EAI Endorsed Transactions
on Game Based Learning

Editorial ICST.ORG

Programming Games for Logical Thinking

H. Tsalapatas[1]

[1]University of Thessaly, Department of Computer Engineering, Telecommunications, and Networks, Iasonos 10, 38333 Volos, Greece

Abstract

Analytical thinking is a transversal skill that helps learners synthesize knowledge across subject areas; from mathematics, science, and technology to critical reading, critical examination, and evaluation of lessons. While most would not doubt the importance of analytical capacity in academic settings and its growing demand for the skill in professional environments, school curricula do not comprehensively address its development. As a result, the responsibility for structuring related learning activities falls to teachers. This work examines learning paradigms that can be integrated into mathematics and science school education for developing logical thinking through game-based exercises based on programming. The proposed learning design promotes structured algorithmic mindsets, is based on inclusive universal logic present in all cultures, and promotes constructivism educational approaches encouraging learners to drive knowledge building by composing past and emerging experiences.

Keywords: analytical thinking, algorithmic thinking, learning-to-learn, knowledge construction, game-based learning, programming.

Received on 27 August 2012; accepted on 27 November 2012; published on 20 March 2013

doi: 10.4108/trans.gbl.01-06.2013.e4

1. Introduction

Analytical and critical thinking are among the transversal learning capabilities that help learners excel in all subject areas. Other learning-to-learn skills that positively affect learner performance in broad academic themes are creative thinking, entrepreneurial thinking, the ability to work effectively in groups as well as independently, and the ability to work across cultures. These soft skills contribute to learner capacity to drive knowledge development and to synthesize new knowledge based on past experiences as advocated by constructivism theories introduced by Papert [5] [6] [3] and extended in practical learning environments by Resnick and the Lifelong Kindergarten Group.

Papert's and Resnick's pedagogical theories suggest that learners already possess knowledge when entering the classroom. The teacher must facilitate the development of new knowledge through activities that engage learners in synthesis and step-wise scaffolding of information.

Papert introduced the concept of "microworlds" [5] [6] [7] which are contained virtual environments that simulate a simplified version of the real world; microworlds present learners with adequate information for solving a logical problem while omitting unnecessary "noise". Microworlds were initially used for geometry learning but are now applied in wide subjects.

Resnick's work in the Lifelong Kindergarten Group [8] [3], as the name implies, draws inspiration from the ways young children learn, by doing and experimenting, through games, and through exploration; for example, by painting with their fingers or by building castles that enables them to understand physics laws. Resnick further suggests that individuals of all ages, including adults, can learn in the same hands-on manner by synthesizing knowledge as opposed to memorising information transferred by an instructor.

This line of thought that promotes hands-on exploration is encapsulated in the Scratch programming environment for children [4], developed by the Lifelong Kindergarten Group, in which learners can snap together

programming commands from a comprehensive toolkit to develop a program. Another programming environment, albeit designed for middle school children, is Alice, which is developed by Carnegie Mellon University [12]. The main advantage of these digital frameworks is their design for deployment specifically addressing children; their ease of use, graphical interfaces, and adaptability makes them effective learning tools for exposing learners in primary and secondary school to programming and to synthesis.

Programming is typically not integrated into formal primary education curricula, which focus mostly on understanding the functionality and purpose of computer systems. Only in upper secondary education do learners get exposed to programming, mostly through pseudo-code that follows closely programming language-like syntax. However, teachers in primary and lower secondary education do use the above mentioned graphical environments in classrooms in the context of their own educational initiatives, share projects in broad communities, and build on each other's projects [18]. Teachers see educational benefits for their learners stemming from these tools towards analytical and critical thinking development.

School curricula mostly do not set quantified educational goals in relation to building analytical capacity or other soft, learning-to-learn skills. This is probably a consequence of the purpose of school curricula, which is to define concrete knowledge on specific subjects that a learner must possess upon completing a particular educational level. In practice, teachers integrate activities towards analytical skill development in mathematics and science education through problem solving. However, the creativity of teachers and learners is evident in activities towards building analytical thinking skills that go well beyond the above traditional routes; they include critical reading, critical examination, evaluation of lessons by learners, and more.

