. They also contribute to the development of code-creative co-solving skills problems.

The typology of educational activities for the present level may be formulated through the most types of a scenario-based teaching planning. Thus, “Activities for the initial knowledge construction,” “Activities for the knowledge construction consolidation,” “Evaluation activities” and “Metacognitive activities” contribute transversally in the cognitive process since a range of teaching strategies are involved. The present level targets apply to older learners since the additional robotic tools (Cubelets, Kibo, Lego®, WeDo™, Lego Mindostorm EV3) bring the user/child to develop construction skills alongside the construction of programming competencies.

4.1.5 Level 5. Co-creative Project-Oriented Robotic Challenge (Collaborative Project Definition, Programming and Construction)

In co-creative project-oriented robotic challenges, learners are encouraged to solve a realistic problem that they can choose in a participative way. These are complex projects with a space for co-creativity for the learner and his ideally large team. Such projects are part of the learning strategies related to project-based learning (project based learning) and help develop creative solutions to real problems (e.g. design and program a robot to facilitate telepresence for a sick child in class) in connection with the community while developing a critical approach to technology (robotics).

In this level, an appropriate teaching plan should incorporate the other levels of the scenario-based approach ("Activities for the initial knowledge construction," "Activities for the knowledge construction consolidation," "Evaluation activities" and "Metacognitive activities") in order to develop programming competencies through construction in a project-based learning environment. The emphasis is especially put on the metacognition process since this function is responsible for developing higher-order thinking skills leading to additional competencies, such as those reflected on Fig. 7.

In order to sustain the development of the 21st century skills (critical thinking, problem-solving, creativity, communication, collaboration, curiosity, initiative, persistence, adaptability, leadership and social awareness), the fifth level of the educational robotics activities is the most appropriate: (5) co-creative robotics project oriented to solve a realistic and challenging (Fig. 7).

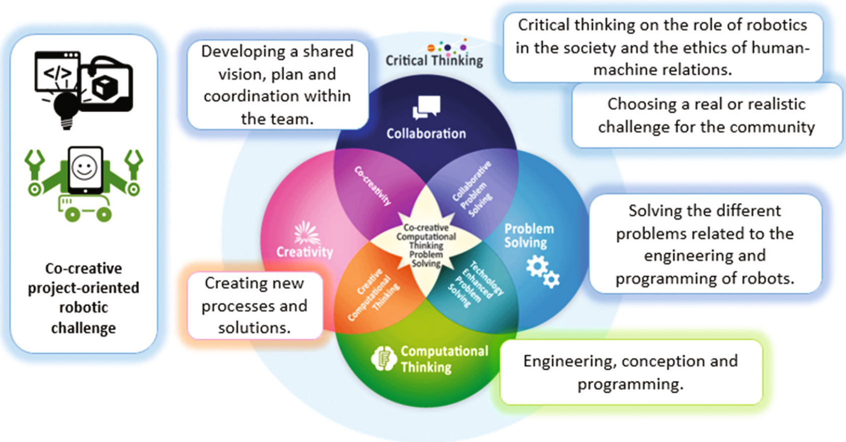


Fig. 7. Sustaining the 21st century skills through co-creative project-oriented robotic challenges.

5 Discussion

We have introduced in this paper a scenario-based approach to design ER activities, which can contribute to the achievement of the learning objectives in K12 education through a constructivist-constructionist approach. Furthermore, we have discussed a taxonomy to differentiate the various uses of ER activities according to the learners' engagement in the co-creative problem solving activity. The biggest opportunities for sustaining 21st century skills are the co-creative project-oriented robotic challenges (fith level of the taxonomy). Despite the opportunities for these complex-oriented activities, some teachers engages involve students in procedural (step by step) robotics activities following very tight instructions which does not allow them to create an ill-defined situation in which they should use their co-creative problem solving strategies. We invite teachers and students to target the co-creative problem solving challenges to develop the full potential of educational robotics for the development of 21st century skills.

References

1. Komis, V., Misirli, A.: The environments of educational robotics in Early Childhood Education: towards a didactical analysis. Educ. J. Univ. Patras UNESCO Chair **3**(2), 238–246 (2016)
2. Mikropoulos, T.A., Bellou, I.: Educational robotics as mindtools. Themes Sci. Technol. Educ. **6**(1), 5–14 (2013)
3. Azhar, M., Goldman, R., Sklar, E.: An agent-oriented behavior-based interface framework for educationa robotics. In: Agent-Based Systems for Human Learning (ABSHL) Workshop at Autonomous Agents and MultiAgent Systems (AAMAS-2006) (2006)
4. Yuen, T.T., Boecking, M., Tiger, E.P., Gomez, A., Guillen, A., Arreguin, A., Stone, J.: Group tasks, activities, dynamics, and interactions in collaborative robotics projects with elementary and middle school children. J. STEM Educ. Innov. Res. **15**(1), 39–45 (2014)
5. Gaudiello, I., Zibetti, E.: La robotique éducationnelle: état des lieux et perspectives. Psychol. Fr. **58**(1), 17–40 (2013)
6. Papert, S.: Teaching children to be mathematicians versus teaching about mathematics. Int. J. Math. Educ. Sci. Technol. **3**(3), 249–262 (1972)
7. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012)
8. Romero, M., Kamga, R.: Usages de la robotique pédagogique en éducation primaire selon son intégration disciplinaire et le développement des compétences du 21e siècle, presented at the Intelligences numériques/Digital Intelligence 2016, Québec, QC (2016)
9. Baltes, J., Anderson, J.: Introductory programming workshop for children using robotics. Int. J. Hum. Friendly Welf. Robot. Syst. **6**(2), 17–26 (2005)
10. Romero, M., Loufane: Vibot the robot. Publications du Québec, Québec (2016)
11. Romero, M., Laferrière, T.: Usages pédagogiques des TIC: de la consommation à la cocréation participative. Vitrine Technologie Éducation, 04 December 2015
12. Romero, M., Dupont, Y.: Educational robotics: from procedural learning to co-creative project oriented challenges with LEGO WeDo, pp. 6159–6163 (2016)
13. Pernin, J.-P., Lejeune, A.: Dispositifs d'apprentissage instrumentés par les technologies: vers une ingénierie centrée sur les scénarios, in Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie, Université de Technologie de Compiègne, pp. 407–414 (2004)
14. Komis, V., Tzavara, A., Karsenti, T., Collin, S., Simard, S.: Educational scenarios with ICT: an operational design and implementation framework. In: Proceedings of Society for Information Technology and Teacher Education International Conference, 2013, pp. 3244–3251 (2013)
15. Graham, C.R.: Theoretical considerations for understanding technological pedagogical content knowledge (TPACK). Comput. Educ. **57**(3), 1953–1960 (2011)
16. Misirli, A., Komis, V.: Robotics and programming concepts in early childhood education: a conceptual framework for designing educational scenarios. In: Research on e-Learning and ICT in Education, pp. 99–118. Springer, Heidelberg (2014)
17. Hesse, F., Care, E., Buder, J., Sassenberg, K., Griffin, P.: A framework for teachable collaborative problem solving skills. In: Assessment and Teaching of 21st Century Skills, pp. 37–56. Springer, Heidelberg (2015)
18. Jonassen, D., Hung, W.: Problem solving. In: Seel, N. (ed.) Encyclopedia of the Sciences of Learning, 2nd edn. Springer, US (2012)
19. Kirschner, P.A., Erkens, G.: Toward a framework for CSCL research. Educ. Psychol. **48**(1), 1–8 (2013)

20. Romero, M., Hyvönen, P., Barberà, E.: Creativity in collaborative learning across the life span. Creat. Educ. **3**(4), 422 (2012)
21. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
22. Meichenbaum, D., Burland, S., Gruson, L., Cameron, R.: Metacognitive assessment. In: Growth Reflection in Children, pp. 3–30 (1985)
23. Vygotsky, L.S.: Mind and Society: The Development of Higher Mental Processes. Harvard University Press, Cambridge (1978)
24. Jerome, L., Algarra, B.: Debating debating: a reflection on the place of debate within secondary schools. Curric. J. **16**(4), 493–508 (2005)
25. Komis, V., Misirli, A.: L'usage des jouets programmables à l'école maternelle: concevoir et utiliser des scenarios éducatifs de robotique pédagogique. In: Scholé, pp. 143–154 (2012)

Assessment of Lower Secondary School Pupils' Work at Educational Robotics Classes

Michaela Veselovská(✉) and Karolína Mayerová

Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Bratislava, Slovakia
{veselovska,mayerova}@fmph.uniba.sk

Abstract. This paper presents our proposal for assessment of lower secondary pupils working with educational robotics LEGO WeDo, using the curriculum we have designed. The curriculum consisted of eleven activities with complex methodical materials that the teachers can incorporate into their ordinary classes. Teachers can use this curriculum within compulsory subject Informatics and therefore it was necessary to create a way to assess pupils' work. We have created evaluation through rubrics, because working with robotic kits encompasses several aspects. The assessment focuses on three main aspects: (a) the construction of the robotic model, (b) program to control it and (c) the presentation of mentioned program. In this paper we also describe the concrete example of assessment of pupils' work using our rubrics.

Keywords: Assessment · Rubrics · Educational robotics · Lower secondary school

1 Introduction

Teaching and learning with educational robotics includes several novel aspects. Assessment is a significant element of robotic-based learning and can add great deal to the whole experience [1]. Results of numbers of publications indicate positive development of knowledge and skills with the use of educational robotics. Pupils acquired new knowledge during classes in programming, geography and robotics [2–4], during learning fundamental principles of evolution [5] and during science classes [6]. Among acquired skills were the science process skills [6], social interaction skills [7] and problem solving skills [4]. Results of many researches, who have focused on educational robotics in concrete school subjects, confirm the importance of the teacher during education with robotic kits. Role of the teacher is crucial for example for solving problems with selected technology. But assessing pupils' work with robotic kits is still unfamiliar for many teachers. They are some cases where teachers used peer assessment [8] with educational robotics. Another teacher describes on his blog[1], how he used rubrics for assessing different jobs in the teamwork. But there are not many publications that discuss assessing educational robotics in schools.

[1] www.legoengineering.com/assessment-tools-for-group-projects/.

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_13

Assessing consists of different strategies, which are occasionally used in regular compulsory education [1, 9]. During the integration of educational robotics into ordinary education teachers should consider which specific educational goals will they assess. They could assess time management, team work, pupils' journals (that contain the design of a robotic model, progress in its construction and the program to control it), robotic model built from selected robotic kit and others. However this assessment should depend on teachers' educational goals.

2 Selected Methods

Creation of assessment for our curriculum was a part of our dissertation research, which we conducted in 5th and 6th grade at a lower secondary school (with pupils aged between 10 and 12) in small town near the capital city in Slovakia. We selected *Design based research* [10] as the main research strategy and we used qualitative methods of data collection and data analysis including observation, focus groups, audio-visual materials (pictures, photographs of pupils' products and recorded videos of pupils' work). We conducted semi-structured interviews with teacher, who taught her pupils following our curriculum in her ordinary classes within compulsory school subject Informatics. Two researches were present in the classroom collecting data. During the third iteration of our design based research we created rubrics for assessment of pupils' work on robotic models that pupils created according their imagination. We tested these rubrics in fourth (last) iteration of our research. In the third iteration we worked with 2 groups of pupils separately (10 pupils = 7 boys and 3 girls; 12 pupils = 7 boys and 5 girls). In the fourth (and last) iteration we worked with 1 group of 13 pupils = 4 boys and 9 girls. During the implementation of our curriculum pupils worked mostly in pairs of two boys or two girls (rarely a pair consisted of a boy and a girl). During the third iteration of our research we used our rubrics for assessment of pupils' robotic models in seventh and tenth activity of our curriculum (we briefly describe this curriculum in the next chapter). During the fourth iteration of our research the rubrics were used to assess the fourth, seventh and tenth activity of our curriculum. We revised our rubrics according results from data analysis of these activities. When we were creating our rubrics, we were guided by the rubrics developed by Mark Gura [1].

3 Curriculum for Educational Robotics with LEGO WeDo

We developed eleven activities with complex methodical materials, which teachers can incorporate into their ordinary classes and provide for the pupils an attractive introduction to educational programming. Each activity is intended for one class session (45 min) and is presented as a methodical material for teacher. In some cases this material includes a worksheet for pupils, possibly instructions for building robotic model and other materials.

1. **What is robot?** – Familiarization with the concept of a robot - with discussion, creation of mind map and specification of its definition.

2. **Create a dream car** – Familiarization with the content of a robotic kit and its programming environment. Building a robotic model according to the instructions and its simple moving.
3. **Even cottage on robotic foot can dance** – Building a robotic model according to the instructions and its programming according to the tasks in worksheet (command wait and repeat).
4. **My own first helper** – Creating a robotic model and a program to control it (command repeat).
5. **Matias' hovercraft from the future** – Familiarization with the motion sensor. Building a simple robotic model according to the instructions and its programming with the tasks in worksheet (commands for control motion sensor).
6. **Martinas' vision of lost vehicle** – Building a part of robotic model according a photograph, completing it using one's own fantasy and controlling it with the tasks in worksheet (exploring differences between selected commands).
7. **The best helper is in the world** – Creating a robotic model and a program to control it according to the requirements (using motion sensor and a motor while building robotic model and using selected commands for programming it).
8. **The move of a modern turtle** – Familiarization with the tilt sensor. Building a simple robotic model according to the instructions and its programming with the tasks in the worksheet (commands for control tilt sensor).
9. **Designing fairy tale** – Creating a sketch of a robotic model and describing its future behaviour in writing, using a worksheet.
10. **Creating a fairy tale** – Building and programming a robotic model according the design from the previous lesson and its introduction to the classmates.
11. **Fairy tale for everyone** – Creating presentation about a robotic model (together with its ups and downs) and delivering it to one's classmates.

Our curriculum used several methods of assessment including peer assessment, norm-referenced assessment, rubrics for assessment pupils' own robotic model, rubrics for assessing worksheet and assessment of pupils' final project. We briefly described these assessments in [11]. In this article we focus on rubrics for assessment of the pupils' robotic model created according to their imagination.

4 Rubrics for Assessment Pupils' Own Robotic Model

Rubrics are tables that contain a list of gradation criteria that provide a base for consistent and objective evaluation. Rubrics in robotics can include several aspects of creation of robotic model and also the process that the pupils used to create it. During forming these rubrics it is important to choose criteria, which are useful and practical for us as teachers [1].

We can create rubrics by listing everything that we require the pupils to do and learn while working on a robotic model. Each of these items has a function of rating criterion and each criterion includes several levels of success [1]:

1. It satisfies the selected criterion very little or not at all.

2. It partially satisfies the selected criterion.
3. It thoroughly satisfies the selected criterion.
4. It exceptionally satisfies the selected criterion.

The teacher then evaluates pupils' performance for each criterion, and assigns the appropriate number of points. Subsequently teacher adds the points for every criteria and based on the school assessment system pupils receive a grade [1].

We created our rubrics based on the observations of pupils' mistakes made most frequently during working on their robotic models. We have described these mistakes in [11–13]. Based on pupils' mistakes we have developed requirements for the content of the construction of a robotic model, its' program and its' description:

- We repeatedly reminded pupils, that their robotic model have to include motor or at least one of two sensors so it can be controlled with software.
- Pupils often did not include sensor in their robotic model, but they wanted to use the sensor parameter in their program.
- Pupils included one sensor in their robotic model, but they did not use it in the program that control led the model.
- Pupils described their imagination about how the robot should work, but not their real program [12].

After we applied mandatory requirements, we observed that pupils satisfied them with different levels of success. This was the first step during creation of the scale for our rubrics. In the second step we revaluated after each iteration of our research those rubrics that were used in the assessment of pupils' work.

These rubrics were primarily formed to assess pupils work on robotic models, that they created only according their imagination, see Table 1. With the created rubrics we assessed pupils' knowledge from three angles: building a robotic model, programming it and presenting the program that controls it. Therefore our rubrics contain three main parts:

- construction of a robotic model – construction,
- program to control a robotic model – program,
- presenting the program to control robotic model – presentation.

Pupils could receive 0–4 points for each part and adding the points for these parts generates the final grade. We also incorporate two bonus points for active cooperation and for collaboration in a group, for exceptional or extraordinary construction of a robotic model or a program to control it.

When we use our rubrics to assess pupils' robotic models, it is necessary to assess every part of these rubrics in comparison with the other parts. For example: when we assess description of program (in our rubrics it is a part of Presentation), we observe, whether pupils correctly describe their program, whether they can control robotic model with it, and also whether this model is reacting to their program. If pupils' robotic model includes a sensor, we assess whether it was pointing the right way (it means for example: motion sensor may be able to "see" something before it, so its' field of vision must not be blocked by other bricks from the robotic kit), whether pupils can work with the

Table 1. Rubrics for assessment pupils' own robotic models

	4	3	2	1	0
Construction	**It includes all** required components and it **is stable** (it does not fall apart when the program is running).	**It includes all** required components, but it **falls apart** when the program is running.	**It does not include all** required components, but it **is stable** (it does not fall apart when the program is running).	**It does not include all** required components and it **falls apart** when the program is running.	It does not include any required components.
Program	**It includes all** required commands, which **are connected in required** way.	**It does not include all** required commands, but they **are connected in required** way.	**It includes all** required commands, but they **are not connected in required** way.	**It does not include all** required commands and they **are not connected in required** way.	It includes very little or none required commands.
Presentation	It includes **precise correct description of all** commands in used program.	It includes **precise correct description of only some** commands used in program.	It includes **partially correct description of all** commands used in program.	It includes **partially correct description of only some** commands used in program.	It does not include correct description of any commands used in program.

selected sensor and also whether they can control with their program the robotic model that uses the sensor.

We used our rubrics during the seventh activity, where pupils were building robotic models according their imagination yet within the selected theme. They were expected to use the motor and a motion sensor in construction of their models (see Fig. 1). In their program they were supposed to use commands such as *count loop*, *motor power, motor on for some time*, *wait for some time* and also the parameter of the motion sensor.

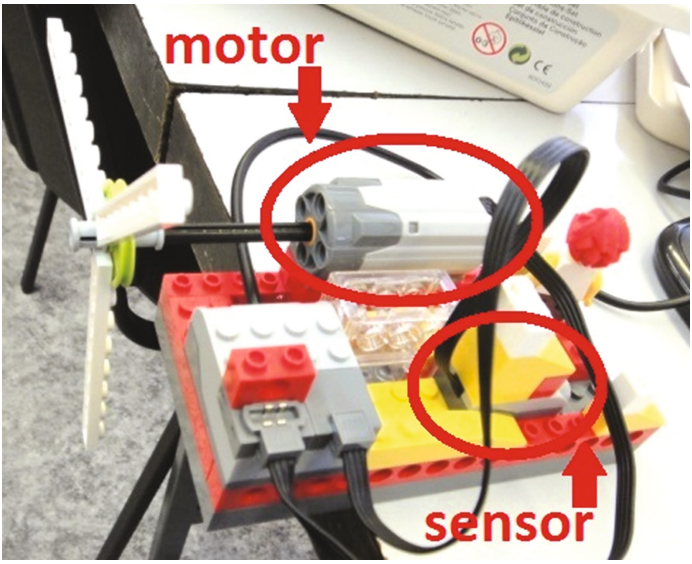


Fig. 1. Robotic model of a ship

In the following example we explain, how we proceeded during assessing a selected robotic model of a ship created by a team of two boys in the seventh activity during the third iteration of our research (identified as the pair number four in the second group of the third iteration).

Verbal description of the program to control the robotic model: *This is a ship, which transfers people from one river side to another. So I show you. Passenger goes into the ship and also in front of the sensor. Then the ship goes for 50 s. Then it stops in order for the passengers to photograph surroundings. Then it goes and stops.*

Construction of this model contained a motor, a motion sensor, and it was stable (it did not fall apart when the program was running), see Fig. 1. Therefore we assessed the construction of this model with 4 points, see Table 1 line called Construction. However program that controlled the robotic model did not contain all required commands, because it did not contain the command *count loop*, see Fig. 2. When we wanted to decide, whether all commands in program were connected in the required way, we had to hear its verbal description from pupils. But in that moment we knew that we were going to assess this program with 3 or 1 points, see Table 1 line called Program. After pupils' verbal description we found out, that commands in pupils' program were connected in the required way. This means that sequence of described commands corresponded to the sequence of commands in the assessed program. But we decided that pupils did not describe all commands in the program. For example they did not mention the command *motor power* and for how long the motor was waiting – the command *wait for some time*. You can see the description of the program. Therefore we assessed this program with 3 points. However the verbal description of the program contained a partially correct description of only some of the commands used in program, so we assessed it with 1 point, see Table 1 line called Presentation. Total number of point which pupils received was 8 and it was 50% of all points. Also we did not give them any of bonus points, because the construction was really simple, program contained redundant commands and we had to remind these pupils to cooperate with each other.

Fig. 2. Program to control the robotic model of a ship

In Table 2. we present the results of assessment of pupils' work in the third and fourth iteration. For each team we show the final number of points for selected activities from our curriculum.

Table 2. Representation of points that teams gained during the third and fourth iterations in the activities where we applied our rubrics

Number of points within rubrics							
	team	**1**	**2**	**3**	**4**	**5**	**6**
3. iteration	**1. group**						
	7. activity	12	4	12	0	12	7
	2.group						
	7. activity	12	10	12	8	10	-
	10. activity	7	9	10	12	7	-
4. iteration	**1.group**						
	4. activity	12	5	11	7	11	12
	7. activity	10	10	10	10	12	6
	10. activity	12	6	7	7	6	10

We also present examples of the results (the robotic model and its program) from the fourth iteration for teams 1, 3 and 6 during the fourth activity (Models – Fig. 3 and programs – Fig. 4); from teams 1, 2, and 5 during seventh activity (Models – Fig. 5 and programs – Fig. 6); from team 6 during tenth activity (Fig. 7).

