# Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education

Michail Kalogiannakis
*University of Crete, Greece*

Stamatios Papadakis
*University of Crete, Greece*

A volume in the Advances in Early Childhood and K-12 Education (AECKE) Book Series

Published in the United States of America by
IGI Global
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com



Library of Congress Cataloging-in-Publication Data

Names: Kalogiannakis, Michail, 1969- editor. | Papadakis, Stamatios, 1974- editor.
Title: Handbook of research on tools for teaching computational thinking in P-12 education / Michail Kalogiannakis and Stamatios Papadakis, editors.
Description: Hershey, PA : Information Science Reference, 2020. | Includes bibliographical references and index. | Summary: “This book examines the implementation of computational thinking into school curriculum in order to develop creative problem-solving skills and to build a computational identity which will allow for future STEM growth”-- Provided by publisher.
Identifiers: LCCN 2019060058 (print) | LCCN 2019060059 (ebook) | ISBN 9781799845768 (hardcover) | ISBN 9781799845775 (ebook)
Subjects: LCSH: Computer science--Study and teaching. | Computer programming--Study and teaching. | Critical thinking--Study and teaching.
Classification: LCC QA76.27 .H3548 2020 (print) | LCC QA76.27 (ebook) | DDC 004.071--dc23
LC record available at https://lccn.loc.gov/2019060058
LC ebook record available at https://lccn.loc.gov/2019060059

This book is published in the IGI Global book series Advances in Early Childhood and K-12 Education (AECKE) (ISSN: 2329-5929; eISSN: 2329-5937)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

# Advances in Early Childhood and K-12 Education (AECKE) Book Series

Jared Keengwe
University of North Dakota, USA

ISSN:2329-5929
EISSN:2329-5937

## Mission

Early childhood and K-12 education is always evolving as new methods and tools are developed through which to shape the minds of today's youth. Globally, educational approaches vary allowing for new discussions on the best methods to not only educate, but also measure and analyze the learning process as well as an individual's intellectual development. New research in these fields is necessary to improve the current state of education and ensure that future generations are presented with quality learning opportunities.

The **Advances in Early Childhood and K-12 Education (AECKE)** series aims to present the latest research on trends, pedagogies, tools, and methodologies regarding all facets of early childhood and K-12 education.

## Coverage

- Performance Assessment
- Reading and Writing
- Pedagogy
- Head Start and Pre-K Programs
- Special Education
- Diverse Learners
- Curriculum Development
- STEM Education
- Standardized Testing
- Early Childhood Education

IGI Global is currently accepting manuscripts for publication within this series. To submit a proposal for a volume in this series, please contact our Acquisition Editors at Acquisitions@igi-global.com or visit: http://www.igi-global.com/publish/.

The Advances in Early Childhood and K-12 Education (AECKE) Book Series (ISSN 2329-5929) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-early-childhood-education/76699. Postmaster: Send all address changes to above address. 

## Titles in this Series

*For a list of additional titles in this series, please visit:*
*https://www.igi-global.com/book-series/advances-early-childhood-education/76699*

***Applying Flow Theory to Strings Education in P-12 and Community Schools Emerging Research and Opprtunities***
Taichi Akutsu (Okayama Prefectural University, Japan & Seisa University, Japan)
Information Science Reference • © 2020 • 151pp • H/C (ISBN: 9781799833598) • US $155.00

***Cases on Emotionally Responsive Teaching and Mentoring***
Ann M. Ellsworth (Montana State University, USA)
Information Science Reference • © 2020 • 296pp • H/C (ISBN: 9781799829713) • US $195.00

***Empowering Multiculturalism and Peacebuilding in Schools***
Soner Polat (Kocaeli University, Turkey) and Gizem Günçavdı (Kocaeli University, Turkey)
Information Science Reference • © 2020 • 378pp • H/C (ISBN: 9781799828273) • US $195.00

***Learning Challenges for Culturally and Linguistically Diverse (CLD) Students With Disabilities***
Soraya Fallah (California State University, Northridge, USA) Bronte Reynolds (California Lutheran University, USA) and Wendy Murawski (California State University, Northridge, USA)
Information Science Reference • © 2020 • 337pp • H/C (ISBN: 9781799820697) • US $195.00

***Inclusive Theory and Practice in Special Education***
Henriette van Rensburg (University of Southern Queensland, Australia) and Shirley O'Neill (University of Southern Queensland, Australia)
Information Science Reference • © 2020 • 425pp • H/C (ISBN: 9781799829010) • US $195.00

***Cases on Teaching Sexuality Education to Individuals With Autism***
Peggy Schaefer Whitby (University of Arkansas, USA)
Information Science Reference • © 2020 • 289pp • H/C (ISBN: 9781799829874) • US $185.00

***Handbook of Research on Software for Gifted and Talented School Activities in K-12 Classrooms***
Shigeru Ikuta (Otsuma Women's University, Japan)
Information Science Reference • © 2020 • 448pp • H/C (ISBN: 9781799814009) • US $245.00

***Incorporating LGBTQ+ Identities in K-12 Curriculum and Policy***
April Sanders (Spring Hill College, USA) Laura Isbell (Texas A&M University, Commerce, USA) and Kathryn Dixon (Texas A&M University, Commerce, USA)
Information Science Reference • © 2020 • 381pp • H/C (ISBN: 9781799814047) • US $185.00



IGI Global
PUBLISHER of TIMELY KNOWLEDGE

701 East Chocolate Avenue, Hershey, PA 17033, USA
Tel: 717-533-8845 x100 • Fax: 717-533-8661
E-Mail: cust@igi-global.com • www.igi-global.com

# Editorial Advisory Board

Nicholas Zaranis, *Department of Education, University of Crete, Greece*

# List of Reviewers

Alexandros Balatsoukas, *University of Groningen, The Netherlands*
Courtney K. Blackwell, *Feinberg School of Medicine, Northwestern University, USA*
Nathalia da Cruz Alves, *Federal University of Santa Catarina, Brazil*
Muhibuddin Fadhli, *Universitas Muhammadiyah Ponorogo, Indonesia*
Andreas Giannakoulas, *University of Macedonia, Greece*
Jean C. R. Hauck, *Federal University of Santa Catarina, Brazil*
Alexandros G. Kapaniaris, *Democritus University of Thrace, Greece*
George Koutromanos, *National and Kapodistrian University of Athens, Greece*
Kleopatra Nikolopoulou, *National and Kapodistrian University of Athens, Greece*
Vassileios Orfanakis, *Hellenic Mediterranean University, Greece*
Emmanouil Poulakis, *University of Thessaly, Greece*
Mustafa Saritepeci, *Necmettin Erbakan University, Turkey*
Dimitrios Stamovlasis, *Aristotle University of Thessaloniki, Greece*
Anastasios Theodoropoulos, *University of Peloponnese, Greece*
Tandra Lea Tyler-Wood, *University of North Texas, USA*
Christiane Gresse von Wangenheim, *Federal University of Santa Catarina, Brazil*
Stelios Xinogalos, *University of Macedonia, Greece*
Hatice Yildiz Durak, *Bartin University, Turkey*
Georgios Zacharis, *Aristotle University of Thessaloniki, Greece*

# List of Contributors


**Alves, Nathalia da Cruz** / *Federal University of Santa Catarina, Brazil* ........................................... 76
**Angelopoulos, Panagiotis** / *Ministry of Education, Greece* ..................................................... 392, 425
**Balatsoukas, Alexandros** / *University of Groningen, The Netherlands* .................................. 392, 425
**Borgatto, Adriano F.** / *Federal University of Santa Catarina, Brazil* ................................................ 76
**Chatzopoulos, Avraam** / *University of West Attica, Greece* ........................................................... 238
**Durak, Aykut** / *Bartin University, Turkey* .......................................................................................... 54
**Giannakoulas, Andreas** / *Department of Applied Informatics, University of Macedonia, Greece* ..... 1
**Gresse von Wangenheim, Christiane** / *Federal University of Santa Catarina, Brazil* ..................... 76
**Hauck, Jean C. R.** / *Federal University of Santa Catarina, Brazil* .................................................... 76
**Kalogiannakis, Michail** / *University of Crete, Greece* ............................................ 104, 238, 289, 339
**Kanaki, Kalliopi** / *University of Crete, Greece* ................................................................................. 104
**Kapaniaris, Alexandros G.** / *Democritus University of Thrace, Greece* ................................. 460, 484
**Lepouras, Georgios** / *University of Peloponnese, Greece* ............................................................... 159
**Lombardi, Gaia** / *Istituto Comprensivo Statale Via dei Salici, Legnano, Italy* ............................... 184
**Molla, Marina** / *9th Primary School of Komotini, Greece* .............................................................. 356
**Niros, Antonios D.** / *Experimental High School of Mytilene of the University of the Aegean, Greece* ........................................................................................................................................ 140
**Nistor, Adina** / *European Schoolnet, Belgium* ................................................................................. 425
**Papachristos, Dimitrios** / *University of West Attica, Greece* ......................................................... 238
**Papadakis, Stamatios** / *Department of Preschool Education, University of Crete, Greece* ............ 31, 289, 339
**Papoutsidakis, Michail** / *University of West Attica, Greece* ........................................................... 238
**Politis, Panagiotis** / *University of Thessaly, Greece* ........................................................................ 200
**Poulakis, Emmanouil** / *University of Thessaly, Greece* .................................................................. 200
**Psaridou, Vasiliki** / *Minority School of Dokos, Greece* .................................................................. 356
**Psycharis, Sarantos** / *School of Pedagogical and Technological Education, Greece* ...................... 238
**Saritepeci, Mustafa** / *Necmettin Erbakan University, Turkey* ............................................................ 54
**Solomou, Efthalia** / *Ministry of Education, Greece* ......................................................................... 392
**Stamovlasis, Dimitrios** / *Aristotle University of Thessaloniki, Greece* ........................................... 104
**Theodoropoulos, Anastasios** / *University of Peloponnese, Greece* ................................................ 159
**Theofanellis, Timoleon** / *School of Pedagogical and Technological Education, Greece* ................ 310
**Topçu, Ahmet** / *MNE, Turkey* ............................................................................................................. 54
**Tsolakis, Savvas** / *University of Thessaly, Greece* .......................................................................... 310
**Tyler-Wood, Tandra Lea** / *University of North Texas, USA* ............................................................ 377
**Voulgari, Evagelia** / *University of Thessaly, Greece* ....................................................................... 310

# Table of Contents

## Section 2
## Teaching and Learning Computational Thinking



## Section 3
## STEM and Educational Robotics

# Detailed Table of Contents



In recent years, several educational games for learning programming have been developed with promising results. The main purpose of this chapter is to present 22 educational games or platforms that aim to cultivate computational thinking through teaching computer programming concepts to primary school students. A short description of each game followed by a comparative analysis of both their game mechanics and their educational aspects is presented. Additionally, less typical functionalities such as online classrooms, the support for learning analytics, and the creation of new levels are analyzed. This chapter could be useful for game designers and IT teachers who would like to use a game-based approach in the teaching process.



The last two decades have necessitated the need for an interdisciplinary approach to mathematics, science, and technology (STEM) as contemporary problems are too multidimensional to be tackled by a single scientific discipline as was the case with classical school curricula. Teaching programming has the potential to contribute to this vision as it is effective in helping students develop critical thinking skills. This work presents an educational approach that combines STEM learning with the basic concepts of programming through the creation of a weather-forecast app for smart mobile devices with the programming environment MIT App Inventor. This approach was implemented with second grade high school students as a school project. The evaluation results are considered encouraging as the students engaged in authentic learning activities and research related to the STEM field while, at the same time,

Coding is a spreading teaching methodology that is involving more students and teachers all over the world. But how can the practice of coding affect the development of computational thinking strategies in early years? The author, a primary school teacher, will investigate the Italian experience, believing that it may constitute an excellent field of study on the matter thanks to the enormous enthusiasm with which coding was received by the teachers, capable of renewing their teaching practices, particularly in primary school. This is a movement born from below, from the spontaneous participation of teachers, and which, in many cases, has been substantiated in what can be defined as unplugged activities, without the use of electronic technological tools.



This chapter focuses on the "unplugged" approach for teaching computational thinking (CT), that is, teaching without the use of computers or digital equipment. After a short discussion of the different definitions of CT, the chapter presents the most well-known tools and methodologies of unplugged philosophy, with a connection to CT concepts. The chapter also summarizes the main advantages of the unplugged approach to CT education and furthermore, the most important design principles of unplugged, kinaesthetic activities. A separate section is dedicated to blended approaches of plugged and unplugged activities and the evaluation of unplugged approaches. While more large-scale implementations are still required to fully evaluate the benefits of unplugged approaches to CT education, existing studies report positive findings, especially in relation to the use of unplugged approaches for CT education. The majority of these resources are available for use by educators free of charge on the internet, which makes them very useful as a CT teaching approach.



In this chapter, the authors present their research on how P12 students apply computational thinking (CT) skills when they are assigned simple science, technology, engineering, mathematics (STEM) problems, which they are called upon to solve with the help of educational robotics (ER) activities. The reason for this research was the high participation and increased interest shown in an ER event, where distributed questionnaires recorded students' views on ER, STEM, and CT. Their answers were the spark to conduct a

The “CCAP” project is an effort to teach in an interdisciplinary way both the teaching subjects of History (the trip of Columbus to discover America) and Informatics (3D modelling and printing). Students of B grade from the Junior High School of Vrilissia (age 13), on a voluntarily basis, separated into groups of 4-6, have created in a 3D design environment instruments used by Columbus during its trip to America, astrolabes, compasses, caravels, etc., as were taught during the subject of History and according to the description of the instruments given by the school book and other resources. The instruments were eventually printed out using the 3D printer in the computer lab. Part of the program was supported through the school’s curriculum hours, and part of the program had to be implemented out of school hours. After the completion of the project, students responded to a questionnaire prepared by the teachers in a Google form format. The most important results of this questionnaire are discussed in this work.



Computational thinking (CT) is increasingly emerging as a thinking skill to support the development of 21st century skills such as critical thinking, creativity, collaboration, or technology literacy, essential for students to become successful in an increasingly complex society. Educators are always looking for new strategies for developing these skills in students. Three-dimensional (3D) printing and scanning technologies are sufficiently mature and economically accessible to be used at the school level. By using 3D technologies, students explore, invent, discover, and engage in real problems and situations. This study explores the use of 3D printing technologies in a secondary school in Athens over the course of two school years. The study investigates if 3D technologies can support the development of CT skills in students.



The subject matter of the chapter is the result of a doctoral thesis conducted in the Department of Preschool Education and Educational Design of the University of the Aegean in Greece. The purpose of this chapter, derived from a corresponding thesis on the issue of folk culture and education, is to explore the role of information technology and information and communication technologies (ICT) in the production of digital educational material. In essence, the work comes to contribute to the scientific debate on whether technology can enhance the relationship between folk culture and education through interactive-multimedia and online technologies. Simultaneously, this project also aspires to contribute to the configuration of the instruction of folk culture through enriched teaching interventions by analog and digital means.

The ongoing developments in the field of information technology and, in particular, information and communication technologies (ICT) combined with the new digital culture that is rapidly emerging on the internet create a new perspective in the teaching of popular culture in primary education. More specifically, the teaching of popular culture can be greatly enhanced by learning experiences based on digital learning objects. Moreover, with the use of IT tools and enriched teaching can enhance the relationship between local folklore research and research work in primary education in the context of flexible zone or cultural programs (school activities). The internet at large and online technologies, Web 2.0, can transfuse special dynamic entities to popular culture evidence.

# Preface

## INTRODUCTION

Digital technologies knowledge, understanding and skills, such as the literacy of coding, will form the foundation of the digital literacies practices that students of all ages will draw upon as they transition to school (Walsh & Campbell, 2018). A reason is that coding-based systems are everywhere in society and student's lives, and students -even young age children- experience and manipulate these systems without recognizing them daily (Lee, 2019). Additionally, as Bers (2018a) notes while more and more people learn to code, computer programming leaves the exclusive domain of computer science and becomes central to other professions and to new ways of thinking, coding takes on the civic dimension of literacy. The ability to code is now accepted as an important skill for students of all ages to learn before they enter the technology driven, disruptive, global labor market that defines the early twenty-first century (Çiftci & Bildiren, 2019). According to Wing (2006), the children should be taught not only reading, writing and arithmetical ability but also the contribution of the information to problem solving process and how to apply Computational Thinking skills and make logical analysis even during early childhood education (Durak, Yilmaz & Yilmaz, 2019).

Computational Thinking focuses on skills children develop from practicing programming and algorithms and enables the development of qualities such as pattern recognition, and logical reasoning (Angeli, 2019). It is not a surprise that Computational Thinking (CT) has been described by Wing (2006) as a necessary skill for all and deemed fundamental to 21st Century professionals (Gomes, Falcão & Tedesco, 2018) as it involves a number of skills, like problem decomposition, developing algorithms, and abstraction (Angeli, 2019).

## OVERVIEW

This book has been dedicated to showcasing recent developments and exemplary uses of digital technology as well as unplugged activities in Developing Computational Thinking skills and programming competence at all levels (preschool, primary and secondary school). The book ends with an overview of the examples of the use of digital technology tools for development of CT, programming, and STEM content knowledge such as the use of digital fabrication technology, 3D modeling software, robotics, etc. What follows is a brief overview of each chapter.

In the first section of the book, in the opening Chapter 1, Andreas Giannakoulas and Stelios Xinogalos present a review of twenty-two educational games or platforms that aim to cultivate CT through teaching computer programming concepts to primary school students. According to the researchers although Computational Thinking (CT) can be applied to various subjects, many researchers agree that the most effective way to promote it to the early ages is through programming activities. However, learning programming is not an easy process for novices and in order to overcome the underlying difficulties different approaches have been proposed, such as the use of educational games. The presentation of the different educational games or platforms is followed by a comparative analysis of important features both regarding the game mechanics and their educational aspects.

Focusing on the use of block-based programming environments to help novice programmers to learn how to program and/or improve their understanding of programming concepts in flexible, easy and fun way, Chapter 2 presents an educational approach that combines STEM learning with the basic concepts of programming through the creation of a weather-forecast app for smart mobile devices with the programming environment MIT App Inventor. This approach was implemented with 2nd Grade High School students as a school project. The evaluation results are considered encouraging as the students engaged in authentic learning activities and research related to the STEM field while, at the same time, enhanced their interest and knowledge in pursuing careers involving programming, science, technology, engineering, and mathematics.