It becomes evident that teachers understand the importance of developing analytical thinking skills and make efforts towards this end for the benefit of their students. However, many point to the lack of educational tools, especially in digital form, that are age appropriate and may be used to support the instructional process [16].

The ability to think analytically is not only desirable in academics. The PISA survey, which documents what learners know and can do every 3 years with a focus on basic academic competencies such as mathematics, science, and reading, includes an analysis by Levy and Murnane that demonstrates a shift in the desired skills for employment from 1960 to 2002 from basic cognitive to analytic [2] [1]. In addition, the Europe 2020 strategy points to the projected increase over the next decade of the demand for highly skilled professionals in the context of a knowledge economy [9]; Europe 2020 includes as a key target the improvement of the performance of secondary education learners in subjects where analytical thinking activities are practiced, namely mathematics and science.

This paper introduces educational frameworks that take the programming towards synthesis skill development paradigm a step further by introducing game-based approaches for developing logical thinking. The proposed approach eases learners into algorithmic thinking by exposing them to logical puzzles. Learners develop intuition on potential solutions through semi-structured exploration and implement visual programs based on past knowledge and emerging experiences.

2. Using programming to teach logic

Programming is an inherently analytical process. The most important aspect of programming is developing an algorithm; in other words a precise, accurate solution to a problem that can be executed in a specific amount of time by following well defined steps.

Reaching a correct algorithm involves structured thinking. By the time a learner is able to write a program that implements an algorithm the learner has already mentally solved the problem. Introducing a series of instructions that explains to someone else, in this case the computer that does not possess intelligence but only executes commands, how to implement an algorithm implies that the learner fully understands the underlying solution and is able to elaborate on it and to document it.

Algorithmic thinking is rooted in universal logic that is present in all cultures. An educational approach that deploys algorithms towards building logical thinking is, for this reason, naturally inclusive. It can be adapted to diverse educational environments engaging learners with varying learning needs. It can be personalized in terms of content, intensity, and focus to meet individual educational goals through the appropriate selection of exercises.

Analysis in the context of programming practices involves identifying implementation goals, understanding the resources available for building a solution, breaking down a problem into smaller parts, solving those, and synthesizing a solution based on the smaller components. This activity promotes the development wide problem-solving skills. Functional programming helps learners identify in an analytical manner repeating sequences of actions that can be replaced by a function in a code segment making the solution more coherent and easier to read and understand. Using programming constructs such as loops helps develop more elegant solutions to a given exercise. On the other hand, object-oriented programming introduces an organization to thinking patterns that is useful for addressing high levels of complexity. Learners in ICT ultimately get exposed to these programming paradigms; they benefit by developing high level problem solving skills.

Algorithmic thinking itself plays a central role in programming education. Learners become familiar with algorithmic patterns that can solve broad categories of logical exercises. Examples of such well known algorithms include divide-and-conquer and reduce-and-conquer, which involve breaking down a problem into smaller instances and tackling those. As a result of the wide applicability of the above thinking frameworks, ICT students become familiar with them early on and refer to them for synthesizing solutions throughout their professional careers.

Algorithms, programming, and logic are interrelated concepts. The following sections analyse pedagogical frameworks that apply programming and algorithmic ideas to wide audiences of young learners aiming to expose them early on to sound thinking mind sets that help them perform well academically and, later on, professionally.

3. Programming as a serious game

The above discussion highlights what the ICT and mathematics communities well understand: that algorithmic thinking forms a foundation for problem solving capacity. However, students get exposed to algorithmic thinking only optionally in upper secondary education in the context of curricula electives that prepare them for engineering or science higher education. Once enrolled in university education, students study deeper algorithmic thinking and logic only in the context of programs leading to engineering or mathematics degrees. As a result, the benefits of algorithmic thinking approaches for building problem solving capacity are lost to wide student audiences. Many of them would consider engaging with algorithms as a difficult task, not related to life out of school, and probably a little boring.

