Scientific Bulletin of the „Petru Maior" University of Tîrgu Mureş
Vol. 9 (XXVI) no. 1, 2012, ISSN 2285 – 438X (Online), ISSN–L 1841 – 9267

GAME-BASED PROGRAMMING TOWARDS DEVELOPING ALGORITHMIC THINKING SKILLS IN PRIMARY EDUCATION

Hariklia TSALAPATAS[#1], Olivier HEIDMANN[#2], Rene ALIMISI[#3], Elias HOUSTIS[#4]

[#1] *Unversity of Thessaly*
Argonafton & Filellinon, 38221 Volos, Greece
[#1]htsalapa@uth.gr

[#2,3,4] *Centre for Research and Technology Thessaly*
Industrial Area of Volos, 38500 Volos, Greece
[#2]olivier.heidmann@gmail.com, [#3]ralimisi@gmail.com, [#4]enh@uth.gr

ABSTRACT

This paper presents cMinds, a learning intervention that deploys game-based visual programming towards building analytical, computational, and critical thinking skills in primary education. The proposed learning method exploits the structured nature of programming, which is inherently logical and transcends cultural barriers, towards inclusive learning that exposes learners to algorithmic thinking. A visual programming environment, entitled 'cMinds Learning Suite', has been developed aimed for classroom use. Feedback from the deployment of the learning methods and tools in classrooms in several European countries demonstrates elevated learner motivation for engaging in logical learning activities, fostering of creativity and an entrepreneurial spirit, and promotion of problem-solving capacity.

Keywords: analytical thinking, computational thinking, problem-solving skills, algorithm, game-based learning

1. Introduction

We live in a 'Creative Society' where it is important to perform problem-solving skills and logical and critical thinking capacity[1]. Despite the existence of various studies that bring into focus the importance these skills [1], their development early in primary school students' life is not representative of its importance.

Nevertheless Science, Technology, Engineering and Maths (STEM) modules can provide opportunities towards the development of these skills, there is evidence that school curricula do not encourage students' meaningful engagement in STEM practices [3]. Many students consider STEM modules as irrelevant with real life and as a result opportunities for active involvement in analytical, critical, computational and problem-solving thinking practices are being missed. This attitude affects students' employment future and opportunities [3], According to recent studies [3] fewer and fewer students pursue a career in STEM and many students lack skills such as critical thinking, analytical thinking and problem-solving skills. In accordance with NESTA organization UK '*faces massive shortages of people with science, technology engineering and mathematical (STEM) training*' [4]. For this reason teacher training is planned aiming at equipping teachers with the necessary skills to motivate their students towards engagement in the area of STEM [4]. Through STEMNET [5] another attempt is made to bring together teachers, employers, and ambassadors and to inspire young students in science, technology, engineering, and mathematics.

But why it is important to encourage students explore the area of STEM? Interestingly, there is a growing demand for employees in STEM occupations who can efficiently deal with emerging problems using analytical thinking skills, technology, mathematics, and science. Educational requirements and credentials for STEM occupations may vary given the position and the nature of the job but all require the ability to think logically and critically and to deal with problems creatively [6] [7]. On the other hand, the cultivation of these skills encourages students to reflect upon their own way of thinking and to develop themselves as critical thinkers; such a capacity is useful in non STEM areas as well [8].

These concerns and facts bring into focus the need to develop training resources and interventions that can prepare teachers, students, and practitioners in exploring the area of science, technology, engineering, mathematics (STEM) in a meaningful way [7]. cMinds builds on this idea and aims at providing the educational community with free learning tools, teaching and learning resources, and facilities for collaboration towards computational and analytical thinking and problem-solving practices early in primary school student's life. The cultivation of these skills is important and can motivate and prepare students to explore with success the area of STEM, to develop themselves as critical thinkers, and to increase possibilities for perspective hiring.