Chapter 3 tries to investigate whether variables such as demographic, academic success or programming related variables are related to Computational Thinking Self-Efficacy level in middle school students. Relational screening model was utilized in this study. As the result of the research, it was found that CT self-efficacy level is at low level. Furthermore, programming experiences of the students are approximately two years. The most important predictor of CT self-efficacy of the students who received programming education is demographic variables.

Chapter 4 discusses the automation of the assessment of algorithms and programming Concepts in App Inventor projects in Middle School. As computer science education makes its way into schools, diverse initiatives worldwide promote computer science education in K-12 often focusing on teaching algorithms and programming with block-based programming languages such as Scratch or App Inventor. However, alternatives to assess the learning of computer science concepts on this educational stage are still scarce. This chapter presents CodeMaster, an automated rubric for assessing algorithms and programming concepts of App Inventor projects on middle school level. The assessment is based on a rubric proposed in alignment with the K-12 Computer Science Framework with satisfactory reliability and validity. The rubric has been automated through a web-based system that allows assessing App Inventor projects through static code analysis. As a result, it can support computer science education in practice providing feedback to students and teachers.

The second section of this book discusses additional aspects of teaching & learning of Computational Thinking aspects to younger children. Chapter 5 presents a part of a wider project regarding the development of a computational thinking assessment tool for first and second grade primary school students. The assessment tool is proposed to be used within the context of the Environmental Study course, which deals with the study of physical and natural science. Its main pillar is the computational environment PhysGramming, which gives students the opportunity to create their own digital games. The relevant research was conducted in primary schools in Crete, Greece, from January to June 2019. The parameters examined were: (a) the validity and the reliability of the results obtained when applying the tool, (b) the students' computational thinking levels, and (c) the correlation of the computational thinking levels with the comprehension of the course content. The chapter presents the proposed assessment tool, as well as the results of the research regarding algorithmic thinking, which is one of the fundamental aspects of computational thinking.

Chapter 6, written by Konstantinos Zacharis and Antonios Niros focuses on a novel didactic approach that is based on Computation Thinking. After a short analysis of this powerful pedagogic frame, the authors list characteristic course design exemplars that embody it to the everyday implementation of school curricula. The authors focus on novel programming environments, as a necessary tool to reinforce and implement learning ideas without using classical programming languages. It is the authors' strong belief that this type of action carries an additive value which deserves further disseminating inside the community.

In Chapter 7, the authors examine the literature on playing digital games, for acquiring basic CT skills or learning basic programming concepts, in regard to the potential positive impacts of gaming on users of school and preschool education, especially with respect to learning, skills enhancement, motivation and engagement. For that purpose, 44 articles were carefully selected from the literature, over the past ten years, covering a wide range of approaches in the area of learning through games. The review indicates that digital games used in Computational Thinking Education provide positive effects, however, the effects are greatly dependent on the context in which the learning is being implemented, as well as on the users using it.

In Chapter 8 the author based on the fact that the generation of the so-called "digital natives" has already developed relative skills while entereing the school, aims to provide an approach to teaching technology, including the basics of computational thinking and coding. Starting from the examination of the situation in the Italian school, of which he is an active part both as a teacher and as a parent, the author presents the main unplugged activities and tools offered to pupils 6-11. Generally proposed in a playful form, with a strong component of creativity and manual skills, often collaborative and with interdisciplinary contents, the unplugged activities allow a transversal work not only between the subjects of study, but also between school orders, so as to guarantee the maximum inclusion of all students.

In the last chapter of this section, Chapter 9 Emmanouil Poulakis and Panagiotis Politis focus on the "Unplugged" approach for teaching CT, i.e. teaching without the use of computers or digital equipment. The chapter shortly discusses the main advantages of the unplugged approach to CT education and furthermore, the most important design principles of unplugged, kinesthetic activities. Moreover, it presents the most well-known tools and methodologies of unplugged philosophy, with a connection to CT concepts. Finally, a discussion is made on mixed approaches of plugged and unplugged activities and the evaluation of unplugged approaches. Summing up all resources and papers presented, the chapter states some interesting conclusions and proposes future research directions.

Following the five chapters dealing with the usage of STEM & Educational robotics as tools for how to cultivate computational and critical thinking skills in P-12 Education in the third section of the book. Chapter 10 presents a study on how P12 students apply the Computational Thinking skills of abstraction, generalization, algorithm, modularity, segmentation, debugging, and collaboration, when they are called to solve simple Science Technology Engineering & Mathematics (STEM) problems by engaging Educational Robotics (ER) activities. The results showed that: i) students may develop CT skills when involved in ER activities, ii) educational robots enhance students' engagement with programming, iii) educational robotics create a more favorable environment for developing students' CT skills. Moreover, robots, ER, and STEM are especially attractive to students due to their playful environment and they engage them in game-based problem-solving activities.

In Chapter 11, the authors present results obtained in the development of a learning experience in computational thinking, using Bee-Bot educational robotics. The experience involved 47 preschoolers of a kindergarten in Crete, Greece during the period 2019- 2020. The study reports statistically significant learning gains between the initial and final assessment of children's computational thinking skills. It was found that children in the treatment group who engaged in the robotic curricular intervention performed better on CT tests. This finding shows that an enhanced teaching experience using robots was beneficial for improving young children's computational thinking skills. The implications for designing appropriate curricula using robots for kindergarteners are addressed. Concluding the authors, claim that when used intentionally and appropriately, technology and interactive media are effective tools to support learning and development. As in recent years, there has been a push to introduce coding and computational thinking in early childhood education, educational robotics is an excellent tool to achieve this.

In the next Chapter 12, Timoleon Theofanellis and Evagelia Voulgari discuss about computational thinking and how educational robotics can help in this direction. For this purpose, the authors use the Arduino platform which is freeware both as hardware and as software. Authors also use a popular "closed system" Lego EV3 and their differences are discussed. They make specific suggestions in teaching with secondary education students. Special reference is made to do-it-yourself as a practice in computational thinking. They also argue why and how computational thinking should be taught and the advantages of this teaching to students.

Chapter 13 explores preservice teachers' attitudes about the usage of educational robotics in preschool education. Their study derives from the need that although educational robotics have become popular worldwide with a broad range of students, including preschoolers, less is known about preschool teachers' perceptions of how robotics technology impacts in learning and its relation to use in the classroom. This is problematic since the authors state we know that teachers' perceptions have a great iñuence on their teaching practices. This study used survey data gathered from 102 students of a department of Preschool Education in a University in Greece. A questionnaire developed by the researchers were used as data collection tool. At the end of the study, it was determined that preservice preschool teachers' attitudes about educational robotics usage in preschool classrooms were positive although they lack in relevant knowledge. These findings are discussed with respect to their educational implications.

The last chapter of this section, Chapter 14, focuses on a short-term collaborative project that was implemented with a thematic and interdisciplinary approach to STEM. Its objectives were to inspire students about STEM and STEM professions and to empower them to cope with stereotypes and discrimination in the professional field. The lack of a school ramp was chosen by the students as the theme of the project for which they worked together in favour of inclusion and diversity in and outside the school. Three schools collaborated in this project: two Minority Primary Schools from Greece and a High school from Turkey. Scratch coding tool was used to define the problem. STEM professionals then visited the schools, presented their profession and worked with pupils by using traditional and modern engineering tools. The collected data was processed in a Mathematical Lab where students worked collaboratively in order to design a ramp.

The last section of this book ($4^{th}$ section) discusses the role of new forms of technology and its implication on the methods of cultivating Computational Thinking skills. Chapter 15 states that digital fabrication offers many opportunities to engage in computational technology for k-12 students. Digital fabrication is defined as "the process of translating a digital design developed on a computer into a physical object" (Berry et al., 2010). Digital fabrication unleashes k-12 students' potential to innovate, create, manufacture, and share products. Products such as unique, artistically designed prosthetic devices housed in digital libraries allow students to share across countries and cultures. Digital fabrication allows students to make valuable and worthwhile products that can truly impact others. Not only can digital fabrication make currently needed products but engaging in activities associated with 3D printing can provide a plethora of much needed 21st century job skills.

Chapter 16, the authors describe the 'CCAP project' an effort to teach in an interdisciplinary way both the teaching subjects of History and Informatics (3D modelling and printing). Students of second grade of Junior High School (ages 13-14), on a voluntarily basis, separated into groups, created in a 3D design environment navigation instruments used by Christopher Columbus to his first trip to discover America. The aim of the project was to support teaching between STEM and Classical subjects, co-create and implement integrated models inspired by STEM and classical disciplines and to investigate if students developed Computational Thinking skills.

In Chapter 17, the authors explore the use of 3D printing technologies in a secondary school of Athens for two school years and investigate if the use of 3D technologies can support the development of Computational Thinking skills in students. 3D printing was used in the context of the cognitive subject of Informatics (inside the curriculum hours), of an interdisciplinary didactic approach between the cognitive subject of Informatics and Ancient Greek and History (both inside and outside the curriculum hours) and of informal teaching where 3D technologies were used by students to support different phases of environmental and STEM school programs (outside the curriculum hours).

Chapter 18 focuses on the relationship between folk culture and the Internet, as the Internet constitutes the basis of new web applications, information, and digital learning objects that can further support the teaching of popular culture, opening up new roads for enriched teaching situations. In the research part of the essay, the mains points of a research are presented as derived from a corresponding doctoral thesis that has been applied for the first time in Greece, as far as we know, in the field of folk culture and education.

This book ends with Chapter 19 focused on the teaching of folk culture with the use of digital enriched tools, based on an educational scenario that supports a guided educational project, aimed at Primary Education students. In the first part of the essay, the theoretical framework that examines the relationship between folklore and education alongside the continuation of the past (traditional culture) to the present, is unfolded. Regarding the creative part of the essay, a complete educational scenario following the logic of a guided educational project is presented. The scenario entitled "Approaching the Work of Folk Painters Yesterday and Today: The Case of Theofilos" is dedicated to the folk painter Theofilos.

## CONCLUSION

Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education examines the recent advancement of digital technologies in P-12 Education, explores the innovative ways that developmentally appropriate methodologies are used to enhance students Computational Thinking skills, selects exemplary cases of teaching interventions, and identifies challenges and future directions for practice, research, and theoretical development in using digital technologies as well as unplugged activities in P-12l education. The 19 chapters together update the field with research-informed, evidence-based design and implementation recommendations in the use of digital technologies and unplugged activities the development of Computational Thinking and coding skills, which will help P-12 teachers to prepare their students for future challenges.

This book is useful for educators, education leaders, education researchers, and stakeholders in P-12 education as well as higher education institutions. Its theoretical and pedagogical frameworks, trends and best practices in using digital technologies and unplugged activities, as well as challenges and future directions identified by the studies in this book will help the target audience gain a comprehensive and deep understanding of using digital technologies and unplugged activities in education and training for the development of students Computational Thinking skills and competence.

*Michail Kalogiannakis*
*University of Crete, Greece*

*Stamatios Papadakis*
*University of Crete, Greece*

## REFERENCES

Angeli, C. (2019). Computational thinking education: Issues and challenges. *Computers in Human Behavior*. Advance online publication. doi:10.1016/j.chb.2019.106185

Bers, M. U. (2018a). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 8. doi:10.20897/ejsteme/3868

Çiftci, S., & Bildiren, A. (2019). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer Science Education*, 1–19.

Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational Thinking, Programming Self-Efficacy, Problem Solving and Experiences in the Programming Process Conducted with Robotic Activities. *Contemporary Educational Technology*, *10*(2), 173–197. doi:10.30935/cet.554493

Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction*, *16*, 77–84. doi:10.1016/j.ijcci.2017.12.005

Lee, J. (2019). Coding in early childhood. *Contemporary Issues in Early Childhood*, 1463949119846541.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. doi:10.1145/1592761.1592779

Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (2011). A CS0 course using Scratch. *Journal of Computing in Small Colleges*, *26*(3), 19–27.

Rizvi, M., & Humphries, T. (2012). A Scratch-based CS0 course for at-risk computer science majors. In *2012 Frontiers in Education Conference Proceedings* (pp. 1–5). 10.1109/FIE.2012.6462491

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017, November 1). Demystifying computational thinking. *Educational Research Review*. doi:10.1016/j.edurev.2017.09.003

Turbak, F., Sherman, M., Martin, F., Wolber, D., & Pokress, S. C. (2014). Events-first programming in APP inventor. *Journal of Computing Sciences in Colleges*, *29*(6), 81–89.

Uludag, S., Karakus, M., & Turner, S. W. (2011). Implementing IT0/CS0 with scratch, app inventor forandroid, and lego mindstorms. In *Proceedings of the 2011 Conference on Information Technology Education*. West Point, NY: ACM. 10.1145/2047594.2047645

Walsh, C., & Campbell, C. (2018). Introducing Coding as a Literacy on Mobile Devices in the Early Years. In Mobile Technologies in Children's Language and Literacy: Innovative Pedagogy in Preschool and Primary Education (pp. 51-66). Emerald Publishing Limited. doi:10.1108/978-1-78714-879-620181004

Weese, J., & Feldhausen, R. (2017). STEM Outreach: Assessing Computational Thinking and Problem Solving. In *2017 American Society for Engineering Education Annual Conference & Exposition (ASEE)*. Columbus, OH: Academic Press.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717–3725. doi:10.1098/rsta.2008.0118

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education*, *14*(1), 1–16. doi:10.1145/2576872

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational Thinking for Teacher Education. *Communications of the ACM*, *60*(4), 55–62. doi:10.1145/2994591

# Acknowledgment

The editors gratefully acknowledge the editorial advisory board, essential contributions from the chapter authors who are the faculty members, researchers, teachers, instructional designers, and thought leaders who have bravely entered the foray of the modern competency-based education movement in P-12 education. Their innovative spirits, scholarship, and leadership create a bright future for competency-based education and for the many students who benefit from alternative, learner-centered approaches to P-12 education.

There are many reviewers to thank and we are grateful to those authors who so willingly reviewed more than one chapter. A particular thank you goes to those reviewers who were not authors but graciously shared their time and expertise to support this cause.

We also wish to thank Jan Travers, Director of Intellectual Property & Contracts IGI Global, Brittany Haynes, Editorial Assistant, IGI Global, Caroline Campbell, Marketing Manager, IGI Global, Charlie Mercer, PsycINFO Book Content Coordinator, Morgan Brajkovich and Maria Rohde, Assistant Development Editors, IGI Global for their extensive (and good-natured) editing support in preparation of the final manuscript and for their support in managing the many details associated with publishing a book.

We would like to thank our families for their on-going help, support and encouragement.

*Michail Kalogiannakis*
*University of Crete, Greece*

*Stamatios Papadakis*
*University of Crete, Greece*

# Section 1
# Teaching and Learning Programming

# Chapter 1
# A Review of Educational Games for Teaching Programming to Primary School Students

**Andreas Giannakoulas**
*Department of Applied Informatics, University of Macedonia, Greece*

**Stelios Xinogalos**
https://orcid.org/0000-0002-9148-7779
*Department of Applied Informatics, University of Macedonia, Greece*

## ABSTRACT

*In recent years, several educational games for learning programming have been developed with promising results. The main purpose of this chapter is to present 22 educational games or platforms that aim to cultivate computational thinking through teaching computer programming concepts to primary school students. A short description of each game followed by a comparative analysis of both their game mechanics and their educational aspects is presented. Additionally, less typical functionalities such as online classrooms, the support for learning analytics, and the creation of new levels are analyzed. This chapter could be useful for game designers and IT teachers who would like to use a game-based approach in the teaching process.*

## INTRODUCTION

Computational Thinking (CT) was first presented by Papert in 1980 and again in 1996 (Papert, 1980; 1996), and since then its definition, teaching, and evaluation have been discussed by many scholars (Hsu, Chang, & Hung, 2018). The concept of CT became popular by Wing (2006). According to Wing (2006), CT is a set of skills, techniques, methods, and attitudes that allow solutions to be addressed in a wide range of problems, not just in the field of information technology. Through CT, people discover new ways to deal with existing problems. Although there are several studies nowadays on CT, the scientific community has not come up with a precise definition and what exactly it means (Moreno-León, Román-

DOI: 10.4018/978-1-7998-4576-8.ch001

González, & Robles, 2018). Thus, several definitions besides the one of Wing (2006) can be found. For example, according to Atmazidou and Demetriadis (2016), *abstraction*, *decomposition*, *generalization*, *algorithmic thinking* and *modularity* are considered as basic CT skills.

Computer programming is considered by many researchers as an effective way to develop high level skills, such as problem solving, logical thinking, critical thinking, and CT (Combéfis, Beresnevičius, & Dagienė, 2016). Although CT can be applied to various subjects such as mathematics or biology (Hsu et al., 2018), most researchers agree that the most effective way to promote CT to the early ages is through teaching programming activities (Moreno-León et al., 2018).

Recognizing the important role of computer programming as a means of promoting CT, many initiatives such as the "Europe Code Week" (https://codeweek.eu/), the "Hour of Code" (https://code.org/learn), the "Scratch Day" (https://day.scratch.mit.edu/) and the "Bebras" competition (https://www.bebras.org/), have been introduced for students of different age groups, aiming at their digital literacy and familiarity with computer programming and CT, through fun and constructive activities. Usually, these initiatives are based on digital platforms where students can develop and improve their CT skills through playing games.

However, learning programming is not an easy process for novice programmers (Zaharija, Mladenović, & Boljat, 2013). The difficulty in understanding the abstract concepts of programming but also the traditional teaching method which is based mainly on the presentation of theoretically programming concepts and the implementation of number and symbol processing programs in professional programming environments reduce students' motivation and interest in learning programming (Brusilovsky, Calabrese, Hvorecky, Kouchnirenko, & Miller,1997).

In order to overcome such difficulties, different approaches have been proposed by researchers (Xinogalos & Satratzemi, 2004) such as: programming microworlds (Xinogalos, Satratzemi, & Dagdilelis, 2006), flowchart-based programming environments (Xinogalos, 2013), educational robotics activities (Atmazidou & Demetriadis, 2016) and more recently educational games (Vahldick, Mendes, & Marcelino, 2014).