Given the educational and professional benefits of sound logical thinking that results from working with algorithms, designing learning frameworks that are based on algorithmic thinking and target all students, as opposed to the mathematically inclined ones, may enhance the analytical thinking capacity of the average learner.

Serious games that deploy programming concepts may be a motivational learning tool towards building critical and structural minds. Most like to solve a puzzle and children even more so. An educational approach that deploys carefully selected logical puzzles has the potential of exposing learners to algorithmic thinking at a young age, well before they would normally be engaged with related learning activities in upper secondary or tertiary education. Using programming as a central activity in a serious game may promote solving skills without using intimidating scientific or ICT terminology.

As an example, consider the following story line aimed for primary education learners: “Santa, with the help of his elves, packs presents to deliver to all children in the world. Being absent-minded, he puts his dirty socks into one of the boxes. Can you help Santa find the box with the dirty socks in the pile of all presents?”

This problem does not have a trivial answer. A solution involves weighing packages against each other to discover the heavier one among otherwise identical boxes.

Many, even at higher educational levels, would probably first attempt an implementation approach such as the following: while there are still packages in the pile the learner picks two, weighs them against each other, and if they have equal weight discards them; the learner repeats the process until he/she discovers a heavier package. This is a correct solution. It exposes students to important analytical thinking as well as programming concepts; namely identifying a repeating sequence of execution steps and nesting the sequence in a loop to avoid a very long solution.

However, a more efficient solution exists in which the number of execution steps is minimized. It involves weighing half of the packages in the pile against the other half, identifying the heavier half, and repeating the process until only one package with higher weight remains. This is a divide-and-conquer algorithm; it exposes learners to advanced programming concepts, including recursion and optimization. Using this exercise in a gaming environment where learners are asked to graphically synthesize a solution to the given puzzle enables them to engage in challenging analytical thinking activities in the context of gameplay.

4. A game-based learning framework that uses programming to build analytical thinking

The above learning design ideas have been applied towards introducing a proof-of-concept game-based virtual learning environment that deploys visual programming to help build analytical thinking skills among young learners in primary education, named cMinds [10] [11] [13] [14] [15] [16] [17]. The gaming aspect of the environment builds on the natural curiosity of children and acts as a motivational feature that promotes engagement with the learning process. The proposed learning framework does not primarily focus on simply building digital competence; rather, the educational objective is to help develop logical minds. As a result, the related environment leans heavily on visual programming. Learners structure solutions by dragging and dropping graphically depicted actions and programming constructs from a toolkit into a programming zone. The environment all but eliminates programming-like syntax maintaining only visual representations of programming structures such as loops, conditionals, and switches and using images for actions and sequential commands. This design choice aims to focus learner attention on logic as opposed to syntactical conventions of programming languages.

The game-based environment exposes learners to logical puzzles. While the puzzles are age appropriate, their complexity is not insignificant. As a consequence, the environment provides learners with the opportunity to develop a basic intuition on potential solutions by experimenting, similarly to drawing sketches on scratch paper. Following learning-by-doing paradigms, the framework includes a semi-structured hands-on exploration area which, similarly to microworlds, presents necessary information for shaping a solution by dragging and dropping while avoiding non-required data.

Once learners have solved the puzzle in a game-like manner through experimentation, they structure a precise visual program that “tells the computer how to solve the problem”; in other words, it consists of an accurate sequence of visually presented commands. The process of building a program is iterative; learners make step-wise process by introducing new commands. The environment allows them

to view the effects of their actions through animated visualizations of the execution of their program. Visual feedback reinforces learner understanding of the solution and provides input for identifying and correcting errors.

Learner interaction with the learning environment does not end with the completion of a correct program. Learners have the option of comparing their solution with an optimal implementation. In the context of this work, optimality refers to the least number of execution steps or, in other words, the fastest algorithm. Learners can see side-by-side the visual code of their own solution and the optimal one; they can further see the visual executions of both programs and critically compare their efficiency. This process enables learners to reflect upon alternative solutions further enhancing their analytical thinking capacity.