In the next section, the practice of 'programming' is discussed from a pedagogical point of view. This is done for two reasons: First, 'programming' is a well-known tool closely related to the area of STEM. Second, programming can be seen as a mean for fostering computational and critical thinking early in young learners' life. The third section focuses on the cMinds methodology. The cMinds activities design and the developed cMinds Learning Suite are brought into focus in the fourth and fifth section. The paper concludes with a discussion on the results derived from all the validation sites.

2. Engagement in 'programming', but how?

Formalizing concepts as algorithms is considered to be a practice with great educational potential when compared mainly to traditional ways that focus on the simple comprehension of things [8]-[10]. Research studies relate closely programming to the development of 'analytical, computational thinking and problem-solving skills' which are seen as essential in 21st century [1][2].

As Papert [10] advocates, there is need to abandon the practice of using the computer to *'program the child'* and to encourage students to *'program the computer'*. The regulation of the computer can help students develop computational thinking skills, engage in scientific practices, and gain an understanding of the way the human-made world has been constructed [11].

Kahn [13] taking into account Papert's work [10][12] considers programming as 'a fertile ground' for fostering general thinking skills, such as: 'problem decomposition, component composition, explicit representation, abstraction, debugging, and thinking about thinking'.' Resnick et al (2007, p.62) [8] connects programming with the ability to '*design strategies (such as modularization and iterative design*)'. Beyond all dispute, programming fosters the development of skills that are transferable and can also be applied to 'non-programming domains' [8].

However, students' engagement in programming is not an easy process. Research shows that students encounter difficulties in understanding what an algorithm is and how programming concepts operate [8],[14]-[18]. They often have serious misconceptions that raise confusion and lead to drop outs.

Studies [8],[14]-[18] identify three basic reasons for these failures and drop outs. First, early programming languages were difficult to master raising students' confusion and discouraging their engagement in the programming process (i.e. complex language-syntax) [8],[14]-[17]. Second, the activities (that students are called to program for) are usually overextended from mathematics and do not meet students' interests and experiences [8],[14]-[16]. Thirdly, the inappropriate mental models and the inadequate support of these through the various programming environments can discourage students and lead to drop outs [14],[16],[17].

Many interesting programming learning environments (i.e. Scratch [18], Alice [19], Cruislet [20],[21], ToonTalk [22], Lego Mindstorms [23]-[24]and more) have been developed and are aimed at dealing with the three previously mentioned problem areas. A comprehensive review of these programming learning environments is out of the scope of this paper. Nevertheless, it is worth referring briefly to some outcomes closely related to the use of such tools. Recent studies demonstrate that the related learning interventions arouse students '*interest and help them build knowledge and gain understanding of abstract phenomena*' [11] as well as to gain an insight of the human-made world[11]. Research also advocates in favour of the development of teaching material and resources to support teachers in integrating such learning tools in the class and in designing relevant educational content [26].

Programming in the context of the cMinds project is seen as a vehicle for triggering learning mechanisms towards analytical and critical thinking. cMinds started with the idea that the formation of an algorithm can more meaningful occur if stages of hands-on practices that allow the intuitively approach of the algorithmic solution pre-exist. Trying to foster analytical thinking and problem solving skills the cMinds Learning Suite introduces an environment in which children learn through playful experimentation, decomposing problems (which are classic logical challenges) and composing algorithmic solutions towards building analytical thinking skills.

In the next section the methodology that was followed in brought into focus. A clear link is made between the methodology and the design of the activities that integrated in the cMinds Learning Suite.