Using educational games in the teaching process can bring many benefits. Educational games incorporate an attractive and interactive learning context that motivates students for practicing with programming and provides them appropriate feedback, challenge and scaffolding (Laporte, & Zaman, 2016; Malliarakis, Satratzemi, & Xinogalos, 2014a). When dealing with primary school students, utilizing educational games as a means of promoting CT is considered even more important. Additionally, in recent years several educational games for learning programming have been developed with promising results (Malliarakis, Satratzemi, & Xinogalos, 2017).

The main purpose of this chapter is to review existing educational games or platforms that aim to teach CT to primary school students and present their main potential in an organized way. The contribution of this study lies in the fact that it could be a useful tool for game designers and additionally for IT teachers who would like to use a game-based approach in the teaching process, in order to select the appropriate game for their course, providing them with a concise and comprehensive overview of the main potentials of the games.

The rest of the chapter is organized as follows: the next section presents briefly the main results of similar studies on games designed to teach basic programming concepts followed by a description of the methodology of this study. After that the main characteristics of twenty two educational games for teaching basic programming concepts are analyzed, followed by a comparative analysis. The next section discusses the results of the comparative analysis and finally conclusions and proposals for further research are presented.

## RELENANT WORK

Table 1 summarizes basic information of relevant work on serious games that focus mainly on learning programming, as well as the review presented in this chapter.

*Table 1. Studies evaluating games for teaching programming*

| Study | Number of games | Categories of evaluation criteria | Games that are common with the review presented in this chapter |
|---|---|---|---|
| Vahldick et al. (2014) | 40 | Type of the game<br>Platform<br>Type of activities – competency<br>Topics covered<br>Programming language | Code Combat, Code Spells, Kodable, Light-Bot 2.0, Program Your Robot, World of Variables |
| Malliarakis et al. (2014b) | 12 | Educational goals<br>Scenario<br>Information resources<br>Topics covered<br>Programming language<br>Programming activities (Type)<br>Special characteristics<br>Generic conditions (e.g. offline, online, blended learning) | LightBot |
| Laporte and Zaman (2016) | 19 | Cognitive components<br>Process components<br>Affective components | LightBot, Kodable, Codespells, Code Combat |
| Combéfis et al. (2016) | 7 | Trained skills<br>Game elements<br>Interaction style<br>Environment<br>Learning approach | Lightbot |
| Eguíluz et al. (2018) | 26 | General characteristics<br>Social Options<br>Recommended ages<br>Information relating to the use in class<br>Engagement<br>CT aspects<br>Design aspects<br>Feedback to user | Kodable, Kodetu, LightBot, SpriteBox |
| Miljanovic and Bradbury (2018) | 49 | Audience<br>Educational content (based on the knowledge areas identified in the ACM Computer Science Curricula 2013)<br>Learning Focus<br>Available Evaluation of games | Code Combat, Software KIDS, World of Variables, BOTS, Program Your Robot, Light-Bot 2.0 |
| Review presented in this chapter | 22 | Scenario<br>Game Mechanics<br>Topics – Programming concepts<br>CT concepts<br>Programming language and editor<br>Program execution – test and debug<br>Support for online classrooms and level editor<br>Available platforms | |

Vahldick et al. (2014) concluded that most of the 40 games they evaluated support only one programming language and that there is a trend for block based manipulation environments.

Malliarakis et al. (2014b) evaluated 12 educational games for computer programming and found that the majority of the games satisfy the examined features to a satisfactory extent, but they cover fairly simple programming concepts. Furthermore, the authors concluded that there are limited experimental studies for these games.

In another study, Laporte and Zaman (2016) identified 10 base problem categories faced by novice programmers, grouped them into three main components groups and recorded how these problems are addressed in a set of 19 programming games. Authors found that the games do not deal to a great extent with problems such as mapping and problem solving strategies or writing correct syntax, which are problems specific to the programming field. On the other hand, the games deal better with problems common to other learning domains such as design, implementation and evaluation.

Combéfis et al. (2016) reviewed 7 educational online platforms or serious games for learning programming. All the reviewed games are visual, 2D and online, while interaction is accomplished through the keyboard or mouse. With the exception of Leek Wars, none of the other games supports collaboration, while the majority of them are designed for a single player.

Eguíluz et al. (2018) reviewed 26 platforms, which aim to teach CT skills mainly on primary and secondary school students. Findings of this research reveal the predominance of the block based visual environments, the absence of learning analytics, the need to assessment improvement in order to include more key indicators in CT skills and moreover the limitation of creating new challenges on these systems.

Recently, Miljanovic and Bradbury (2018) examined 49 serious programming games and concluded that the games focus mainly on problem solving and fundamental programming concepts and less on data structures, development methods and software design. Furthermore the majority of the games are not multiplayer.

All the aforementioned studies present games that can be used in a wide range of age groups. The main difference between this chapter and previous studies is that the authors focus on the comparative analysis of twenty two games that aim to teach programming concepts mainly to primary school students. Twelve of these games are not included in the reviewed studies.

## RESEARCH METHODOLOGY

In order to achieve the main goal of this chapter, three research questions were defined:

**RQ1:** Which educational games for teaching programming or else CT aspects to primary school students are available?

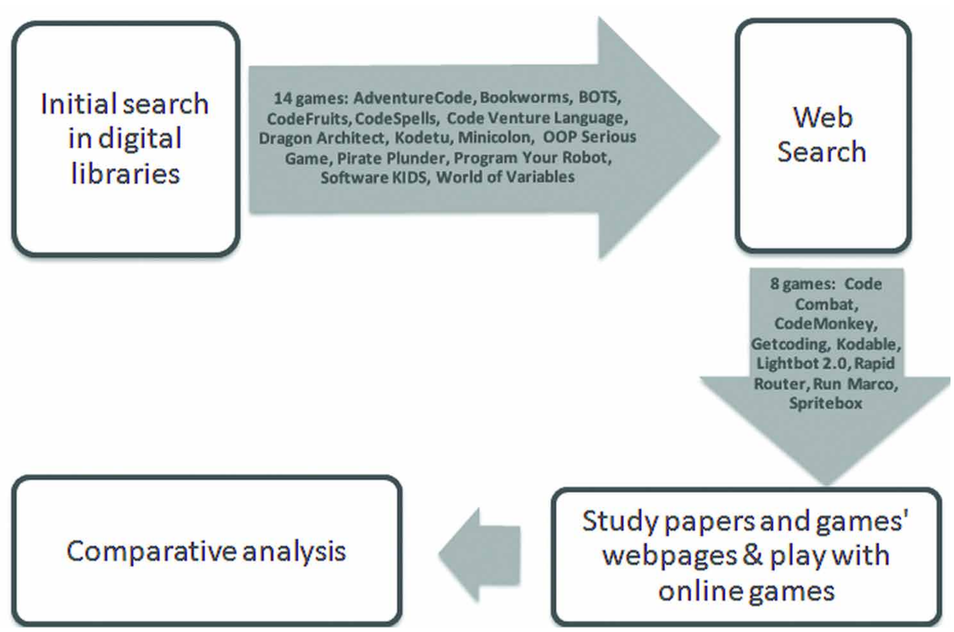
**RQ2:** Which are the most important features of these games both regarding their game mechanics and their educational aspects?

**RQ3:** Which are the open problems and what features are currently missing from these games?

To find games suitable for this study a research was carried out in digital libraries like ACM Digital Library, IEEE Xplore, Scopus, and Google Scholar. Terms like *games*, *teaching programming*, *teaching Computational Thinking*, *teaching programming concepts*, *Computational Thinking*, *learning programming*, *primary school*, *primary students*, *young students* and *novice programmers* were used in various combinations. Next, the studies that describe or evaluate games focusing mainly on teaching basic programming concepts to primary school students were selected. Games developed for secondary or higher education students, according to their designers or their educational content, were excluded. Fourteen games were identified. After that, the web was searched with the same keywords and 8 more games or platforms were found. In order to capture the potential of the games and achieve a more accurate comparative analysis, the collected papers that describe or evaluate each game and its' webpage (if available) were studied and finally we played each game in order to gain personal experience. The entire process of the study is summarized in Figure 1.

The evaluation criteria for the comparative analysis of the games refer both to features that are important for a successful educational game and learning of programming and CT concepts. The categories of "Scenario" and "Game Mechanics" include criteria that are widely referenced in well-known SGs design frameworks (de Freitas & Jarvis, 2006; Kiili, 2005; Sanchez, 2011; Yusoff, Crowder, Gilbert, & Wills, 2009), while the categories "Topics – Programming concepts" and "Programming language and editor" include criteria from the CMX framework that refers to SGs for programming (Malliarakis et al., 2014a).

*Figure 1. Study process*




## GAMES FOR TEACHING COMPUTATIONAL THINKING

This section introduces twenty two educational games designed to teach basic programming concepts to novice programmers. The authors of this chapter focus mainly on games that in their estimation are suitable for primary school students.

## AdventureCode

AdventureCode (Scheuerman, 2015) is an educational game designed to introduce elementary school aged children to CT concepts through playing. Players drag and drop code blocks creating algorithms, in order to guide a character through a maze to reach a goal. The game introduces players to concepts like *decomposition* and *algorithmic thinking* by guiding them through a narrative driven series of puzzles.

## Bookworms

The purpose of Bookworms (Korpi, 2014) is to develop young children's *logical thinking abilities* and inspire them to practice with programming on their own. Understanding block type coding and the ability to modify it are the goals of the game. The gameplay is divided into three phases. In the first phase the player should place towers in a path in order to stop termites walking on it, while in the second phase the player helps bookworms to collect resources efficiently, writing or changing a script that the bookworms will execute. Finally, in the third phase the player has to group different objects according to their common properties.

## BOTS

BOTS (Hicks, Catete, & Barnes, 2014) is a web based game designed to teach basic programming concepts and fundamental ideas of problem-solving to novice programmers. Students attempt to guide a robot character through maze-like puzzles, designed on a grid-based 3D environment. The goal of each puzzle is to press several switches within the environment, which can be done by placing an object or the robot on them. To do this, players drag and drop visual commands to the programming area. The game provides five types of commands: action, control, functions, variables, and parameters. Moreover, students can play expert designed puzzles, create new puzzles, and play puzzles created by other players.

## Code Fruits

Code Fruits (Goyal, Chopra & Mohanan, 2017) is a gesture based interactive game which implicitly helps K-12 students to learn abstract concepts of CT and write code in an easier, playful and engaging way. The game implements a scenario where the player helps the fruit truck to collect fruits from the town's abandoned orchards which would then be delivered to the local old age home. Hand gestures of two types representing the different elements of the programming language such as functions and parameters are used as input interaction method in order to move the truck to its destination. The Kinect SDK was used to design custom hand gestures which are mapped to the functions used in the game.

## CodeCombat

CodeCombat (https://codecombat.com/) is a role-playing game (RPG) that introduces students to object oriented programming (OOP) in a fun way. The user interface is divided into two parts: a code editor and a maze. At each level of the game, the player undertakes to successfully perform a mission in which s/he is required to perform a series of actions such as collecting gems, killing various monsters and eventually moving to the exit of the level. The player writes commands in the code editor in order to guide the avatar in the maze and is gradually introduced to new concepts such as loops, conditionals, and variables. Incorrectly typed commands and errors (such as ordering the avatar to run against a wall) will cause the avatar to lose points and die after many errors.

### CodeMonkey

CodeMonkey (https://www.playcodemonkey.com) is a fun and educational game-based environment that teaches kids basic programming concepts. In each level of the game, students write programs using a real open-source programming language called CoffeeScript, in order to guide a monkey to collect a number of bananas. Students need to write code or choose from available commands, to make the monkey move on the screen.

### CodeSpells

CodeSpells (Esper, Foster, Griswold, Herrera, & Snyder, 2014) is a 3D immersive video game that is designed to teach elementary students Java and basic computer programming concepts. CodeSpells embodies the idea of "spells" (small Java programs) as both an artifact that players can see, read and write and also an executable object that players can use to interact with the world and solve problems. In order to fully immerse the players, a spellbook which might be found in any other fantasy RPG has been designed. The spellbook provides information about how spells work and when they might be useful. Players can use it, along with the Non-Player Character's advice, to navigate through quests. Most important though is the spellbook's dual purpose as a learning resource. Players are encouraged to read through the spellbook, but also to execute spells to test them out.

### Code Venture Language

Code Venture, (Hussain, Fergus, Al-jumeily, Pich, & Hind, 2015) aims to improve the programming skills of kids and their logical thinking by providing better feedback to their teachers. According to its designers the game offers a number of activities, where the player is allowed to do a number of steps in order to reach the desired goal. Teachers have access to a web interface where they can create classrooms, add/remove students, or monitor each child's progress, while students access the game using their login data only through mobile or tablets.

### Dragon Architect

In Dragon Architect (Bauer, Butler, & Popovic, 2017) (http://dragonarchitect.net/play/latest/), players write code to control a dragon that builds 3D structures in a cube world. The user interface is separated into two parts: an area where players can build their code and an area which represents a 3D environment affected by their code. The player writes programs using the Blockly library, in order to move the dragon in three dimensions and remove cubes of various colors. Initially the game offers puzzles that introduce the idea of assembling and running code, as well as the code blocks for moving the dragon and placing cubes. After that, the player can experiment and build in the sandbox completing other puzzle sequences and making more code blocks available. Players also, can switch between sandbox and puzzles at any time.

## Getcoding

Getcoding (https://www.hepis.gr/getcoding) is a game that aims to teach basic *programming concepts* to novice programmers such as sequences, loops, conditionals and furthermore flowcharts. The player needs to program the hero (Thales or Irida) of the game to move on a route in order to collect a gold coin, using a simplified visual programming language that allows him/her to visually create a program as a flowchart. Players can use commands such as move forward, turn right, turn left, check whether there is a path or an obstacle in front, left, or right of the hero, and also commands for loops.

## Kodetu

Kodetu (Eguíluz, Guenaga, Garaizar, & Olivares Rodríguez, 2017; Gal, Hershkovitz, Eguíluz, Guenaga, & Garaizar, 2017) (http://kodetu.org/), is an online game platform that teaches basic programming concepts to novice programmers. The game implements a scenario where the player guides an astronaut through the maze-shaped spaceship to reach a goal without falling into space. To do this, the player joins visual blocks that represent special commands in a workspace. There are blocks to go forward, turn right, turn left, check whether there is a path in front, left, or right of the astronaut, and perform loops to move towards the goal. Kodetu is based on "Maze", a basic application made with Blockly.

## Kodable

Kodable (https://www.kodable.com/) is an educational game for teaching basic programming concepts in kindergarten or elementary school students. The basic idea behind this game is that the Faze family has landed abnormally on the planet Smeeborg and begins exploring the planet. In the first section of the game the player drags and drops simple visual commands in a limited editor area, in order to program the Fuzzes to escape from a series of linear mazes, collecting at the same time coins and points. In the second part of the game, which is the planet Asteroidia, the player ejects colored balls from a spaceship in order to eliminate a set of asteroids. Finally, in the last section of the game, named the Bug World, the players build various new towers that best fit their defensive needs, which are essentially objects from the class tower, in order to protect the Fuzz family.

## Lightbot 2.0

Lightbot 2.0 (http://armorgames.com/play/6061/light-bot-20) is an online flash game designed by Danny Yaroslavski that introduces the player into basic programming concepts such as sequencing, functions, recursive loops and conditionals. The player programs a robot to navigate a maze and light up all the blue tiles in a fantastic world of square tiles, dragging icons from a toolbar and arranging them in an area of empty slots. Icons represent specific commands for the robot such as "move forward", "turn right", "jump", etc. Two distinct versions of Ligthbot which differ from version 2.0 - in terms of the topics covered - are Lightbot 1.0 and Lightbot Hour of Code.

## MiniColon

MiniColon (Ayman, Sharaf, Ahmed, & Abdennadher, 2018) is an interactive, gesture-based game, where the Kinect Sensor device is used to control the gameplay instead of playing it using a normal PC. The main idea of MiniColon is that the main character, a carrot, is trapped on an island and in order to escape, has to collect some fruits and supplies. The carrot is moving forward all the time during the gameplay. The player controls the motion of the carrot in order to achieve the goal of each level. MiniColon aims at teaching children basic programming concepts like sequencing, conditionals, and iteration.

## OOP Serious Game

OOP Serious Game (Lotfi & Mohammed, 2018) is a 2D game that teaches and evaluates knowledge gained about OOP concepts. The game implements four different scenarios in four corresponding levels. In each level the player should complete some tasks in order to understand a concept concerning OOP. In the first level the player perceives the notion of an "Object", in the second level the player understands the notion of "class" and in the last two levels the player is taught the principle of "inheritance" and "polymorphism" respectively. A number of animals play the role of the main characters of the game.

## Pirate Plunder

Pirate Plunder (Rose, Habgood, & Jay, 2018) is a novel educational programming game that introduces loops, custom blocks and clones in a game-based Scratch-like setting. The main aim of the game is to teach players to identify and correct code smells, enhancing their abstraction and decomposition skills. The player using blocks programs a pirate ship to navigate to a position around a grid and collect a treasure chest, collecting at the same time items (coins) and interacting with obstacles. The game provides two kinds of levels, the "Tutorial levels" and the "Challenges". Tutorial levels introduce new blocks, with a parrot character explaining the functionality of each block. After that, the players use these blocks to complete a set of challenges before trying the next tutorial. The Pirate Plunder layout and functionality is similar to that of Scratch 2.0.

## Program Your Robot

Program Your Robot (Kazimoglu, Kiernan, Bacon, & Mackinnon, 2012) is an online flash game developed as an academic project. The game aims at teaching basic programming concepts and various CT skills such as algorithm building or debugging. The player programs a robot to navigate in a series of platforms and reach a "teleporter", dragging icons from a toolbar and arranging them in an area of empty slots. The icons represent specific commands for the robot such as "move forward", "turn right", "turn left", etc.

## Rapid Router

Rapid Router (https://www.codeforlife.education/) is an online free game aligned to the UK Computing Curriculum that aims to teach children the basic principles of programming. The player needs to help "Dee" – the main character - drive the van on a road and make all the deliveries. At the first levels of the game the player programs the van using Blockly commands, such as “move forwards” and “turn right”, and later s/he writes lines of code in Python.