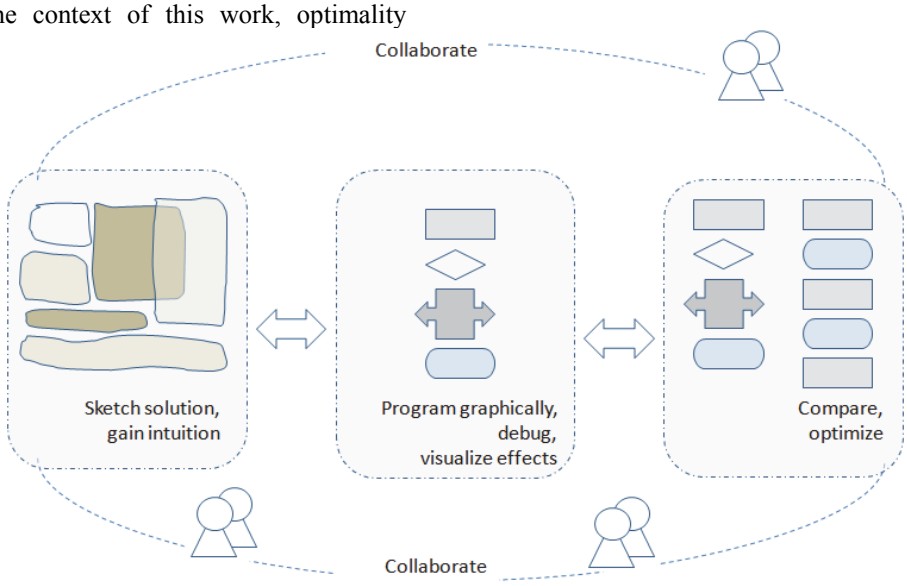

Figure 1. A game-based learning framework for building analytical thinking skills through experimentation, solution structuring, and optimization.

The proof-of-concept virtual environment is demonstrated in the following figure that depicts the graphical presentation of the Santa learning puzzle, including the toolkit of commands on the bottom, the code developed by the learner on the left, and the visual execution of the program on the centre and right.

Figure 2. The Santa Claus puzzle in the cMinds learning environment [13] [17].

5. Educational benefits

The proposed game-based learning framework for analytical skill development, which is validated through the cMinds implementation, offers value-adding educational benefits to learners.

The hands-on, semi-structured exploration provides an experimentation ground in which learners are encouraged to think creatively. It encourages thinking out of the box and playing with a variety of solutions not only for reaching a correct implementation but also for enhancing it through more efficient sequences of commands. It is an important step towards synthesis of a structured answer and it eases learners through game-play into valuable logical exercises.

The framework follows a low entrance high ceiling approach. Most children are eager to play in the hands-on exploration area. By developing intuition on a problem solution they feel more confident to take on the challenge of building a structured program. Through better understanding of the problem at hand the exploration empowers children to engage with more advanced analytical thinking processes. Logical puzzles are structured with several levels of difficulty encouraging learners to solve more complex instances of a puzzle once they have successfully completed an entry level, easier to solve variation.

Graphical programming emphasises analytical thinking through minimal use of programming syntax. It exploits the structured nature of programming towards developing critical thinking and departs from typical design of programming environments for children in which syntax is a significant parameter [10] [17].

Real time visual feedback in the form of animated program execution allows learners to make a connection between cause and effect of their actions in a vivid manner. In correlation with the iterative solution synthesis process visual feedback helps children learn from past efforts and apply newly developed experience towards improving their solutions. Learners scaffold knowledge step by step gaining confidence with each choice that takes them closer to a correct implementation.