3. Methodology

cMinds adopts an end-user centred implementation approach. The project starts by analyzing typical strengths and weaknesses in the area of STEM education and more specifically in the teaching of algorithmic and computational thinking in the Greek, Romanian, Czech and Swedish school curricula. The analysis ensures that the learning intervention will be well in line with the needs of educational community that takes part in the project. Based on the findings of this analysis, the cMinds didactical framework for analytical and computational skill building was developed. cMinds didactical framework follows an active and constructionist approach to learning [10][12] according to which students create solutions exploring the underlying scientific concepts. The cMinds framework brings together collaborative learning practices, explorative approaches to learning and problem-based learning theories. The cMinds methodology encourages an iterative and step-wise development of a solution allowing learners to plan their design strategy, construct their solutions, learn by their mistakes, correct errors, explore multiple solutions and build incrementally their analytical thinking capacity. This process is described below:

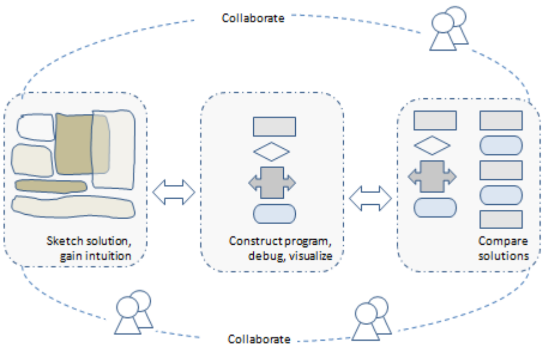

Fig. 1– cMinds methods towards analytical skill building

This theoretical framework is the backbone of the cMinds project and it set a basis whereupon the cMinds visual learning environment and corresponding learning activities were designed. In the following section the cMinds activities design and the underpinning rationale is brought into focus.

4. Activity Design

Three main factors influenced the selection and the design of the cMinds activities. First, an attempt was made to tie basic algorithmic techniques that draw upon problem solving theory to the cMinds learning activities [27]. The three basic algorithmic techniques that were selected are: the brute force, the divide and conquer, and the decrease and conquer one. Each model/technique exposes learners to a different way of working with data. Problems, puzzles, and logical challenges to which a characteristic solution can be developed based on the above algorithms were identified for proof-of-concept implementation that demonstrates the benefits of engaging young learners in algorithmic thinking [27].

From a wider pool of logical puzzles a final short list was structured that included problems solvable with the use of simple serial commands, conditional statements, switches/cases, and while loops [27]. Selecting logical puzzles with age appropriate complexity for the targeted primary education learners was important for challenging but not overwhelming learners.

The activities/problems were further selected among widely known logical puzzles which are often already used in-class for teaching. This selection criteria ensures that both teachers and pupils are already familiar with the problems introduced by the learning environment and can engage meaningfully in the process of decomposing each problem to its components [27].

In a last step the graphical representation of the components of the activities and the programming tools were carefully considered. The process described above led to the selection of the following problems:

1) The tutorial area: The students are encouraged to start their educational journey with cMinds from the tutorial activity. The tutorial aims at explaining basic programming concepts such as serial commands, conditional statements, while loops, and switches/cases. A simple scenario was selected that introduces a story for attracting learner interest while at the same time it maintains a simple environment and presents just enough information for enabling learners to solve the problem at hand. This allows learners to focus on the solution and build confidence on the use of programming concepts in a simplified simulation of the outside world. Specifically, a robot is placed close to an apple tree. Several exercises engage learners to make the robot pick apples with varying degrees of difficulty in what to pick and how; for example to pick all apples, pick only apples that are ripe, throw away rotten apples, etc [28].

2) The 'Math activity': This activity aims at familiarising students with serial commands while helping them execute basic mathematical operations. The activity helps learners develop a general methodology for addressing operations by bringing together several mathematical concepts including estimation, approximation, subtraction, and division. Graphically, a number line graduated to a scale provides students with a spatial representation of mathematical concepts allowing them to visualize the starting conditions and objectives of mathematical exercises. Learners are asked to instruct the robot to move from the starting to the target position on the line with as few steps as possible from pre-defined increments [28].