## Run Marco

Run Marco (https://www.allcancode.com/web) is an educational game that aims to teach the player basic programming concepts, such as sequencing, iteration and conditions. Players should guide Marco or Sophia - the main character – on a path, in order to find his (her) friends in the jungle. For this purpose, the players “drag and drop” visual blocks that represent commands such as “step forward” and “turn left” on an editor area.

Back in 2018 the platform provided tools, such as class management, the ability to implement lesson plans, as well as authoring tools to adapt them to a class’s needs, creation of new activities, and a dashboard which allowed monitoring students’ progress in the game in real time as recorded in a study by Giannakoulas and Xinogalos (2018). It should be noted, however, that at the time of writing this chapter the game does not allow the use of these authoring tools.

## Software KIDS

Software KIDS (Ramirez-Rosales, Vazquez-Reyes, Villa-Cisneros, & De Leon-Sigg, 2016) is a serious game that aims to teach basic concepts of *OOP* and *Software Engineering* (SE) to children older than eight years old, through Android devices. The game is divided in ten levels where the player learns about the stages of the software development, new vocabulary about SE and OOP, and the differences between objects, functions and attributes. Each level is limited in time to a range between 60 and 90 seconds, depending on the played level. To progress between levels it is necessary to answer correctly. When the player does not achieve all correct answers on time, or has more mistakes than allowed, s/he can start again. Levels have a simple operation, based mainly on selecting-and-dragging blocks and selecting-true-or-false options.

## SpriteBox Code Hour

SpriteBox (https://spritebox.com/index.html) is a logic puzzle game from the creators of Lightbot that combines the fun of *platformer* games (like Minecraft and Super Mario Bros) with learning the basics of coding. The player needs to help a sprite in order to free his bottled-up friends. The player controls the movement of the hero in various platforms using arrow keys and collects stars during his journey. In order to overcome some platforms that are out of reach or impassable, the player should find a yellow SpriteBox, jump, and bump it. This will call over a little ghost-like character, who will try to help the player continue the journey in the game. The players program the ghost-like character, using icon or textual commands in order to modify the game environment, building bridges, ladders, and more and making previously impassable sections passable.

## World of Variables

World of variables (Zapušek & Rugelj, 2013) is based on visualizations of different variable types and on the explanation of the assignment statement. The game takes place on a planet far away in the universe where inhabitants have a problem making an order from the chaotic mess that dominates on the planet. They also need help with transporting some goods to the other planets. The goods that are transported in the game have similar properties as variables in programming. In this way students can learn the concept of a variable in an intuitive and motivating setting. The game consists of four parts.

## COMPARATIVE ANALYSIS

### Scenario

Table 2 presents characteristics of the games regarding their *scenario* and the target group ages which are suitable for them. The majority of the games utilize a fantastic scenario in which the player programs the movement of a character (like a fish, a robot, an astronaut etc) in a simulated world. In addition, Bookworms and Kodable incorporate three different scenarios, while in CodeCombat the player undertakes to successfully perform a mission that requires actions such as collecting gems, killing various monsters and eventually moving to the exit of the level. CodeSpells embodies the idea of "spells" (small Java programs) as both an artifact that players can see, read and write and also an executable object that players can use to interact with the world and solve problems. Furthermore, OOP Serious Game offers activities related to OOP principles and Spritebox allows the player to modify the game's environment.

Several games offer players the choice between two or more available *avatars*, which makes it more possible for the player to identify with the main character in the game, while in Kodable and Pirate Plunder the player can change or customize the avatar. Additionally, in CodeCombat the player can buy additional items to upgrade or to unlock new avatars.

Bookworms, CodeCombat, Kodable, Pirate Plunder and Program Your Robot, are the only games which include *enemies* that the player has to avoid or to defense while in some games the player collects *objects* or avoids *obstacles*.

Regarding the type of the *programming activities* offered, in the majority of the games the activities remain within the domain of designing an algorithm, while some games incorporate activities related to fixing errors in a buggy program.

### Game Mechanics

Table 3 presents game mechanics used to support the player or to reward him. The majority of the games are classified as *puzzle games* with CodeCombat and CodeSpells being additionally *Role Playing Games* (RPG). Spritebox is a *platform game* besides a puzzle game, while CodeFruits and MiniColon are classified as *gesture based games*.

Almost all the games provide some kind of support to the player to solve the activities. Several games, incorporate *tutorial levels* that support the player whenever a new programming concept is introduced, providing a step-by-step guidance or making available only the code constructs required for solving the specific level. In addition, *informational messages* before running an activity, is a common method used in many games, while hints, dialog boxes, help windows or video which explain the functionality of the user interface or the new concepts introduced, are rarely used.

Only eight games *reward* players with a total score for their performance. Additionally the majority of them reward the player after the successful completion of each activity displaying appropriate mes-

Table 2. Scenario of games - Ages

| Games | Age<br>* denotes authors' estimation | Variety of activities | | Select avatar | Collect objects | Avoid obstacles | Enemies | Programming Activities | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Program the movement of a character | Other | | | | | Design an algorithm | Fix errors in a program | Other |
| Adventure Code | 6-12 | √ | | | | | | √ | | |
| Bookworms | 7-12 | | Build defense towers;<br>Collect resources;<br>Upgrade defense towers | | | | √ | | | Understand a given script;<br>Improve a given script;<br>Find common properties between objects |
| BOTS | 12-18 | √ | | | | | | √ | √ | |
| CodeCombat | 9+ | √ | | √<br>(customize) | √ | | √ | √ | | |
| CodeFruits | 8-10 | √ | | | | | | √ | | |
| CodeMonkey | 9+ | √ | | | √ | √ | | √ | √ | |
| CodeSpells | 9+ | | Embodies the idea of "spells" | | | | | √ | √ | |
| Code Venture | 5-11 | √ | | √ | | | | √ | | |
| Dragon Architect | 9+ | √ | | | | | | √ | | |
| Getcoding | 10+ (*) | √ | | √ | | | | √ | | |
| Kodetu | 8+ (*) | √ | | √ | | | | √ | | |
| Kodable | 5-11 | √ | Build defense towers;<br>Destroy asteroids | √<br>(customize) | √ | | √ | √ | | Assign values to a variable;<br>Create/modify Object properties |
| Lightbot 2.0 | 9+ (*) | √ | | √<br>(hour of code) | | √ | | √ | | |
| MiniColon | 8-9 | √ | | | √ | | | √ | | Execute a snippet of code;<br>Trace a code snippet written in Python |
| OOP Serious Game | 11+ (*) | | Understand the notion of "object", "class", "inheritance" and the principle of "polymorphism" | | | | | | | Create objects;<br>Classify objects to classes;<br>Move animals to the corresponding parent classes;<br>Move methods to perform animals' actions |

continues on following page

*Table 2. Continued*

| Games | Age * *denotes authors' estimation* | Variety of activities | | Select avatar | Collect objects | Avoid obstacles | Enemies | Programming Activities | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Program the movement of a character** | **Other** | | | | | **Design an algorithm** | **Fix errors in a program** | **Other** |
| **Pirate Plunder** | 9+ | √ | | √ (customize - upgrade) | √ | √ | √ | √ | √ | |
| **Program Your Robot** | 10+ (*) | √ | | | √ | | √ | √ | | |
| **Rapid Router** | 5-16 | √ | | | | √ | | √ | | |
| **Run Marco** | 6-12 | √ | | √ | | √ | | √ | | |
| **Software Kids** | 8+ | | Ten different activities in corresponding levels | | | | | | | Ten different activities in each level |
| **Spritebox** | 5+ | √ | Players control the main character with arrow keys; Players program a little ghost-like character | √ | √ | √ | | √ | | |
| **World of variables** | 8+ | | Choose appropriate container, label and content for a variable; Place the goods on appropriate space ship; Answer questions about initialization of variables; Place goods to the right places | | | | | | | Declaration of a variable; Understand the several types of variables; Initialization – storing a value in memory; Implicit value assignment to the variable location |

sages while four games incorporate a *three star reward system* depending on how well the activity ended. Lightbot 2.0 also somewhat rewards the player counting the number of commands used for a solution, while BOTS rewards students after an activity with medals based on the length of their programs.

Additionally, six games offer an integrated reward system of achievements, while many games encourage the player to give a *better solution*, rewarding him/her with a better score or with appropriate messages.

## Topics: Programming Concepts

Table 4 presents the programming concepts covered by each one of the games presented in this chapter. With the exception of OOP Serious Game and Software Kids which support only activities related to *OOP*, and World of variables, which is based on visualizations of different variable types and on the explanation of the assignment statement, the majority of the rest games support *procedural programming* activities. OOP activities are also included in the games Bookworms, CodeCombat, CodeMonkey, CodeSpells and Kodable.

Bookworms, CodeFruits, CodeMonkey (in its free version which is presented in this study), Dragon Architect, Pirate Plunder and Spritebox don't support if statements, while Lightbot 2.0 does not support iterative structures, proposing the use of recursion instead. Recursion is also supported on Pirate Plunder and Program Your Robot. Only six of the games incorporate activities with nested if statements, while the majority of them support loops with a predefined number of iterations. Also, variables are rarely covered.

Some games like CodeCombat, CodeMonkey, Kodable and Rapid Router are embedded into a broader learning platform, which provides extensive support through a series of ready-made lesson plans that include activities based on game philosophy. Also, Getcoding is accompanied by a book (Kalovrektis, 2015) with theory and exercises inseparably connected to the game.

## CT Aspects

The vast majority of the games support the development of *algorithmic thinking* through programming activities where the students must develop algorithms in order to complete the tasks assigned to them (see Table 2). With the exception of the games Adventure Code, Code Venture, OOP Serious Game, Software Kids and World of variables all the other games support *pattern recognition* and *modularity*, usually through activities where the player has to recognize the existence of a *pattern* for solving a task, which has to be coded and repeated using loops, reducing repetitive code (see Table 4). Furthermore, most games support *abstraction* and *generalization* through the use of *functions* (see Table 4) while some other games in this direction, incorporate activities related to fundamental concepts of *OOP* such as *classes* and *objects* (see Table 2 and Table 4). Finally, with the exception of OOP Serious Game, Software Kids and World of variables, all the other games support *decomposition* through activities where the students have to break down an algorithm into smaller parts.

## Programming Language and Editor

Table 5 presents features regarding the editor and the languages supported by each game. Most of the games include an *editor* that allows the player to write a program, using mainly a block-based programming environment. Moreover, in some games like Kodable and Lightbot 2.0, the player "drags and drops" visual icons (commands) in an area with only a few slots available, forcing this way the player to write

*Table 3. Game Mechanics*

| Games | Scaffolding/ Support | | | | Score | Reward after an activity | | | Reward better solution | Achievements |
|---|---|---|---|---|---|---|---|---|---|---|
| | Tutorial levels | Messages before an activity | Hints (Audio or visual cues) | Other support | | Messages | Three stars per level | Other | | |
| **Adventure Code** | | | √ | A friendly character provides helpful feedback;<br>A voice narrates the story | | | | | | |
| **Bookworms** | √ | | | Help windows | | | √ | | √ | √ |
| **BOTS** | √ | √ | | Dialog boxes | | | | Players earn platinum, gold, silver or bronze medals;<br>Scoreboards | √<br>A reward system shows the least number of commands needed to solve the puzzle and the players' number of commands used in the level | |
| **CodeCombat** | √ | √ | √ | | √<br>(gems and experience) | √ | | Players earn gems, experience points, and occasionally a new item | √ | √ |
| **CodeFruits** | | | | | | | | With a short story or an anecdote by the old granny after completing a level | | |
| **CodeMonkey** | √ | √ | | | √<br>(requires account) | √ | | | √ | √ |
| **CodeSpells** | | | | Offers a spellbook with ready java code | | | | | | √<br>Additional Spellbook pages are unlocked once the student collects certain badges |

*continues on following page*

Table 3. Continued

| Games | Scaffolding/ Support | | | | Score | Reward after an activity | | | Reward better solution | Achievements |
|---|---|---|---|---|---|---|---|---|---|---|
| | Tutorial levels | Messages before an activity | Hints (Audio or visual cues) | Other support | | Messages | Three stars per level | Other | | |
| **Code Venture** | | | | | √ | | | | | |
| **Dragon Architect** | | | | | | √ | | | | |
| **Getcoding** | | | | Video | | √ | | | √ | √ |
| **Kodetu** | | √ | | | | √ | | | | |
| **Kodable** | √ | | √ | | | √ | | | √ | |
| **Lightbot 2.0** | √ | | | | | | | Counting the number of commands used | | |
| **MiniColon** | | | | | | √ | | Sound referring to winning | | |
| **OOP Serious Game** | | | | | √ | | | Bonus | | |
| **Pirate Plunder** | √ | | | | √ Coins and stars | | √ | | √ | |
| **Program Your Robot** | | | | Video | √ | | | | √ | √ |
| **Rapid Router** | | √ | | | √ (for each level) | √ | √ | | √ | |
| **Run Marco** | √ | | | | | | √ | | √ | |
| **Software Kids** | | | | | √ | | | | | |
| **Spritebox** | √ | √ | | A little ghost-like character helps the player continue his journey | √ | | | Sound referring to winning | | |
| **World of variables** | | | | | | | | | | |

*Table 4.Programming concepts*

| Games | Sequence commands | Simple if | If…else | Nested if | For | Do while or Repeat until | Nested loops | Recursion | Functions | Object Oriented | Variables |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Adventure Code** | √ | | | | | | | | | | |
| **Bookworms** | √ | | | | | √ | | | | √ | |
| **BOTS** | √ | √ | | | √ | √ | √ | | √ | | √ |
| **CodeCombat** | √ | √ | √ | √ | √ | √ | √ | | √ | √ | √ |
| **CodeFruits** | √ | | | | √ | | √ | | √ | | √ |
| **CodeMonkey** | √ | | | | √ | | | | √ | √ | |
| **CodeSpells** | √ | √ | √ | √ | √ | √ | √ | | √ | √ | √ |
| **Code Venture** | √ | | | | | | | | | | |
| **Dragon Architect** | √ | | | | √ | | √ | | √ | | |
| **Getcoding** | √ | | √ | √ | √ | √ | √ | | | | |
| **Kodetu** | √ | √ | √ | √ | | √ | | | | | |
| **Kodable** | √ | √ | | | √ | | | | √ | √ | √ |
| **Lightbot 2.0** | √ | √ | | | | | | √ | √ | | |
| **MiniColon** | √ | √ | | | | √ | | | | | |
| **OOP Serious Game** | | | | | | | | | | √ | |
| **Pirate Plunder** | √ | | | | | √ | | √ | | | |
| **Program Your Robot** | √ | | √ | | √ | | | √ | √ | | |
| **Rapid Router** | √ | √ | √ | √ | √ | √ | √ | | √ | √ | √ |
| **Run Marco** | √ | √ | √ | √ | √ | √ | √ | | | | |
| **Software Kids** | | | | | | | | | √ | √ | |
| **Spritebox** | √ | | | | √ | | √ | | | | |
| **World of variables** | | | | | | | | | | | √ |

more efficient programs or even use specific programming concepts such as functions and recursive calls. In a similar way, the games BOTS and Spritebox, allow the player to "drag and drop" visual commands and arrange them in an editor area. Dragon Architect, Kodetu, Rapid Router and Run Marco are based on Blockly, where the player builds a program by linking visual code blocks, while Pirate Plunder uses a selected set of blocks relevant to the game play like in Scratch.

Some games allow the player to write commands in a *real textual programming language*. Specifically, in CodeCombat the player has the choice to write code in Python, Javascript or Coffee Script, while in CodeMonkey the player writes lines of code in CoffeeScript. Additionally, in Spritebox the player besides the visual commands, has the choice to write a program in Java or Swift arranging visually the corresponding commands. Furthermore, in CodeSpells the player writes programs in Java and in the last levels of Rapid Router the player has the ability to write commands in Python and see the equivalent visual program in Blockly. Additionally, in Kodable the player has the ability to see the corresponding Javascript code when s/he creates a new object and furthermore s/he can modify the avatar's properties (for example, the body color) using JavaScript, while in Getcoding the player arranges visual commands forming a flow chart that corresponds to his/her program.

Finally, in the games CodeFruits and MiniColon the player writes lines of code using hand gestures as the basic input method.

*Table 5. Language supported*

| Games | Programming Language | Visual Symbol Icons | Visual Text Blocks | Other Representation |
|---|---|---|---|---|
| **Adventure Code** | | √ | | |
| **Bookworms** | | √ | | Free form drawing tool to design a path |
| **BOTS** | | | √ | |
| **CodeCombat** | Python; Javascript; Coffee Script | | | |
| **CodeFruits** | | | | Hand gestures which represent different functions or parameters of a programming language |
| **CodeMonkey** | Coffescript | | | |
| **CodeSpells** | Java | | | |
| **Code Venture** | | √ | | |
| **Dragon Architect** | | √ | | |
| **Getcoding** | | | √ | |
| **Kodetu** | | √ | | |
| **Kodable** | Javascript | √ | | |
| **Lightbot 2.0** | | √ | | |
| **MiniColon** | Python | | | |
| **OOP Serious Game** | | √ | | |
| **Pirate Plunder** | Scratch | | | |
| **Program Your Robot** | | √ | | |
| **Rapid Router** | Python | √ | | |
| **Run Marco** | | √ | | |
| **Software Kids** | | √ | | Select true or false options |
| **Spritebox** | Java; Swift | √ | | |
| **World of variables** | | | | √ |

## Program Execution: Test and Debug

Facilities offered by each game in order to test and debug a solution are presented in Table 6. The majority of the games provide mechanisms in order to test and debug the players' program. Ten out of twenty two games *highlight the command* that is being executed. Furthermore, seven games allow program execution at *different speeds* while in Pirate Plunder and Run Marco the block execution is slowed down to make it easier for the player to track and trace the errors in the program. Fifty percent (50%) of the games provide the player with *corrective feedback* corresponding to an error in the program, while Kodable highlights the location of the error.

In the majority of the games testing and debugging is only possible through running the program, and receiving real-time feedback about how the code is affecting the avatar's actions. In addition, Program Your Robot incorporates a "debug mode", which allows the player to debug his/her solution (choosing a debug button) and in the case of a logical error, explanatory messages appear for helping him/her in correcting it (Kazimoglu et al., 2012). BOTS, Dragon Architect and Rapid Router are the only games that allow a *step-by-step execution* of commands.