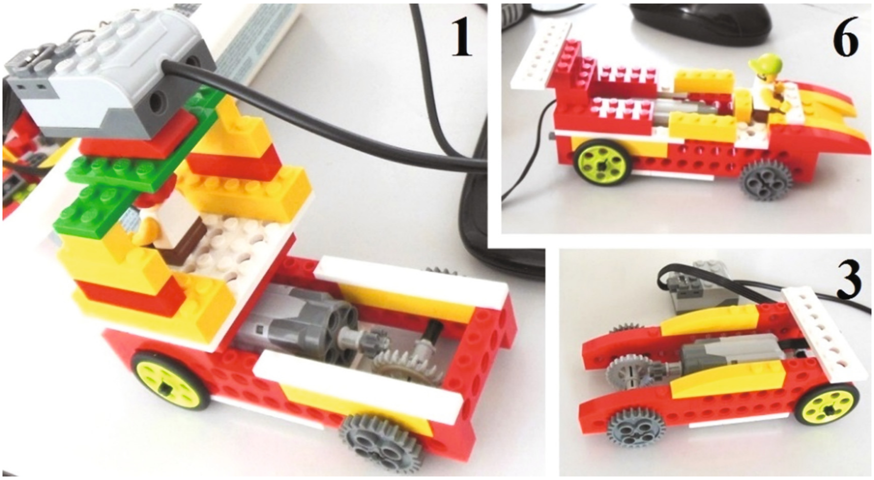


Fig. 3. Robotic models of teams: 1, 3 and 6 from fourth iteration and fourth activity

Fig. 4. Programs for robotic models (from Fig. 3) of teams: 1, 3 and 6 from fourth iteration and fourth activity

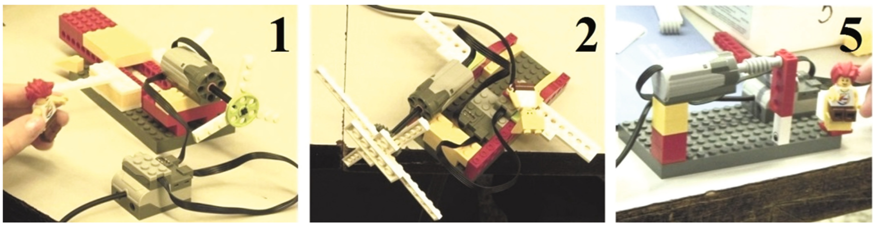


Fig. 5. Robotic models of teams: 1, 2 and 5 from fourth iteration and seventh activity

Fig. 6. Programs for robotic models (from Fig. 5.) of teams: 1, 2 and 5 from fourth iteration and seventh activity

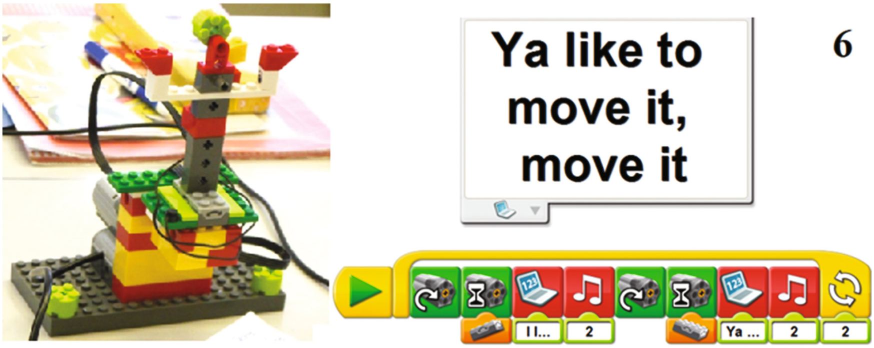


Fig. 7. Example of robotic model and program of team 6. It is a figure Julien from Fairy-tale Madagaskar.

5 Conclusion

Educational robotics can help in developing several learning goals in many school subjects. But still it is not common in compulsory education. Inseparable part of the integration of educational robotic into education is the assessment of pupils' work. In this article we have described rubrics for assessment of pupils' robotic models, which they constructed and programmed according to their imagination but within the selected theme. We created these rubrics by observing the mistakes that pupils made most often during creating robotic models and the programs to control them in most of our activities. We hope that these rubrics can be used as inspiration for assessment of pupils' work in other robotic classes.

Acknowledgments. Implementation of this article has been financially supported by grant UK/403/2016.

References

1. Gura, M.: Getting started with LEGO robotics: a guide for K-12 educators. International Society for Technology in Education (2011)
2. Barker, B.S., Ansorge, J.: Robotics as means to increase achievement scores in an informal learning environment. J. Res. Technol. Educ. **39**, 229–243 (2007)
3. Nugent, G., et al.: The effect of 4-H robotics and geospatial technologies on science, technology, engineering, and mathematics learning and attitudes. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 44–452. AACE, Vienna (2008)
4. Nugent, G., et al.: The use of digital manipulatives in k-12: robotics, GPS/GIS and programming. In: FIE 2009 Proceedings of the 39th IEEE International Conference on Frontiers in Education Conference, pp. 1–6 (2009)
5. Whittier, L.E., Robinson, M.: Teaching evolution to non-English proficient students by using lego robotics. Am. Sec. Educ. **35**, 19–28 (2007)
6. Sullivan, F.R.: Robotics and science literacy: thinking skills, science process skills and systems understanding. J. Res. Sci. Teach. **45**, 373–394 (2008)
7. Mitnik, R., Nussabaum, M., Soto, A.: An autonomous educational mobile robot mediator. Auton. Robots **25**, 367–382 (2008)
8. Catlin, D.: Using Peer assessment with educational robots. In: Cao, Y., Väljataga, T., Tang, J., Leung, H., Laanpere, M. (eds.) ICWL 2014. LNCS, vol. 8699, pp. 57–65. Springer, Heidelberg (2014). doi:10.1007/978-3-319-13296-9_6
9. Mayerová, K.: Edukačná robotika v informatickej výchove na 1. stupni ZŠ. Bratislava: Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave, Bratislava (2013)
10. Creswell, J.: Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research. Pearson Education, New Jersey (2008)
11. Veselovská, M., Mayerová, K.: Assessing robotics learning at lower secondary school. In: Information and Communication Technology in Education 2015, Ostrava, pp. 240–247 (2015)
12. Veselovská, M., Mayerová, K.: Programming with motion sensor using LEGO WeDo at Lower secondary school. In: International Journal of Information and Communication Technologies in Education, vol. 4, pp. 40–52. Pedagogical Faculty, University of Ostrava, Ostrava (2015)
13. Veselovská, M., Mayerová, K.: Pilot study: Educational robotics at lower secondary school. In: Constructionism and Creativity 2014, pp. 418–424. Austrian Computer Society, Vienna (2014)

Educational Robotics and Programming

The Use of Robotics in Introductory Programming for Elementary Students

Lito Athanasiou, Paraskevi Topali(✉), and Tassos A. Mikropoulos

Department of Primary Education, University of Ioannina, Ioannina, Greece
Litoulath@yahoo.gr, paraskevi.topali@gmail.com, amikrop@uoi.gr

Abstract. Studies have shown that teaching programming to students is often a difficult task. The programming language itself is not as much a challenge as the concepts and structures, which define it. This paper explores the use of educational robotics to introduce basic programming concepts through meaningful teaching and learning activities with the hands-on use of bee-bots. The results from an empirical study show that students can successfully develop algorithmic thinking and programming skills based on their knowledge acquired by the bee-bots. Thanks to the tactile interaction with robots, students developed their creativity and imagination as built systems with Lego WeDo, all the while enjoying the course through teamwork activities.

Keywords: Educational robotics · Programming · Programming structures

1 Introduction

Today's learning environments focus on developing problem solving skills, critical thinking and creativity [1]. STEM, the acronym that refers to the academic disciplines of Science, Technology, Engineering and Mathematics, when blended with multi-disciplinary approaches, paves the way to engineering education and the cultivation of computational thinking. Hence, educational robotics is considered an effective learning tool because computational thinking and engineering skills are embedded in a single framework [2]. In fact, in recent years, educational robotics activities have been gaining more and more ground as the field of robotics provides opportunities for students to explore how technology works in real life, as well as develop their critical thinking and construction skills. Hence, robotics is introduced in educational settings through a multidisciplinary approach. It is used as a cognitive tool, as an instructional technique, or even as a subject area in itself [3]. As a whole, given that students learn best through their personal involvement in problem solving and creation, robotics provides the appropriate learning framework.

1.1 Robotics in Schools

The field of robotics primarily provides learning incentives because it brings together many areas of the curriculum [4]. Learning is fun and students engage in activities that sharpen their thinking (through observation, design, calculation, measurement and the

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_14

chance to test their projects in a real life context), and imagination, while equally developing their social skills and team spirit [5]. The application of educational robotics has yielded significant positive results in the development of technology literacy and the ability to problem solve [6]. Sullivan describes improved observational skills, the capacity to make appropriate calculations, creativity building and the intuitive assessment of hypotheses and variables [5]. More so, positive results are observed in the social skills of children and the development of team spirit and mutual collaboration [7–9].

However, it seems that educational robotics is a field not often utilized as a cognitive tool in specific subjects, especially at younger ages. Barak and Zadok note that the benefits of educational robotics are relevant to concepts in Physics and Technology, as they are strongly associated with problem solving [10]. Mikropoulos and Bellou have also proposed educational robotics as mind tools in Physics and Computer Science education through meaningful learning activities [11].

1.2 The Benefits of Robotics in Learning Programming

Programming cultivates and develops cognitive skills and is the basis for the development of strategic thinking and finding solutions to problems [12]. In short, programming builds skills that can easily be applied to other disciplines and fields [13] such as the ability of communicate one's ideas and thoughts in a relevant context that is extremely important.

Often enough, students face difficulties in understanding programming language and its structures [14]. Many students find it challenging to create and complete rational computational programming. In particular, students report difficulties in understanding the concepts of programming and algorithms, in addition to solving simple problems and hesitate to apply leadership skills such as taking the initiative and risk to make confident decisions [15, 16].

There are many advantages to introducing basic programming terms especially to young students via robotics. The main benefit however is the direct feedback students receive when writing code, which is an important factor in meeting the student's desire for success and creating motivation for learning [17, 18]. In fact, robotics and programming, when combined, provide an explanation to abstract causes rendering these immediately observable phenomena. Meanwhile, through the execution of programming, results are reflected in a robot, a physical and tangible system [5]. Students are equally involved in debugging procedures and must control variables, make hypothetical predictions and assess these scientifically. In addition, students are engaged with a physical model that shows observable physical behavior (sound, movement), and performs meaningful activities [19], which is significantly more attractive to students in comparison to focusing only on computer screens [17]. Nuget and colleagues in a study conducted on elementary students that aimed to foster problem-solving skills with robots, found positive results in the understanding of programming concepts [8].

This brief literature review reveals that despite the number of studies conducted there is still a lack of empirical data concerning the use of robotics in introductory programming especially for elementary school students.

2 Methodology

2.1 Objective and Research Questions

The aim of the present study is to investigate the contribution of robotics in the introduction of basic programming concepts. Specifically, the empirical study investigated (a) the potential of simple educational robots in teaching basic programming structures and (b) the learning outcomes when presenting students with a visual programming environment through play and interactive games.

2.2 Sample

The empirical study was conducted in the context of a one-week summer school. The duration of each day's course was seven academic hours (7 × 45 min). The attendees were seven 9–12 year old elementary school children. None of the students had any prior experience in programming. The research can be considered as a case study, because of its small sample, the intensive courses, and the in depth study.

2.3 Procedure

Within the setting of the one-week summer school, we set out to design an introduction to basic programming course. Robotics activities were conceived which aimed to facilitate the understanding of algorithmic structures, constants and variables, as well as iteration and selection structures. The activities were taught with the use of bee-bots (https://www.bee-bot.us/), and learning was assessed with Lego WeDo (https://education.lego.com/en-gb/lesi/elementary/lego-education-wedo).

The introduction to programming course started from the first day with the use of bee-bots as a general teaching aid. The initial objective was the development and cultivation of algorithmic thinking through handling basic commands (forward, backward, right, left) and their serial organization in performing a specific action. Students within the first two days were divided into two or three groups, depending on the activity with the bee-bots. Once the students became familiar with the bee-bot operation, they were asked to perform the following play activities:

1. Sequential structure: the robot must correctly follow the route outlined on the floor (Fig. 1 left).
2. Car racing: Students are divided into two groups and each group, with its robot, stands in front of two identical routes. Each group has to correctly calculate the robot's required movements in order to complete the route, in other words provide the bee-bot with the proper sequence of commands in order to execute the route correctly and quickly (Fig. 1 right).
3. Dancing: This collaborative activity included three synchronized bee-bots. Each one of the three groups had to program their robot to run a specific route in synchrony

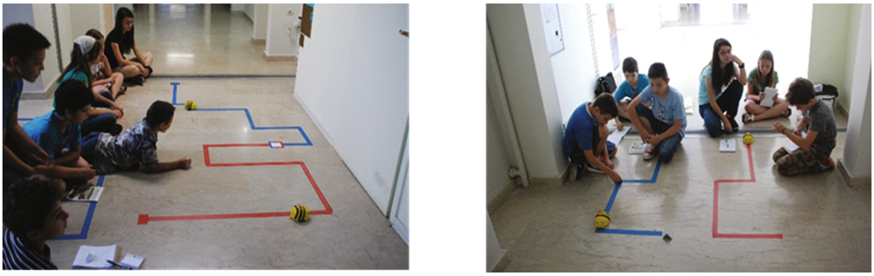


Fig. 1. Learning activities with the bee-bots (left: sequential structure, right: car racing).

and similar orientation as the other two robots. The objective was to create a choreography where the final result would be three bee-bots placed in similar proximity yet facing one another (Fig. 2).

Fig. 2. Collaborative learning activity (dancing).

Over the following three days, the students worked with the Scratch programming environment. The process was organized into the following phases:

1. Phase 1: presentation of the environment and familiarization with Scratch. The students explore its possibilities and perform guided activities.
2. Phase 2: observation and experimentation with existing games. Studying program codes and trying to understand the reasoning behind programming. The students create small codes and are introduced to the concept of repetition structures. The students are required to modify their programs using repetition structures to optimize their code.
3. Phase 3: discussion on the different command structures provided by Scratch, how they work and their value in coding. Variables are presented versus constants and their usefulness in program coding is discussed.
4. Phase 4: introduction to the concept of selection structure. Students discover how to choose and change the direction of the software code. Each choice leads to a different

set of commands that needs proper planning. This gives the students opportunities for experimentation and the possibility to create small applications and simple games.

5. Phase 5: debugging activities. The students identify and resolve errors that exist in codes.

The researchers provided scaffolding in the form of code templates of similar problems, when needed.

On the last day, once students became comfortable with the use of Scratch, they attempted to connect it with the Lego Education WeDo Construction Set. Initially, students were presented with readymade Lego WeDo Construction Sets to induce their interest. Then, they were split into two groups and were asked to use Lego blocks that had already been combined with respective sensors to design various robot-animals. The first group was provided with the precise steps they had to follow to build the robot-animals, while the second group was given a picture with the desired final structure and these students would have to find the right way to connect the pieces. Once both groups had finished, the robot-animals were programmed in Scratch in order for the concepts taught in the previous days to be assessed.

Students were evaluated based on their participation in bee-bot activities and the projects they created in the Scratch programming environment. A questionnaire was designed to investigate their attitudes towards the project in which they participated and their overall behavior was observed throughout the five learning phases.

3 Results

Almost all the students performing the bee-bot activities and those in Scratch, created complete programs and everyone used correct algorithmic structures. The students gradually understood the new concepts and worked to make the necessary changes. The study of the children's projects shows that they were able to solve a problem in more than one ways. Regarding teamwork activities students collaborated successfully, assisting and complementing each other.

The basic programming learning objectives set out by the bee-bot activities were achieved. Robotics gave students tangible feedback on the effectiveness of the programs they created. Any errors and deviations from the expected results lead to creative thinking about modifications and reprogramming. Students were able to resolve the challenges presented to them in an average of three efforts. Even the bee-bot dance activity which was the most demanding as it combined spatial and temporal synchronization of the robots, was completed correctly on the students' third attempt.

The results were similar for the activities in the Scratch environment. Table 1 shows the levels of algorithmic thinking of the students. These five levels follow the "Hierarchical Assessment of Programming" model that is based on the SOLO (Structure of the Observed Learning Outcomes) taxonomy and takes into account the declarative, structural, procedural and strategic types of knowledge. The five hierarchical levels are determined both by the development of algorithmic thinking and programming skills [20].

Table 1. HAP model to assess qualitatively the algorithmic thinking of students.

Students	1	2	3	4	5	6	7
Number of tasks	3	4	3	4	3	6	3
Level 1: Partial understanding of task				✓			
Level 2: Understanding of task but algorithm remains incomplete							
Level 3: Solutions sought in a general context without consideration of specific conditions	✓	✓			✓		
Level 4: The solution's organization and presentation approach is fully structured and satisfactory			✓				✓
Level 5: The optimal algorithm is selected and implemented						✓	

Figures 3 and 4 show samples of the students' projects. Student 1 (Fig. 3 left) seems to understand the task, but it finds it difficult to establish a correct and complete solution to the problem. The student is able to use the commands, but falls back on the correct manipulation of the code. The code could be improved significantly. Student 2 in his "star fight" project (Fig. 3 middle) performs successfully all the programming steps. He uses the commands correctly, but does not use the correct algorithm. Student 3 in his packman game (Fig. 3 right) uses the correct programming structures. The student seems to understand complex commands such as "report" and "when I receive" and when challenged, he looks for an alternative solution. There is room for improvement in order to avoid the use of unnecessary commands.

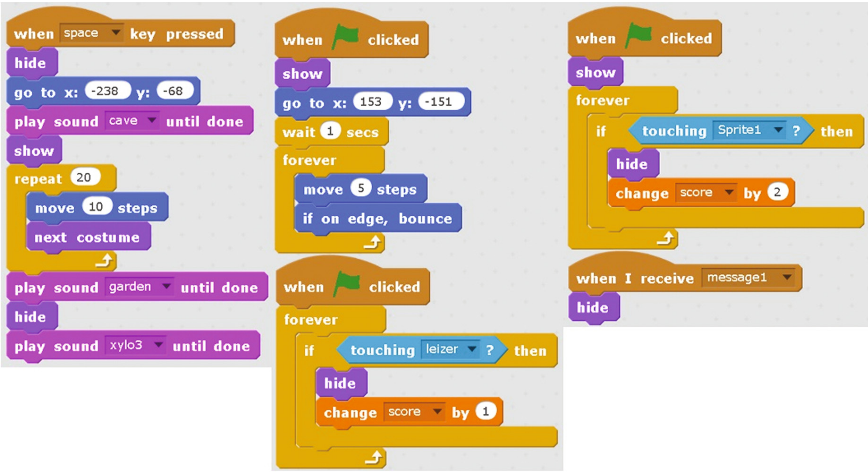


Fig. 3. Three code samples.

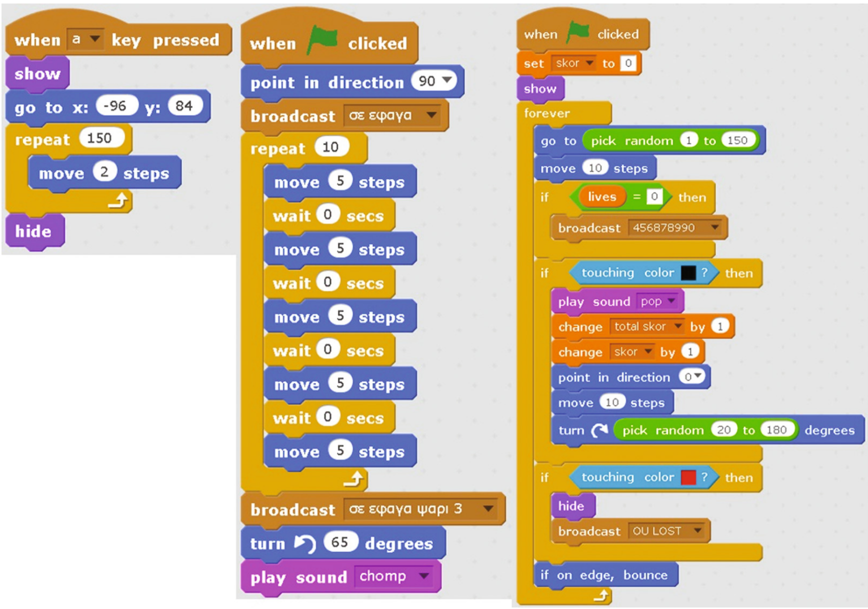


Fig. 4. Three more code samples.

Student 4 (Fig. 4 left) seems not to understand the project as a whole and hence he does not present a complete picture in her code. At a very early stage, there is the correct use of the commands, however, the result shows a weakness to combine commands in the proper way. Student 5 in his underwater world (Fig. 4 middle) understands the initialization and selection commands and he uses them to complete successfully the task. However, the main challenge is in fully understanding the use of the repeat structure in order to avoid the use of unnecessary command combinations. Student 6 in his table tennis project (Fig. 4 right) uses the appropriate commands in the motions, looks, sensors and control categories as well as in complex commands for the creation of the best possible algorithm. Student 7 utilized the motion commands, selected key look commands and initialized the figures. He understood each task and designed correctly a structured solution to the entire problem.

In general, considering the results in the Scratch environment, each student applied successfully their knowledge, analyzed the problem, tried to find the appropriate programming structure for their script. The students reached up to the sixth level of the revised Bloom cognitive taxonomy, since they designed and developed their applications by themselves. The teacher only used the scaffolding technique in order for each child to reach its own zone of proximal development. Finally, it should be noted, that one student exceeded our expectations, by actively using the camera to create his own game.

All students found the Lego WeDo activity very interesting while they enjoyed to a lesser extent the bee-bots activity. This can be interpreted because of the fewer construction and programming features bee-bots have in comparison to Lego WeDo. Unexpected results nonetheless were brought to light at the end of the course. Five out of six students to a high degree felt familiar with the use of bee-bots while only half-expressed confidence in working with Lego WeDo. Overall, all the students expressed positive feedback regarding their participation in the program, its content and assignments.

4 Discussion

The aim of this case study was to investigate the contribution of robotics in introductory programming for elementary school students.

Students, with the successful completion of the bee-bot activities were able to link abstract code with real knowledge. They used reflective processes and saw immediate results. The immediate feedback they received from the robots they built enabled them to make corrections and optimize their coding. Students composed a code to create a specific structure and set out rules to express an algorithm. Moreover, their creative thinking and imagination were cultivated, mainly through the final tasks with the Lego WeDo package. The students seem to understand the programming concepts and independently explored the new knowledge step by step through personal participation while assessing every decision before implementing it. Students successfully created their own applications in the Scratch environment.

The findings of this study are consistent with the results of other studies which demonstrate that the use of educational robotics makes teaching programming code more effective. It emerged that working with a physical model that presents observable physical behavior is more attractive to students than to solve problems in common programming environments [19]. It also appeared that elementary school children actively involved in robotics education approached the technology analytically and cultivated problem-solving skills and algorithmic thinking [8].

The results have been encouraging. The field deserves further investigation. The limitations of this study include the small number of participants and the restricted time duration of the intervention.