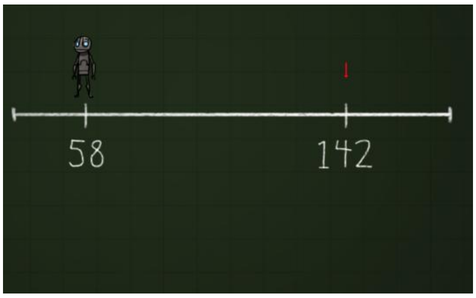

Fig. 2– The Math activity

3) *The 'Santa Claus' Socks'*: This learning exercise draws inspiration from classic puzzles that expose learners to recursive divide-and-conquer and decrease-and-conquer algorithms. The activity further introduces learners to the operation of the While-loop. According to the playful scenario, Santa mistakenly put his dirty socks into one of the otherwise identical gift boxes. Learners must instruct the robot to identify the one heavier box that contains the socks by cleverly weighing the boxes against each other. This puzzle can be solved in more than one ways. A straightforward but suboptimal solution can involves iteratively weighing two boxes at a time. A more efficient solution based on divide-and-conquer principles involves iteratively weighing half of the boxes against the other half. The first solution is a good example of loops while the second is algorithmically sounder. Both implementations are supported through the cMinds Learning Suite [28].

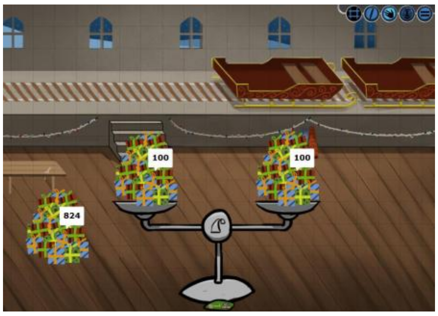

Fig. 3– The Santa Claus activity

4) *The 'Water Jugs'* is a brute-force exercise in which learners instruct the robot to fill a water container with the help of two smaller ones with specific volumes. Through this activity students engage in a logical challenge that promotes mathematical thinking. At the same time they practice simple serial commands [28].

Fig. 4– The Jugs activity

5) *The 'River Crossing'* puzzle is a typical rule-driven exercise. It draws inspiration from the classic puzzle that asks learners to help a wolf, a sheep, and a cabbage cross a river on a single raft based on restrictions on leaving any two of the above unattended on the same river bank. Both this version of the puzzle as well as a more complex one in which a group of grown-ups and children must cross a river taking into account constrains and restrictions have been implemented. Through this activity students are encouraged to build a smart and flexible algorithm using loops and conditional statements [28].

Fig. 5– The River Crossing activity

6) *The 'Friezes Activity'* is a pattern recognition exercise in which learners must discover series of repeating shapes in a sequence. A 'frieze' is constituted of geometrical shapes of many colours. The problem introduces students to the way conditional constructs operate and conditions evaluate to true or false. Each level of difficulty introduces additional friezes raising the difficulty level through more complex sequences [28].

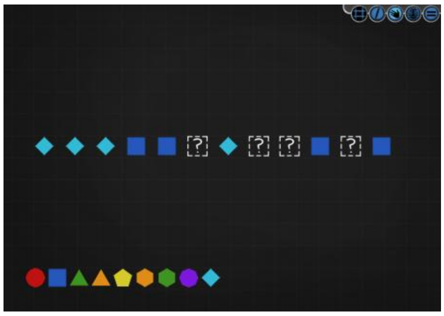

Fig. 6 – The Freezes activity

7) *The 'Eggs Activity'* is a pattern recognition and sorting exercise in which learners categorize objects based on specific characteristics. According to the storyline, learners must instruct the robot to pick an egg from a box, recognize the pattern drawn on the egg, and put it into a matching destination box. The problem introduces students to the way cases/switches and conditional constructs operate [28].

Fig. 7 – The Easter eggs activity

In the next section, the cMinds Virtual Environment is presented. The environment brings together all the learning activities but does not stop there. It also integrates a range of supporting resources and facilities for exploration, collaboration, and communication.