By introducing learners to the concept of optimization the learning framework promotes the exchange of ideas, findings, and solutions. It is designed for deployment in the classroom in the context of wider collaboration and promotes peer learning. It highlights the knowledge learners already possess when walking into the classroom and empowers them to extend it by combining it, synthesizing it, and applying it in new learning scenarios. It encourages children to drive the knowledge building process and to learn from each other in an inclusive environment where entrepreneurial thinking and innovative solutions are positively regarded and rewarded.

6. An evaluation framework

Validating the impact of the proposed game- and programming-based learning framework towards analytical skill development among primary education students involves the application in real-life learning experiments of the proof-of-concept virtual environment that implements the proposed hands-on, learning by doing, explorative educational design. Validation engages representatives of the target group, namely primary education learners and their teachers. Well designed activities are deployed in classrooms in the context of collaborative learning aiming to provide objective feedback on a number of issues related to the relevance, effectiveness, and ease of use of the proposed methodologies digital tools. For example:

- Are learners motivated to engage in analytical thinking activities that involve synthesizing solutions to logical puzzles?
- Do learners engage longer with the learning process and specifically with learning activities focusing on building analytical thinking as a result of the proposed learning framework?
- Does hands-on exploration and solution synthesis in a virtual environment enable learners to more effectively analyze information and build knowledge than off-line approaches? Are learners able to produce more innovative and out-of-the-box solutions to problems?
- Are learner attitudes towards subjects related to analytical thinking such as mathematics and science positively affected?
- Is the integration of digital environments into classroom practices natural, easy, and effective?
- Are teachers eager to use the tools?
- Do teachers perceive educational value for their students?
- Do teachers believe that learning sessions become richer?

Given the young age of learners and the fact that the answers to several of the above questions are related to perceptions, experiences, and attitudes a qualitative evaluation approach offers the potential of producing true insight on the effects of the proposed methodologies and tools on learning experiences of the target group. An evaluation process that involves participatory observation as well as interviews with teachers and learners is used. This strategy allows for open ended feedback by users, both children and instructors, on the deployment of game-based pedagogies and programming in learning. It further provides the opportunity to experience directly the reactions of the users and gain insight on benefits and potential challenges related to the integration of the proposed learning approaches into existing educational practices.

Early evaluation results are encouraging and demonstrate positive reactions. Some key findings related to learner experiences include:

- The game-based approach that revolves around logical puzzles succeeds in engaging learners that are otherwise intimidated by analytics in mathematics and science to engage meaningfully in the learning progress
- The low ceiling entrance with gradual increase of the difficulty of the exercises helps learners build self-confidence resulting in the successful completion of exercises by learners in all complexity levels
- The semi-structured exploration provides a stepping stone towards synthesizing a structured solution that is appreciated by learners
- The virtual environment that includes real-time graphical feedback enables learners to experiment fast with a wide choice of potential solutions enriching thinking pathways; off-line approaches do not offer this speed of experimentation
- Learners is several cases develop innovative, surprising solutions, demonstrating entrepreneurial thinking
- Learners are able to engage in advanced algorithmic thinking, for example divide-and-conquer exercises, as a result of game-play that eases them into complex notions
- Some learners express the desire to become program developers, demonstrating positive attitudes and impressions in relation to programming; this demonstrates that the use of programming in the context of gaming scenarios can introduce naturally young learners to technology education

In relation to teacher perspectives:

- Teachers need a certain level of support given that they themselves are often unfamiliar with programming concepts as a result of their mostly pedagogical background

- Teachers are eager to introduce the learning framework in the classroom; this is a consequence of the focus of the environment on learning puzzles, many of which are already used off-line in classroom activities
- Teachers are very proud of their students being able to progressively produce innovative solutions to logical puzzles
- Teachers point to the enrichment of learning processes through digital tools; this reaction is the result of the scarcity of tools available for educational use in classrooms
- Teachers point to the improvement of math scores as a result of using the proposed learning method

7. Conclusions

This paper presents a pedagogical framework that deploys programming concepts in a game-based environment aiming to enhance analytical thinking capacity among primary education learners. The framework departs from past pedagogical paradigms that expose learners to programming with the primary objective of building digital skills. Programming is used in this work as a means to build structured thinking. The framework deploys semi-structured, game-based experimentation before exposing learners to solution synthesis aiming to build an initial intuition. Optimization is introduced as a means for further enriching analytical thinking through reflection and comparison of solutions. Early evaluation of a proof-of-concept implementation of the proposed learning paradigm demonstrates positive reactions of learners and teachers, enrichment of classroom practices through easy integration, elevated motivation of learners to engage in analytical subjects in school, and gradual enhancement of learner problem-solving capacity.