References

1. Moyer, L.A.: Engaging students in 21st century skills through non-formal learning. Ph.D. thesis, Virginia Polytechnic Institute (2016)
2. Eguchi, A.: Robotics as a learning tool for educational transformation. In: 4th International Workshop Teaching Robotics, Teaching with Robotics and 5th International Conference Robotics in Education, Padova, Italy, pp. 27–34 (2014)
3. Alimisis, D., Kynigos, C.: Constructionism and robotics in education-the TERECOP project proposal. In: Alimisis, D. (ed.) Teacher Education on Robotics - Enhanced Constructivist Pedagogical Methods, pp. 11–26. ASPETE, Athens (2009)
4. Johnson, J.: Children, robotics, and education. Artif. Life Robot. **7**, 6–21 (2003)
5. Sullivan, F.R.: Robotics and science literacy: thinking skills, science process skills and systems understanding. J. Res. Sci. Teach. **45**(3), 373–394 (2008)
6. Anagnostakis, S., Michaelides, P.G.: Laboratory of educational robotics' - an undergraduate course for primary education teacher – students. In: 3rd International Conference on Hands-on Science, Braga, Portugal, pp. 329–335 (2006)
7. Mitnik, R., Nussbaum, M., Soto, A.: An autonomous educational mobile robot mediator. Auton. Robots **25**(4), 367–382 (2008)
8. Nuget, G., Barker, B., Grandgenett, N., Adamchuk, V.: The use of digital manipulatives in K-12: robotics, GPS/GIS and programming. In: Atman, C. (ed.) Proceedings of the 39th IEEE International Conference on Frontiers in Education Conference, pp. 302–307. IEEE Press, Piscataway (2009)
9. Owens, G., Granader, Y., Humphrey, A., Baron-Cohen, S.: LEGO® therapy and the social use of language programme: an evaluation of two social skills interventions for children with high functioning autism and asperger syndrome. J. Autism Dev. Disorders **38**(10), 1944–1957 (2008)
10. Barak, M., Zadok, Y.: Robotics projects and learning concepts in science, technology and problem solving. Int. J. Technol. Des. Educ. **19**(3), 289–307 (2009)
11. Mikropoulos, T.A., Bellou, I.: Educational robotics as mindtools. Themes Sci. Technol. Educ. **6**(1), 5–14 (2013)
12. Papert, S.: Mindstorms: Children, Computers and Powerful Ideas. Basic Books, New York (1980)
13. Pirolli, P., Recker, M.: Learning strategies and transfer in the domain of programming. Cogn. Instr. **12**, 235–275 (1994)
14. Barreto, F., Benitti, V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**, 978–988 (2012)
15. Jimoyiannis, A.: Using SOLO taxonomy to explore students' mental models of the programming variable and the assignment statement. Themes Sci. Technol. Educ. **4**(2), 53–74 (2011)

16. Vrachnos, E., Jimoyiannis, A.: Design and evaluation of a web-based dynamic algorithm visualization environment for novices. Procedia Comput. Sci. **27**, 229–239 (2014)
17. Barnes, D.J.: Teaching introductory Java through LEGO MINDSTORMS models. In: 33rd SIGCSE Technical Symposium on Computer Science Education (2002). https://www.cs.kent.ac.uk/pubs/2002/1517/content.pdf
18. Carnegie Mellon Robotics Academy: Introduction to Programming EV3 Curriculum. http://www.education.rec.ri.cmu.edu/content/lego/ev3/
19. Lawhead, P.B., Bland, C.G., Barnes, D.J.: A road map for teaching introductory programming using LEGO© mindstorms robots. In: Proceedings of ITiCSE-WGR 2002 Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education, pp. 191–201. ACM, New York (2002)
20. Bellou, I., Mikropoulos, T.A.: The hierarchical assessment of programming knowledge model. In: Proceedings of the 4th Conference on Computer Science education, pp. 111–120. HAICTE, Patras (2008)

The Combined Use of Lego Mindstorms NXT and App Inventor for Teaching Novice Programmers

Stamatios Papadakis[1(✉)] and Vasileios Orfanakis[2]

[1] Department of Preschool Education, University of Crete, Crete, Greece
stpapadakis@gmail.com
[2] 1st Vocational School of Agios Nikolaos, Crete, Greece
vorfan@gmail.com

Abstract. Both in Greece and abroad, students in school and in introductory computer science courses perceive programming as a difficult task. Introductory programming courses are often disappointing both for students and for teachers. One of the major factors to which these difficulties in learning programming has been attributed is the traditional approach to teaching the fundamentals of programming, which is unable to provide students with an interesting and richly stimulating environment through which problems and concepts are the subject of investigation in a creative and enjoyable way. In contrast, several studies claim that teachings robotics is suitable to students regardless of age and background and is a way of encouraging learning. This paper presents an alternative use of robotic Lego Mindstorms constructions and the visual programming environment App Inventor for teaching programming with the goal of understanding basic programming structures.

Keywords: Lego Mindstorms · App inventor · Robotics · Programming

1 Introduction

Several studies indicate that programming is, for most students, an unattractive activity [1, 2] and often novice programmers hold misunderstanding and misconceptions [3]. Since "programming" includes a large number of different activities and various concepts, the studies in the literature describe a large number of different problems of student's learning to program [3]. Additionally, the "traditional" teaching approach to programming is likely to deter rather than attract novice developers [4]. There has also been extensive work done to investigate alternative ways of programming, for example programming by demonstration, programming by example, programming with examples, graphical programming languages, and animated programming languages [5]). A different approach to the teaching of students and beginners programming is one that utilizes educational robotics in combination with visual programming. Teaching robotics is suitable for students regardless of age and background and is a way of encouraging learning [6]. Students view robotics more as a game than as a learning tool while the majority has already played with a robot before [7].

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_15

This paper presents a pilot scheme which aims to use robotic Lego Mindstorms sets and the App Inventor visual, "drag-and-drop" building block visual programming environment. Our aim is the creation of a mobile application for Android smart mobile devices for controlling the robot. By using toys as a spark for learning we hope to create positive incentives and motivation in students in order to lead them to easier understanding of basic programming structures.

2 Difficulties in Teaching and Learning Programming

It is generally accepted that the teaching and learning of programming is normally characterized by some difficulties, which mainly occur in the construction of an algorithm or program [8]. Various studies have shown that students have difficulties in understanding the concepts and structures of programming [9]. Other difficulties students encounter while learning to program, is a general problem of orientation; difficulty to instruct the machine about the solution of a problem; and tendency to converse with a computer as if it was a human [4].

One of the major factors which have been found to be a source of difficulties in learning to program is the fact that the classic teaching approach emphasizes on work with numbers and symbols in an abstract way ignoring students' needs for learning through concrete action [10]. According to the classical teaching approach, students are taught a general purpose language (Pascal, Basic, C, Java, etc.) [9]. This method of teaching is problematic [11], as those general-purpose programming languages include multiple commands which in combination with the strict structural details constitute a large amount of information that must be mastered by the students, forcing them often, to be more concerned with the language technical details and not focus on fundamental concepts and programming techniques. Additionally, the common exercises in programming are not of interest to students, since they involve the processing of numbers and symbols [12].

A constructive approach to learning programming requires appropriate teaching environments that on the one hand help students to solve problems, and effectively address the aforementioned misconceptions and difficulties on the other [13, 14]. [15] suggest the combined use of appropriate activities and solving selected problems in a computer lab in real programming environments. This approach emphasizes a pedagogical design for teaching novice programmers as the emphasis has shifted from teaching a strict language syntax to the development of critical and analytical thinking through problem solving.

The "drag-and-drop" visual programming environments such as App Inventor and Scratch are considered easy to learn for learners of all age groups, educational backgrounds and interests as they allow users to experiment with programming structures simply by joining pieces of code in a way similar to connecting Lego bricks [14]. According to the same researchers, this approach is ideal for beginners in programming as they are offered the opportunity to focus on the structure of solutions rather than drafting programming commands. Besides, the goal is not the teaching of programming concepts, but helping students to build the necessary conceptual frameworks in which to exercise programming activities [13].

3 Educational Robotics

Educational robotics is an educational environment, where the student is able, with the help of a simple programming language to compose and direct a technological entity [16]. It has its roots in the late sixties and early seventies when Seymour Papert developed the idea of Logo as a computer language and the Turtle as a device that children could control with their Logo programs [17]. Robotics can be integrated as a complementary activity to the subject of Information and Communications Technology (ICT) and in particular to the programming lesson. The use of robots in the introduction to programming is considered to be potentially positive, since it can lead - among others things - to the understanding of an accurate and logical command language [18], while basic programming structures, such as control and repetition, can be accessed through experimental programming of model behavior. In such a process there is no right or wrong outcome, only learning opportunities [19]. [20] report that educational robotics provides a way of approaching ICT that can arouse students' interest while bringing them in contact with important concepts of computer science. Additionally, the "playful" aspect of programmable robotic models encourages students to be more creative by approaching the programming of the robot as a recreational and enjoyable pastime, significantly enhancing their willingness to deal with programming [7]. Therefore, [6] report that the teaching of robotics is suitable for students regardless of age and background and is a way of encouraging learning. At the same time, we must also not overlook the potential for understanding and assimilating technical knowledge [18], while students can learn more about the real world by working with robotic constructions.

Lego's educational robots have been systematically used to introduce beginner students to programming [7]. Lego Mindstorms can have sensors to perceive events or factors in the environment (temperature, distance, obstacles, light intensity, etc.). They may also have a drive mechanism (motor) which sets the whole structure or a part of it in motion [21]. A typical behavior of a robotic construction is reacting to a potential stimulus [14]. Interfacing the robot with a computer can be done via a USB port or Bluetooth wireless communication. [6] report that Lego Mindstorms reinforce the main objectives of teaching programming such as documentation and discovery, learning a new symbol system, communication between machines and learning algorithms. Their advantages are that many students are familiar with Legos from an early age and they are considered an affordable solution [6]. This approach to teaching programming with Lego Mindstorms can help to eliminate the weaknesses of traditional teaching methods and create the right learning conditions in order to make teaching more effective [9]. Results of pilot studies show that utilizing Lego Mindstorms or similar sets and tools can be a very good educational tool, which under certain conditions can be valuable aids to the teacher and put the theories of knowledge construction through research, testing and disposal into practice [1]. However in Greece, unlike abroad, teaching robotics is mainly restricted to university institutions or specialized courses, while it is absent in secondary level education [22, 23].

4 The App Inventor for Android Programming Environment

Research findings show that students have increased incentives to use handheld forms technologies [24, 25] while carefully designed educational activities which make use of smart mobile devices encourage students to engage with them [26].

The App Inventor for Android (AIA) is a free web-based programming environment with building blocks [27]. The biggest advantage of AIA is that because it requires no special programming or other knowledge and because of the playful nature and the intuitive “drag-and-drop” command environment, it is quite easy to learn compared to other programming environments [28]. Applications created with AIA can be installed on any Android device. AIA provides additional incentives for students in relation to Scratch and Alice, because of the portability and practicality of the applications created with it, as students are able to see and run the applications directly on their smart mobile devices [29].

In the US, AIA has been used in primary and secondary education for the last 4 years [30, 31], and several universities have updated their curriculum by introducing AIA for the teaching of programming (http://mobile-csp.org/) [32].

5 Teaching Proposal

The teaching proposal that we present was conducted during the 2014–2015 school year in a General Lyceum (Upper Secondary School) in Heraklion, Crete, with the involvement of first-year students who attended the optional subject entitled “Computer Applications”. 24 students (11 boys and 13 girls) between 15 and 16 years old participated in the activity. All students were taught some introductory lessons in programming in the third grade of Gymnasium. During the activity, students worked a total of 5 two-hour sessions based on specially designed worksheets. The activity combined Lego Mindstorms and the AIA programming environment enabling students to see how the robot works, interact with it and check the results of test data inputs. In other words, they escape the simple transmission of knowledge to more “real” uses of programming structures. In implementing activities, we focused our attention on both teaching the principles of programming and developing construction skills. For the purpose of this teaching proposal, we used the concept of Pedagogical Content Knowledge (PCK) as proposed by [33] and reformulated by [3] (Fig. 1).

Through the activities we tried to cover both the basic programming structures and to build on the individual parts of the robot (motor and sensors). Our worksheets include the use of ready-made command blocks which students can experiment with by changing the values of various parameters and creating new command groups. Our approach focuses on problem-solving skills and developing algorithms, not on learning the programming language itself. In any case, the relationship between the physical structure and control program is easy to understand, while the ease of use of AIA is not expected

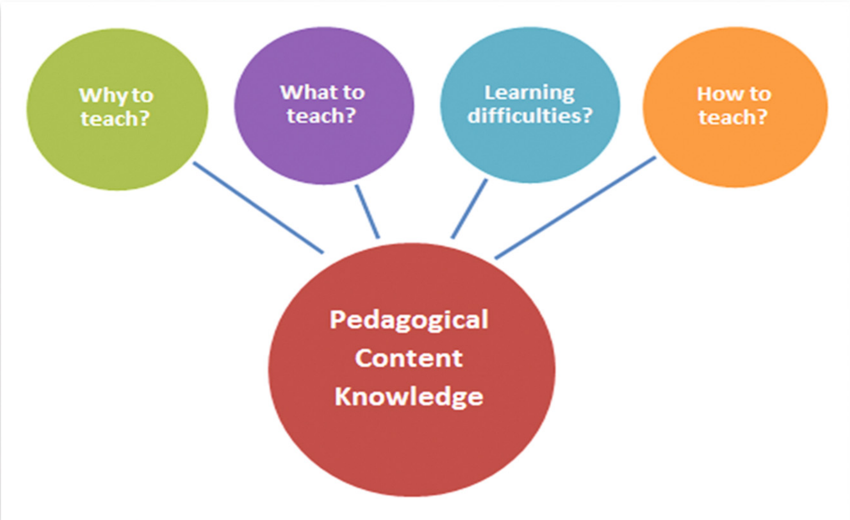


Fig. 1. Pillars of the teaching intervention

to cause cognitive overload to students [34]. The only possible obstacle is the foreign language or syntax of the development environment.

The method we used was based on group work. Students were divided into 6 groups of fours. For the group formation, we took into account that the smaller the group, the greater the potential for interaction among members [35]. In each group, we wanted to have a range of student levels, mixed by intellectual ability or achievement level. So what we did was that we handed out sheets of paper with directions/material on it and a puzzle piece attached. While appearing to be a random selection to the students, we determined which students will come together into a particular group. In each group, roles were given to each member. So in each group, we had the "moderator of the discussion", the "researcher", the "secretary" and the "presenter" [36]. As teachers we avoid the temptation to "lead" the groups. We just monitored and assisted the groups as needed. For example, we didn't answer student questions unless the group members were unable to resolve the issue by themselves [37]. We mainly moved among the groups to assure that they were actively engaged in their roles and following designated procedures.

In summary, from the teaching point of view, the students' process development included the following four distinct stages (Fig. 2).

5.1 Activities

The project-based method we followed, it was based on the idea that in programming we can distinguish two kinds of knowledge, namely the program generation and the program comprehension [3]. In the first case, the student should be coached in the process of problem solving, reflection on this process, and in the development of algorithmic ways of thinking. In the second case, the student is asked to give a demonstration of her/his understanding of how a given program works [3].

According to the curriculum, 16 teaching hours must be devoted to the teaching of programming in the First Grade. The intervention was organized in five two-hour modules, so as to be sufficient time for observation, interaction, and completion of the tasks that has been assigned to the students. The students' work consisted of two individual educational activities: (a) familiarity with hardware and software and (b) the main activity.

The first activity was implemented in the first two-hour meeting, during which the researchers briefly presented the LEGO Mindstorms NXT kit. The researchers supplied each group with a pre-manufactured vehicle, containing the basic building blocks for time economy reasons. The students modifying the robot by connecting sensors (touch, distance, light, color, sound) and output devices (lamps), experimented with its operation and programming in order to understand the usage of the training package. The second (main) activity, which was more complex, implemented during the next meetings, as the students worked with worksheets. The students were asked to develop a mobile application for smart mobile devices with Android Operating System. The purpose of the application is to control the robot in two different ways: either remotely via the Bluetooth ports of the portable device and the robot, or through voice command. Through the completion of this activity we aimed at enhancing students' problem solving skills and offering the students a subject, which includes aspects of different disciplines; use of modularity and transferability of the knowledge/skills; and the opportunity to work with a multi-disciplinary subject. Initially the students were involved in the creation of the robot's guidance application. App Inventor's programming environment was

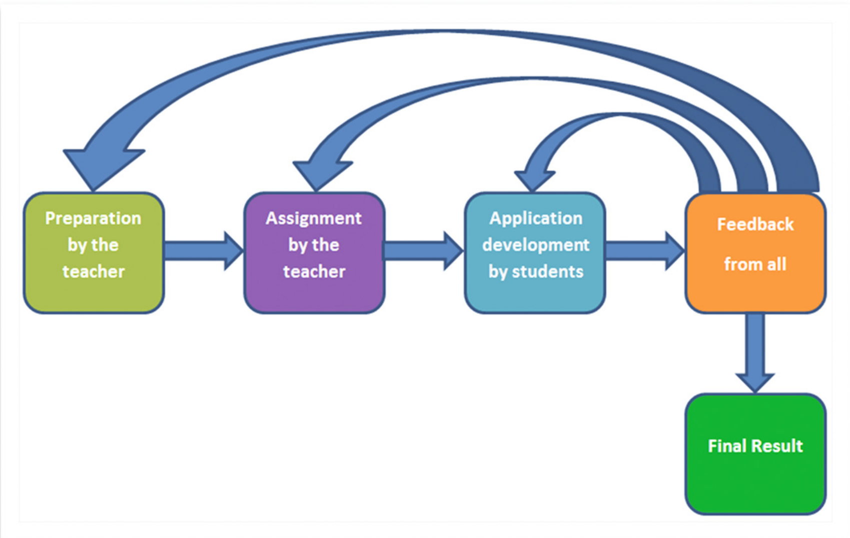


Fig. 2. Students' development process model

used for its creation. The design part of the application (3 screens, one menu and a screen for each guidance selection) is shown in Fig. 3.

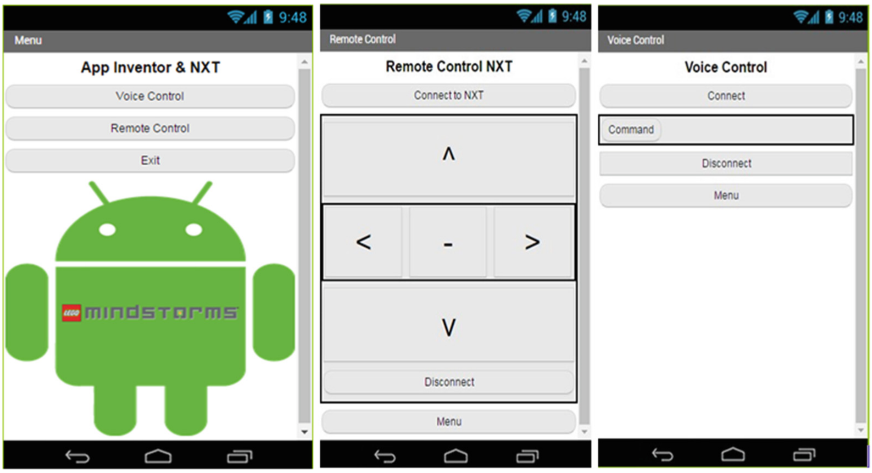


Fig. 3. App design in App Inventor

Fig. 4. Photos from students' activities inside and outside the school laboratory

The work and activity sheets related to: (a) the design of the user interface in AIA, (b) the programming of the Lego Mindstorm in AIA and (c) understanding the basic principles of robotics. Thus, in worksheet 1 students created and interconnected the three screens that would be used. Then, in worksheet 2 students were involved in the programming of objects placed on their screens. Issues relating to the Lego sensors and their use are explored in other worksheets such as worksheet 3. Finally, programming structures, such as the selection structure, are approached with introductory activities and further expanded on in corresponding worksheets.

Photos from indoor and outdoor student activities are shown in Fig. 4.

6 Evaluation of Educational Activity

For the evaluation of the activity effectiveness in enriching students' knowledge before and after the implementation of the teaching intervention data were collected through semi-structured interviews. Students were asked to provide open responses articulating their views concerning issues related to the teaching objects. Prior to the main study, a pilot study was conducted with a small number of students (who were not part of the main sample) in order to test the intelligibility of the interview scheme. Each interview lasted for 15–20 min.

First, the students were asked if they were familiar with basic programming terms such as algorithm, control flow, loop, variable, logical operation and then to express in their own words how they understood them. Questions have been raised as: How you understand the term "variable"? Can you explain in your own words how a variable works? The students' responses constitute the basic themes of the data analysis. The method applied for data analysis was a qualitative one [38]. In this way, the empirical data were condensed and grouped according to the coherent comments of each student. Subsequently, the different responses were compared and a limited number of categories were produced for each of the themes.

The analysis of children's answers concerning the basic programming terms revealed that students were confused. For example, despite the fact that some students correctly pointed out that a variable is an identifier bound to a piece of storage in the main memory of the computer, the majority of them (70%) identified incorrectly the term "variable". No statistically significant differences were revealed between the different types of students' conceptualizations in relation to age and gender.

At the end of the teaching intervention, all students were again asked to answer the same questions at the beginning of the study. The analysis of their responses showed that significant progress had been made with respect to their knowledge of the basic programming terms. The students through their answers seemed to understand how variable works, or the role of a subprogram as a piece of code that has been named and can be referred to by that name (called) as many times as is needed. Similarly, their knowledge about the data types revealed an impressive degree of improvement. It is worth mentioning that there were no significant gender differences in their responses.

The majority of students, in their assessment at the end of the activity, noted that they enjoyed the activities. The students positively commented that they obtained information in both different contexts outside the "formal" classroom environment and that the information was not only limited to textbooks and standard teaching material. In general, and despite some technical problems, all students were very enthusiastic about the activity. They also said that they hope to repeat the activity in the next school year, as all of them noted that they acquired new knowledge in a pleasant and creative way.

Just like any new piece of technology, in the case of mobile technology and robotics, there are certain drawbacks associated with this innovation [39–46]. In this activity, we observed three major problems: the high cost of robots, the extra time, space, work and competence required from teachers and schools, and the shortage of curriculum materials and guidelines. Another challenge is that this activity demands the use of a number of smart mobile devices. In the most likely scenario, according to which the school unit cannot provide students with smart mobile devices, a possible solution would be the students to use their own smart de-vices in the lab. However, there are a number of bureaucratic issues that must be addressed particularly in the Greek educational system.

7 Conclusions

In this paper, we presented a combined approach to utilizing educational robotics and visual programming through Lego Mindstorms and App Inventor for teaching and understanding of basic programming structures. Students through their involvement seem to have understood the basic concepts of programming and technology as they engaged in practical work in an interdisciplinary authentic environment. The results of the intervention produced positive outcomes but its short duration poses some questions: To what extent students feelings and attitudes would be the same if they had used this technology for a longer period? The greater familiarity with this technology could replace students' initial enthusiasm with boredom or with more enthusiasm?