Also, BOTS offers some more facilities compared to the other games. Specifically, in the activities where students must fix errors in a buggy program, the game highlights students' changes to differentiate them from the original code, helping this way the students, to keep track of changes. Another significant feature is that the game displays the original code that the player deletes as being crossed out while in case of new code that has been added from the player the code is disappeared, allowing this way the player not to lose focus (Liu, Zhi, Hicks, & Barnes, 2017)

Finally, ten games allow the player to stop or pause the program execution, while some of them allow level restart.

## Support for Online Classrooms and Level Editor

Table 7 presents some enhanced features of the games such as support for online classrooms or creating customized game levels. Five games give the teachers access to a *dashboard* where they can create *online classrooms*, providing detailed students' progress. In addition, in Kodable the teacher can choose and assign specific activities of the game to a class and moreover CodeCombat allows the teacher to use practice levels that adapt to each student's needs, when they struggle with a level.

Nine games support the creation of new levels. Furthermore, in AdventureCode, CodeCombat, Dragon Architect, Kodetu and Lightbot 2.0, the players can create their own puzzles trough a *level editor* tool and share them with friends and the community. In Kodable after completing the first lesson of conditions, the Maze Maker tool is unlocked and gives the player the opportunity to create new mazes, while in Rapid Router both the teacher and the student can create new levels. Additionally, in BOTS the players can share the puzzles they create with their peers, but also they can play and evaluate their friends' puzzles and improve past solutions.

## Available Platforms

Table 8 presents the available platforms that are mainly used for the distribution of each game. Several games are on-line accessible free, through a modern web browser while some of them are also available as mobile Android or iOS apps. However, some of them provide freely only some of their levels or features and require a subscription in order to have full access. For Lightbot 2.0, which is available through the web, there are other versions of the game e.g., Lightbot Hour of Code (www.lightbot.com), Lightbot Jr 4+ which are available through mobile apps for Android or iOS. Furthermore, CodeFruits, CodeSpells and MiniColon are distributed only through a desktop version, while Kodable also supports this feature.

Table 6. Program execution – Test and debug

| Games | Highlighting the command being executed | Run a program at different speeds | Corrective feedback in case of an error | Testing and debugging | | | | Program execution | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Through running the program | Fix errors in a given program | Step-by-step execution | Other | Stop | Restart level | Pause |
| **Adventure Code** | | | | | | | | | | |
| **Bookworms** | | √ Player can speed up the animation | √ (partially) | | | | Try and error approach | √ | √ | |
| **BOTS** | √ | | √ Pop up red messages explain errors | √ | √ | √ | | √ | | |
| **CodeCombat** | √ | | √ | √ | | | | √ | √ | |
| **CodeFruits** | | | | √ | | | | | √ | |
| **CodeMonkey** | √ | √ | √ | √ | | | | √ | √ | |
| **CodeSpells** | | | | | | | . Players can execute spells to test them out using the spellbook; Players can modify existing code through the in-game IDE | | | |
| **Code Venture** | | | | | | | | | | |
| **Dragon Architect** | √ | √ | | √ | | √ | | | | √ |
| **Getcoding** | | | √ | √ | | | | | √ | |
| **Kodetu** | √ | | √ Messages explain the errors | √ | | | | √ | | |
| **Kodable** | √ | √ | √ Highlights the location of the error | √ | | | | | | |

continues on following page

*Table 6. Continued*

| Games | Highlighting the command being executed | Run a program at different speeds | Corrective feedback in case of an error | Testing and debugging | | | | Program execution | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Through running the program | Fix errors in a given program | Step-by-step execution | Other | Stop | Restart level | Pause |
| **Lightbot 2.0** | √ | √ | | √ | | | | √ | | |
| **MiniColon** | | | √ With appropriate messages | √ | | | Code tracing levels | | | |
| **OOP Serious Game** | | | | | | | | | | |
| **Pirate Plunder** | | Block execution is slowed down to make it easier for the player to debug the program | √ Through the green parrot avatar | √ | √ | | | | | |
| **Program Your Robot** | | | √ | √ | | | Debug mode | √ | | |
| **Rapid Router** | √ | √ | √ | | | √ | | √ | | |
| **Run Marco** | √ | Block execution is slowed down | | √ | | | | | | |
| **Software Kids** | | | | | | | | | | |
| **Spritebox** | √ | √ Pressing Space key | | √ | | | | | | √ |
| **World of variables** | | | | | | | | | | |

*Table 7. Online classrooms – Creation of new levels*

| Games | Online classrooms | Create new levels |
|---|---|---|
| **Adventure Code** | | √ |
| **BOTS** | | √ |
| **CodeCombat** | √<br>Teacher's dashboard provides detailed student progress, completion time and overall class statistics, and also allows teachers to assign and manage assignments | √ |
| **CodeMonkey** | √ | √<br>(With subscription) |
| **Code Venture** | √<br>Teacher can add/remove pupils to a class and check their progress and scores;<br>Teacher can change the difficulty for each pupil if she/he feels they are struggling | Not Mentioned |
| **Dragon Architect** | | √<br>The players can share what they build in the sandbox in a communal gallery and browse, view and download others' creations. |
| **Kodetu** | | √<br>The player can create levels and publish them to the games community |
| **Kodable** | √ | √ |
| **Lightbot 2.0** | | √ |
| **Rapid Router** | √ | √ |

## CONLUSION

Throughout this chapter a comparative analysis of twenty two educational games that aim to teach basic programming concepts and cultivate CT aspects to novice programmers was presented. Regarding the scenario, we can conclude that all the games utilize an interesting scenario in order to motivate the students in carrying out programming activities in a playful way. The majority of them do not offer a wide variety of activities, implementing mainly a scenario in which the player must program the movement of a character in order to reach a specific destination with only seven games offering additional types of activities, like fixing the errors in a given program, creating or modifying objects etc. However, a more active interaction of the player with the virtual world is a feature that seems to be missing from these games.

Regarding the games' genre, the majority of them are classified as puzzle games. Furthermore, almost all the games support the player during game play with different ways in order to pass a level. Among others, audio and visual cues, tutorial levels, appropriate help messages and video explanations regarding the rules of the game, or the functionality of the user interface and the programming constructs being introduced, are the main techniques used.

As for rewarding methods, only in eight games there is a total score. Furthermore, seven games reward the player by displaying messages after successfully completing an activity and four games use a three star system depending on how well the activity ended. Furthermore, some games provide an integrated reward system of achievements.

Regarding the educational content, with the exception of two games focused solely on teaching OOP and one involving only activities related to the different types of variables, all the other games support sequence commands execution. The majority of them support iterative structures while ten games support if statement and functions. Furthermore, Lightbot 2.0, Pirate Plunder and Program Your Robot are the only games that cover recursion and seven games support activities related to OOP.

Additionally, the games' activities are characterized by reasonable increasing complexity and they do not require prior knowledge from the player. Therefore, taking into account the target age of the games according to their designers, the authors believe that all of them could be used to teach CT aspects to primary school students, although some of them, such as CodeSpells, could also be used at a higher level of education.

*Table 8. Available platforms*

| Games | Web | Android | iOS | Desktop | Freely available online |
|---|---|---|---|---|---|
| **Adventure Code** | √ | √ | √ | | |
| **Bookworms** | √ | Not mentioned | Not mentioned | Not mentioned | |
| **BOTS** | √ | | | | |
| **CodeCombat** | √ | | | | √ |
| **CodeFruits** | | | | √ | |
| **CodeMonkey** | √ | | | | √ |
| **CodeSpells** | | | | √ | |
| **Code Venture** | √ (teacher) | √ (student) | √ (student) | | |
| **Dragon Architect** | √ | | | | √ |
| **Getcoding** | √ | √ | 5.1.1 or later | | √ |
| **Kodetu** | √ | | | | √ |
| **Kodable** | √ | | 9.0 or later | √ | √ |
| **Lightbot 2.0** | √ | √ (in other versions) | √ (in other versions) | | √ |
| **MiniColon** | | | | √ | |
| **OOP Serious Game** | | √ | √ | | |
| **Pirate Plunder** | √ | | | | |
| **Program Your Robot** | √ (not this time) | | | | |
| **Rapid Router** | √ | | | | √ |
| **Run Marco** | √ | √ | √ | | √ |
| **Software Kids** | | √ | | | |
| **Spritebox** | √ | √ | √ | | √ |
| **World of variables** | √ | | | | |

Regarding CT, it was found that the games support the cultivation of skills like *algorithmic thinking, decomposition, pattern recognition* and *modularity, abstraction* and *generalization.*

As far as the programming language of the games is concerned, there is a predominance of block based environments, where the program is written visually, allowing this way the students to practice with programming without syntax errors or memorizing commands (Vahldick et al., 2014). Additionally Rapid Router and Getcoding allow bidirectional conversion between visual and textual programming language while, flowcharts, a visual tool widely used in learning programming at the conceptual level (Eguíluz et al., 2018), are used only in Getcoding.

The majority of the games provide facilities in order to test and debug a solution. The most common techniques used by the games to help players track and trace the errors in the program are: highlighting the command that is being executed, offering choices for different speed at program execution, corrective feedback in case of an error. In the majority of the games, testing and debugging is only possible through running the program and watching how the code is affecting the avatar's behavior.

Furthermore, only five games support online classrooms providing an online dashboard where the teacher derives information especially detailed in some cases about students' progress while nine games allow building new levels through a game level editor. User generated content such as new level creation, could contribute in replay ability of the game and increase players' interest (Hicks et al., 2014), players' experience and not limit the learning cycle (Eguíluz et al., 2018). Additionally, the authors' perception is that this feature (creation of new levels) combined with the existence of ready-made educational material like exercises related to the game for each lesson, would motivate teachers to incorporate such games in their courses and furthermore, can allow teachers to create their own curriculum adapted to the needs of a class.

Regarding the available gaming platforms, the most common way of distributing such games nowadays is mainly via web platforms, while some of them are also distributed through mobile apps.

With the exception of CodeCombat none of the games presented in this study support collaboration between players during game play and have been designed for a single player. Multiplayer or collaborative games are more motivating and engaging than single player games (Combéfis et al., 2016), offer guidance to the players by getting help from the community (Bauer et al., 2017) and moreover, create an environment that is safe and socially stimulating for all students entering computer science (Barnes, Richter, Chaffin, Godwin, Powell, Ralph, Matthews, & Jordan, 2007). Therefore, this feature should be taken into account when designing an educational game (Malliarakis et al., 2014a).

Moreover, even the games that support online classrooms and monitoring students' progress, do not offer or offer to a small degree learning analytics (Malliarakis, Satratzemi, & Xinogalos, 2014c). Pirate Plunder is the only game that seems to offer analytics for several players' actions concerning the games' section change, the time spent on a level or the number of attempts on a level. In addition, CodeMonkey reports the last time that the player finished a level and Rapid Router, reports the total time needed for the player to finish a level and the exact time s/he started and finished it.

Learning analytics could assist teachers effectively in monitoring students' progress, to investigate users' behavior and how these games facilitate their learning (Eguíluz et al., 2018) and more importantly to support the teachers in evaluating students' comprehension of the concepts taught (Malliarakis et al., 2014c).

In addition, these games do not offer assessments for the knowledge gained by the players (Eguíluz et al., 2018). Although some games somehow evaluate the players' knowledge, either through the player's score or by counting the number of commands used or by assigning a number between one to three stars depending on the solutions' efficiency, assessment in these games should be improved (Eguíluz et al., 2018).

## FUTURE RESEARCH DIRECTIONS

This study could be a starting point for researchers, to carry out experimental studies for investigating the games effect on students' comprehension of programming concepts and moreover to investigate teachers' acceptance of utilizing such games in the teaching and learning of programming and the acquisition of CT skills. This could contribute to the limited number of empirical research that investigate the true impact of educational games on teaching and learning programming, especially to small aged students (Fesakis, Gouli, & Mavroudi, 2013).

Moreover, it would also be interesting for researchers to identify open problems and features that are currently missing from the games analyzed, such as learning analytics, collaboration during game play, enhanced interaction between the player and the virtual world, support for creating customized lesson plans, and incorporate them into a new, more integrated educational game platform that supports the teaching of CT more effectively.

## REFERENCES

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75B*, 661–670. doi:10.1016/j.robot.2015.10.008

Ayman, R., Sharaf, N., Ahmed, G., & Abdennadher, S. (2018). MiniColon; Teaching Kids Computational Thinking Using an Interactive Serious Game. In. Lecture Notes in Computer Science: Vol. 11243. *Serious Games. JCSG 2018*. Springer; doi:10.1007/978-3-030-02762-9_9

Barnes, T., Richter, H., Chaffin, A., Godwin, A., Powell, E., Ralph, T., ... Jordan, H. (2007). Game2learn: a study of games as tools for learning introductory programming concepts. In *Proceedings of the ACM Special Interest Group on Computer Science Education.*. New York, NY: ACM.

Becker, T. (2010). The Character of Successful Trainings with Serious Games. *International Journal of Emerging Technologies In Learning, 5*(SI3). Retrieved April 17, 2012, from http://online-journals.org/i-jet/article/view/1498

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages: A way to learn programming principles. *Education and Information Technologies*, *2*(1), 65–83. doi:10.1023/A:1018636507883

Combéfis, S., Beresnevičius, G., & Dagienė, V. (2016). Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, *10*(1), 39–60. doi:10.15388/ioi.2016.03

de Freitas, S., & Jarvis, S. (2006). A Framework for Developing Serious Games to meet Learner Needs. *Interservice/Industry Training*, *Simulation and Education Conference*.

Eguíluz, A., Garaizar, P., & Guenaga, M. (2018). An Evaluation of Open Digital Gaming Platforms for Developing Computational Thinking Skills. In *Simulation and Gaming*. InTech. doi:10.5772/intechopen.71339

Eguíluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodríguez, C. (2017). Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis. *IEEE Transactions on Emerging Topics in Computing*, 1–1. doi:10.1109/TETC.2017.2768550

Esper, S., Foster, S. P., Griswold, W. G., Herrera, C., & Snyder, W. (2014). *CodeSpells: bridging educational language features with industry-standard languages*. Koli Calling. doi:10.1145/2674683.2674684

Fesakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97. doi:10.1016/j.compedu.2012.11.016

Gal, L., Hershkovitz, A., Eguíluz, A., Guenaga, M., & Garaizar, P. (2017). *Suggesting a Log-Based Creativity Measurement for Online Programming Learning Environment*. doi:10.1145/3051457.3054003

Giannakoulas, A., & Xinogalos, S. (2018). *A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. In Education and Information Technologies*. Springer. doi:10.100710639-018-9702-x

Goyal, S., Chopra, S., & Mohanan, D. (2017, October). CodeFruits: Teaching Computational Thinking Skills Through Hand Gestures. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play* (pp. 291-298). ACM. https://dl.acm.org/citation.cfm?id=3131335

Hicks, A., Catete, V., & Barnes, T. (2014). Part of the game: Changing level creation to identify and filter low-quality user generated levels. In *Proceedings of the international conference on the foundations of digital games* (pp. 242-245). ACM.

Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. doi:10.1016/j.compedu.2018.07.004

Hussain, A. J., Fergus, P., Al-jumeily, D., Pich, A., & Hind, J. (2015). Teaching Primary School Children the Concept of Computer Programming. *2015 International Conference on Developments of E-Systems Engineering*, 180–184. 10.1109/DeSE.2015.57

Kalovrektis, K. (2015). Getcoding for everyone - The first steps in the world of the Code. Academic Press.

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522–531. doi:10.1016/j.procs.2012.04.056

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and Higher Education*, *8*(1), 13–24. doi:10.1016/j.iheduc.2004.12.001

Korpi, J. (2014). *Teaching programming to children through games* (Master of Science Thesis). Tampere University of Technology.

Laporte, L., & Zaman, B. (2016). Informing Content-driven Design of Computer Programming Games: a Problems Analysis and a Game Review. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (p. 10). ACM. 10.1145/2971485.2971499

Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, *27*(1), 1–29. doi:10.1080/08993408.2017.1308651

Lotfi, E., & Mohammed, B. (2018). Teaching Object Oriented Programming Concepts Through a Mobile Serious Game. *Proceedings of the 3rd International Conference on Smart City Applications*. 10.1145/3286606.3286851

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014a). Designing educational games for computer programming: A holistic framework. *Electronic Journal of e-Learning*, *12*(3), 281–298.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014b). Educational games for teaching computer programming. In *Research on e-learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 87–98). Springer.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014c). Integrating learning analytics in an educational MMORPG for computer programming. In *Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies*. Athens, Greece: IEEE Computer Society Press.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2017). CMX: The effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, *10*(2), 219–235. doi:10.1109/TLT.2016.2556666

Miljanovic, M. A., & Bradbury, J. S. (2018). A Review of Serious Games for Programming. In Lecture Notes in Computer Science: Vol. 11243. *Serious Games. JCSG 2018*. Cham: Springer. doi:10.1007/978-3-030-02762-9_21

Moreno-León, J., Román-González, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. *2018 IEEE Global Engineering Education Conference*, 1684-1689. 10.1109/EDUCON.2018.8363437

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, *1*(1), 95–123. doi:10.1007/BF00191473

Ramirez-Rosales, S., Vazquez-Reyes, S., Villa-Cisneros, J. L., & De Leon-Sigg, M. (2016). A serious game to promote object oriented programming and software engineering basic concepts learning. *Proceedings - 2016 4th International Conference in Software Engineering Research and Innovation, CONISOFT*, 97–103. 10.1109/CONISOFT.2016.23

Rose, S. P., Habgood, J., & Jay, T. (2018). Pirate plunder: game-based computational thinking using scratch blocks. In *Proceedings of the 12th European Conference on Games Based Learning*. Academic Conferences and Publishing International Limited, 556-564.

Sanchez, E. (2011). *Key criteria for Game Design. A Framework. MEET Project*. European Commission.