5. The cMinds Learning Environment

From a pedagogical point of view, the cMinds learning suite takes a top-down approach that guides children through the step-wise solution of a problem from the beginning to the end. To achieve this objective, the pilot application starts by introducing children to basic programming concepts through the tutorial area. Subsequently, students are encouraged to select an activity from the logical puzzle set and to explore its solution initially through hands-on exploration and subsequently through visual programming [28].

Hands-on practice is possible in a semi-structured exploration area. Learners are encouraged to solve a problem through a classical drag and drop interface. Each hands-on area is different and specific to the selected activity [28]. For example, in the Friezes activity learners must first recognize patterns in a sequence themselves before developing a program that instructs the robot to solve the puzzle generically and automatically; for the Santa activity, learners may experiment by weighing different number of boxes against each other with the purpose of understanding the divide-and-conquer algorithm before developing a program that finds the heavier box through minimum execution steps.

Once a solution is intuitively synthesized in the hands on area, learners are encouraged to move to the 'robot phase'. This area is named after the star character, the robot, which is a human-like representation of a computer. Learners are encouraged to build a program that instructs the robot (i.e. computer) to precisely solve the problem. Learners can address the orders to the robot by dragging and dropping instructions from a toolbox into a script area. In order to avoid exposing students to a complex programming language-like syntax, commands are graphically represented by images and icons. Learners are provided with the opportunity to focus on the development of an algorithmic solution without being distracted by issues closely related with syntactical errors. The execution of the script can be visualised in the 'effect zone'. Learners may modify the script upon visualization to correct errors or to implement a more efficient solution [28].

The learning activity concludes with the 'comparison zone' where students are may compare their solution to 'optimal' algorithms. This process facilitates the reinforcement of learning outcomes in the context of class collaboration. Learners are encouraged to discuss their choices/ solutions and to reflect upon their own thinking and the thinking of others.

From a technical point of view, the cMinds Learning Environment has been coded in Flash, as the Flash browser plug-in is already installed in more than 97% of computers and is supported widely by popular desktop operating systems. Furthermore, as the project's code can be compiled as an Adobe AIR application, the learning environment could also be deployed natively on any platform that doesn't support the Adobe Flash Player, e.g. an iPad.

The cMinds Learning Suite does not stand alone; supporting material targeting teachers in the form of learning sheets, scientific links, good practice videos, and collaboration facilities is introduced to support the teaching process and to facilitate smooth integration of cMinds outcomes into classrooms, enhancing teachers' skills, and encouraging them to take an active role in designing similar learning interventions tailored to the specific needs of their pupils.

6. Evaluation design and on-going feedback

Validation activities engaging learners and teachers are carried out in primary in Greece, Sweden, Romania, and the Czech Republic. The activities aim to develop field input on the relevance, acceptance, and effectiveness of the proposed methodologies and tools towards building analytical thinking skills in primary education. Furthermore, they aim to evaluate the whether the proposed learning interventions can be effectively integrated into existing school practices aiming to enhance classroom learning experiences.

The validation activities follow a 'teaching experiment' approach and occur in four stages. The first stage involves tool introduction; the teachers demonstrate the cMinds Learning Suite to the students aiming to familiarize them with the multiple functionalities, services, and tools. In a second stage students are encouraged to practically work on the activities integrated in the cMinds Learning Suite. This stage may vary in duration and can be repeated several times. For example, it may last one school hour, or 45 minutes, or may be repeated over several sessions in a period that covers several months. Teachers act as facilitators of the learning process and as an observer. They document in writing in an on-going manner perceptions and findings with the objective of identifying benefits of and recommending enhancements to the cMinds learning intervention. Interviews and informal discussions with the teachers provide additional input. As a last step, reports, observations, and statements are analysed. The results affect system and application design and implementation cycles and are integrated into the software tools. New versions of the software as well as observations on pedagogical deployment trigger new evaluation cycles.