Moreover, we should keep in mind that in a constructivism approach in teaching programming, the aim is not to teach programming concepts, but to educate students to build the necessary conceptual frameworks to practice programming activities [47]. Computational Thinking and Educational Robotics have a natural symbiotic relationship and can work together to offer exciting educational opportunities for K-12 Education [17].

References

1. Elefterioti, E., Katatrantou, A., Panagiotakopoulos, C.: Using Lego Mindstorms NXT for teaching programming in an interdisciplinary context: a pilot study. In: Jimoyiannis, A. (ed.) Proceeding of the 7th PanHellenic Conference with International Participation on "ICT in Education", Korinthos, Greece (2010, in Greek)
2. Kay, J.: Contextualized approaches to introductory computer science: the key to making computer science relevant or simply bait and switch? In: SIGCSE 2011, Dallas, Texas, USA (2011)
3. Saeli, M., Perrenet, J., Jochems, W.M., Zwaneveld, B.: Teaching programming in secondary school: a pedagogical content knowledge perspective. Inf. Educ. Intl. J. **10**(1), 73–88 (2011)

4. Forte, A., Guzdial, M.: Computers for communication, not calculation: media as a motivation and context for learning. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (2004)
5. Moström, J.E.: A study of student problems in learning to program. Thesis, Department of Computing Science, Umeå University, Umeå, Sweden (2011). ISBN 978-91-7459-293-1. http://umu.diva-portal.org/smash/get/diva2:447104/FULLTEXT02
6. Harisis, C., Mikropoulos, T.A.: Robotics, visual programming and basic programming structures. In: Komis, V. (ed.) Proceedings of the 4th Panhellenic Conference “Didactics of Informatics”. New Technologies Publications, Patra (2008, in Greek)
7. Atmatzidou, S., Markellis, H., Dimitriadis, S.: Using LEGO Mindstorms in primary and high school: the game as a trigger to learning. In: Komis, V. (ed.) Proceeding of the 4th PanHellenic Conference on “Didactics of Informatics”, Patra, Greek (2008, in Greek)
8. Efopoulos, V., Evaggelidis, G., Dagdilelis, V., Kleftodimos, A.: The difficulties of Novice Programmers. In: Proceeding of the 3rd PanHellenic Conference on “Didactics of Informatics”, Korinthosm, Greece (2005, in Greek)
9. Kagkani, K., Dagdilelis, V., Satratzemi, M., Evaggelidis, G.: A case study of teaching programming in secondary education with LEGO Mindstorms. In: Jimoyiannis, A. (ed.) Proceeding of the 3rd PanHellenic Conference on “Didactics of Informatics”, Korinthos, Greece (2005, in Greek)
10. Moro, M., Alimisis, D.: From the Logo Turtle to the Tiny Robot Turtle: practical and pedagogical issues. In: 5th Conference in Syros – Information & Communication Technology in Education, Syros, Greece (2009)
11. Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., Miller, P.: Mini-languages: a way to learn programming principles. Educ. Inf. Technol. **2**(1), 65–83 (1999)
12. Xinogalos, S.: Scenarios of teaching programming in secondary education. In: Proceedings of the 2nd Panhellenic Conference of Educators for ICT “Utilizing ICT in Teaching Practice, Syros, Greece (2003, in Greek)
13. Tsovolas, S., Komis, V.: Programming robotic constructions: a case study of elementary students. In: Komis, V. (ed.) Proceedings of the 4th Panhellenic Conference “Didactics of Informatics”. New Technologies Publications, Patra (2008, in Greek)
14. Karatrantou, A., Tachos, N., Alimisis, D.: Introduction to basic principles and programming structures with robotic construction LEGO Mindstorms. In: Tzimogiannis, A. (eds.) Proceedings of the 3rd Panhellenic Conference “Didactics of Informatics”. University of Peloponnese, Corinth, 7–9 October 2005 (2005, in Greek)
15. Tzimogiannis, A., Politis, P., Komis, V.: Study of representations graduates Lyceum students on the concept of variable. In: Tzimogiannis, A. (ed.) Proceedings of the 3rd National Conference “Didactics of Informatics”, pp. 61–70, Corinth (2005, in Greek)
16. Korres, G.: Educational Robotics Laboratory using LEGO Mindstorms NXT. In: Proceeding of the 5th PanHellenic Conference of Informatics Teachers (Workshop), Ioannina, Greece (2011, in Greek)
17. Catlin, D., Woollard, J.: Educational robots and computational thinking. In: Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, pp. 144–151 (2014)
18. Komis, V.: Introduction to Educational Applications of Information Technologies and Communications. New Technologies, Athens (2004, in Greek)
19. Frangou, S., Papanikolaou, K.: Educational use of robotic systems. In: Grigoriadou, M. (ed.) Proceedings of the 5th Panhellenic Conference of Didactics of Informatics, EKPA, Athens (2010, in Greek)

20. Nikolos, D., Misirli, A., Davrazos, G., Bakopoulos, N., Komis, V.: Familiarity with the programming language Scratch and robotic pack Lego WeDo. In: Proceedings of the 2nd National Conference, "Integration and Use of ICT in the Educational Process", Patras, Greece (2011, in Greek)
21. Papadakis, S., Orfanakis, V.: A new programming environment for teaching programming. A first acquaintance with enchanting. In: The 2nd International Virtual Scientific Conference - Scieconf 2014, pp. 268–273. EDIS - University of Zilina, Slovakia (2014)
22. Panhellenic Union of Informatics Teachers: Educational robotics in Primary school (2010, in Greek). http://dide.ilei.sch.gr/keplinet/education/docs/pekap_edurobotic.pdf
23. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013, in Greek)
24. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: programming for all. Commun. ACM **52**(11), 60–67 (2009)
25. Bradley, C., Holley, D.: Empirical research into student' mobile phones and their use for learning. Intl. J. Mob. Blended Learn. **3**(4), 38–53 (2011)
26. Zaranis, N., Kalogiannakis, M., Papadakis, S.: Using mobile devices for teaching realistic mathematics in kindergarten education. Creative Educ. **4**(7A1), 1–10 (2013). Special Issue in Preschool Education
27. Wolber, D.: A blocks language for mobile phones: App Inventor for Android. In: Canessa, E., Zennaro, M. (eds.) mScience: Sensing, Computing and Dissemination. ICTP—The Abdus Salam International Centre for Theoretical Physics, Trieste (2010)
28. Papadakis, St., Kalogiannakis, M., Zaranis, N.: Creating applications for smart mobile devices with AppInventor. In: Proceedings of the 7th National Conference of Computer Teachers' "Informatics in Primary and Secondary Education. Challenges and Prospects", University of Macedonia, Thessaloniki (in Greek) (2013)
29. Papadakis, S., Kalogiannakis, M., Orfanakis, V., Zaranis, N.: Novice programming environments, scratch & App Inventor: a first comparison. In: Proceedings of the 2014 Workshop on Interaction Design in Educational Environments, p. 1. ACM (2014)
30. Roy, K.: App Inventor for Android: report from a summer camp. In: Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE 2012). ACM, New York (2012)
31. Hsu, Y.-C., Rice, K., Dawley, L.: Empowering educators with Google's Android App Inventor: an online workshop in mobile app design. Br. J. Educ. Technol. **43**(1), E1–E5 (2012)
32. Liu, J., Lin, C-H., Potter, P., Philip, E., Zebulun, H., Barnett, D-B., Singleton, M.: Going mobile with App Inventor for Android: a one-week computing workshop for K-12 teachers. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013), NY, USA, pp. 433–438 (2013)
33. Grossman, P.L., Lynn, P.: The Making of a Teacher: Teacher Knowledge and Teacher Education. Teachers College Press, Columbia University, New York (1990)
34. Papadakis, St., Kalogiannakis, M., Zaranis, N., Orfanakis, V.: Using Scratch and App Inventor for teaching introductory programming in secondary education: a case study. Int. J. Technol. Enhanced Learn. 8(3/4), 217–233 (2016)
35. Dunham, K.: Helping students design hypercard stacks. Learn. Lead. Technol. **23**(2), 6–8 (1995)
36. McIntyre, T.: Competitive vs. cooperative learning formats, Accessed 10 Sept 2016. http://www.behavioradvisor.com/CoopLearning.html
37. Scholten, B., Whitmer, J., Hypermedia projects-metastacks increase content. Focus Learn. Lead. Technol. **24**(3), 59–62 (1996)

38. Cohen, L., Manion, L., Morrison, K.: Research Methods in Education. Routledge, London (2007)
39. Petrovic, P.: Program your NXT robot with imagine. In: Hungarologo 2007, XII, Nemzeti konferencia a Logo programnyelvroel logo-pedagogiarol, Budapest, pp. 16–27 (2007)
40. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc. (1980)
41. Resnick, M., Silverman, B.: Some reflections on designing construction kits for kids. In: Proceedings of Interaction Design and Children Conference. Boulder, CO (2005)
42. Papadakis, S., Kalogiannakis, M., Zaranis, N.: Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. Int. J. Mobile Learn. Organ. **10**(3), 187–202 (2016)
43. Papadakis, St.: Creativity and innovation in European education. 10 years eTwinning. Past, present and the future. Intl. J. Technol. Enhanced Learn. **8**(3/4), 279--296 (2016)
44. Papadakis, St., Kalogiannakis, M., Zaranis, N.: Comparing tablets and PCs in teaching mathematics: an attempt to improve mathematics competence in early childhood education. Preschool Prim. Educ. **4**(2), 241–253 (2016)
45. Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., Vassilakis, K.: Digital student conference platform implementation: the case study of the “research project” course. Open Educ. **12**(2), 5–23 (2016)
46. Papadakis, S., Kalogiannakis, M., Zaranis, N.: Designing and creating an educational app rubric for preschool teachers. Education and Information Technologies, 1-19 (2017). 10.1007/s10639-017-9579-0
47. Pair, C.: Programming, programming languages and programming methods. In: Green, T., Hoc, J.-M., Samurçay, R., Gilmore, D. (eds.) Psychology of Programming. Academic Press, New York (1990)

Educational Robots Driven by Tangible Programming Languages: A Review on the Field

Theodosios Sapounidis[1,2(✉)] and Stavros Demetriadis[1]

[1] Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{teo,sdemetri}@auth.gr

[2] Department of Philosophy and Education, Aristotle University of Thessaloniki, Thessaloniki, Greece

Abstract. This paper presents a comprehensive literature review on tangible programming languages which are designed to program real robots and robotic mechanisms. Tangible programming interfaces appear to be more accessible to novice programmers and possibly reduce the age threshold for participation, making this way robot programming an educational toy even for preschool and elementary students. Moreover, this paper makes a short examination on the resent research findings on the field of tangible robot programming and argues that the combination of tangible programming and robot construction may offer unique opportunities for educational robotics.

Keywords: Tangible programming · Educational robotics · Interfaces in education

1 Introduction

Educational robotics (ER) is a new innovative tool for education and learning introduced in many schools with the scope to enhance higher level thinking skills and abilities and thus help students solve complex problems in other domain of knowledge [1]. With ER children build robotic entities and program by means of a simple programming language. These two procedures appear to be an ideal platform to motivate, collaborate and finally create new knowledge [2]. Several studies indicate that the ER activities have positive effects on the level of cooperation between the students, the development of critical thinking skills and problem solving [3]. An important pedagogical aspect in ER is that knowledge is established through play within a context that reinforces the reasoning of students about the decisions they make in problem solving challenges [4, 5].

To design and program a robot to do even a simple task, can prove to be extensively demanding for students, not only because of the construction of the robot itself, but also for the effort needed to develop the corresponding programming code. Focusing on programming, studies report the beneficial impact of ER on learning [6, 7] and conceptualize the increased interest of students during the learning activities toward the completion of the educational goals.

Tangible user interfaces (TUIs) for programming on the other hand appear to be a field that offers unique opportunities for learning. In many cases researchers tried to

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_16

teach either simple programming concepts by guiding graphical representations on a computer screen (e.g. Algoblock [8]) or simulate and program advanced concepts, like physical motion using TUIs (e.g. Topobo [9]). In almost all cases researchers took advantage of the fact that TUIs convert physical object into programming elements and that may reduce the age threshold for participation in activities [10]. Moreover, similarly to the construction of the robot, tangible interfaces may offer equal opportunities for participation and active learning [11]. Consequently, the combination of robot construction and programming through tangible interfaces might be an attractive, easier, collaborative way to lead young student and novice adults to the world of ER.

Therefore, this paper makes a review on the field of tangible programing languages that are dedicated to program real robotic constructions. Moreover, it provides a short research review on the field providing insight to the possible benefits of this kind of tools.

2 Tangible Languages for Robot Programing

All systems are presented according to the chronological order of their development and at the end a table summaries their features.

2.1 Tortis – Slot Machine

The Slot machine [12, 13] is the first system constructed by Radia Perlman. It introduced plastic cards that could be piled into three colored stacks (red, green and blue). On the left side of each stack there is a "Do it" button. When someone presses that button, a turtle robot carries out an action, which is visible on the picture of each card of the first stack. When an action is carried out a lamp turns on below this card. The Slot machine offered the direct ability of handling the program either by adding or by rearranging and even by taking out tangible cards. Perlman also introduced special colored cards in order to "call" the program from the other stacks and so, she addressed the procedure "call" [13].

2.2 Tangible Programming for Trains

Tangible Programming for trains is a programming language for trains which created in 1998 by Genee Lyn Colobong and Martin [14]. This language became commercial product by LEGO on 2003 as the "Lego intelli – train". Children were able to program the train's behavior using placing cubic – commands in line. For example, if the train passes the "reverse", then "toot sound" and then "stop" bricks. As a result will go backwards, will make the corresponding sound and it will stop until children press the "go" button. Even when the train is in motion children are able to add and rearrange the bricks and this way alter the program and the behavior of the train. Evaluating the system with real users, it was observed that the construction of the line and the placement of commands was not an instinctive process for a 4 year-old child because, in order to realize what exactly the bricks do, children used to place the bricks in front of the train triggering derailments.

2.3 Tangible Programming Using "Strings"

With this system, users can create simple programs that control robotic games [15]. The system itself consists of two separate surfaces. The first surface represents events and the second actions. The program is formed by relating events with actions, which are carried out in correspondence to events. The connection between events and action are represented with physical strings which are actually cables looking like strings.

After making the appropriate connections – relations between events (which are created by the robot's sensors) with the actions (which are carried out in correspondence to the events) the program is ready to be uploaded to a remote robot's memory. As a result, the robot operates according to the way described by the connections of the "strings".

2.4 Tangible Programming Brick

McNerney's [13] research on tangible user interfaces began in 2000 with the creation of the system "Tangible Programming Brick" that was another environment for tangible programming. The designers decided to create a one-dimension system through which users would make a sequence of commands by putting one Lego brick on top of another. In order to make the system more powerful the bricks were manufactured in a way that it was feasible to get a SIM card from the one side. Initially, this method aimed to give the potential to acquire parameters but in the process the designers realized that they could also include some other features in the slots, such as switches, sensors etc.

Finally, the results of a limited research have shown that the system was too complex for users under the age of 6.

2.5 Electronic Blocks - roBlocks

The Electronic blocks [16] is a system that lets very young children to develop programmable robotic constructions and machineries. Stacking the electronic blocks one on top of the other children are able to create programmable little robotic vehicles and simple constructions. The idea behind electronic blocks is based on three types of blocks (action, sensor and logic blocks). The action blocks make an action like move, light on etc. Sensor blocks detect events like sound, light. Finally, logic blocks play an intermediate role in the middle of action (on the bottom) and logic blocks (on the top) handling and transmitting the signals between them. Consequently, children can easily make a small robot that moves when concurrently light and sound is sensed. The pilot research conducted showed that even very young children are able to use the electronic blocks.

The roBlocks [17] is a later version similar to Electronic blocks with more functions. The evolution of roBlocks is also known today as Cubelets.

2.6 GameBlocks

This system [18] has three parts: cubes, rails, and a humanoid robot. The program is formed on a specific surface (the rails) using cubic command blocks. The relative position of the cubes describes a logical sequence, which is the program itself. The system totally has six commands (without separate parameters) to control a humanoid robot. The offered commands are: ahead, back, body to the left, body to the right, head to the left and head to the right. An interesting feature of this system is that no kind of incorporated electronics is required within the blocks as long as the identification process take place on the rails with external circuitry. Consequently, the computational power is reduced and so no computer is required since the necessary calculations might be easily supplied by microcomputers, which are incorporated in the base of the system.

2.7 Tern – Tangicons

Tern [19] is other educational tangible programming tool that consists of compact physical objects without incorporated electronic circuits and feeds. The programming elements resemble puzzle pieces and are able to control a Lego Mindstorms RCX or the iRobot Create. For the program identification users use a portable scanning system on the top of the construction. After scanning and identification of the commands topology the program is uploaded to the robot for execution. Tangicons [20] system is based on the same principles as Tern.

2.8 The PROTEAS Kit

PROTEAS (PROgramming TangiblE Activity System) kit, is an assembly of tangible and graphical tools for robot programming. It contains two tangible (T-ProRob and T_Butterfly) [21] and one graphical (V-ProRob) [22] interface. The T-ProRob (Fig. 1) subsystem may program the behavior of a NXT Lego robot and consists of 44 cubic shaped commands and parameters. 28 of these cubes represent commands and 16 smaller cubes represent parameters. Users may program the robot by snapping together the desired commands and parameters. To execute the program user has to press the "run button" on the top of the basis ("master box"). T_ProRob supports loops and conditional statements in any form and simultaneously permit users to save their code on dedicated for this purpose cubes.

2.9 Algorithmic Bricks

This is a tangible programming language, based on electronic circuits, that programs a line - tracer robot [23]. Each brick has USB connectors to ensure connectivity among the bricks and microcontrollers for the necessary computational power. The program may be extended in multiple horizontal rows instead of making one long row of commands. Finally, the bricks' hardware, facilitate stacking to simulate the use of parameters. After the program completion, the robot must be connected to the bricks to download the program for execution.

Fig. 1. A program with T-ProRob.

2.10 Dr. Wagon

Dr. Wagon [24] is a quite new tangible programming toy that includes a base block, a series of wooden programming bricks and a wagon-shaped robot. The robot is accompanied with accessories to facilitate customizable appearance. The system supports commands like move forward, turn right/left, light on, make sound. Also users may use advanced programming structures like repeat and conditional statements. The system has magnets on the command blocks thus (a) it may give haptic feedback that the connection between the blocks has been made and (b) reinforces the electrical connectivity among the cubes. To better visualize the use of repeat and conditional statements the system uses stretchable blocks that may encapsulate the desired commands.

2.11 Robo-Blocks

Robo Blocks [25] is an example of active programming with embedded electronic circuits like, T-ProRob and Algorithmic Bricks. The system controls the behavior of a floor robot by snapping and adjusting the available commands blocks. Each command block has magnets and so can easily snap together.

The available commands offer movement control like, move forward/backwards, turn right/left. Some commands are also equipped with a knob and seven segment display to adjust for example for how long the robot will move forward. The robot is also equipped with a pen that may leave a trail on the floor much like the Logo turtle. Consequently, many shapes may be created by the users like rectangles, squares, polygons and of course letters.

2.12 KIBO

KIWI is the ancestor of KIBO [26] which started at Tufts University. The system is based on a robot with accessories to support personalization and a collection of wooden bricks that represent the available commands. Each wooden brick has a separate barcode sticker and for the program creation, users need to assembly the bricks in a logical sequence. For the program recognition, users need to scan each wooden block using the scanning system which is embedded on the robot. With this system users may explore concepts like parameters, loops and conditions.

2.13 T-Maze, E-Blocks, TanProRobot

These three systems were designed by the same researchers [27, 28]. T-Maze is a tangible - graphical game where children try to guide a virtual character to escape form a virtual maze. Initially, the system was based on image recognition like Quetzal and Kibo. Later researchers illustrated E-Blocks. In this case, instead of wooden blocks and visual recognition, researchers used active cubes with embedded electronics as the carriers of the programming language. Finally, their latest release is TanProRobot an active programming language with a real robot. This particular language aims to introduce children to the concepts of event handling using the robot sensors.

2.14 Primo

Primo [29] is a tangible toy which consists of a robot (cubetto), a programming board and a set of wooden instruction blocks. The system support simple commands like turn right/left, go forward and function. After snapping the wooden blocks on the programming board, children by pressing the run button, instruct the base board to recognize the blocks and transmit the program for execution to the robot.

2.15 Code-a-Pillar

Code-a-pillar [30] is a new toy by Fisher-Price for preschoolers. The system consists of a motorized head and 8 command segments that can be connected to the head. The combination of the connected segments is simultaneously the robot and the program itself. The head segment features lights, sounds and blinking eyes to make the construction attractive for children. The available command segments light up as the corresponding action happens and are “move forward” (3), “turn left” 90° (2), “turn 90 right” degrees (2) and “make sound” (1).

2.16 Development of Tangible Programming Languages

Table 1 compiles the system characteristics of the aforementioned tangible programming languages.

Table 1. Features of the tangible programming languages.

Characteristics	Tangible programming languages														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Supports “Loop”	X		X				X	X	X	X		X	X		
Supports “IF”			X				X	X	X	X		X	X		
Supports “Call”	X						X	X	X					X	
Supports parameters			X				X	X			X	X			
Supports different robotic constructions			X	X			X	X			X				X
Support robot personalization										X		X			
Interaction on the interface (e.g. on cube)		X	X	X				X			X				X
Supports different learning styles (e.g. GUI)							X	X							
Computer independent		X	X	X		X			X	X	X	X	X	X	X

1 = Tortis – Slot machine, 2 = Tangible Programming for trains, 3 = Tangible Programming Brick, 4 = Electronic Blocks – roBlocks, 5 = Tangible Programming using “strings”, 6 = GameBlocks, 7 = Quetzal – Tern – Tangicons 8 = PROTEAS kit 9 = Algorithmic Bricks 10 = Dr. Wagon 11 = Robo-Blocks 12 = KIBO 13 = T-Maze E-Blocks 14 = Primo 15 = Code-a-pillar

Although a lot of work appears to have been done in the field of construction current development efforts on tangible robot programming seam to focuses on: (a) the way these systems may effectively represent programming concepts (b) to make systems robust, autonomous and “user-friendly” (c) make constructions simple and effective (d) to reduce the cost of development and (e) to give teachers the opportunities to create a positive learning atmosphere in the classroom using accessories and new features for the robots.