Scheuerman, J. (2015). AdventureCode: Computational Thinking Through Games. In S. Carliner, C. Fulford & N. Ostashewski (Eds.), *Proceedings of EdMedia 2015--World Conference on Educational Media and Technology* (pp. 1832-1837). Montreal, Quebec, Canada: Association for the Advancement of Computing in Education (AACE). Retrieved December 10, 2019 from https://www.learntechlib.org/primary/p/151458/

Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. In *Frontiers in Education Conference (FIE)* (pp. 1–7). IEEE. 10.1109/FIE.2014.7044114

Wing, J. M. (2006, March). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Xinogalos, S. (2013). Using Flowchart-based Programming Environments for Simplifying Programming and Software Engineering Processes. In *Proceedings of 4th IEEE EDUCON Conference* (pp. 1313–1322). Berlin, Germany: IEEE Press. 10.1109/EduCon.2013.6530276

Xinogalos, S., & Satratzemi, M. (2004). Introducing novices to programming: a review of teaching approaches and educational tools. In *Proceedings of the 2nd International Conference on Education and Information Systems, Technologies and Applications* (*Vol. 2*, pp. 60-65). Academic Press.

Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education*, *47*(2), 148–171. doi:10.1016/j.compedu.2004.09.005

Yusoff, A., Crowder, R., Gilbert, L., & Wills, G. (2009). A Conceptual Framework for Serious Games. *The 9th IEEE International Conference on Advanced Learning Technologies*, 21-23. 10.1109/ICALT.2009.19

Zaharija, G., Mladenović, S., & Boljat, I. (2013). Introducing basic programming concepts to elementary school children. *Procedia: Social and Behavioral Sciences*, *106*, 1576–1584. doi:10.1016/j.sbspro.2013.12.178

Zapušek, M., & Rugelj, J. (2013). Learning programming with serious games. *EAI Endorsed Transactions on Game-Based Learning*, *1*(1), e6. doi:10.4108/trans.gbl.01-06.2013.e6

## ADDITIONAL READING

Bauer, A., Butler, E., & Popovic, Z. (2015). Approaches for teaching computational thinking strategies in an educational game: A position paper. *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond),* 121-123.

Bauer, A., Butler, E., & Popovic, Z. (2017). Dragon architect: open design problems for guided learning in a creative computational thinking sandbox game. In *Proceedings of FDG'17*, *Hyannis, MA, USA*, August 14-17, 2017, 1-6. DOI: 10.1145/3102071.3102106

Fesakis, G., & Serafeim, K. (2009). Influence of the Familiarization with "Scratch" on Future Teachers' Opinions and Attitudes about Programming and ICT in Education. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE-2009), Paris, France*, 6-8 July, 2009, Vol II, ACM, New York, NY, USA, pp. 258-262. 10.1145/1562877.1562957

Gibson, B., & Bell, T. (2013). Evaluation of games for teaching computer science. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, (pp. 51–60). New York, NY, USA: ACM. 10.1145/2532748.2532751

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. doi:10.1016/j.compedu.2018.07.004

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Kiernan, M., Kazimoglu, C., Bacon, L., & Mackinnon, L. (2014). Developing an Educational Game to Support Cognitive Learning. *Compass: Journal of Learning and Teaching*, *5*(9), 2014.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014a). CMX: Implementing an MMORPG for learning programming. In *Proceedings of 8th European Conference on Games Based Learning* (pp. 346–355). 9–10 October 2013, Berlin, Germany.

Papadakis, S. (2018). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Piteira, M., & Haddad, S. R. (2011). Innovate in your program computer class: an approach based on a serious game. In *Proceedings of Workshop on Open Source and Design of Communication*. pp. 49–54 (2011) 10.1145/2016716.2016730

## KEY TERMS AND DEFINITIONS

**Computational Thinking:** Is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could execute.

**Computer Programming:** Is the process of designing and building an executable computer program to accomplish a specific computing result.

**Novice Programmer:** A computer programmer who is not experienced in programming.

**Primary Education:** Is typically the first stage of formal education, coming after preschool and before secondary school.

**Serious Game (SG):** Is a game designed for a primary purpose other than pure entertainment.

# Chapter 2
# Evaluating a Teaching Intervention for Teaching STEM and Programming Concepts Through the Creation of a Weather-Forecast App for Smart Mobile Devices

**Stamatios Papadakis**
https://orcid.org/0000-0003-3184-1147
*Department of Preschool Education, University of Crete, Greece*

## ABSTRACT

*The last two decades have necessitated the need for an interdisciplinary approach to mathematics, science, and technology (STEM) as contemporary problems are too multidimensional to be tackled by a single scientific discipline as was the case with classical school curricula. Teaching programming has the potential to contribute to this vision as it is effective in helping students develop critical thinking skills. This work presents an educational approach that combines STEM learning with the basic concepts of programming through the creation of a weather-forecast app for smart mobile devices with the programming environment MIT App Inventor. This approach was implemented with second grade high school students as a school project. The evaluation results are considered encouraging as the students engaged in authentic learning activities and research related to the STEM field while, at the same time, enhanced their interest and knowledge in pursuing careers involving programming, science, technology, engineering, and mathematics.*



DOI: 10.4018/978-1-7998-4576-8.ch002

## INTRODUCTION

Equal digital citizenship is a presumptive value in the modern world. But to be digitally enough in the modern world requires not only access to and use of ICTs (Information and Communication Technologies) but also appropriate knowledge, skills and attitudes (Vuorikari, 2015). Besides, over the last decade there is an urgent need for an effective and interdisciplinary approach to Mathematics, Science and Technology (STEM) while modern problems are complex and multi-dimensional to be addressed by a single science discipline as was the case with classical school curricula for decades (Morrison & Bartlett, 2009). STEM seeks to transform the traditional teacher-centered teaching and learning approach to an alternative approach where problem-solving, creative thinking, discovery-exploration learning, and other high-level cognitive skills play a key role, while a direct and active learner involvement is required to obtain a subsequent behavior change (Dorouka, Papadakis & Kalogiannakis, 2019). This approach also avoids some of the common challenges to the use of creative programming (Wing, 2006). Within this approach of programming, which is not only focused on the techniques to code a program, different components which are related to the creative problem-solving process such as collaboration, creativity, communication, innovation, and critical thinking (Orfanakis & Papadakis, 2016) are utilized.

The development of STEM and programming skills to successfully negotiate the 21st-century information society, besides learning, has become a key competence that is addressed to all students. The reason is that they provide them with the necessary skills for their future profession (European Schoolnet, 2014). However, while one might expect students would be more likely to embrace programming courses in the context of formal education, evidence around the globe suggests that such a trend is not observed. Data from around the world, specifically in the United States and Europe, show that an increasing number of students are choosing not to pursue the programming disciplines in secondary schools and higher education (Papadakis & Kalogiannakis, 2017).

Computing has been introduced as a mandatory subject in primary and secondary school Greek curricula for decades. With the recent Computer Science curricula in Primary and Secondary Education (http://ebooks.edu.gr/new/), even the rigorous curriculum of the Greek school enables the teacher to plan and control his/her activities, to make use of readymade appropriate educational material and computer resources and to create his/her material as well. Teachers are also suggested to use real-world examples that demonstrate the interdisciplinary nature of modern disciplines and/or scenarios that enrich students' knowledge from previous classes and courses in math, physics, and electronics. This approach can be a very powerful way to engage students in STEM and coding activities and to help them appreciate why an understanding of STEM and coding matters to their lives. This science-based approach is not just logical; it engages students hands-on and inquiry-based activities to develop apps using developmentally appropriate learning programming environments for novices such as MIT App Inventor 2 (Papadakis et al., 2017).

This paper presents an educational intervention that combines the basic concepts of programming and STEM learning via the creation of a weather-forecast application (app) for smart mobile devices using the MIT App Inventor 2 programming environment. This paper is an extension of the author's paper presented at the 6th Panhellenic Scientific Conference 'Integration and Use of ICT in the Educational Process' in the Greek language. In this article, the author presents the rationale and design of the study, as well as the relevant result. The rest of the paper is organized as follows. The next section presents the current trends of empirical research in the necessity of programming, the difficulties that novice programmers experience during programming courses as well as the use of block-based programming environments as an alternative and efficient teaching approach in programming education. Section 2 describes the teaching intervention. Finally, Sections 3, 4 and 5 present the limitations, the results and conclusions of the current study.

## THEORETICAL FRAMEWORK

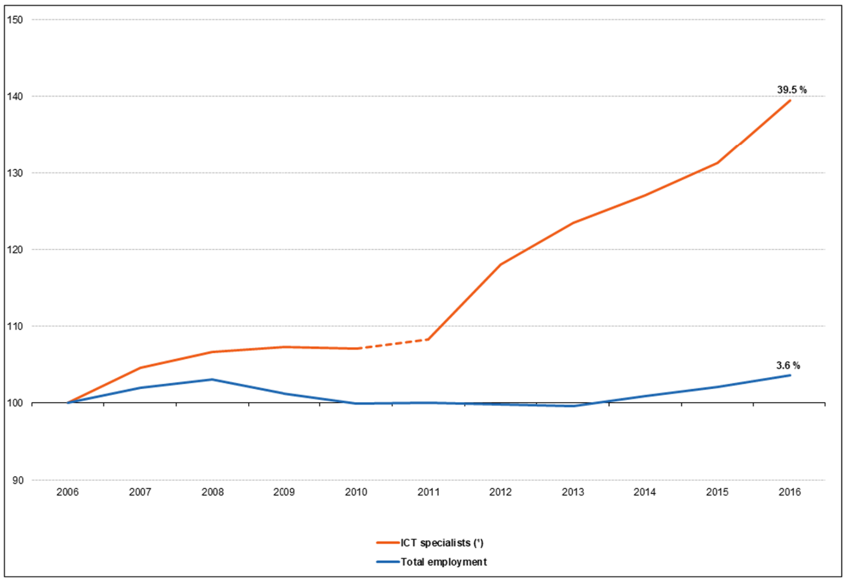
### The Necessity of Programming

The 21st-century skills are referred to a wide range of skills, such as learning and innovation skills; information, media and technology skills (Qian & Clark, 2016, p.51). As stated by the Organization for Economic Cooperation and Development (OECD, 2013), the growing need for more sophisticated skills suggests that individuals with poor 21st-century skills are more likely to find themselves at risk of unemployment and social exclusion as these skills are considered as universal in our digitized life. Programming and Computational thinking (CT) have become the center of attraction for many researchers around the world in the last decade due to the role they have to play for our 21st-century learners. Today students must understand that technology is everywhere (Wang, 2017). Furthermore flexibility, critical thinking, problem-solving, collaboration, and communication are skills that are likely to be useful in future careers. The Computer Science Teachers' Association (CSTA) taskforce in the United States supports the argument for the development of coding and computational thinking (CT) skills by pointing to their value for increasing students' higher-order thinking skills and general problem-solving abilities (Falloon, 2016, p.580). Thus, they are no longer considered the stereotypical impression of the specialized skill required by software engineers (Hsu, Chang & Hung, 2018).

Computer Science (CS) is an academic discipline that builds its own body of knowledge and can equip and motivate students to become independent thinkers, innovators, and developers of new ideas and technologies. In studying CS, students of all ages expect to gain skills, knowledge, and a way of thinking about and solving problems: Computational Thinking (CT). As Chen and Huang (2017) state programming is an excellent way to develop CT skills, because it involves problem-solving using computer science concepts such as abstraction, debugging, remixing and iteration. CT cognitive and metacognitive strategies such as abstraction and modeling intersect with CS concepts such as algorithms, automation, and data visualization (Csizmadia et al., 2015). For Wing (2006, p.33), CT represents 'a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.' This demand is not only relevant to science, technology, engineering, and mathematics jobs and disciplines (STEM) but nearly all jobs and positions worldwide (Kalogiannakis and Papadakis, 2018a, 2018b). As Cheng (2019) points out different domains such as robotics, mobile programming, etc., can be used by students and/or novice programmers to improve their understanding of programming concepts and develop their CT skills.

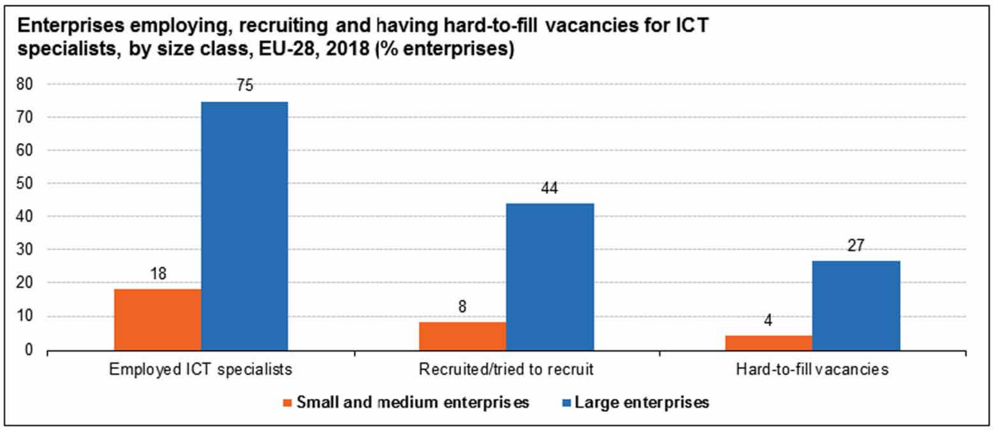
Besides gaining theoretical knowledge, students need to be prepared to accept positions in the information technology (IT) field that does not exist yet (Jordaan, 2018). The IT sector is characterized as an area with considerable potential for professional development, as it generates economic growth, and as a result, jobs, with the potential to drive radical technological change and an economic revival in all of the countries (European Schoolnet, 2014) (see Figure 1).

*Figure 1. Persons employed as ICT specialists and total employment, EU-28, 2006–2016 Source: Eurostat, 2019a).*



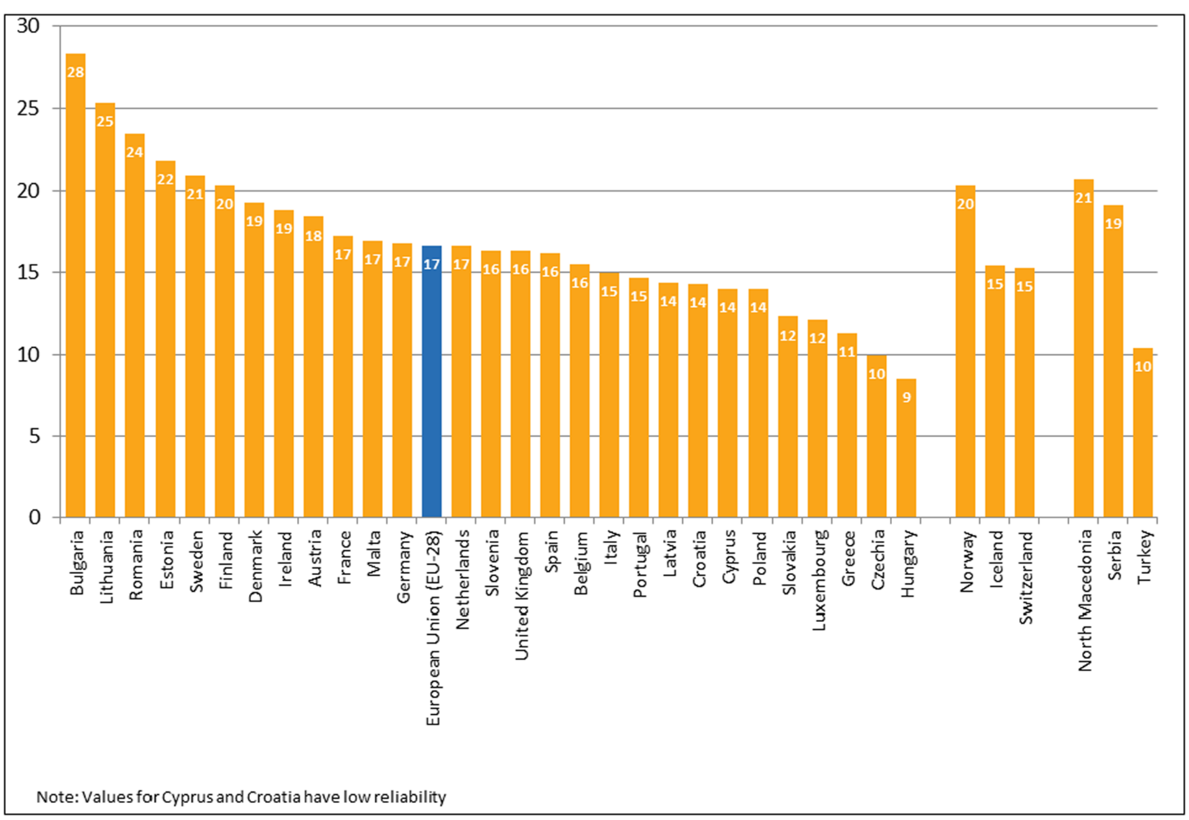

According to the British Institute SFIA Foundation (Skills Framework for the Information Age), there are 86 different professional skills related to IT that corresponding to 290 different specialties (SFIA Foundation, 2017). Some of these areas include mobile application developers, web developers, cloud architects, software engineers, etc. According to Eurostat, in 2017, more than half of EU enterprises that recruited or tried to recruit ICT specialists had difficulties in filling ICT vacancies (Eurostat, 2019b) (see Figure 2).

*Figure 2. Enterprises employing, recruiting and having hard-to-fill vacancies for ICT specialists, EU-28 Source: Eurostat, 2019b*



At the same time, across the EU-28, most jobs for ICT specialists in 2018 were held by men; In 2018 around 8.9 million people were employed in the European Union (EU) as ICT specialists. The profession was predominately male, as only 17% (1.5 million) of the ICT specialists were women (Eurostat, 2019b) (see Figure 3). The low presence of women in the ICT industry is partially the result of previously low enrolment in technological and scientific careers (Cussó-Calabuig et al., 2018). In pursuit of a professional and academic career in the STEM fields, women are significantly under-represented in science and engineering throughout Europe, despite these sectors offering well-paid opportunities. The gender-based digital divide is particularly pronounced in science, technology, engineering, and mathematics (STEM) disciplines, with only 29 in 1,000 female graduates holding a computer science degree, and - of those 29 - only four choose to work directly with ICT (Papadakis, Tousia & Polychronaki, 2018; Sangiuliano, 2015). The role that school is called upon to play is therefore crucial to breaking down gender stereotypes, giving students, and especially females, greater freedom of choice for education and employment (Papadakis, 2018a; Sangiuliano, 2015). In this context, it is particularly important to proactively educate teachers and students regarding the potential gender dimension in STEM education and professional fields. For example, it has been found that evidence-based teaching practices such as exploratory methodologies stimulate girls' interest in science, tech, engineering, and math - fields where they're still in the minority (Sangiuliano, 2015).