Acknowledgements

This work has been funded support from the European Commission. This communication reflects the views only of the author and the Commission cannot be held responsible for any use which may be made of the information contained therein. I would further like to acknowledge the teachers that use the proposed methods and tools in the context of pilot learning activities: Argyro Paraskeva, George Metaftsis, Aristea Veremi, Ingegard Hansson, Susi Sundstrom, Hana Pakandlova, and Ioan Konig.

References

[1] What Students Know and Can Do, PISA Survey 2009, on-line at: http://pisa.oecd.org
[2] What Makes a School Successful, PISA Survey 2009, on-line at: http://pisa.oecd.org
[3] Lifelong Kindergarten, on-line at: http://llk.media.mit.edu/index.php
[4] Scratch, on-line at: http://www.scratch.mit.edu/
[5] Papert, S. (1980), Mindstorms: Children, Computers, and Powerful Ideas, New York, Basic Books
[6] Papert, S. (1993), The Children's Machine: Rethinking School in the Age of the Computer. New York. Basic Books
[7] Ackerman, E. (2001), Piaget's constructivism, Papert's constructionism: what's the difference?, online at: http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf
[8] Resnick, M. (2007), All I Really Need to Know (about creative thinking) I Learned (by studying how children learn) in Kindergarten'. Proceedings of the 2007 Conference on Creativity and Cognition, Washington DC, USA (pp. 1-6)
[9] Europe 2020 Strategy, on-line at: http://ec.europa.eu/europe2020/index_en.htm#
[10] The cMinds Project: Teaching Programming towards Building Early Analytical, Structural, and Critical Minds, Comenius Action project 2010-2012, on-line at: http://cminds.org
[11] Tsalapatas, H., Heidmann, O., Alimisi, R., Tsalapatas, S., Florou, C., Houstis, E. (2012) Game-based Learning towards Building Early Analytical Thinking Skills through Visual Programming, EduLearn 2012 Conference
[12] An Educational Program that Teaches Students Computer Programming in a 3D Environment, http://alice.org
[13] Tsalapatas, H., Heidmann, O., Alimisi, R., Florou, C., Tsalapatas, S., Houstis, E. (2012), Supporting Primary School Students in Developing Computational Thinking Skills through the cMinds Learning Suite, INTED Conference, March 5-8, 2012, Valencia, Spain
[14] Tsalapatas, H., Heidmann, O., Alimisi, R., Tsalapatas, S., Florou, C., Houstis, E. (2011), Programming Concepts as a Means of Fostering Analytical Thinking in Primary School Students, EduLearn Conference, July 4-7, 2011, Barcelona, Spain
[15] Tsalapatas, H. (2012), Developing Analytical Thinking Capacity through Serious Games and Programming, Educa On-line Conference, November 28-30, 2012, Berlin, Germany
[16] Tsalapatas, H., Heidmann, O., Tsalapatas, S, Alimisi, R., Florou, C., Houstis, E. (2011), Visual Programming towards the Development of Early Analytical and Critical Thinking, Future of Education Conference, June 16-17, 2011, Florence, Italy
[17] Tsalapatas, H., Heidmann, O., Alimisi, R., Florou, C., Tsalapatas, S., Florou, C., Houstis, E. (2012), Game-based Learning towards Building Early Analytical Thinking Skills through Visual Programming, EduLearn Conference, July 2-4, 2012, Barcelona, Spain
[18] The Scratch Community, on-line at: http://wiki.scratch.mit.edu/wiki/Scratch_Community