2.17 Research on Tangible Programming

Recently, a number of scientists tried to study the impact of tangible programming languages in relation to traditional graphical. These studies frequently performed usability evaluations comparing tangible and isomorphic graphical languages. The main purpose was to clarify the circumstance under which tangible interfaces offer more benefits in the field of robot programming. In detail Horn et al. [11] compared a tangible programming language against a similar graphical. The study performed in informal learning setting at the Boston Museum of Science. Researchers concluded that the tangible programming language had some advantages in relation to the graphical. In particular it appeared to be more attractive and supportive for active collaboration.

Subsequently, Horn et al. [31] conducted a second study in a kindergarten using the same tangible and graphical language. Implementing qualitative analysis, it was concluded that if 5–6 year old children are given access to the appropriate technology they can create simple programs and may acquire understanding on concepts related with domain of robot programming.

Kwon et al. [23] performed a comparison between Algorithmic Bricks (tangible) and Scratch with the scope to realize whether the tangible tool was appropriate to perform ER activities. The results showed that although there were no significant differences at the domain of usability and performance, the tangible programming language was effective for early stages of learning robot programming.

Sapounidis and Demetriadis [32] explored children's opinions regarding tangible and graphical programming. The results showed that the tangible interface was considered more attractive especially for girls, more enjoyable and finally easier to use for younger children.

Finally, using the same tools Sapounidis et al. [33] measured three variables associated with children performance upon tasks and four variables related with performance during free interaction. Data analysis showed that children produced fewer errors, made more effective debugging and younger children in particular needed less time to accomplish the robot programming tasks with the tangible subsystem. Moreover, during free interaction, elder children were more engaged, created more complicated programs and explored different commands and parameters more actively in the tangible case.

3 Conclusion

In this article, we made a review on the field of tangible programing languages for robot programming. Moreover, we offered a short research review on the field providing insight to the possible benefits of tangible languages. We argue that the combination of tangible interfaces and robots opens new avenues in the field of ER.

Undoubtedly educational robotics is a new innovative instrument for education and learning which is already present in many schools. The scope of ER is definitely to enhance higher level thinking skills and abilities through robot development and programming. The programming languages used for ER at the moment are based mostly on graphical user interfaces. Simultaneously the field of tangible programming appears to by continuously evolved and might offer a new way of interaction between robots and students.

Although, TUIs designers developed a series of systems, there are a lot of things to be done, not only at the domain of development, by also at the domain of research. On the one hand we need more tools to be available in schools and on the other hand we needed to provide empirical evidences that may clarify the circumstance under which robotics and tangibles languages offers more benefits in a real class context. Even though tangibles are believed to be more efficient (easier for younger children etc.) than graphical user interfaces, there is limited research that systematically explores the cognitive and social advantages of TUIs compared to traditional GUI solutions.

References

1. Blanchard, S., Freiman, V., Lirrete-Pitre, N.: Strategies used by elementary schoolchildren solving robotics-based complex tasks: innovative potential of technology. Procedia Soc. Behav. Sci. **2**(2), 2851–2857 (2010)
2. Barell, J.: Problem-based learning: an inquiry approach. Corwin Press, Thousand Oaks (2007)
3. Atmatzidou, S., Demetriadis, S.: Evaluating the role of collaboration scripts as group guiding tools in activities of educational robotics: conclusions from three case studies. In: 2012 IEEE 12th International Conference on Advanced Learning Technologies, pp. 298–302 (2012)

4. Castledine, A., Chalmers, C.: LEGO Robotics: an authentic problem solving tool? Des. Technol. Educ. **16**, 19–27 (2011)
5. Hussain, S., Lindh, J., Shukur, G.: The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data. Educ. Technol. Soc. **9**(3), 182–194 (2006)
6. Atmatzidou, S., Markelis, I., Demetriadis, S.: The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning. In: Workshop Proceedings of International Conference on Simulation, Modeling and Programming for Autonomous Robots, pp. 22–30 (2008)
7. Nourbakhsh, I., Crowley, K., Bhave, A., Hamner, E., Hsiu, T., Perez-Bergquist, A., Richards, S., Wilkinson, K.: The robotic autonomy mobile robotics course: robot design, curriculum design and educational assessment. Auton. Robots **18**(1), 103–127 (2005)
8. Suzuki, H., Kato, H.: AlgoBlock: a tangible programming language, a tool for collaborative learning. In: Proceedings of 4th European Logo Conference, pp. 297–303 (1993)
9. Raffle, H. S., Parkes, A. J., Ishii, H.: Topobo: a constructive assembly system with kinetic memory. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 647–654 (2004)
10. Marshall, P.: Do tangible interfaces enhance learning? In: Proceedings of the 1st International Conference on Tangible and Embedded Interaction, pp. 163–170 (2007)
11. Horn, M., Solovey, S., Crouser, R., Jacob, R.: Comparing the use of tangible and graphical programming languages for informal science education. In: Proceedings of the 27th International Conference on Human Factors in Computing Systems, pp. 975–984 (2009)
12. Kelleher, C., Pausch, R.: Lowering the barriers to programming. ACM Comput. Surv. **37**(2), 83–137 (2005)
13. McNerney, T.: Tangible computation bricks: building-blocks for physical microworlds. In: Proceedings of CHI 2001 (2001)
14. McNerney, T.: From turtles to tangible programming bricks: explorations in physical language design. Pers. Ubiquitous Comput. **8**(5), 326–337 (2004)
15. Patten, J., Griffith, L., Ishii, H.: A tangible interface for controlling robotic toys. In: Proceedings of CHI 2000 Conference on Human Factors in Computing Systems, pp. 277–278 (2000)
16. Wyeth, P., Purchase, H.: Using developmental theories to inform the design of technology for children. In: Proceedings of the Conference on Interaction Design and Children, pp. 93–100 (2003)
17. Schweikardt, E., Gross, M.: roBlocks: a robotic construction kit for mathematics and science education. In: Proceedings of the 8th International Conference on Multimodal Interfaces, pp. 72–75 (2006)
18. Smith, A.: Using magnets in physical blocks that behave as programming objects. In: Proceedings of the 1st International Conference on Tangible and Embedded Interaction, pp. 147–150 (2007)
19. Horn, M., Jacob, R.: Designing tangible programming languages for classroom use. In: Proceedings of the 1st International Conference on Tangible and Embedded Interaction, pp. 159–162 (2007)
20. Scharf, F., Winkler, T., Herczeg, M.: Tangicons: algorithmic reasoning in a collaborative game for children in kindergarten and first class. In: Proceedings of the 7th International Conference on Interaction Design and Children, pp. 242–249 (2008)
21. Sapounidis, T., Demetriadis, S.: Touch your program with hands: qualities in tangible programming tools for novice. In: 15th Panhellenic Conference on Informatics (PCI), pp. 363–367 (2011)

22. Sapounidis, T., Demetriadis, S.: Exploring children preferences regarding tangible and graphical tools for introductory programming: evaluating the PROTEAS kit. In: IEEE 12th International Conference on Advanced Learning Technologies (ICALT), pp. 316–320 (2012)
23. Kwon, D.-Y., Kim, H.-S., Shim, J.-K., Lee, W.-G.: Algorithmic bricks: a tangible robot programming tool for elementary school students. Educ. IEEE Trans. **55**(4), 474–479 (2012)
24. Chawla, K., Chiou, M., Sandes, A., Blikstein, P.: Dr. Wagon: a 'Stretchable' toolkit for tangible computer programming. In: Proceedings of the 12th International Conference on Interaction Design and Children, pp. 561–564 (2013)
25. Sipitakiat, A., Nusen, N.: Robo-Blocks: designing debugging abilities in a tangible programming system for early primary school children, pp. 98–105 (2012)
26. Sullivan, A., Elkin, M., Bers, M.U.: KIBO robot demo: engaging young children in programming and engineering. In: Proceedings of the 14th International Conference on Interaction Design and Children, pp. 418–421 (2015)
27. Wang, D., Zhang, C., Wang, H.: T-Maze: a tangible programming tool for children. In: Proceedings of the 10th International Conference on Interaction Design and Children, pp. 127–135. ACM, Ann Arbor (2011)
28. Wang, D., Zhang, L., Xu, C., Hu, H., Qi, Y.: A tangible embedded programming system to convey event-handling concept. In: Proceedings of the TEI 2016: Tenth International Conference on Tangible, Embedded, and Embodied Interaction, pp. 133–140 (2016)
29. Primo toys. https://www.primotoys.com/
30. Fisher-price Code-a-pillar. http://www.fisher-price.com/en_US/brands/think-and-learn/index.html
31. Horn, M., Crouser, R., Bers, M.: Tangible interaction and learning: the case for a hybrid approach. Pers. Ubiquit. Comput. **16**(4), 379–389 (2012)
32. Sapounidis, T., Demetriadis, S.: Tangible versus graphical user interfaces for robot programming: exploring cross-age children's preferences. Pers. Ubiquit. Comput. **17**(8), 1775–1786 (2013)
33. Sapounidis, T., Demetriadis, S., Stamelos, I.: Evaluating children performance with graphical and tangible robot programming tools. Pers. Ubiquit. Comput. **19**(1), 225–237 (2015)

Learning Programming with Educational Robotics: Towards an Integrated Approach

Marios Xenos(✉), Nikoleta Yiannoutsou, Marianthi Grizioti, Chronis Kynigos, and Sofia Nikitopoulou

Educational Technology Lab, Department of Pedagogy, School of Philosophy, National and Kapodistrian University of Athens, Athens, Greece
{mariosxenos,nyiannoutsou,mgriziot, kynigos,sophieniki}@ppp.uoa.gr

Abstract. Despite the fact that it has been a few years since robotics entered the school and offered new learning opportunities, educational robotics usually is offered in the context of extra-curricular activity (e.g. a "club") which addresses a limited number of students and participation is based on student personal interest. In this paper we explore the potential of ER when it is integrated in the typical school curriculum. In the study we report here, we integrated ER in the computer science curriculum and all students of a 9th grade class engaged with robotics activities. The rationale underlying the study is that robotics can be used as a medium to motivate students in engaging with programming and support them to negotiate real life problems. Analysis of the data collected, indicate that ER when integrated with the computer science curriculum, can create a rich learning environment where programming is contextualized and students are highly motivated to engage and negotiate important STEM concepts.

Keywords: Constructionism · Programming · Educational robotics

1 Introduction

It has been over three decades since Papert and his Constructionism Theory expressed the important role that computation technology plays in overcoming mind difficulties when students construct their own knowledge [1]. A few years later computers entered everyday life and became a necessity for everyone. Later on, we started talking about digital fluency [2] and presented the potential uses of robotics in education [3]. Nowadays, the World Robot Olympiad has been enriched with many different categories in order to motivate the participation of students of all education levels [4] indicating the role of Educational Robotics (hereafter ER) in the school.

In many countries the ER program is offered as a separate domain or in special school "clubs" where a limited number of students participate. In this paper we present a study which takes place in real classroom settings and focuses on integrating robotics in the curriculum of Computer Science (hereafter CS). The study design does not try to replace a

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_17

part of CS curriculum with another approach using ER but it offers an additional environment to enrich and support CS concepts when all students of a class are engaged. The aim of the study is to identify outcomes, difficulties and advantages in this integrated approach.

2 The Study

The study was conducted at a Junior High School in Attiki (Greece) in the context of ER4STEM – a three-year project, funded by the European Commission –which focuses on the development of an integrated framework of the different European approaches in using Robotics in Education. Six 9th grade classes, of 65 students in total participated in the study, which lasted 6 school hours distributed in 6 weeks time. The participants worked with robotics for the first time but they had short previous experience in programming with blocks using Scratch. The research method we used is that of Design-Based Research which includes the design of a pedagogical intervention and its evaluation in real classroom settings with the aim to refine the initial pedagogical design [5]. Design – Based Research focuses mainly on the collection and analysis of qualitative data as the objective is to identify the main characteristics and the different facets of the designed intervention when implemented with students.

2.1 The Task

As we mentioned earlier in this paper the task was mainly oriented towards programming but it did not exclude an engineering aspect. Specifically, the study included a task, which consisted of two main activities, an engineering activity and a programming activity.

The engineering activity was the first activity of the task in the context of which students were expected to adjust an ultrasonic sensor in a preassembled vehicle. The challenge for the students was to keep the construction robust and reliable so that the robot could use the sensor for indentifying obstacles. To this end, students were expected to identify the robot shape, the motion direction and the sensor's best position for obstacle identification. An additional requirement was, for other parts of the robot to not interfere with the sensor and the sensor should not intercept the functioning of the robot.

The programming activity was about defining the behavior of the robot so as to be able to identify obstacles while moving. In this activity, the teacher gave to the students a predefined simple program which allowed the robot to detect obstacles. Students were asked to adjust the program so that the robot would detect an obstacle at a specific distance and react avoiding the obstacle with a predefined movement (i.e. performing a U turn and stopping). The programming concepts involved in this activity include sequential command execution (sequence structure) and a real world calculation problem, as students should test and explore if their robot detected with accuracy the obstacle and if this behavior was reliable. Here, the command execution time affects the next robot action, a problem which was expected to be identified and rectified by the students. Regarding the robot reaction, students were asked to move the robot by 4 floor plates, which required relating the program (commands, parameters and values) with the robot's real world movement. The final phase of this activity involved students extending the program so that the vehicle could act more realistically by avoiding efficiently any obstacle encountered. In all phases, teacher had designed some additional sub-tasks to engage students that would come quickly to a solution and so have free time (e.g. program improvement, robot fine-tuning).

2.2 Robotic Kit

In the study, students used Lego NXT (one robotic kit per group) for the engineering activity and Lego programming language for the programming activity.

2.3 Data Collection

During the study, the data collected involved screen capturing during the programming activity and researcher observation notes after the end of each session. The teacher acted also as researcher. At the beginning and at the end of the study, students filled in questionnaires where they expressed their opinions about the actual activity and about robotics, mathematics and science, in general. In the middle of the study, each group prepared a self - evaluation document and shared it with other groups. Finally after the workshops, the teacher - researcher interviewed selected groups of students.

3 Results

The analysis of the collected data provides us with strong evidence regarding the acceptance of the new field of robotics and the level of student engagement in the educational process. It also gives some indications about the knowledge and the experience students earned. Next we present some indicative data from student learning activity which shows their active engagement with programming and engineering concepts.

3.1 The Human Body Analogy: Experimenting with the Sensors

During the first activity, when students tried to assemble the sensor, the identification of the best sensor position was not a difficult task as students considered sensor as it was "eyes" of a human body. A couple of groups believed that the "eyes" (sensor) should be positioned with accuracy lined up with robot body in order to work properly. The following extract is indicative:

St1: Let's put it (i.e. the sensor) exactly here (they point at the front of the vehicle)
St2: But there are no holes here (available sockets)
St1: If we put it in the back? Will it work? Or it will be head-butt?
St3: Let's do what we can for now (i.e. put in the back) and will see…
St2: Should we ask the teacher?
St1: Ask what?
St2: If it is important where we put…

In the extract we presented above we can identify two important aspects. One is the use of the analogy of the human body directing students to place the sensor in the front of the vehicle. The other interesting aspect involves the learning process taking place during student experimentation with the sensor. As we mentioned earlier, students formulated their initial hypothesis about the position of the sensor based on the analogy of the human body. However students could not test this hypothesis as when they observed the actual structure of the robot they realized that they could not find a slot for placing the sensor. When encountered with this impasse they followed another direction for their hypothesis formulation by focusing on the affordances of the vehicle (they identified a socket in the middle of the vehicle). As this option sounded counter-intuitive one student suggested to experiment by testing this option ("let's see what we can do") whereas the other suggested to seek confirmation from the teacher (Fig. 1).

Fig. 1. Students trying to position the sensor

From this simple example the conclusions we can draw are the following: a simple engineering problem can offer rich learning opportunities as students can draw from their experience to make conjectures, then they need to analyze and observe the affordances of the robot to refine these conjectures. Another important aspect in this process is that the physical construction (i.e. placement of the sensor) is tightly linked to the behavior of the robot (if it can detect obstacles or not).

3.2 Facing the Limitations of Sequential Programming

During the programming activity students were engaged with the reliable detection of an obstacle. Students for the first time were presented with a tangible and visible result of a sequential command execution in real world: the robot could not check the sensor input until it finishes the previous action. The robot-programming environment provides ready-made solutions for this issue but the study was designed so as to not present them in order for the students to indentify the situation and give their own solutions.

Only 5 out of the 22 groups indentified the problem and suggested a solution. The rest groups attributed the problem to the functioning of the sensor or to its positioning.

An interesting observation is that most of the students that provided a successful explanation were not good in programming before the study.

St1: Sir, (i.e. teacher) it is doing OK
T: What?
St1: It detects the obstacle
T: At 20 cm distance? (20 cm was the given distance)
St2: Yes Sir, didn't you see it?
St1: Not exactly at 20 cm. A little shorter
T: Is it always the same distance?
St1: Hmmm, every time is less. Maybe we could increase the distance (i.e. at the program)?
St2: The program is fine. This (points to the sensor) is the problem

From the student discussion in the extract above it appears that when students discuss with the teacher they purposefully focus on the result of the behavior (i.e. detecting the obstacle) and not on the conditions of this detection (in 20 cm). This becomes evident when the teacher asks clarifications. At this point students seem to have identified the problem: i.e. "the distance is a bit shorter" and "every time the distance is less". From this point we see that students differentiate suggesting two directions for repairing the problem: one refers to the program suggesting to increase the given distance which would repair the first part of the identified problem (the distance covered is less than 20 cm) but not the second one (every time is less). The other student considers that the problem is related with the sensor and not with the program (Fig. 2).

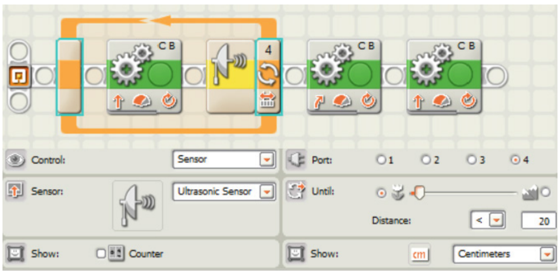


Fig. 2. The result of programming the sensor

Although students in the above extract did not solve the problem at this point, it appears that the structure of the activity and the use of robotics created a learning environment where a) the limitations of sequential programming were clearly demonstrated b) students observing robot behavior could identify in a structured discussion with the teacher the actual limitations of the robot's behavior and c) students could follow different clues in order to pursue solutions to the problem (i.e. the sensor or the program).

3.3 Translating Real World Distances into Programming Values

The next part of the programming activity involved the accurate movement of the robot on 4 floor plates. The strategy that all groups followed was calculations via experimentation. Some groups tried random program values for the whole movement (4 plates). Other groups performed continuous tests on the floor for one plate and then they adjusted their calculation for the 4 plates. None of the groups tried to calculate the distance covered by the robot in relation to the wheel rotation. Even after strong teacher prompting for a computational solution, only few students managed to relate the wheel circumference with the travelling distance.

3.4 From Sequential Programming to Loops

The last problem that students had to solve is rather the most interesting in respect to programming and robotics. The obstacle identification should be continuous in order to build a "smart" vehicle. The verbalization of the problem by the teacher helped students to identify the general direction for solution seeking, because when teacher used phrases like "it (the robot) has to do it continuously", "unlimited", "forever", soon students came to the solution of using the "loop structure". However, not all groups put the loop block at the correct position in the program. Some groups put the loop after the obstacle recognition thinking that robot has to find an obstacle and then search for the next one (Fig. 3). Other groups put the loop in the correct position but included only some commands (blocks) and not all. Some other groups put an "empty" loop after initial program thinking that robot can identify the correct use automatically (Fig. 4). Students had none or limited knowledge of nested structures, so faulty usage of loop was not a surprise. This problem was posed in order to engage students with more advanced programming structures and to showcase mainly the realistic robotic aspect giving an almost human behavior to the robot.

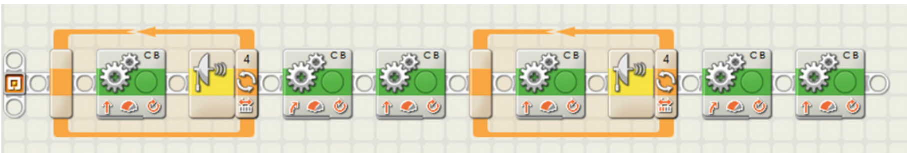


Fig. 3. Incorrect use of the second loop: new loop and new commands after the initial one

Fig. 4. Incorrect use of the second loop: putting the loop after the initial program

3.5 Student Views About Robotics

Important data was taken by the answers that students gave to anonymous questionnaire after the study. In the evaluation of the task, 90% of students rated the interest of the activity with 4 or 5 out of 5. One student gave negative rate and wrote that “it was funny but I was not interested in it”. More elaborate information about the participant’s opinions regarding the task is found in open questions. Specifically, at the question “What have you learned about yourself?” only 6 students did not answer or answer negatively. The rest of the students gave positive and in many cases long answers:

> “I learned to program a robot which can be very useful for me later in my career or in my life generally”
> “Through robotics I understood how things work in everyday life and I realized that robotics can make us smarter”
> “Eventually, I realized that I do like problem solving projects such as building a robot and make it functional”
> “Although I don’t like mathematics, I managed to use it and I realized that if you work hard you will succeed”

At the question “What have you learned about working with robots?” only 4 (out of 65) students did not answer. The rest of students gave positive answers focusing on the learning results or on the difficulties of the activities:

> “It is you that can program them whatever you like and they cannot step away from your orders”
> “Although it is sometimes difficult in the end the result is great”
> “Working with robots is very interesting. It teaches you how to think in alternative ways in order to achieve your goals”
> “I realized that nothing is impossible and with a little hard work I can construct and program a robot”

The overall student interest in robotics and also the importance of robotics in everyday life is designated at the general evaluation answers:

> “It was a very interesting course in the field of Computer Science which helped us to understand how things work and how robots are tightly connected with mathematics and physics”
> “At the beginning, I wasn’t interested at all. But afterwards I understood how important robots are and how important is to collaborate well”
> “I had a great time and it was the first time that I programmed. This workshop helped me understand is the importance of mathematics, which I didn’t find important before”.