*Figure 3. ICT specialists are predominantly male (Source: Eurostat, 2019c)*



## Novice Programmers

It is accepted that to succeed in the workplace, today students and potential employees must possess a base of skills such as communication, creative and critical thinking skills, as described as 21st-century skills. In many cases, the use of ICT in schools simply reinforces old ways of teaching and learning (Papadakis, Kalogiannakis, Orfanakis, & Zaranis, 2014, 2016, 2017). On the other hand, by their very nature, activities such as computer programming are particularly important for the 21st-century student competing in a global economy. By learning how to program students cultivate a variety of skills, such as being able to express their views more fully and creatively, to develop a logical, rational way of thinking and understanding how the machine works, which are scattered throughout their daily lives. In recent years, there has been a growing interest in the field of STEM at the international level. That interest derives from the overwhelming need to cultivate critical and creative thinking for students and future professionals. It is widely known that the motivation for introducing and teaching programming in schools is not fueled by the need to create a generation of programmers. Similarly, implementing STEM concepts the intention is not to promote engineering and related skills but to equip students with the skills and abilities they'll need for the future workplace.

Although, in the last two decades, there has been a widespread interest in equipping primary and secondary school-age students with the necessary knowledge and skills required for programming (Cheng, 2019), recent studies have underlined the fact that the dropout and failure rates are much higher in computer science (CS) classes as compared to other classes. Various other researchers have also highlighted, paradoxically, the lack of interest many students show in CS classes. This is not following the increasing demand for information and communication technology (ICT) specialists (Bennedsen and Caspersen, 2007; Watson and Li, 2014). For instance, the number of ICT specialists in the European Union (EU) grew by 36.1% from 2007 to 2017, over ten times as high as the increase (3.2%) for total employment (Eurostat, 2019a). The lack of interest many students show even for introductory programming courses is usually attributed to the fact that the students find programming difficult as well as full of tedious theoretical concepts and methodologies (Bennedsen and Caspersen, 2007). Of course, programming is a hard skill to learn. As Blikstein (2018, p.32) summarizes, programming requires students to:

- understand what computers can do and how they run programs
- develop, trace, and debug code
- steer away from several categories of misconceptions
- manage cognitive load
- understand the terminology, notations, and conventions
- from other scientific domains and understand their internal overlaps and contradictions with CS
- work in long-term projects and environments that are drastically different from that of conventional classrooms

Novice programmers range from being the ones who have never previously experienced programming to those who may have some basic background in programming, attained informally or via formal study in pre-university contexts (Shi et al., 2018). Often, during programming courses, students or novice programmers fail to apply even simple ideas to simple problems (Guzdial, 2018). Furthermore, in the 'traditional' or 'classical' approach to programming, novice programmers spend all their time trying to learn a rigid syntax of general-purpose programming languages or examples of exercises that are hostile and alien to them (Blikstein, 2018; Kelleher & Pausch, 2005). Concepts such as vectors, formulas, and equations seem to the eyes of young students as a world cut off from their regular lives, interests and the outside world.

Additionally, in general-purpose programming languages like Java, novice programmers engaged by learning strict syntax rules rather than to a deeper understanding of programming constructs and abstract concepts (Zapušek and Rugelj, 2013). Thus, novice programmers usually fail to combine abstract concepts into the design of a program, even though they know the syntax or the semantics of programming statements (Chiu, 2014). Furthermore, Guzdial (2018) points out that programming languages might lead to misconceptions among students because of flawed transfer from mathematics. He states, for instance, that 'Python functions have mutable data, use the '=' sign in a non-mathematical way, and this has side effects.' For the reasons mentioned above, it is easily understandable why learning computer programming is difficult and problematic for many novice programmers (Shi et al., 2018).

As CS education expands, teachers of CS are challenged to help students develop a definitive and concrete understanding of basic programming concepts (Qian and Lehman, 2017). Exploratory environments, tools, and programming languages have offered new opportunities to design more effective learning experiences. But, developing new pedagogies and approaches for teaching CS concepts is a challenge. There are several explanations for that (Blikstein, 2018), p.4):

- students must have a well-developed mental model of what computers are how to run the program and interpret its output.
- programming languages and related software tools are always changing, often leading to new pedagogies and methodologies that are harder to orchestrate.
- CS courses might not follow the same patterns as traditional school disciplines.

A solution to overcome these difficulties is to enhance the learning procedure with authentic learning experiences (Ouahbi et al., 2015). Recent research has supported the idea that this pedagogical methodology is likely to increase students' motivation and excitement for programming while it can bridge the gap between the digital and the real world, the digital youth culture, and the educational system. As in other areas of education, learning technologies are also gaining ground in STEM subjects: the integration of Web tools, multimedia applications, social media, and other digital services can enhance students' motivations and contribute to better learning outcomes (Cinganotto, 2017). Mobile technology may be a useful tool for teaching students STEM concepts. Studies have found that students like to create their applications and services for smart mobile devices, an activity that provides added value and meaning throughout the course work for students (Orfanakis & Papadakis, 2016; Papadakis et al., 2017).

## Programming Environments for Novices

In the last decades, researchers and educators are looking for didactic models using a variety of programming languages and environments, tools, techniques, appropriate and authentic examples as well as changes in the study content (Kelleher and Pausch, 2005). One of the most successful attempts to teach programming to novices is through the development and use of specialized environments for novices (Guzdial, 2004). As Guzdial (2004, p.127) points out, 'each novice programming environment is attempting to answer the question, what makes programming hard?' The programming environments for novices or 'initial learning environments' (ILE) (Fincher et al., 2010) are educational environments that aim to foster immediate engagement in an attractive activity, while allow novice programmers (irrespective of age, gender, and educational background) to create interactive applications, project, apps with elegant graphical user interfaces (GUIs). They provide novices with a visual toolkit to help improve their efforts by enabling them to quickly piece together relatively complex pipelines with only a few lines of code. Blocks-based programming environments are growing in popularity and are increasingly being used in formal introductory programming contexts (Weintrop, 2015)

These blocks-based programming environments consist of the use of graphical blocks instead of text commands (Papadakis, 2018b). This new graphical user interface (GUI) format allows novices to use a mouse pointer or their finger in the case of Scratch 3.0 to select commands from a menu of choices shown on a display screen and put together similar to a Jigsaw puzzle. As Weintrop and Wilensky (2015, p.200) note, in such environments, learners can assemble functioning programs by snapping together instructions and receiving visual (and sometimes audio) feedback informing the user whether a given construction is valid. In general, these environments do not require any specialized expertise beyond creative thinking to tell the computer what to do, step-by-step in a logical order. Block-based visual programming languages have the advantage of using shapes that fit properly only when they make a logical sequence of orders (Papavlasopoulou et al., 2019). This approach increases student interest in introductory programming and successfully lower the barriers faced by novice programmers. Some of the most successful environments are Scratch, Kodu, Logo, Greenfoot, and Alice (Maloney et al., 2010; Brennan and Resnick, 2012).

As educators and researchers, we must acknowledge that education and learning must keep pace with rapid technological changes. The latest entry into the world of ILE is the free web-based software called MIT App Inventor for Android (AI2). AI2 has taken advantage of the increasing popularity of smart mobile devices in students' daily life as well as the new role of these devices as learning tools in a variety of formal, non-formal and informal learning environments (Papadakis et al., 2014). Similar to Scratch, the free visual programming language developed by the MIT Media Lab, AI2 can help novice programmers to learn to program more easily compared to text-based programming languages and thus lowering the barrier to programming (Cheng, 2019). This learning environment is a very recent development, as it was created as a pilot project in Google's lab, in 2009. It is also widely known, that one of the core aspects of a learning activity is the fact that the problem should be meaningful to the learners and tailored to their needs. In AI2, the novice programmers do not create programs that run on personal computers, but on smart mobile devices running the Android operating system (smartphones/tablets) (Bers, 2017). This is one of the greatest advantages of using AI2 in the learning process. Students explore issues and construct artifacts that matter to them.

## MIT App Inventor

The MIT App Inventor (http://appinventor.mit.edu/explore/) is a programming environment designed for people with no previous programming experience to help them easily create mobile applications (apps). As already mentioned, it was initially created as a Google Labs research project by Professor Hal Abelson and a selected group of developers from Google Education and then 'adopted' by the MIT center for mobile learning as an open-source software package (http://appinventor.mit.edu/appinventor-sources/). Despite its relatively short period of existence, it is quite popular among certain groups of users. At the time of writing this article, (October 2019), there were almost 1.1 million monthly active users of a total of 8.2 million registered users who come from 195 countries (MIT App Inventor, 2019). The number of apps built is 34 million. In the latest version of App Inventor entitled App Inventor (AI2), the creation of a mobile application (app) can be done through a browser. This relieves the novice programmer from the extra effort to download/install any software (like Java in the first version of App Inventor) and make the necessary configurations such as the need to set or change the PATH system variable for Java. The AI2 environment consists of the following parts:

- The designer section allows the selection of a wide collection of components such as buttons, labels, sliding bars, and more for enabling the visible or non-visible parts of a mobile app. For instance, non-visible components are the accelerometer, the sound, and the orientation sensor. Instead visible components are the buttons, the images, the labels, etc. Once a component has been selected and added in the application user interface the programmer can either modify or change its properties and attributes.
- The block editor uses commands blocks. The programmer uses these blocks to define how the application responses to a variety of events that may be either user-driven (tap on the screen) or system-driven (e.g. changes in the status of the device accelerometer).

The AI2 allows the programmers to control app real-time functionality during development by using either the AI2 built-in emulator or an Android smart mobile device (phone or tablet). However, this programming environment also has limitations. Compared to the conventional textual programming languages and/or object-oriented programming languages, AI2, for instance, does not allow programmers to create their custom components (classes) and/or to handle the priority of the events (Gray et al., 2012; Wolber, 2011).

## Description of the Teaching Intervention

Our networked society creates new challenges, but it also offers a wealth of technologies that can support innovative pedagogies to improve teaching and learning (Dewey, 2018). In this study, we describe the introduction of a novel teaching intervention to improve student exposure to programming and CT skills and STEM concepts. During the intervention, students had to create a weather forecast app for smart mobile devices with the Android operating system. It is an idea tested in the 2017-2018 school year that combines the basic concepts of CT and programming along with the STEM concepts. The educational intervention was based on the social cognitive theory and modern approaches to 'information processing'. Besides, the teaching practice was student-centered, as, in all stages of the project implementation, the teacher followed a repertoire of instructional techniques that involved students in the educational process, such as brainstorming, exploratory approach, blackout, and error management.

The teacher, using appropriate supporting examples, sought for the students to realize and understand the various STEM concepts as well as to be able to evaluate and organize these concepts in mental structures. The concepts discussed in educational practice were negotiated through teamwork on app development. The MIT App Inventor 2 (AI2) was the programming environment, whose features were used as a springboard for young student's cognitive development, helping them build confidence and learn basic programming. At the same time, the use of real-world data enabled students to understand the interplay between empirical and theoretical probability, to understand the usefulness and importance of STEM concepts. Brief implementations in the AI2 environment in the computer lab provided students with the opportunity to create, test and debug their projects during various stages of app development while serving as a common reference point, a way of developing a comprehensive understanding of programming and STEM concepts. In practice, the following were utilized:

- the 'MIT App Inventor' programming environment from the Massachusetts Institute of Technology (MIT) (http://appinventor.mit.edu/explore/),
- the 'OpenWeatherMap' weather forecast service which provides free weather data (https://openweathermap.org/) through a free API (Application Programming Interface),
- the 'Appinventor.org' website, which pushes for free public access to educational materials (http://www.appinventor.org/course-in-a-box-intro),
- the 'IconArchive' website for the use of free additional multimedia content (http://www.iconarchive.com/),
- the 'Panhellenic Repository of Open Educational Practices for Primary and Secondary Education' (Photodentro) for finding open educational practices and resources (http://photodentro.edu.gr/oep/) and,
- the 'Advanced Electronics Scenarios Operating Platform' (AESOP) by the Institute of Educational Policy (http://aesop.iep.edu.gr/) for finding a selection of learning scenarios.

## The Different Stages of the Teaching Intervention

Activity 1: [Problem identification (the creation of a meteorological application)]

- Type of activity: group discussion, presentation, question and answer, brainstorming.
- Classroom organizing: working in groups.
- Learning objective: focusing on students' perceptions of utilizing an interdisciplinary approach for introductory programming, CT, and STEM concepts.
- Digital archives and educational content:
    - http://aesop.iep.edu.gr/node/18609
    - http://aesop.iep.edu.gr/node/11425
    - http://aesop.iep.edu.gr/node/14025
    - http://photodentro.edu.gr/oep/r/8532/493?locale=en
    - http://photodentro.edu.gr/oep/r/8532/505?locale=en
- Description: Through a variety of educational techniques, the teacher attempts to stimulate students' interest in CT and STEM objects, resulting in the development of a weather forecasting app (see Figure 4a, b).
- Outcomes of the activity: Engaging students in using CT and STEM concepts and developing their reasoning ability.

Activity 2: [Downloading data from the internet - API]

- Type of activity: group discussion, presentation, experimentation.
- Classroom organizing: working in groups.

- Learning objective: familiarizing students with free data and the multiple methods of data collection as well as data management.
- Digital archives and educational content: https://openweathermap.org/api
- Description: Students understand the availability of information on the web, either available freely or via subscriptions. They understand how APIs work and how data is retrieved.
- Outcomes of the activity: Enriching students' knowledge of online data (see Figure 5).

*Figure 4. The weather forecasting app at the stage of (a) a satellite location and (b) the presentation of the meteorological data*