The validation activities are integrated into classroom practices in the context of a blended learning design that includes class instruction and virtual tool deployment. Evaluation feedback obtained by teachers over a period of 15 months is encouraging and brings into focus three main issues:

First, teachers identify a clear link between the cMinds learning methodologies and digital tools and the development of analytical and critical thinking skills through their integration into classroom activities. The following statement exemplifies this fact:

‘Learners’ critical thinking is the most relevant outcome of this application. In addition, students have to think, foresee, try out and think again to accomplish the task. Putting on these ‘thinking hats’ in order to create a working algorithm is the most successful approach of re-shaping and improving the way children think critically’ (Ioan, teacher at the Economic College Transylvania, Romania)

Second, the cMinds Learning Suite can be integrated easily into mathematics and ICT modules enhancing learning experiences; by reinforcing students’ interest cMinds contributes to learner motivation and engagement in STEM. As Ioan stated: *‘The cMinds virtual learning suite can be integrated especially into computer and math lessons but depending on the teacher’ s imagination these methodologies and tools can be integrated in science activities as well. The limit is your imagination after all’*.

Last, apart from the direct relation of cMinds to STEM courses the focus of the learning games on creative and entrepreneurial thinking exemplified the relation of the proposed explorative knowledge building methodologies to wider subject areas. These may range from language learning to humanities and even arts. For example, the cMinds activities inspired in one primary school dramatization of the storylines by learners. Students took roles that were closely associated with the cMinds puzzles, for example the robot, the sheep, the wolf, and the cabbage, and acted out the learning scenarios by presenting in action the solution to a given problem. The dramatization of the execution by the computer of a program allowed learners to gain deeper understanding on algorithmic design. Teachers facilitated and recorded this session.

Fig. 7 –Role playing for River Crossing activity

Fig. 8– Dramatising the tutorial area

7. Discussion

The cMinds technology-enhanced learning intervention aims to provide students and teachers in primary education with rich, technology-enhanced opportunities to develop problem-solving skills and analytical thinking capacity. Teachers can easily recognize the educational benefits offered by the cMinds Learning Suite in developing creative thinking mindsets.

The organization of cMinds learning sessions into hands-on exploration, structured solution synthesis, and comparison of solutions supports the easy introduction of learners into math and science activities and gradually increases the intensity of the exercises aiming at challenging learners to increase their knowledge. This low-entrance high-ceiling approach engages learners, increases their confidence, and motivates their long-term participation in the learning process.

The transfer from the initial active exploration in the hands-on zone for developing intuition on problem objectives and potential solutions to the structured environment of the robot programming zone where learners must develop a precise solution script is a logical step in which learners need instructional support. Early learner efforts in visual programming may naturally result into unfinished or erroneous scripts that learners develop and correct in a step-wise iterative manner that exploits the cMinds

Learning Suite visual feedback. This is not a surprising finding as literature demonstrates that the introduction of novices to programming is not straightforward; rather, it takes time [14] and many students struggle to understand how an algorithm operates [14]. However, the fact that learners gradually enhance their programming skills through cMinds and develop often out-of-the-box solutions demonstrates that the environment facilitates the development of critical and creative thinking.

The provision of support by the teachers plays a crucial and significant role as it can trigger mechanisms towards the elimination of students' confusion and misunderstandings [26]. Teachers can facilitate the learning process and support students in overcoming cognitive obstacles and successfully engage in computational and analytical thinking practices. Teachers need themselves support in the understanding of programming concepts and principles towards explaining this knowledge to their pupils. Supporting teachers is particularly important when their background is not on ICT.

To help teachers build capabilities and skills instructional guidance was provided in meetings with teachers either online or face-to-face. These practices had a great impact on teachers' attitude. Teachers advocated that after the 'supportive meetings' they felt more confident and able not only to integrate the tools in the class but also to train their colleagues in using cMinds in the classroom. Furthermore, instructional support material was developed in the form of videos, learning sheets, and a wiki [27].

Learning sheets describe in a step-wise manner end-to-end learning activities for classroom deployment that are built around the cMinds learning games. They introduce learning objectives, demonstrate the use of the basic features of the tools towards meeting these objectives, and suggest collaboration activities for reinforcing knowledge [27].