4 Concluding Remarks

The data presented earlier showed that an integrated approach of robotics in the curriculum of computer science can offer a learning environment which is engaging for all students and provides opportunities for negotiation and elaboration of important programming concepts. Furthermore, students of both sexes even if they believed that robotics and programming does not interest them, they managed to have a fruitful collaboration and engaged actively with the learning activities. Most of the participants realized the different domains integrated in robotics and they stated that they learned a lot of different things. For some students, robotics was a motive to get involved with programming and realize an authentic aspect of it. Because of prior knowledge and study design itself, we cannot argue that students learned or understood programming structures even if they think so. However, they had the chance to practice and test CS concepts, explore the results and indentify the connection between programming and real world. On the other hand, it seems that students have a difficulty to transfer prior programming knowledge from one programming environment (Scratch) to another (LEGO), so maybe we have to rethink and reconsider about the tools that typical curriculum uses. A possible side benefit is that all participants stressed the role of collaboration and the importance of team spirit for goal achievement. Our analysis of questionnaires, student constructions and researcher reflection notes, indicates that integration of educational robotics in school curriculum may contribute in:

- Engaging all students in programming especially when they believe it is a boring and difficult subject.
- Presenting career opportunities in the field of computer science and technology
- Demystifying domains that by default are considered as difficult (robotics, mathematics)
- Helping children to understand how things work in everyday life
- Helping students in developing problem solving skills

In conclusion, integrating Educational Robotics into the typical school curriculum seems to be beneficial for a number of reasons. Well-designed activities can result to a rich outcome not only in a specific domain but in many aspects of learning process. It is critical to examine and perform more focused analysis in each aspect of this learning process, in order to prepare a new enriched curriculum.

Acknowledgments. The research reported here has received funding from H2020 project Educational Robotics for STEM: ER4STEM (No. 665972).

References

Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books Inc, New York (1980)

Resnick, M.: Rethinking learning in the digital age. In: Kirkman, G. (ed.) The Global Information Technology Report: Readiness for the Networked World. Oxford University Press (2002)

Malec, J.: Some thoughts on robotics for education. In: AAAI Spring Symposium on Robotics and Education (2001)

World Robot Olympiad. http://www.wroboto.org

Cobb, P., Confrey, J., diSessa, A., Lehrer, P., Schauble, L.: Design experiments in educational research. Educ. Researcher **32**(1), 9–13 (2003)

Short Papers Reporting Good Practices or Work in Progress (Presented in the Conference as Posters)

Design Requirements for Educational Robotics Activities for Sustaining Collaborative Problem Solving

Raoul Kamga[1(✉)], Margarida Romero[1,2], Vassilis Komis[2], and Anastasia Mirsili[2]

[1] Université Laval, Québec, Canada
{raoul.kamga-kouamkam.1,margarida.romero}@ulaval.ca, Margarida.Romero@fse.ulaval.ca
[2] University of Patras, Patras, Greece
{komis,amirsili}@upatras.gr

Abstract. The objective of this study is to identify the instructional design requirements of Educational Robotics (ER) activities for sustaining collaborative problem solving (CPS) skills. In CPS activities, learners work in teams towards the analysis, the identification and the solution of a problem. PISA and most of the 21st century skills frameworks have promoted the importance of CPS for 21st century education. In this study, we aim to analyze ER opportunities for 21st century education, in general, and CPS skills, according to the PISA CPS framework. To achieve this, we first review the design requirements of the learning activities that have been identified in the literature as sustaining CPS. Afterwards, we identify the design requirements of ER activities that can sustain the development of CPS. Activities engaging learners in authentic ill-defined problems, with a certain level of complexity are identified as design requirements that have the possibility to promote CPS in ER activities.

Keywords: Educational robotics activity · Robots · Collaborative problem solving · Design

1 Complex and Collaborative Problem Solving for the 21st Century Challenges

Education should contribute to the preparation of learners to the 21st century challenges and to the continuous evolution of digital technologies [1]. Considering the increasing complexity and uncertainty in the professional sphere, problem solving is identified by several studies as one of the key skills in learning and professional contexts. The Future of Jobs Report released by the World Economic Forum [2] not only identifies complex problem solving as the most important skill to develop in 2015 but also predicts its importance for Horizon 2020. In the report, complex problem solving is defined as the “capacities used to solve novel, ill-defined problems in complex, real-world settings”. The capacities to solve ill-defined, novel and complex problems could be enhanced by the collaborative problem solving (CPS) learning activities, in which a group of persons are involved in the analysis, the identification and the solution of a problem. CPS is an important skill that should be nurtured from the early stages of education, through adapted levels of complexity and simplification of real-world settings. At the educational

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_18

level, CPS has been identified by different studies in relation to the learners' capacity to develop and sustain their reasoning, decision making, cognitive thinking, social and communication skills [3, 4].

2 Collaborative Problem Solving (CPS) Skill

According to PISA [5] CPS is "the capacity of an individual to: recognise the perspective of other persons in a group; participate as a member of the group by contributing their knowledge, experience and expertise in a constructive way; recognise the need for contributions and how to manage them; identify structure and procedure involved in resolving a problem; and, as a member of the collaborative group, build and develop knowledge and understanding". PISA proposes a similar definition with an emphasis on the "appropriate action to solve the problem". PISA defines CPS as the capacity "[…] to effectively engage in a process whereby two or more agents attempt to solve a problem by sharing the understanding and effort required to come to a solution and pooling their knowledge, skills and efforts to reach that solution." [5]. PISA's definition has three components:

- "Establishing and maintaining shared understanding": shares knowledge with other team members and consider their views in order to reach an agreement on the understanding of the problem;
- "Taking appropriate action to solve the problem": acts effectively in order to solve the problem;
- "Establishing and maintaining team organisation": structures the team so that its operation is efficient to solve the problem.

While the Griffin, McGaw and Care [11] and PISA definitions consider individual-team coordination in terms of understanding, knowledge and contributions, the solution is not explicitly defined as the result of an iterative process of ideas, generations and failures which are recognized as key components of problem solving in ill-defined situations. Thereafter, we propose to extend the PISA definition of CPS by integrating a fourth component: "Co-regulating the interdependent efforts of the teammates in an iterative approach of design-thinking and prototyping where failure is valued for improving the solution and learning from failure". In accordance to this addition, we propose to identify CPS as the ability to identify a problematic situation for which the process and solution are not known in advance, and the capacity of a team to determine, build and enforce a solution in an effective way.

3 Design Requirements for Educational Robotics (ER) Learning Activities Sustaining the CPS

In this section, we introduce the design requirements for ER learning activities sustaining the CPS based on the PISA approach and the instructional design characteristics that have been related to the development of CPS in the learning science studies. ER is the educational use of robotics technologies to achieve a learning goal or to develop a skill in formal and informal learning environments [6]. The use of educational robotics in the classroom is based on Papert [7]. According to Eguchi [8], there are three approaches: goal-oriented approach, theme-based curriculum approach and project-based learning. The last two approaches are used for the lessons in class. Whatever approach is used, it is possible to develop the CPS skill. However, what are the specificities of the activities to be undertaken that can support the development of this skill?

Based on the components identified through the literature review on the design requirements for the learning activities aiming to support and sustain learners' development of the CPS and the components that we offer after analyzing extended matrix of CPS for PISA 2015, we identify in this section the design requirements of ER activities for sustaining the CPS. Some requirements are generic to all matrix elements and some of them are specific to the matrix elements. The term OPT represent optional element.

- **General characteristics:**
  - Organization of the learners as a small team.
  - Novel task: problem situation in which the solution (or part of it) is not known in advance.
  - Practical activity: an activity in which the team members have to handle physical components of the project.
  - Complex problem: a problem whose solution is not obvious or whose resolution is not to simply apply a formula.
- (1) **Establishing and maintaining a shared understanding.** In order to support the first component there needs to be:
  - An ill-structured problem
  - Meaningful activities or contextualised activities
  - A complex activity
  - A multidisciplinary activity or an activity that requires more skills (OPT)
  - An activity that requires interdependence (OPT)
  - A real-world activity (OPT)
- (2) **Taking appropriate actions to solve the problem.** In order to support the second component there needs to be:
  - An interactive activity
  - A multitask activity
  - An ill-structured problem
  - A real-world problem
  - An exploration activity (OPT)
  - A meaningful activity or contextualised activity (OPT)
- (3) **Establishing and maintaining team organisation.** In order to support the third component there needs to be:
  - A multitask activity
  - A meaningful activity or contextualised activity
  - A real-world activity
  - An interdependence activity (OPT)
  - A dynamic activity (OPT)
- (4) **Iterative co-regulation of intermediate solutions.** In order to support the fourth component there needs to be:
  - An ill-structured problem
  - An exploration activity (OPT)

4 Discussion

During the ER activities, the level of student engagement depends on the type of activities in which they are involved [9]. Similarly, the level of support that the ER activities contribute to the development of CPS for students depends on the type of activities in which they are incurred. Certain ER activities do not support CPS. For example, the activities in which students do not handle physical objects or those related to a procedural use of robots [9] do not support the development of CPS. In school, the development of ER activities to support the CPS is not always easy as there are very few canvas to guide teachers [10] or education consultants in the design and implementation of these activities. The elements to be included in the design of ER activities to support the CPS are little known. In schools, learners' CPS development requires a mid or long-term period of time; learners should persevere because these elements will enable them to move from a prototype solution to another in order to solve the problem. ER activities that aim to support the CPS should consider the grade level of the students to whom these activities are intended, their life contexts and the type of robotics technology to use during the activity.

References

1. Gu, J., Belland, BR.: Preparing students with 21st century skills: integrating scientific knowledge, skills, and epistemic beliefs in middle school science curricula. In: Ge, X., Ifenthaler, D., Spector, J.M. (eds.) Emerging Technologies for STEAM Education. Educational Communications and Technology: Issues and Innovations, pp. 39–60. Springer, Switzerland (2015)
2. World Economic Forum (WEF). The Future of Jobs Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution (2016)
3. Kelson, A., Distlehorst, L.: Groups in problem-based learning (PBL): essential elements in theory and practice. Probl.-Based Learn. Res. Perspect. Learn. Interact., 167–184 (2000)
4. Kim, M., Tan, H.T.: A collaborative problem-solving process through environmental field studies. Int. J. Sci. Educ. **35**, 357–387 (2013)
5. OCDE. Pisa 2015: draft collaborative problem solving framework (2013)
6. Romero, M., Kamga, R.: Usages de la robotique pédagogique en éducation primaire selon son intégration disciplinaire et le développement des compétences du 21e siècle (2016)
7. Papert, S.: Constructionism: A New Opportunity For Elementary Science Education. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, Cambridge (1986)
8. Eguchi, A.: What are students learning from educational robotics?-different approaches to educational robotics. In: Society for Information Technology & Teacher Education International Conference, vol. 2009, pp. 3547–3554 (2009)
9. Romero, M., Dupont, Y.: Educational robotics: from procedural learning to co-creative project oriented challenges with LEGO WeDo, pp. 6159–6163 (2016)
10. Williams, D., Ma, Y., Prejean, L.: A preliminary study exploring the use of fictional narrative in robotics activities. J. Comput. Math. Sci. Teach. **29**, 51–71 (2010)

Hedgehog Light – A Versatile, White Box Educational Robotics Controller

Clemens Koza, Wilfried Lepuschitz(✉), Martin Wolff, Daniel Frank, and Gottfried Koppensteiner

Practical Robotics Institute Austria (PRIA), Vienna, Austria
{koza,lepuschitz,wolff,frank,koppensteiner}@pria.at

Abstract. Robotics curricula that include programming generally require a robotics controller as core platform. The controller and its capabilities shape the offered curriculum in terms of programming means, and therefore age appropriateness. A closed, black-box controller also limits the uses outside the context of such a curriculum. The Hedgehog Light controller aims to imply as few limitations as possible by using open source methodology, the open and well-documented Raspberry Pi, a multi-platform protocol and programming interface, and that can mostly be built in fabrication laboratories (fab labs).

Keywords: Controller · Raspberry Pi · White box · Open source · Programming

1 Introduction

For implementing a robotics curriculum, educators need to choose a controller as their platform. Many factors, such as price, size, processing power, means of programming, peripheral connectivity, and integration with building blocks, influence this decision.

For example, controllers used in the toy-like robots introduced by Saleiro et al. [1] are extremely optimized for cost, providing no capabilities beyond those needed for their proposed elementary school use. On the contrary, a robotics kit targeted at tertiary education commonly include a controller able to support complex use cases. These represent examples of programmable robots and complete starter kits, respectively, in the terminology introduced by Hilal et al. [2]. Hedgehog Light tries to fit diverse use cases without being too complex or expensive.

2 Architecture

The general concept of Hedgehog Light (see Fig. 1) involves the usage of a smartphone as High Level Control (HLC) that can be used as programming device and for hosting any additional high level programs as well as the Hedgehog app for manually controlling motors and servos attached to the controller.

The Low Level Control (LLC) controller is composed of the Software Controller (SWC, a Raspberry Pi) and the pluggable Hardware Controller (HWC) [3], with a STM32F4 microcontroller as its core.

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_19

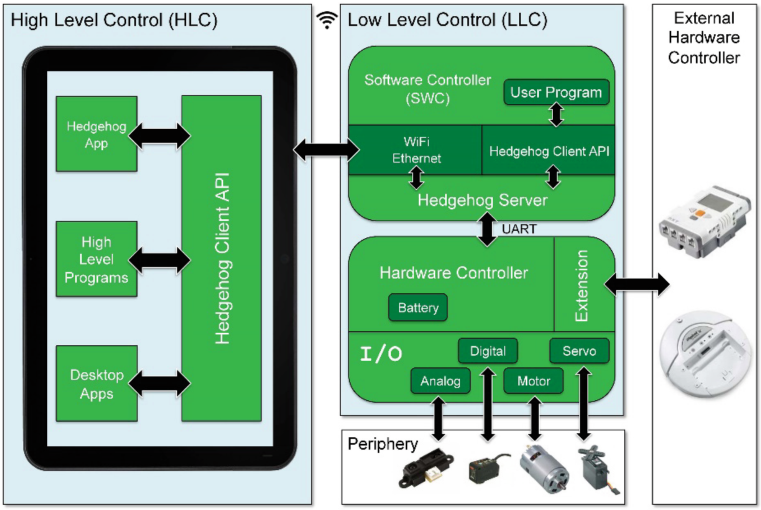


Fig. 1. Architecture of Hedgehog Light.

The HWC is responsible for time-critical and periphery tasks, such as generating pulse-width modulated signals for motors and servos, and reading analog and digital sensors. Four motors and servos, as well as sixteen analog or digital sensors can be connected to one controller. Additional interfaces, such as universal synchronous/asynchronous receiver/transmitter (USART) or inter-integrated circuit (I2C), can be used for connecting complex external hardware, e.g. an iRobot Create.

The SWC interfaces with the HWC via USART and hosts a server that allows both local and remote client programs to access the hardware. The used Raspberry Pi 3 Model B offers Ethernet for stationary and WiFi for mobile robotics projects.

User programs communicate with the Hedgehog Server using a message protocol based on ZeroMQ [4] and Protobuf [5] for efficient serialization. Both of these libraries target many programming languages. Clients can choose to directly compose protocol messages, or use an API library, which currently exists for Python and Java.

3 Maker Aspects and Open Source

The Hedgehog Light controller is designed to be assembled with equipment as it is typically available at fab labs. A laser cutter is needed for the acrylic case of the controller and a reflow oven is required for SMD (surface mounted device) assembly.

Parts that need to be purchased include the battery, Raspberry Pi, SMD parts and pin headers for the HWC board, acrylic sheets and screws for the case, and the HWC printed circuit board (PCB). To keep the controller compact, it was necessary to design a 4-layer PCB, which is not so easily manufactured in fab labs.

The controller software consists mostly of free, open source software: Raspbian is used as the operating system; additional free software packages are installed from Raspbian and PyPI (Python Package Index) repositories, or in the case of original Hedgehog software from GitHub. The microcontroller firmware also relies solely on free software libraries. The Hedgehog software is licensed under the AGPLv3.

4 Capabilities and Use Cases

Hedgehog Light was designed as an educational controller aiming to support curricula for diverse audiences. It already supports graphical programming and multiple programming languages and, in line with maker principles, exposes almost all of its functionality for curious examination and customization.

4.1 Graphical Programming Using Pocket Bot

Based on Pocket Code, a graphical programming environment for Hedgehog Light called “Pocket Bot” was created (see Fig. 2). A program consists of a background and different objects, making up the “stage”. Both of these can have scripts that are triggered by different events, such as program starting, or an object being tapped on screen. Scripts are composed of sequential blocks that each accomplish a task.

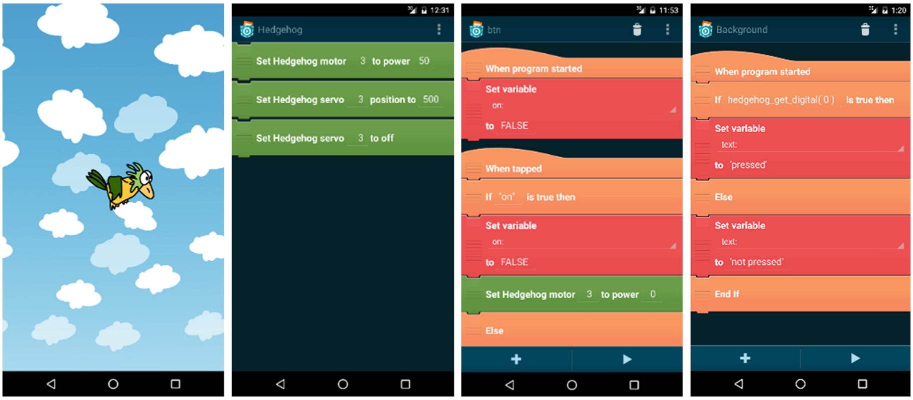


Fig. 2. From left to right: the stage of the Pocket Code demo application; Hedgehog-specific programming blocks; a script for toggling a motor; and usage of sensor functions.

Pocket Bot extends Pocket Code to communicate with the Hedgehog Light controller, introducing new blocks and functions. Users can learn about imperative programming, including loops and conditionals, and robotics concepts, such as kinds of actuators and sensors, without having to learn a textual programming language.

4.2 Textual Programming Using Python

Python is a powerful, multi-paradigm programming language and at the same time easy for beginners to learn [6]. As it is the language used for most of Hedgehog Light’s software, it was also the first language to get a Client API library. As before, users can learn about programming and robotics concepts, but using a highly relevant textual programming language. Beginners are confronted with minimal language overhead when writing first commands for their robot. Advanced users can benefit from Python's rich standard library and collection of third party packages.

4.3 Microcontroller Programming

Besides the microcontroller firmware being open source, the SWC also includes the whole toolkit for compiling and deploying a modified or alternate firmware to the HWC. Hedgehog Light can be used for microcontroller programming without having to set up cross-compilation toolchains. Microcontroller programming in C contrasts with high level programming in Python due to manual memory management, the lack of operating system features, and interrupt handling as a new way of concurrency.

5 Conclusion and Future Work

Hedgehog Light is a versatile controller ideal for incorporation into robotics kits. An IDE with basic support for Python as well as controlling motors and servos exists, but configuration and display of sensors is missing. The controller was yet used only during one workshop and it was shown that it is a stable device. The usage in further workshops will deliver insights regarding the applicability of Hedgehog in education.

Acknowledgment. The authors acknowledge the financial support from the European Union's Horizon 2020 research and innovation program under grant agreement no. 665972.

References

1. Saleiro, M., Carmo, B., Rodrigues, J.M.F., du Buf, J.M.H.: A low-cost classroom-oriented educational robotics system. In: 5th International Conference on Social Robotics, pp. 74–83 (2013)
2. Hilal, A.R., Wagdy, K.M., Khamis, A.M.: A Survey on Commercial Starter Kits for Building Real Robots. German University in Cairo (2007)
3. Koza, C., Krofitsch, C., Lepuschitz, W., Koppensteiner, G.: Hedgehog: a smart phone driven controller for educational robotics. In: 6th International Conference on Robotics in Education, pp. 50–52 (2015)
4. Hintjens, P.: ZeroMQ: Messaging for Many Applications. O'Reilly Media, Inc., Sebastopol (2013)
5. Protocol Buffers. https://developers.google.com/protocol-buffers/docs/overview
6. Donaldson, T.: Python as a first programming language for everyone. In: Western Canadian Conference on Computing Education (2003)

Using LEGO Mindstorms as an Instructional Tool to Teach Science in Primary Education

Marievie Panayiotou(✉) and Nikleia Eteokleous-Grigoriou

Frederick University Cyprus, Nicosia, Cyprus
{pre.mp,n.eteokleous}@frederick.ac.cy

Abstract. This paper reports a case study concerning the use of LEGO Mindstorms Robotics as an instructional tool in science education. In particular, the paper briefly delineates a teaching class module in the field of Science Education for the primary level. It is the authors' expectation to evaluate the proposed methodology under a large-scale study designed to measure the effectiveness of this approach in terms of learning outcomes, students' satisfaction, enjoyment and perceived usefulness.

Keywords: LEGO Mindstorms · Educational robotics · Evaluation robotics · Science education

1 Introduction

Nowadays, robotics has been integrated in several areas, such as medicine, industry, space, military and everyday life. The modern trend wants to exploit robotics in educational context as an important tool from kindergarten to university education. The integration of robotics in education has been reported as a new promising approach to boost the learning process, particularly within the science, technology, engineering and mathematics (STEM) fields [1–3].

Over the past years, many different research approaches have been made in order to clarify the educational benefit, resulting from the integration of robotics in education. Most research efforts show positive results arising from such integration and usually focus on developing certain skills, such as problem solving, deductive ability, critical thinking, socializing and the understanding of various concepts [4, 5]. Moreover, other researchers suggest that the use of robotics develops scientific process skills [6], enhances the motivation for learning [2], improves the learning experience' quality [7], and have positive effects at the cooperation level [8]. However, Benitti (2012) [9], during her literature review, stated that the empirical evidences supporting the effectiveness of educational robotics are still limited, while other researchers dispute the positive contribution of robotics in the educational process [4, 10].

The need to prepare students for a technological culture that requires problem solving skills, new ideas and innovative products, is accumulating and growing. The National Science Foundation highlighted the need to increase the interest in the STEM fields by funding a larger number of programs that might encompass more

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_20

undergraduate students [11]. To improve students' performance and interest in the STEM fields through educational robotics, students should become involved with related activities starting from a very early age. Bearing these ideas in mind, this paper presents the design of an educational module that supports hands-on activities using Lego Mindstorms in the field of science education.

2 Development of an Educational Lesson Plan

This section briefly delineates the design of five activities of a typical educational module, "Light Concepts", relying on the LEGO Mindstorms technology. The proposed activities are tailored to overcome the challenges of teaching light concepts to primary students. It is noteworthy that the activities of educational robotics vary depending on the level of students, the educational objectives and the exploited robotic technology. For the herein proposed activities, the students involved have attended a 7 h NXT programming course. During that course, the students assembled a robot with a built-up light sensor.

"Light Concepts" is an educational module usually being taught in the fifth-grade of elementary education. Its educational goals are as follows:

- Understanding light propagation in linear way and to all dimensions;
- Understanding light' interaction with different materials and sort them using the terms transparent, semi-transparent and opaque;
- Understanding the light phenomena such as reflection, diffusion and absorption.

This module has been thoroughly studied, and while some works try to understand students' common misconceptions about light [12, 13], others try to find out ways to overcome such misconceptions by implementing conceptual change [14, 15]. Anderson and Smith, after researching fifth-grade students' misconceptions, concluded that students' beliefs about light, and particularly its motion, are often vague and confusing [13]. Students usually believe that eyes perceive objects directly, rather than detecting light reflected by those objects. Students could not distinguish light source and light reflection, as some of them tent to believe that everything they see with light is a light source [12].

The proposed teaching class module comprises five activities, supported with robotics technologies, tackling each of the "Light Concepts" educational goals, as follows:

1. **Understanding light propagation in a linear way and in all dimensions -** Using light sensor' readings for each tab, students will have to discuss and make conclusions about light propagation;
2. **Understanding light's interaction with different materials and sort them using the terms transparent, semi-transparent and opaque** - The aim of the activity is to observe that a different amount of light entering the robot depending on the material that is placed in front of the light source;
3. **Understanding light' reflection** - Students will examine reflection phenomena by observing reflected light rays' direction and quantity;

4. **Understanding the light diffusion** - With this activity, students are expected to understand that a rough surface behave differently when in contact with light than the mirrors do, while still reflecting some light;
5. **Understanding light diffusion and absorption** - The activity seeks to stimulate children to understand that the materials can absorb and reflect light rays depending on their properties, namely the colour.

Conventional science teaching aims to provide students observation-based data in laboratory experimental settings. The approach discussed here, provides support in to how a robotic device can be a complementary addition to the experimental space. It can alter students' perception by accessing different type of information both through light intensity measurements and via the observation of the robot's function. Comparing to the solidary use of a light sensor, the observation of a following action as a response to the environmental change, by the robot, offers students' access to different type of exploring data in respect to the common eye, which can lead them to a higher level of explanatory reasoning regarding light phenomena. What is more, the proposed undertakings are expected to enhance students' understanding as students need to use their new obtain knowledge to program the robot perform specific tasks. As Herbart (1895) suggested, the practical application of the new knowledge is significant and should be included in educational practice [15].

3 Conclusion

This paper presented a novel methodology to redesign a typical educational module, "Light Concepts", using the LEGO Mindstorms robotics technology. Five activities were proposed to enhance students' understanding of the most common educational goals concerning this module. As future work, the proposed module will be evaluated under a population of 96 primary students and compared with a control group. It is expected that the results of this study shall encourage towards the adoption of robotics as an educational tool to foster science education.

References

1. Fagin, B., Merkle, L.: Measuring the effectiveness of robots in teaching computer science. ACM SIGCSE Bull. **35**(1), 307–311 (2003). ACM
2. Mauch, E.: Using technological innovation to improve the problem solving skills of middle school students. Clearing House **75**(4), 211–213 (2001)
3. Robinson, M.: Robotics-driven activities: can they improve middle school science learning? Bull. Sci. Technol. Soc. **25**(1), 73–84 (2005)
4. Hussain, S., Lindh, J., Shukur, G.: The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data. Educ. Technol. Soc. **9**(3), 182–194 (2006)
5. Karatrandou, A., Panayiotakopoulos, C., Pierri, E.: Understanding the concepts of Physics in elementary school using Lego Mindstorms: a case study. In: Proceedings for 5th ETPE Conference, Thessaloniki, Greece, 5–8 October 2006

6. Sullivan, F.R.: Robotics and science literacy: thinking skills, science process skills and systems understanding. J. Res. Sci. Teach. **45**(3), 373–394 (2008)
7. Catlin, D., Blamires, M.: The principles of educational robotic applications (ERA): a framework for understanding and developing educational robots and their activities. In: Proceedings for Constructionism 2010 the 12th EuroLogo Conference, Paris, France, 16–20 August 2010
8. Mitnik, R., Nussbaum, M., Soto, A.: An autonomous educational mobile robot mediator. Auton. Robots **25**(4), 367–382 (2008)
9. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012). Lindh & Holgersson (2007)
10. Boykin, K.E.: Engineering Math Based Bridge Program for Student Preparation. ICETI (2010)
11. Şahin, Ç., Ipek, H., Ayas, A.: Students' understanding of light concepts primary school: a cross-age study. Asia Pac. Forum Sci. Learn. Teach. **9**(1), 1–19 (2008)
12. Anderson, C.W., Smith, E.L.: Children's Conceptions of Light and Color: Understanding the Role of Unseen Rays. Research Series No. 166 (1986)
13. Grigorovitch, A.: Children's misconceptions and conceptual change in Physics education: the concept of light. J. Adv. Nat. Sci. **1**(1), 34–39 (2014)
14. Neale, D.C., Smith, D., Johnson, V.G.: Implementing conceptual change teaching in primary science. Elementary School J. **91**, 109–131 (1990)
15. Herbart, J.F.: The Science of Education: Its General Principles Deduced from Its Aim and the Aesthetic Revelation of the World, vol. 19. DC Heath & Company (1895)

Robotics Poetry…

Angeliki D. Theodosi(✉) and Ermioni S. Deli

Ralleia Primary Experimental Schools, Faculty of Primary Education,
University of Athens, Athens, Greece
atheo@sch.gr, ermioni.delh@gmail.com

Abstract. The connection between art and technology may for some people be already known, but for the general public is not obvious nor widely spread. Thus, the question "But what kind of relationship may exist between Kavafis and robotics?" wasn't heard as absurd, but justified in a first level, as it is difficult for someone to relate two phenomenally unconnected fields: Robotics and Literature. Even more, Robotics with Kavafis' poetry, whose thematology may be deeply well-timed and eternal, but reaches the reader through people and language from a different era. And all this, taking place in the primary level of education. However, the effort which was made by the Robotics Club, which consists of students of 5th and 6th grade of Ralleia Model Experimental Primary Schools, was exactly towards the connection of art with technology and, in particular, Kavafis' poetry with robotics.

Keywords: Robotics · Art · Workshop · Poetry · Primary education

1 Introduction

The influence of the technological evolution nowadays is rapidly growing in almost all aspects of everyday life. Everyone, youngsters and adults, receive daily a hail of stimuli and information which is difficult to elaborate and handle. Technology has considerably changed the way we perceive life, environment and reality. Even space, time and distance are different from their conventional meanings for all of us and additionally, for the young pupils who are characterized as "digital natives".

Art has already begun the procedure of osmosis with technology since 1922 and after, with Marcel Duchamp's work "Rotary glass plates", which is interactive and the viewer-user is actively involved in its development. Since then, the artistic works-compositions-settlements, are found in everyday, non-artistic places and can have an interactive role; the materials are no longer classical and delicate but even industrial materials, products or procedures are used, as the technological innovations can become a new means of expression and art.

Through experience, youngsters daily observe the robotic heroes of cartoons which have been produced with 3d animation techniques, have been designed by artists who act in a contemporary field of art which uses the latest achievements of computer technology. People with vision and knowledge work in this field, as for example the unforgettable professor of Carnegie Mellon with the subject of Interaction between Humans

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_21

and Computers, Randy Pausch (Pausch 2009) who shared the dream and the vision of his childhood to work in the field of animation in his famous “Last Lecture”. He was glad to see his childhood dream finally come true, as with his team and lab contributed to the production of the graphics of a very famous animation film. Consequently, the vision of a child in such a sensitive age could become a life aim. In this sense, the creation, the imagination and the vision and inspiration of children of this age, can be developed if proper and structured incentives are offered to them.

The project took place in the academic year 2013–2014 in Ralleia Schools as part of the activities of Ralleia Robotics Club in cooperation with Ralleia Arts Club. Robotics Club pupils’ were 22 pupils (13 boys and 9 girls), their ages were between ten and eleven (10–11) years old (5th and 6th graders). The pupils-members of the club had already introduced to robotics with Lego NXT, thus there was a primitive background knowledge of the material. By primitive knowledge we mean than they had completed the construction of a human like robot with Lego NXT using the touch and the ultrasonic sensors.

2 Materials and Methods

At the present project, robotics were used as a vehicle for exploring and combining different fields of knowledge under a constructivist view of learning (Papert 1980; Alimisis 2009). Teachers followed the scaffolding learning approach as they did not provide any prepared solutions to any stage of the project. Learners were encouraged to express themselves and unleash their imagination. They could share experiences and exchange opinions and ideas on the tasks they had to fulfill. This process produces a bit of noise but created also a pleasant and constructionist/constructivist learning environment for the pupils as they were experienced the feeling of being creative while having multiple degrees of freedom. Tutors mainly acted as facilitators and mentors mostly rather than a source of knowledge.

Two Lego NXT 2.0 packages, Lego bricks, and simple materials such as cork, string, plasticine, wire, and aluminum were used. The number of members of the club exceeded the provided resources, better ratio of Lego packages and pupils would be six instead of two packages for 22 pupils.

3 In Action

According to Hellenic school curriculum for primary education, the poems written by Kavafis which are proposed to be taught are “The horses of Achilles”, “Deisis” and “A house with garden”. Among these, the poem “The horses of Achilles” was chosen for the particular project of the robotics club and it was actually a surprise for the pupils themselves their engagement with poetry in a technology oriented club. Nevertheless, everything was settled when the aim of the project was presented. The aim was to meet this great Alexandrian poet and the particular elements of his poetic work, which inspired them to create and to use the poem theme as a stimulus for a robotic creation. The project was scheduled for eight ninety minute lessons.

During the first 90 min session, the presentation of the poet and the study of the poem took place. Pupils were impressed by the fact that the poet chose to write in Greek although he received English education (Daskalopoulos and Stasinopoulou 2002). On the first level, the language of the poet made an impression to the pupils although the particular poem does not include any difficult expressions like those usually appear in other Kavafis' poems. At this point pupils were organized into six groups according to their own preferences. They connected the pieces of knowledge from other subjects of the school curriculum. They read the particular lyrics of rhapsody R of Iliad where the mourning of horses is described. Afterward, they located the elements-features that the poet attaches to the horses, working through worksheets (group activity). The divided those features into two categories, the one concerning the movement and the other concerning the emotions.

During the second 90 min meeting, pupils were asked to imagine and describe how the horses of Achilles would be if they were robotic (individual activity). They described qualities that the horses had which were a mixture of real qualities of robots and qualities given to the horses by the poet. In this way, they were led to meanings such as "Artificial Intelligence", the use of which would be a prerequisite for the horses' perception or even emotions. They also referred to the experiences of the pupils regarding the robotic heroes of the virtual worlds, of animation and even of the robotic toys they used to play with. The intelligence and the emotion which did not exist suddenly becomes the object of study. The laws of robotics (Asimov 1982) which the robots must follow become objects of speculation and study, as the pupils study the horses' properties. As a result, they found out a contradiction of their mission (participation in a war) and the non-human made laws which they had to obey. By locating this contradiction, the need for consistency to specifically defined rules and orders which do not take back one another becomes very clear.

During the third 90 min meeting, and having completed the description of the horses in their written assignments, the students were divided into groups and art workshops took place at the arts classroom. Children studied drawings and artwork (paintings, sculptures and installations) on the horse (Leonardo Da Vinci, Franz Marc, Wassily Kandinsky, Marino Marini, Jannis Kounellis, John Parmakelis) (Chapman 1993). In particular, for the Da Vinci works and techniques they browsed the website "Da Vinci Science Center (DSC)" for contemporary artistic structures with the support of technology ((DSC), n.d.).

Students initially made sketches-draft plans of robotic horses in watercolor with pencil, pastel and charcoal. Next, they were asked to construct draft plans of them with simple materials such as cork, string, plasticine, wire, aluminum. At that part, they had to think creatively and with the materials they had, to achieve as many of the desired qualities as possible. Every kind of material, with its texture and its color, became a different part of the construction of a robotic horse. The shapes, the colors and the texture of every composing part of their models were carefully chosen and had a particular meaning. The pupils unhurriedly and naturally mentioned the various parts, as for example which had defined as sensors, and furthermore which color and which material they have used for each one of them.

The following five meetings were focused on the construction and programming of robotic horses both with simple pieces of assembly as well as with Lego NXT. The constructions took place in the computer lab and the pupils continued working into 4 member groups. Due to lack of resources (two Lego NXT packages for 22 pupils), groups were divided to two (2) mechanics groups with four members each, two (2) programmers groups with four members each, and two (2) reviewers group with three members each. During the construction of NXT horse robot, for each NXT package, programmers and a mechanics group were cooperating. When the mechanics team formed their robotic horse, the programmer's team took action in order to program it and check its stability and functionality. For more than once, the mechanics team had to modify the construction of the robotic horse in order to be stable and functional. The final result was the construction of a robotic horse by using the LEGO MINDSTORMS NXT 2.0 (Fig. 1). The pupils created an original piece of work whose movement was based on the use of touch sensors. They manage to program the horse moving forward. The touch sensors were attached to horse's legs and whenever the left leg was touching the ground the attached touch sensor was pressed the motor which controls the right leg was activated very similar toy to a man-like pace. The form of the construction was such that left no doubt to anybody. They had managed to create their own robotic horse.

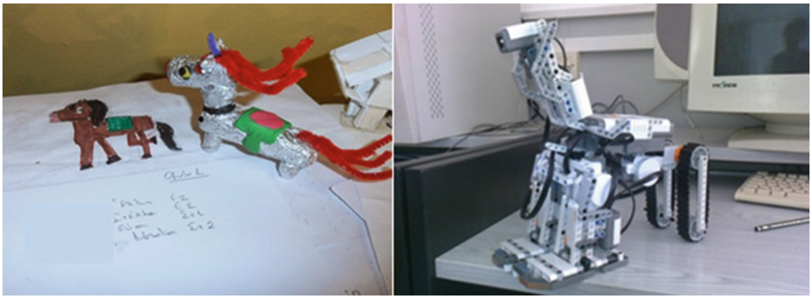


Fig. 1. Robotic horses and lego NXT

The evaluators' teams had to evaluate the final products of the mechanics and programmers teams not only by the aesthetical result but also by testing the horses' stability and functionality under various conditions e.g. movement on a rough ground. Furthermore the evaluator teams had to evaluate the cooperation between the members of the teams and the contribution of each of the member to the final result. The project was concluded with the completion from the pupils of the evaluation sheets given by the tutors.

4 Conclusions

Through this activity of the Robotics Club, the children were invited to combine knowledge and skills from different cognitive fields using their imagination, creativity and critical thought. The young pupils inspired, imagined and created their own robotic horses by discovering and applying knowledge. The evaluation of the functionality of the robotic horses and of the collaboration of the teams by teams of pupils was a totally new experience for all the pupils involved. This experience was considered positive by the pupils. The limited resources were the major problem during the implementation of this project and this was the only complain expressed by the pupils on the evaluation sheets. The discussion, the exchange of experiences and opinions and the collaborative effort helped in the final produced result the work of the team as well as every pupil separately. This was a very encouraging fact that stimulates us to repeat similar projects and evaluate it in multiple aspects. The pupils and tutors themselves considered it as a different and original experience, an attempt which gave them both Ithaca and the beautiful voyage.

References

Pausch, R.: The Last Lecture. Patakis, Athens (2009)

Papert, S.: Mindstorms. Basic Books, New York (1980)

Alimisis, D.: Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods. ASPETE, Patras (2009)

Asimov, I.: Runaround. The Complete Robot, Granada (1982)

Chapman, L.H.: Teaching-art Approaches to Artistic Treatment. Nefeli, Athens (1993)

D.V.S.C. (DSC), www.davincisciencecenter.org

Programming Constructs in Curriculum for Educational Robotics at Lower Secondary School

Michaela Veselovská(✉) and Karolína Mayerová

Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Bratislava, Slovakia
{veselovska,mayerova}@fmph.uniba.sk

Abstract. In this article we present programming constructs taught to lower secondary school pupils in the context of educational robotics curriculum. It consists of eleven activities with robotic kit LEGO WeDo. Students develop many important skills, such as problem solving skills, programming skills, design and ICT competencies. The development of this curriculum was a part of our dissertation research. The students were in 5th and 6th grade (10–12 years old), studying the school subject Informatics. We used qualitative methods of data collection and data analysis. In this article we present programming constructs such as count loop, conditions, variables and parameters that most of the pupils in our classes acquired. These constructs were based on analysis of programming environment for robotic kit LEGO WeDo.

Keywords: Educational robotics · Programming constructs · Lower secondary school · LEGO WeDo

1 Introduction

Results of numbers of publications indicate positive development of knowledge and skills with a use of educational robotics [1]. Students acquired new knowledge during classes on programming, geography and robotics [2–4]. We also agree with the authors [5] statement that the LEGO Education WeDo is an easy-to-use robotics platform that introduces young students to hands-on learning through LEGO bricks and the easiest form of graphical programming software. It is a fun and simple way to get younger students exposed to basic engineering concepts at an early age. In our research we iteratively designed, implemented and verified curriculum for primary and lower secondary school pupils. The curriculum consists of a series of activities for pupils from 6 to 12 years and methodological materials for teachers with robotics kit LEGO WeDo. In this article we focus on eleven activities for lower secondary school pupils. Our curriculum is based on a number of educational principles. In our activities we supported some constructionist ideas [6]: *learning by doing, hands-on activities*; *genuine achievement and own solutions, problem finding*; *hard fun and playful learning*; *learning through designing and creating*; *freedom to make mistakes*; *teamwork, communication, collaboration, sharing work and ideas*. Other principles we followed were “Creative Science for Everyone” [7]: focus on theme, combine art and engineering, support storytelling

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_22

and organize exhibitions. While designing activities for our curriculum we also followed the design principles of learning activity [8] and many others recommendation from Gura [9] and Resnick [10]. Our curriculum introduces one of possible ways for introduction to programming for lower secondary school pupils that is appropriate and interesting. Another learning objective in our curriculum is the development of computational thinking skills through educational robotics. This curriculum covers some topics prescribed in the National educational program for Informatics in lower secondary school in Slovakia (further abbreviated as NEP) that (among other topics) defines which programming constructs should be taught in compulsory education.

2 Methodology

In our dissertation research we had one main question: *What should be specific objectives, content and form of curriculum for educational robotics for 10–12 years old pupils in compulsory school subject Informatics?* In our research and during the development of our curriculum for educational robotics we used qualitative methods of data collection and data analysis [11], including observation (field notes, transcriptions and drawings), focus groups, audio-visual materials (pictures, photographs and recorded videos of pupils' products). Our research strategy was the design based research [12]. We conducted our research in an ordinary lower secondary school in a small town Stupava in Slovakia and our curriculum was designed for ordinary students. Our curriculum was taught by a teacher during school subject Informatics in her classes. There were also two researchers, which were collecting data. We tested our curriculum in four cycles during four years with 92 pupils (49 boys and 43 girls) aged between 10 and 12. In the first iteration of our design based research we worked with 3 groups of pupils separately (14 pupils = 7 boys and 7 girls; 11 pupils = 7 boys and 4 girls; 10 pupils = 4 boys and 6 girls). In the second iteration we worked with 2 groups of pupils separately (11 pupils = 7 boys and 4 girls; 11 pupils = 6 boys and 5 girls). In the third iteration we worked with 2 groups of pupils separately (10 pupils = 7 boys and 3 girls; 12 pupils = 7 boys and 5 girls). In the fourth and last iteration we worked with 1 group of 13 pupils = 4 boys and 9 girls. During the implementation of our curriculum pupils worked mostly in pairs of two boys or two girls, with an occasional pair of a boy and a girl. After every activity from our curriculum we analysed pupils' products such as created robotic models, programs to control them, worksheets and recorded videos. From this data we identified which programming constructs were acquired by most of the pupils. We also found out a lot of about pupils' misconceptions and also which programming constructs they acquired with little or no problems.

3 Programing Constructs in Our Curriculum

Our curriculum consists of eleven activities. Each activity takes 45 min, which is one school hour. In this chapter we present themes for each activity with programming constructs. Every activity includes programming constructs from previous activities. These programming constructs are in accordance with NEP in Slovakia. The first activity does not present any programming constructs – it is an introduction to the realm of robotics. In the last three activities we do not specify the constructs because the pupils select themselves which of the constructs they have learned are appropriate for the task they are working on.

1. **What is a robot? –** Familiarization with the term of robot through a discussion.
2. **Making a dream toy car –** Examining commands for motor control and playing sounds.
3. **Even cottage at robotic foot can dance –** Numerical parameter; Counted loop (and infinite repeat).
4. **My first personal assistant –** Creation of own program.
5. **Matthew's hovercraft from the future –** Sensor parameter that can acquire numerical values; Sensor parameter that can acquire 2 states (sensor can/cannot notice a change); Creation of own program with given commands.
6. **Martina's notion of lost shipping vehicle –** Text parameter; Sensor parameter that can acquire numerical values; Sensor parameter that can acquire two states (sensor can/cannot notice a change); Creation of own program.
7. **The creation of the best helper in the world –** Creation of own program with given commands.
8. **The movement of modern turtle –** Sensor parameter; Creation of own program with given commands.
9. **We propose a fairy tale landscape –** Creation of description of own program.
10. **We are building a fairy tale landscape –** Creation of own program according to own written instruction.
11. **Fairy tale for others –** Evaluation of own work.

Here is an example of one task from activity 5, where pupils were exploring and describing behaviour of robotic model controlled with program in Fig. 1.

Fig. 1. One of the programs which pupils were exploring in activity 5

4 Conclusions

One of the aims of our research was to develop curriculum for ordinary pupils (not just for the gifted ones) at the lower secondary school level. Most of the pupils managed to get acquainted with all programing constructs we introduced. During the development of our curriculum there were also situations where pupils had problems with understanding some of the programing constructs (for example in activity 5 the most frequent misunderstanding was with combination of command *waiting* and parameter of motion sensor). We describe some of the identified mistakes in pupils' work in [13].

Acknowledgments. Implementation of this article has been financially supported by grant UK/403/2016.