A **good practices video gallery** includes "how to" instructions on the features of the cMinds Virtual Learning Suite with a focus on educational use. The videos start by describing the functionality of the tutorial area, move on to the problem analysis visualization zones, and close with demonstrations of the collection of proof-of-concept learning puzzles. They further include a visual glossary of programming concepts demonstrated through the tutorial learning activities. Finally, they provide good practice examples of the use of the cMinds tools and methods in the classroom in the context of evaluation sessions.

A **wiki** is available to teachers for collaboration and information sharing purposes. It provides background information that motivated the introduction of the cMinds project; it describes learning needs in science and technology, game-based pedagogical methodologies, and technology-enhanced learning activities; it provides scientific links on active, explorative, and game-based learning; it provides references to cMinds supporting material; and more. While the content targets teachers, learners may also use the wiki towards exploring meaningfully programming concepts and computational thinking [27].

8. Conclusion

This paper presented cMinds, a learning intervention which provides students and teachers with opportunities to reflect upon logic challenges and to develop problem-solving skills and analytical thinking capacity through technology-enhanced, game-based learning. The cMinds Learning Suite invites students to analyze problems, to identify core components of the solution, to critically snap together the different components, to optimize their solutions, and to reflect upon their thinking. On-going evaluation feedback from the education community over a 15 month period is positive. It demonstrates that cMinds contributes achieves its objectives of cultivating the analytical and transversal learning skills that young learners need to succeed in today's and tomorrow's society and knowledge economy.

Acknowledgement

This project has been funded with support from the European Commission. This publication 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.

We would like to further recognize the school teachers and researchers that deploy or support activities in primary education classrooms: Spyros Tsalapatas, Chryssa Florou, Argyro Parasekeva, Georgios Metaftsis, Aristea Veremi, Sussi Sundstrom, Hana Pakandlova, Ivana Konvalinkova, Ingegard Hansson, Sven-Olof Larsson, Ioan Konig, Maria Laslo, Ana-Maria Moldovan, Ana Carmen Tot and John Stav.

References

[1] Resnick, M. (2007a), '*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)

[2] Resnick, M. (2007b). *Sowing the Seeds for a More Creative Society*. International Society for Technology in Education, 18-22

[3] Kuenzi, J. (2008) *' CRS Report for Congress: Science, Technology, Engineering, and Mathematics (STEM) Education: Background, Federal Policy, and Legislative Action'* available online at: http://www.fas.org/sgp/crs/misc/RL33434.pdf (last retrieved on 6th April, 2012)

[4] NESTA Objectives, available online at: http://www.nesta.org.uk/areas_of_work/public_