References

1. McDonald, S., Howell, J.: Watching, creating and achieving: creative technologies as a conduit for learning in the early years. Br. J. Educ. Technol. **43**(4), 641–651 (2012)
2. Barker, B.S., Ansorge, J.: Robotics as means to increase achievement scores in an informal learning environment. J. Res. Technol. Educ. **39**(3), 229–243 (2007)
3. Nugent, G., et al.: The effect of 4-H robotics and geospatial technologies on science, technology, engineering, and mathematics learning and attitudes. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 447–452. AACE, Vienna (2008)
4. Nugent, G., et al.: The use of digital manipulatives in k-12: robotics, GPS/GIS and programming. In: FIE 2009 Proceedings of the 39th IEEE International Conference on Frontiers in Education Conference, pp. 1–6 (2009)
5. Scaradozzi, D., Sorbi, L., Pedale, A., Valzano, M., Vergine, C.: Teaching robotics at the primary school: an innovative approach. Procedia Soc. Behav. Sci. **174**, 3838–3846 (2015)
6. Papert, S.: The Eight Big Ideas of the Constructionist Learning Laboratory. Unpublished Internal Document, South Portland, Maine (1999)
7. Rusk, N., et al.: New pathways into robotics: strategies for broadening participation. J. Sci. Educ. Technol. **17**(1), 59–69 (2008)
8. Pasch, M.: Teaching as Decision Making: Instructional Practices for the Successful Teacherv. Addison Wesley Publishing Company, Reading (1991)
9. Gura, M.: Getting Started with LEGO Robotics: A Guide for K-12 Educators. International Society for Technology in Education, United States of America (2011)
10. Kafai, Y., Resnick, M.: Constructionism in Practice. Lawrence Erlbaum Associate, Mahwah (1996)
11. Creswell, J.W.: Educational Research: Planning, Conducting, and Evaluating Quantitative. Addison-Wesley, Upper Saddle River (2002)
12. Wang, F., Hannafin, M.J.: Design-based and technology-enhanced learning environments. ETR&D **53**(4), 5–23 (2005)
13. Veselovská, M., Mayerová, K.: Programming with motion sensor using LEGO WeDo at Lower secondary school. Int. J. Inf. Commun. Technol. Educ. **4**(3), 40–52 (2015). Ostrava: Pedagogical Faculty, University of Ostrava

Intensive Robotics Education Approach in the Form of a Summer Camp

Anton Yudin[1(✉)], Anatolii Vozhdaev[1], Dmitriy Sukhotskiy[2], Maria Salmina[3], Tatiana Sukhotskaya[4], and Vladimir Sukhotskiy[4]

[1] Bauman Moscow State Technical University, Moscow, Russia
skycluster@gmail.com
[2] Stankin Moscow State Technological University (Stankin MSTU), Moscow, Russia
[3] Lomonosov Moscow State University, Moscow, Russia
[4] State Budget Vocational and Educational Institution "Vorobyovi Gori", Moscow, Russia
http://www.bearobot.org

Abstract. The paper summarizes approaches and results of an intensive robotics educational programme carried out by the authors in the form of a summer camp. The paper proves that using digital fabrication machines for building a robot in a team for competitions is an outstanding tool of technical, engineering education. Results show high students' interest and educational results of the realized 2-week intensive course.

Keywords: Educational robotics · Intensive education · Digital fabrication · Human capability development

1 Introduction and General Idea

The maker movement amazes ordinary people around the world by showing creativity of enthusiasts who build gadgets for fun. It might seem childish from the first glance but when one thinks of an affordable 3D-printer as being the result of such work the point might switch to potential technological breakthroughs hiding in the passionate community.

Vast potential of the growing maker culture could benefit the whole society, but some steps are needed to be taken before the quantity becomes quality. While digital fabrication laboratories grow in numbers as fast as the maker movement itself as they provide the actual means for the artists, it is also important not to forget that knowledge is as important as creativity when we speak about technology.

Common technical education is not very responsive to the cutting edge results, so it might take time to import the best practices of the maker movement and possibilities of digital fabrication to prepare even better engineers. This paper presents another step towards this goal.

The paper concentrates on the very early results (this text was prepared while still in the camp and concluded within several days after it finished) of a summer

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_23

camp organized by a group of educators near Moscow, Russia in July 2016. The prior successful experience of organizing technically oriented education activities using competitive approach [1] was employed as the basis for the camp's idea. The camp itself could be described as a mix of introduction lectures, uncommon after school activity environment, digital fabrication equipment, robotic competition and intensive schedule.

While robotic competition puts tasks for participants and motivates to win the friendly design race, tutors and teachers are there to help learn during this process. Most of the learning is therefore hidden when seen from the point of the common "lesson-like" school approach.

The result of the camp – a unique robotic project, is achieved through practical work and cooperation between the participants themselves and the "transparent" educational staff.

2 Student Participants

The authors belong to a group of educators who proposed the initial idea and worked on the core aspects of the educational programme for the camp [2,3]. This work was then adapted to the needs of the universities interested in finding better prospective students among the schoolchildren and teenagers.

The result of the joint effort was a selection process for secondary school students having another 1 or 2 years before entering a university. The selection step for the camp was regulated by universities and was carried out in the form of mathematics/physics/programming written competition, which is common as part of admission process to many higher education institutions.

The typical written competition attracted around 500 participants. 60 of them were selected upon their results. Most of the selected students being well educated in mathematics, physics and programming didn't have prior experience in robotics or any engineering practical work.

The author group was responsible for the camp's general educational framework (lectures, practical know-hows, guidance) and for competition organization (preparation of the rules, referees and competition conduction).

3 Schedule

The camp took place in an ordinary summer vacation centre for children. The already well established means of the centre were used to do the basic day-to-day planning. The centre provided meals, sports and recreational activities.

The camp duration of 14 days was also the result of a predefined and nonflexible duration of the summer vacation centre's typical session. The given time was then divided into 3 periods: "arrival and getting acquainted", "qualification phase" and "competition phase". The first period is self-explaining, as most of the participants didn't know each other before. The last two will be explained later. At this time one can think of the qualification phase as of a more guided educational programme and the competition phase to be more of a DIY (do-it-yourself) style when guidance is very limited to provoke cooperation within the participants' community.

4 Means and Equipment

The equipment used during the camp is presented in Fig. 1: 1 - laser cutter; 2 - 3D-printers; 3 - drilling machines; 4 - soldering equipment, oscilloscope, signal generator, magnetometer; 5 - plastic bending machine; 6 - a starting kit given to each teams' disposal on day 3.

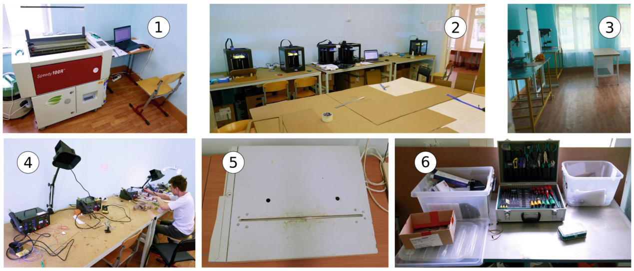


Fig. 1. The equipment.

The standard starting kit for each team contained such components for robot building as: 2 geared motors for movement, 2 cheap actuators, 2 wheels, 1 caster wheel, a number of switches and buttons for robot's remote control, 1 power supply (5 V and 12 V output), a terminal block, 2 Arduino boards, a motor driver board, a relay board, 3 servo motors, a number of distance and line sensors.

Other materials were common to all teams. For example, for laser cutting there were plywood, acrylic glass, cardboard and fibreboard (orgalite).

Teams were not allowed to use any other materials (besides those available in the workshop), nor were they allowed to add components to the standard starting kit. If they ended up with a burned Arduino board they were not given a spare one.

5 The Competition

Robotic competition is the core of the presented educational approach. It motivates students to learn by themselves more than in any other environment. It proves to be true for long-term courses [4] based on such competitions as Eurobot and also for such intensive study short-term courses [5] as the camp being described.

The "qualification" phase of the competition was introduced to fill in the gaps of practical engineering knowledge for most of the participants. This phase

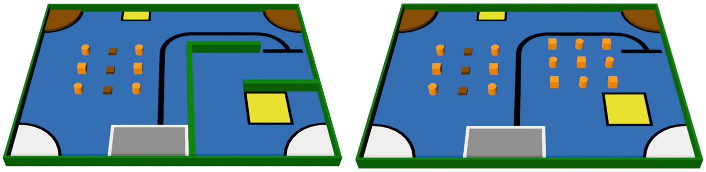


Fig. 2. The playing field for the qualification phase.

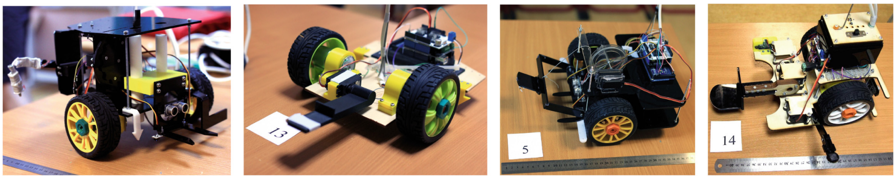


Fig. 3. Examples of robots built during the camp.

was guided by the experienced teachers and 6 tasks of different difficulty were completed one by one in no special order by the students. Teams are free to rebuild and redesign their robots when passing qualification tasks. Work on the tasks in this phase "prepared" the participants for the second "do-it-yourself" competition phase.

In the end of the qualification phase the ratings are used to put teams on a match table for the competition phase. While most of the tasks looked same in competition the robots' mechanics had to be adapted and changed to be able to solve all of the tasks during a timed match. Competition phase's tasks are solved by a single robot controlled with a remote.

The playing field used in the camp's competition phase is shown in Fig. 2 (qualification setup on the left, competition setup on the right).

Finally the rating is composed of points for qualification, points from the jury and points for winning play-off matches in the competition.

6 Conclusion

The paper summarizes the authors' experience in conducting an intensive robotics education summer camp based on using CNC machines for building a robot in a team for competitions.

The results of the educational programme are outstanding as all of the participants, mostly having no prior experience in robot building, succeeded in 2 weeks time to finish and present a unique robotic project able to solve qualification and competition tasks (for 4 out of 15 resulting robots see Fig. 3).

The teaching experience gained during the camp proves the idea of merging digital fabrication and robotics competition to give enormous possibilities for educators to teach and for participants to learn. Thus the work is actually a proof of well-established approach to technical, engineering education.

Further analysis and precisions of the educational programme itself are planned to be published as well. Authors also plan to integrate previous work [6] in future programmes for even better results.

References

1. Yudin, A., Sukhotskiy, D., Salmina, M.: Practical mechatronics: training for mobile robot competition. In: Dessimoz, J.-D., Balogh, R., Obdzralek, D. (eds.) Proceedings of the 6th International Conference on Robotics in Education, RiE 2015, HESSO.HEIG-VD, Yverdon-les-Bains, Switzerland, 20–23 May 2015, pp. 94–99. Roboptics Editions, Cheseaux-Noreaz, Switzerland (2016). ISBN: 978-2-9700629-5-0
2. Sukhotskiy, D., Yudin, A.: Startup robotics course for elementary school. In: Obdržálek, D., Gottscheber, A. (eds.) EUROBOT 2010. CCIS, vol. 156, pp. 141–148. Springer, Heidelberg (2011). doi:10.1007/978-3-642-27272-1_13
3. Yudin, A., Kolesnikov, M., Vlasov, A., Salmina, M.: Project oriented approach in educational robotics: from robotic competition to practical appliance. In: Merdan, M., Lepuschitz, W., Koppensteiner, G., Balogh, R. (eds.) Robotics in Education. AISC, vol. 457, pp. 83–94. Springer, Switzerland (2017). doi:10.1007/978-3-319-42975-5_8
4. Salmina, M., Kuznetsov, V., Poduraev, Y., Yudin, A., Vlasov, A., Sukhotskiy, V., Tsibulin, Y.: Continuous engineering education based on mechatronics and digital fabrication. In: Dessimoz, J.-D., Balogh, R., Obdzralek, D. (eds.) Proceedings of the 6th International Conference on Robotics in Education, RiE 2015, HESSO.HEIG-VD, Yverdon-les-Bains, Switzerland, 20–23 May 2015, pp. 56–57. Roboptics Editions, Cheseaux-Noreaz, Switzerland (2016). ISBN: 978-2-9700629-5-0
5. Yudin, A., Salmina, M., Sukhotskiy, V., Dessimoz, J.-D.: Mechatronics practice in education step by step, workshop on mobile robotics. In: Proceedings of 47th International Symposium on Robotics in the Era of Digitalisation, ISR 2016, Munich, Germany, 21–22 June, 2016. VDE VERLAG, Germany (2016). ISBN: 978-3-8007-4231-8
6. Vlasov, A., Yudin, A.: Distributed control system in mobile robot application: general approach, realization and usage. In: Obdržálek, D., Gottscheber, A. (eds.) EUROBOT 2010. CCIS, vol. 156, pp. 180–192. Springer, Heidelberg (2011). doi:10.1007/978-3-642-27272-1_16

Anthropomorphic Robots and Human Meaning Makers in Education

Karolina Zawieska(✉) and Agnieszka Sprońska

Industrial Institute for Automation and Measurements PIAP, Warsaw, Poland
{kzawieska,aspronska}@piap.pl

Abstract. This paper provides a theoretical discussion of the use of anthropomorphic social robots in education. In particular, this work addresses the use of human-like robots in the role of meaning disruptors. While social robots are often used to imitate human roles to a varying degree, be it teachers, assistants or peers, the underlying assumption here is that anthropomorphic robots are creations that may be assigned entirely new roles. Also, this paper argues that rather than aim at the close resemblance of human characteristics in the robot's form and behaviour, anthropomorphic robots may deliberately exploit the divergence between the robot's characteristics and performance and the human frame of reference. This is how both teachers and school students will become active meaning makers and an increased interest in robotics will be maintained over time. The main subject of reflection is the difference between the human and human-like.

Keywords: Meaning-making · Anthropomorphism · Educational robotics

1 Introduction

There has been a wide consensus on using a degree of anthropomorphism in social robots to facilitate human-robot interaction [7]. Such thinking applies also to educational robots that include a variety of platforms, from toy-like social robots to humanoids [13]. The use of the anthropomorphic forms and behaviours implies also assigning human-like roles to robots. This is particularly true in non-technical education, where robots are used as tutors or peers [12]. Also, the resemblance of humanness is present to the entire contexts of use, since anthropomorphic robots are often expected to fit into the existing human physical and social contexts. Thus, rather than bring new elements, educational robotics often reinforces the existing practices as well as it uses the human frame of reference we are all familiar with.

The question about motivation and methods for creating human-like robots continues to pose challenges to the robotics and Human-Robot Interaction (HRI) researchers. For the purposes of this work, it is important to emphasise that anthropomorphic robots are by definition supposed to resemble human characteristics in a given social environment (the very term "anthropomorphism" derives from the Greek "anthropos" and "morphe", i.e. "man" and "form" respectively [7]). A discrepancy between the robot design and the human frame of reference, including a mismatch between the robot's abilities and

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9_24

users' expectations, are generally viewed as serious obstacles to the successful human-interaction and the integration of social robots into our societies [7, 9]. Excessive expectations have also been viewed as an issue in educational robotics [3], and the need to reproduce human-like behaviours in educational robots has been expressed [12].

It has been argued that most of technologies, robotics included, recently used in schools simply reinforce the old ways of teaching and learning and fails to inspire critical thinking and creativity, among other skills [1]. In line with the previous discussion [13], this work proposes to deliberately exploit the anthropomorphism-related divergences, and use them as a means to foster the human ability to make meanings and engage with educational robots over an extended period of time.

2 Anthropomorphism: Meanings Made

This research follows the classical version of symbolic interactionism developed by George H. Mead [10], i.e. one of the main frameworks in social science. According to symbolic interactionists, people do not react "automatically" towards external or internal stimuli that act upon them but they actively engage in the process of interpretation. Thus, people ascribe meanings to things they deal with and act based on the meanings those things have for them. The meanings are shared within specific socio-cultural contexts and the act of constructing meanings occurs in the course of social interaction [5, 10]. Following this logic, anthropomorphisation is viewed here as an inherently interpretive process: While the robot design has an important role in generating the anthropomorphic effect, it is only in the eyes of the human observer that the robot becomes human-like. Thus, people actively ascribe meanings to the robot's appearance and behaviour, rather than merely respond to the anthropomorphic cues. This is particularly true for such concepts as "humanness" and "human-likeness" that are highly subjective and culture-dependent and they defy any definite definition. In this sense, robot users are meaning makers who need to consciously reflect on the anthropomorphic robot's roles and characteristics as well as their own role in the process of human-robot interaction.

Such a perspective is coherent with the constructivist and socio-constructivist paradigms that have been widely used in education, with meanings being its central concept. However, with the advent of new technologies and socio-economic changes there has also been a need to develop new educational paradigms that go beyond constructivism [6]. This is why this paper proposes to use the anthropomorphic robots as tools that not only encourage the human ability for interpretation but also challenge such an ability and the existing meanings.

3 Anthropomorphic Robots: Meanings Disrupted

When describing humans as capable of constructing meanings, Mead did not argue human conduct is always reflective and conscious [4]. According to Mead reflective consciousness arises when a given object calls conflicting tendencies of action, i.e. contact experiences do not match people's expectations [4]. This is because such situations first require people to evaluate their initial expectations and then to construct new meanings, if needed, to resolve a mismatch. In Mead's words, "As long as action with respect to objects proceeds uninterruptedly, we are unaware of the meaning or content of these objects. When, however, an object calls out conflicting tendencies of action, we are 'thrown back upon an analysis of (our) spontaneous acts and therefore upon the objects which get their content from them" ([11] quoted in [4, p. 249]). In education, the examples of similar approaches include educational strategies that in order to foster creativity in students rely on "divergent thinking", including in the programming education [2] and with the use of LEGO robotics [8].

In general, the use of robots in education, from the robotic kits to humanoid platforms, is known for the positive impact it has on the students' interest and engagement, at least for a short period of time. One could argue that when using anthropomorphic robots, the reason for an increased interest lies not so much in the novelty of robots as in an illusion of the human-like life they give. Such an illusion relies not only on the similarities between the human and human-like but also divergences (which is why it remains an illusion and not reality). The ability of anthropomorphic robots to disrupt the existing meanings, along with the constructive role of the conflicting tendencies, leads to the argument that anthropomorphic robots can be successfully used in education to foster the ability to make meanings and maintain an increased interest in robots over time. However, one should be careful not to generate excessive people's discomfort or provoke rejection of the robotics technologies due to the Uncanny Valley effect or the incoherent robot's design and functionalities. The further research include translating such an approach into the robot design and specific educational strategies and programmes.

4 Human Meaning Makers

When using the anthropomorphic robots as meaning disruptors, it is important to emphasise that the main focus is not so much on the robots as the human actors who interact with the robot, to a varying degree. Firstly, the principle of divergence applies not only to the robot but also human actors: Just as the anthropomorphic robot is not-quite-human, teachers and students also engage in the interactions and roles that may go far beyond the practices they normally follow when interacting with other people in a given educational context. For example, the teachers may be expected to talk to the robots, and hence engage in pretend play in the class. Also, the meanings ascribed to anthropomorphic robots are constructed and shared within larger social groups that include teachers, students, parents and the schools staff. In other words, the perception and image of the robot is never an outcome of only the robot design or the robot's interactions with single users but also a result of interactions between different social actors (for example, a parent may never come to see the robot in the classroom but he or she may significantly influence children's expectations towards the robot while discussing such a topic at home). In this sense, the robot brings new elements not only to social interactions but also to the entire social networks. Another element that may challenge the existing practices is a degree of adaptation needed to use the robot in a given physical and social context (e.g. the classroom settings may require changes to allow the robot to easily move around; also, the teachers who participate in the study may need other teachers' help to modify their regular teaching agenda and the corresponding activities). Last but not least, the meanings ascribed to anthropomorphic robots tend to change over time. This may include both an increased as well as decreased degree of anthropomorphisation (e.g. a person may become emotionally attached to the robot or quite the opposite, notice the robot's limitations that makes it far from being human-like). While such a tendency has often been seen as an obstacle in HRI research, it is seen as an asset here: Rather than search for fix roles and applications for robots in education, with the goal to achieve the close resemblance to humans, one may use anthropomorphic robots as a catalyst for the ever ongoing process of meaning-making. In this sense, educational robotics may go far beyond the classroom and foster life-long learning skills.

5 Conclusions

This paper discussed the potential of the anthropomorphic social robots for being used as tools that foster the human ability to make meanings. This is due to possibility to exploit divergences between the human and human-like as well as the known and the new that come with the anthropomorphic robots. Such an approach opens the door to development of new educational paradigms and further reflection on the distinctively human traits – or lack of.

Acknowledgements. This paper is based on the results of Phase III of the National Programme "Safety and working conditions improvement", funded in the years 2014 – 2016 in the area of research and development works by the Ministry of Science and Higher Education/The National Centre for Research and Development. The Programme coordinator: Central Institute for Labour Protection - National Research Institute.

References

1. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013)
2. An, J., Park, N.: The effect of EPFL programming based on CPS model for enhancing elementary school students' creativity. In: Park, J.H., et al. (eds.) Embedded and Multimedia Computing Technology and Service: EMC 2012, pp. 237–244. Springer, Heidelberg (2012)
3. Baxter, P., et al. The wider supportive role of social robots in the classroom for teachers. In: 1st International Workshop on Educational Robotics at the International Conference on Social Robotics. Paris, France (2015)
4. Biesta, G.J.J.: Mead, intersubjectivity, and education: the early writings. Stud. Philos. Educ. **17**(2–3), 73–99 (1998)
5. Blumer, H.: Symbolic Interactionism: Perspective and Method. University of California Press, Berkeley (1986)
6. Brown, T.H.: Beyond constructivism: exploring future learning paradigms. Educ. Today **2**(2), 1–11 (2005)

7. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots: concepts, design, and applications. Technical report CMU-RI-TR-02–29, vol. 1, pp. 16–17 (2002)
8. Goodwin, C.B., Gustavson, M.: Science, creativity and the real world: lessons learned from the U.S. homeschool community. In: Demetrikopoulos, M.K., Pecore, J.L. (eds.) Interplay of Creativity and Giftedness in Science, pp. 257–266. Springer, Heidelberg (2016)
9. MacDorman, K.F.: Subjective ratings of robot video clips for human likeness, familiarity, and eeriness: An exploration of the uncanny valley. In: Proceedings of the ICCS/CogSci-2006 Long Symposium: Toward Social Mechanisms of Android Science (2006)
10. Mead, G.H.: Mind, Self, and Society: From the Standpoint of a Social Behaviorist. University Of Chicago Press, Chicago (1934)
11. Mead, G.H.: Suggestions toward a theory of the philosophical disciplines. The Philos. Rev. **9**(1), 1–17 (1900)
12. Mubin, O., et al.: A review of the applicability of robots in education. J. Technol. Educ. Learn. **1**, 1–7 (2013)
13. Zawieska, K., Duffy, B.R.: The social construction of creativity in educational robotics. In: Szewczyk, R., Zieliński, C., Kaliczyńska, M. (eds.) Progress in Automation, Robotics and Measuring Techniques, pp. 329–338. Springer, Heidelberg (2015)

Author Index

D. Alimisis et al. (eds.), *Educational Robotics in the Makers Era*,
Advances in Intelligent Systems and Computing 560, DOI 10.1007/978-3-319-55553-9