services_lab/past_projects_public_services_lab/stem_education
[5] Stemnet Website : http://www.stemnet.org.uk/
[6] TED: The Editor's Desk (2007) retrieved online at: http://www.bls.gov/opub/ted/2007/jun/wk4/art04.htm & http://www.ctstemjobs.org/resources/CT_STEM_Employer_skills_survey_summary_report.pdf .
[7] A New Impetus for European Cooperation in Vocational Education and Training to Support the Europe 2020 Strategy, communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee, and the Committee of the Regions, online at: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2010:0296:FIN:EN:PDF
[8] Resnick et al (2009) *Programming for all* available online at http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf
[9] Gal-Ezer, J. and Harel, D. (1998). 'What (Else) Should CS Educators Know?'. Communications of the ACM, 41 (9), pp. 77–84.
[10] Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*, New York, Basic Books.
[11] Bers, M.U., Ponte, I., Juelich, C., Viera, A. & Schenker, J. (2002). *Teachers as Designers: Integrating Robotics in Early Childhood Education*. In-formation Technology in Childhood Education Annual, 2002(1), 123-145. AACE.
[12] Papert, S. (1993). *The Children's Machine: Rethinking School in the Age of the Computer*. New York. Basic Books. 1993.
[13] Kahn, K. (2004). *ToonTalk-Steps Towards Ideal Computer-Based Learning Enviroments. In: Mario Tokoro and Luc Steels* (eds), A learning Zone of One's Own: Sharing Representations and Flow in Collaborative Learning Enviroments, Ios Pr Inc.
[14] Guzdial, M. (2003). *Programming Environments for Novices.* Available a thttp://coweb.cc.gatech.edu/mediaComp-plan/uploads/37/novice-envs2.pdf
[15] Alimisi R. and Winters N., (2010). '*Programming Playfully for a Real-life Problem: Conditional Statements on the Stage of Scratch'*, in J. Clayson and I. Kalas (eds.) *Proceedings for Constructionism 2010*, 16-20 August, 2010, Paris, France, ISBN 978-80-89186-65-5 (Proc), ISBN 978-80-89186-66-2 (CD)
[16] Doukakis D., Tsaganou G., Grigoriadou M. (2007). *Using Animated Interactive Analogies in Teaching Basic Programming Concepts and Structures*. Proceedings of the ACM Conference on the State of: Informatics Education Europe II, Thessaloniki, Greece, pp. 257-265.
[17] Soloway, E. and Spohrer, J. (eds) (1989). *Studying the Novice Programmer*. Hillsdale, NJ, Lawrence Erlbaum Associates.
[18] Maloney, J., Peppler, K., Kafai, Y., Resnick, M., and Rusk, N. (2008). *Programming by Choice: Urban Youth Learning Programming with Scratch*. Available at: http://web.media.mit.edu/~mres/papers/sigcse-08.pdf
[19] Cooper, S., Dann, W. and Pausch, R.(n.d). Teaching Objects-first In Introductory Computer Science. Available online at http://www.alice.org/publications/TeachingObjectsfirstInIntroductoryComputerScience.pdf
[20] Alexopoulou, E., Kynigos, C., Markopoulos, C. (2007). Changing a Half-baked 3D NavigationalGame'. *EuroLogo Conference*, Bratislava. Available at http://www.di.unito.it/~barbara/MicRobot/AttiEuroLogo2007
[21] Alexopoulou, E. and Kynigos, C. (2008), Half-baked Games as a Context for Understanding Conditional Constructs' (in greek). Available at: http://www.etpe.gr/files/proceedings/21/1223368001_DIDINFO08_71_80.pdf
[22] Kahn, K. (1995). ToonTalk™ – *An Animated Programming Environment for Children*. Available at: www.toontalk.com
[23] Martin, F. (1996). *Kids Learning Engineering Science Using LEGO and The Programmable Brick*. Unpublished paper; presented at the 1996 annual meeting of the American Educational Research Association. Available at http://www.media.mit.edu/˜fredm/
[24] Alimisis, D., Moro, M., Arlegui, J., Frangou, S. and Papanikolaou, K. (2007). *Robotics & Constructivism in Education: the TERECoP project*. In Proceedings of EuroLogo 2007. Edited by I. Kalas. Bratislava, August. pp. 39.
[25] Alimisis, D., Arlegui, J., Fava, N., Frangou, S., Ionita, S., Menegatti, E., Monfalcon, S., Moro, M., Papanikolaou, K., Pina, A., (2010). *Introducing Robotics to Teachers and Schools: Experiences from the TERECop Project'*, Constructionism 2010, Paris, France, 16–20 August 2010 available online at: http://www.etlab.eu/images/arthra/alimisis2010.pdf
[26] Laurillard, D., (2008). *The Teacher as Action Researcher: Using technology to Capture Pedagogic Form*, Studies in Higher Education, 33(2), 139-154.
[27] Tsalapatas, H., Heidmann, O., Alimisi, R., Florou, C., Houstis, E. (2012), *Supporting primary school students in developing analytical and computational thinking skills through the cminds learning suite.* In the proceedings of INTED Conference, 4-7 March, Valencia.
[28] cMinds website: www.cminds.org