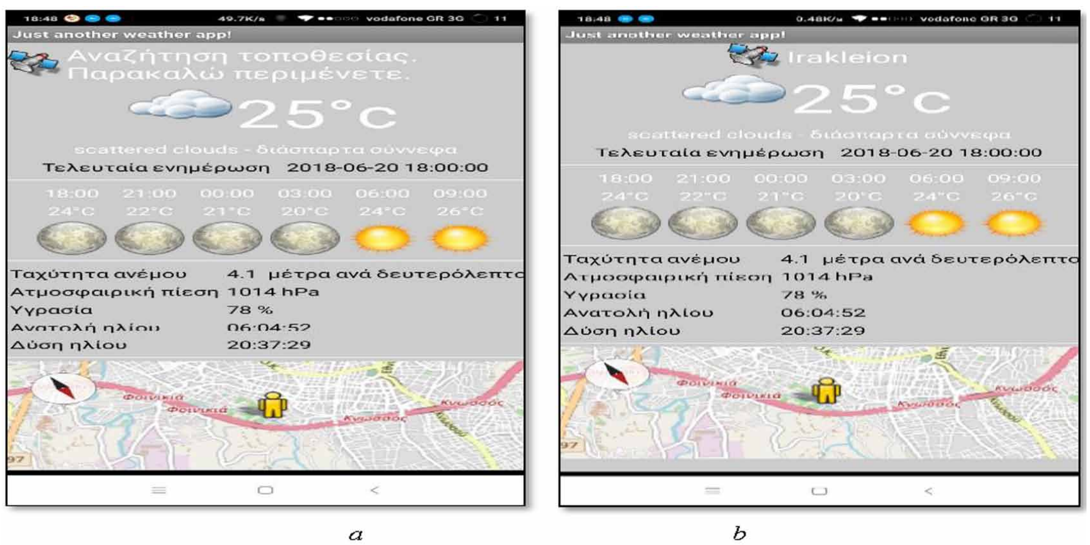



*Figure 5. Example of meteorological data via API from OpenWeatherMap*


{"cod":"200","message":0.0036,"cnt":40,"list":[{"dt":1485799200,"main":{"temp":261.45,"temp_min":259.086,"temp_max":261.45,"pressure":1023.48,"sea_level":1045.39,"grnd_level":1023.48,"humidity":79,"temp_kf":2.37},"weather":
[{"id":800,"main":"Clear","description":"clear sky","icon":"02n"}],"clouds":{"all":8},"wind":{"speed":4.77,"deg":232.505},"snow":{},"sys":{"pod":"n"},"dt_txt":"2017-01-30 18:00:00"},{"dt":1485810000,"main":
{"temp":261.41,"temp_min":259.638,"temp_max":261.41,"pressure":1022.41,"sea_level":1044.35,"grnd_level":1022.41,"humidity":76,"temp_kf":1.78},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"clouds":{"all":32},"wind":
{"speed":4.76,"deg":240.503},"snow":{"3h":0.011},"sys":{"pod":"n"},"dt_txt":"2017-01-30 21:00:00"},{"dt":1485820800,"main":
{"temp":261.76,"temp_min":260.571,"temp_max":261.76,"pressure":1021.34,"sea_level":1043.21,"grnd_level":1021.34,"humidity":84,"temp_kf":1.18},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":68},"wind":
{"speed":4.71,"deg":243},"snow":{"3h":0.058},"sys":{"pod":"n"},"dt_txt":"2017-01-31 00:00:00"},{"dt":1485831600,"main":
{"temp":261.46,"temp_min":260.865,"temp_max":261.46,"pressure":1019.95,"sea_level":1041.79,"grnd_level":1019.95,"humidity":82,"temp_kf":0.59},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":68},"wind":
{"speed":4.46,"deg":244.5},"snow":{"3h":0.05225},"sys":{"pod":"n"},"dt_txt":"2017-01-31 03:00:00"},{"dt":1485842400,"main":
{"temp":260.981,"temp_min":260.981,"temp_max":260.981,"pressure":1018.96,"sea_level":1040.84,"grnd_level":1018.96,"humidity":81,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13d"}],"clouds":{"all":80},"wind":
{"speed":4.21,"deg":245.005},"snow":{"3h":0.19625},"sys":{"pod":"d"},"dt_txt":"2017-01-31 06:00:00"},{"dt":1485853200,"main":
{"temp":262.308,"temp_min":262.308,"temp_max":262.308,"pressure":1018.1,"sea_level":1039.77,"grnd_level":1018.1,"humidity":91,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13d"}],"clouds":{"all":88},"wind":
{"speed":4.1,"deg":249.006},"snow":{"3h":0.535},"sys":{"pod":"d"},"dt_txt":"2017-01-31 09:00:00"},{"dt":1485864000,"main":
{"temp":263.76,"temp_min":263.76,"temp_max":263.76,"pressure":1016.86,"sea_level":1038.4,"grnd_level":1016.86,"humidity":87,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13d"}],"clouds":{"all":68},"wind":
{"speed":3.87,"deg":254.5},"snow":{"3h":0.21},"sys":{"pod":"d"},"dt_txt":"2017-01-31 12:00:00"},{"dt":1485874800,"main":
{"temp":264.182,"temp_min":264.182,"temp_max":264.182,"pressure":1016.19,"sea_level":1037.77,"grnd_level":1016.19,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":76},"wind":
{"speed":3.67,"deg":257.001},"snow":{"3h":0.1375},"sys":{"pod":"n"},"dt_txt":"2017-01-31 15:00:00"},{"dt":1485885600,"main":
{"temp":264.67,"temp_min":264.67,"temp_max":264.67,"pressure":1015.32,"sea_level":1036.94,"grnd_level":1015.32,"humidity":86,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":88},"wind":
{"speed":3.61,"deg":262.503},"snow":{"3h":0.1425},"sys":{"pod":"n"},"dt_txt":"2017-01-31 18:00:00"},{"dt":1485896400,"main":
{"temp":265.436,"temp_min":265.436,"temp_max":265.436,"pressure":1014.27,"sea_level":1035.76,"grnd_level":1014.27,"humidity":90,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":80},"wind":
{"speed":3.67,"deg":266.5},"snow":{"3h":0.1625},"sys":{"pod":"n"},"dt_txt":"2017-01-31 21:00:00"},{"dt":1485907200,"main":
{"temp":266.104,"temp_min":266.104,"temp_max":266.104,"pressure":1013.1,"sea_level":1034.62,"grnd_level":1013.1,"humidity":90,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":80},"wind":
{"speed":3.81,"deg":269.004},"snow":{"3h":0.1025},"sys":{"pod":"n"},"dt_txt":"2017-02-01 00:00:00"},{"dt":1485918000,"main":
{"temp":266.904,"temp_min":266.904,"temp_max":266.904,"pressure":1011.96,"sea_level":1033.47,"grnd_level":1011.96,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":76},"wind":
{"speed":4.26,"deg":274.002},"snow":{"3h":0.12},"sys":{"pod":"n"},"dt_txt":"2017-02-01 03:00:00"},{"dt":1485928800,"main":
{"temp":268.102,"temp_min":268.102,"temp_max":268.102,"pressure":1011.23,"sea_level":1032.62,"grnd_level":1011.23,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13d"}],"clouds":{"all":76},"wind":
{"speed":4.4,"deg":283.501},"snow":{"3h":0.13},"sys":{"pod":"d"},"dt_txt":"2017-02-01 06:00:00"},{"dt":1485939600,"main":
{"temp":270.269,"temp_min":270.269,"temp_max":270.269,"pressure":1010.85,"sea_level":1032.1,"grnd_level":1010.85,"humidity":92,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13d"}],"clouds":{"all":64},"wind":
{"speed":4.53,"deg":297.5},"snow":{"3h":0.1875},"sys":{"pod":"d"},"dt_txt":"2017-02-01 09:00:00"},{"dt":1485950400,"main":
{"temp":270.585,"temp_min":270.585,"temp_max":270.585,"pressure":1010.49,"sea_level":1031.65,"grnd_level":1010.49,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13d"}],"clouds":{"all":76},"wind":
{"speed":4.31,"deg":302.004},"snow":{"3h":0.065},"sys":{"pod":"d"},"dt_txt":"2017-02-01 12:00:00"},{"dt":1485961200,"main":
{"temp":269.661,"temp_min":269.661,"temp_max":269.661,"pressure":1010.22,"sea_level":1031.49,"grnd_level":1010.22,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":68},"wind":
{"speed":4.91,"deg":296.5},"snow":{"3h":0.0825},"sys":{"pod":"n"},"dt_txt":"2017-02-01 15:00:00"},{"dt":1485972000,"main":
{"temp":269.155,"temp_min":269.155,"temp_max":269.155,"pressure":1009.95,"sea_level":1031.3,"grnd_level":1009.95,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":80},"wind":
{"speed":5.7,"deg":310.501},"snow":{"3h":0.11},"sys":{"pod":"n"},"dt_txt":"2017-02-01 18:00:00"},{"dt":1485982800,"main":
{"temp":268.056,"temp_min":268.056,"temp_max":268.056,"pressure":1011.21,"sea_level":1032.49,"grnd_level":1011.21,"humidity":89,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":68},"wind":
{"speed":5.56,"deg":333},"snow":{"3h":0.225},"sys":{"pod":"n"},"dt_txt":"2017-02-01 21:00:00"},{"dt":1485993600,"main":
{"temp":265.803,"temp_min":265.803,"temp_max":265.803,"pressure":1013.79,"sea_level":1035.06,"grnd_level":1013.79,"humidity":83,"temp_kf":0},"weather":[{"id":600,"main":"Snow","description":"light snow","icon":"13n"}],"clouds":{"all":8},"wind":

Activity 3: [Global Positioning System - GPS]

- Type of activity: group discussion, presentation, demonstration, experimentation.
- Classroom organizing: working in groups.
- Learning objective: an interdisciplinary approach to Science, Technology, Engineering and Mathematics (STEM).
- Digital archives and education content:
  - Finding geostatistical satellites http://www.sdtv.gr/google/ex3.htm
  - Finding geographical coordinates https://maps.google.com/
  - Meteorological data extraction based on geographical coordinates https://openweathermap.org/current
- Description: Students understand what the Global Positioning System is and how it is used as an X, Y, Z coordinate measurement and positioning system for a Global Cartesian Geocentric Reference System. They learn how to use this data to extract location information.
- Outcomes of the activity: Students understand the importance and use of the GPS and make the necessary mental connections to use it in the design of a weather-forecast app.

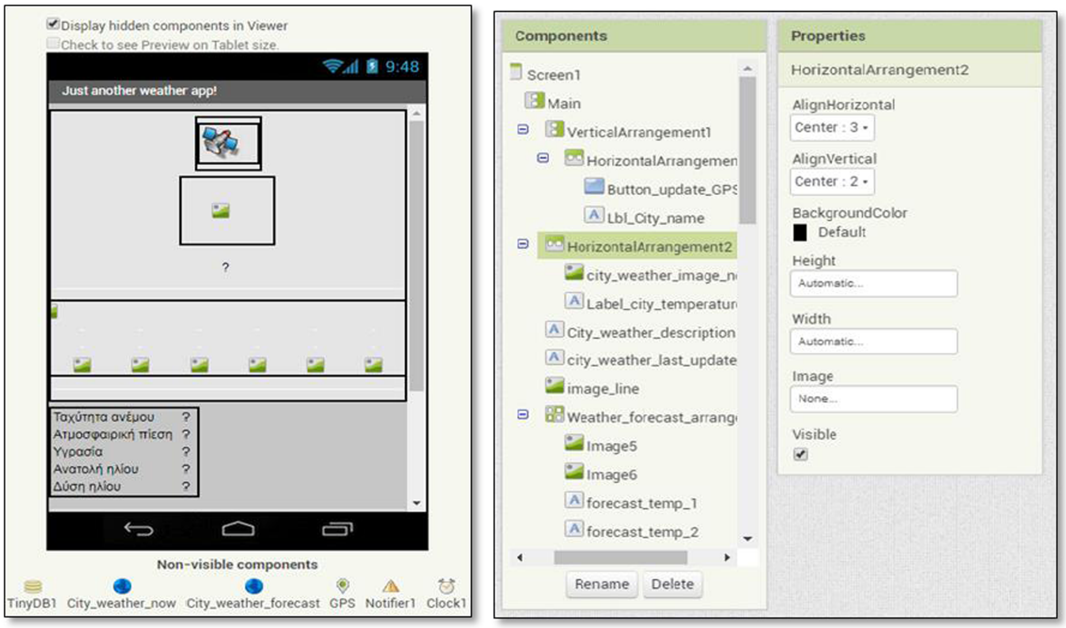
Activity 4: [Graphical User Interface Design]

- Type of activity: group discussion, presentation, brainstorming, experimentation.
- Classroom organizing: working in groups.
- Learning objective: creating a weather-forecast app interface (see Figure 6).
- Description: Students design the graphical interface of the weather-forecast app.
- Outcomes of the activity: students understand the various parameters involving in the designing of the GUI of a mobile app such as the data availability, the use of multimedia content, the data types, the limitations of mobile devices, etc.) (see Figure 7).

Activity 5: [Coding]

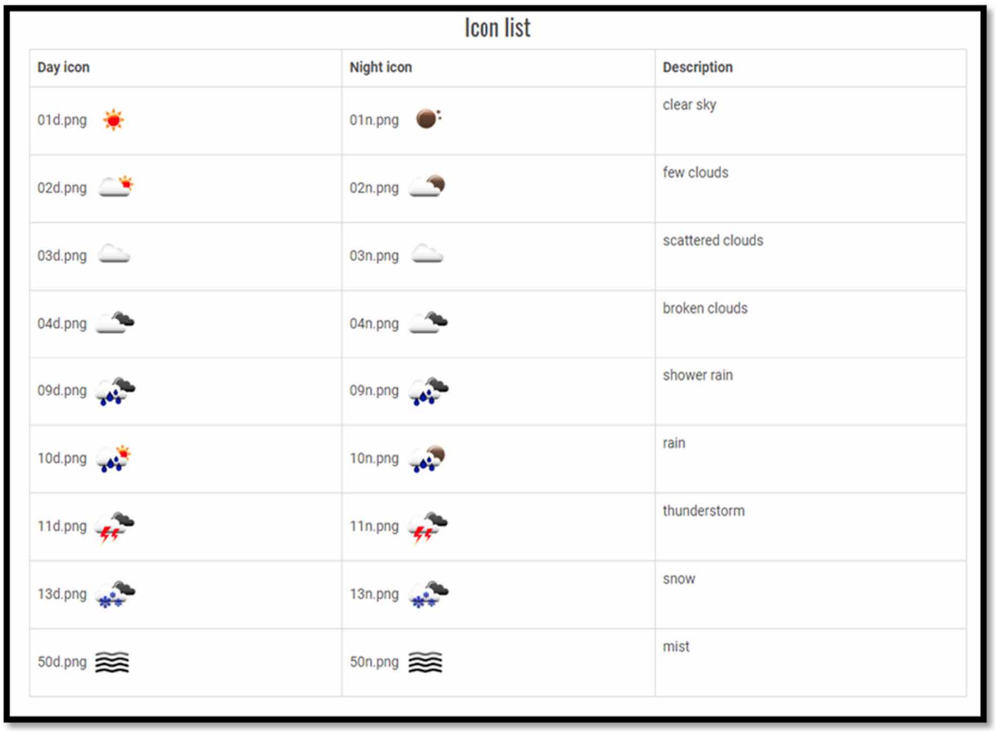
- Type of activity: group discussion, presentation, brainstorming, experimentation.
- Classroom organizing: working in groups.
- Link to the teaching objective: create a code for the proper functioning of the weather forecast app.
- Learning objective: Students create the necessary code for the proper functioning of the weather-forecast app.
- Outcomes of the activity: students practice their coding skills (like program debugging, testing, code reuse, maintenance, etc.). Students learn by transforming and refining their prior knowledge into more sophisticated programming skills such as data management and data storage while they understand the capabilities of mobile technology (see Figures 8, 9 and 10).

*Figure 6. Design of the graphical user interface (a) and the configuration of the properties of the various components of the app (b)*




*a* *b*

*Figure 7. Free weather description icons via the OpenWeatherMap service*



Icon list

| Day icon | Night icon | Description |
|---|---|---|
| 01d.png | 01n.png | clear sky |
| 02d.png | 02n.png | few clouds |
| 03d.png | 03n.png | scattered clouds |
| 04d.png | 04n.png | broken clouds |
| 09d.png | 09n.png | shower rain |
| 10d.png | 10n.png | rain |
| 11d.png | 11n.png | thunderstorm |
| 13d.png | 13n.png | snow |
| 50d.png | 50n.png | mist |

*Figure 8. Code example for GPS management*

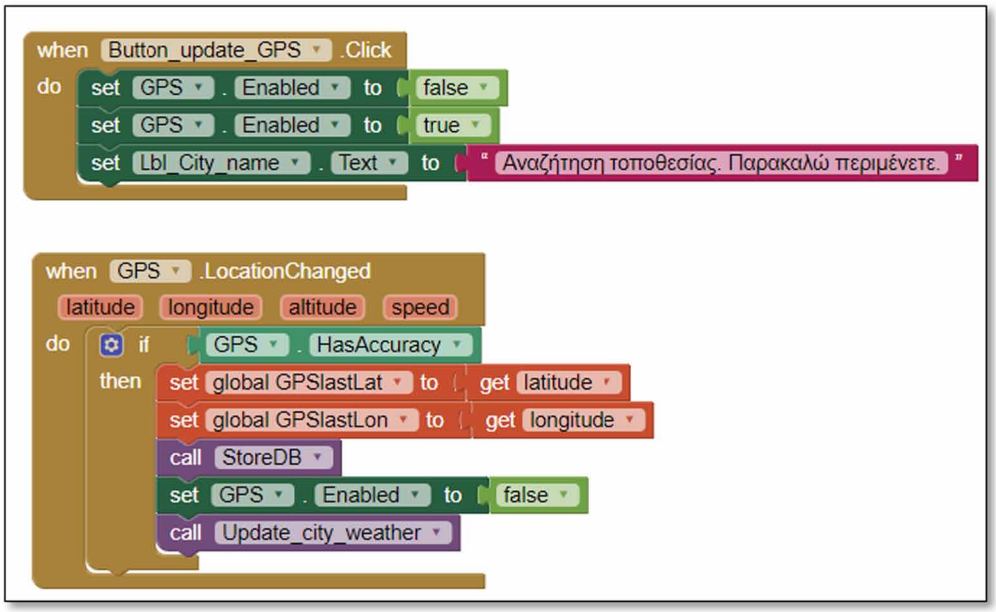



*Figure 9. Code example for the conversion of the Unix time code (or POSIX time or UNIX Epoch time) in a format compatible with standard date/time*

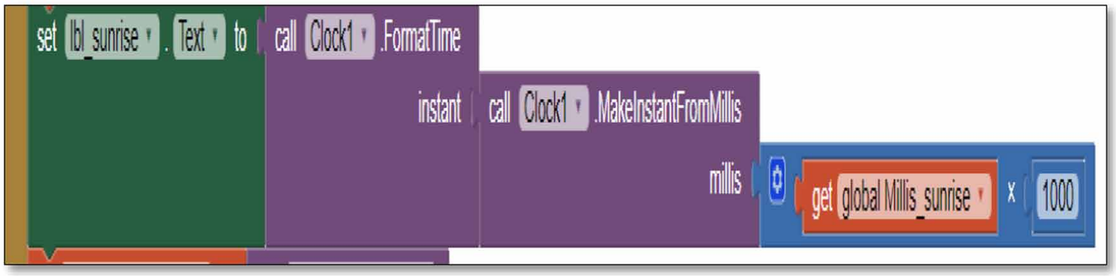



*Figure 10. Code example for geographical coordinate storage*

Activity 6: [Debugging - application optimization, generating executable code (.apk format)]

- Type of activity: group discussion, presentation, brainstorming, experimentation
- Classroom organizing: working in groups
- Learning objective: correction of the design bugs, optimization of the app. Generating executable code (.apk format).
- Description: students correct or optimize the code for the proper functioning of the weather-forecast app and create the executable code (.apk format) to install the app on smart mobile applications.
- Outputs of the activity: students practice their coding skills and learn how to create an executable version of the app. Students deepen their knowledge of problem-solving steps. software optimization issues (e.g. repetitive code snippets, process calls, etc.).

## Methodology

The sample of the study consisted of 23 students (10 boys, 13 girls) who were studying in the second grade in the 2017-2018 school year in a General High School (Lyceum) in Crete, Greece. To verify the effectiveness of the proposed teaching intervention (improving students' knowledge of programming and understanding of STEM concepts as well as their attitude towards them), a standard experimental procedure was followed which included a pre-test and a post-test without a control group (Papadakis et al., 2014, 2016, 2017). In both phases, the students in the sample were asked to answer three different questionnaires. The first questionnaire measured students' attitudes, perceptions, and motivations towards programming. The Computer Attitude Scale (CAS) (Loyd & Gressard, 1984) was used as the basis for the questionnaire creation. Kleinschmager & Hanenberg's (2011) work was also used as a basis for testing students' knowledge and understanding of the basic programming structures. To test STEM knowledge, the researcher developed a specially designed questionnaire following the scientific methodological approach for its creation and validation (Cohen, Manion & Morrison, 2002).

To ensure the clarity and validity of the questions asked, the questionnaire was reviewed independently by two teachers with expertise in programming as they had taught the same course in previous years in different schools. This is a common approach for developing and improving achievement tests (Orfanakis & Papadakis, 2014, 2016). The questionnaires were field-tested in a pilot survey carried out with 20 students from the same school who did not participate in the study to judge their representativeness. This approach was chosen to minimize potential problems related to questionnaire comprehension, the translation from the English to the Greek language and the conduct of the study. The respondents were asked to comment on the content of the tool and the phraseology of the items. Minor changes were made based on the comments and suggestions of the participants. No other practical and implementation problems were encountered in the pilot study and no items were removed from the questionnaire. Reliability analysis, to determine the internal consistency of the questionnaires, were evaluated by Cronbach's alpha. The Cronbach's alphas for the pre-test and post-test for the questionnaires ranged from 0.80 to 0.84 suggesting acceptable reliability. Statistical analysis of data was done using IBM SPSS Statistics for Windows software (version 23.0). A p value of 0.05 was adopted as the minimum level of statistical significance for the statistical control of the data.

## Impact of the Course on Students' Attitudes and Knowledge on Programming and STEM Concepts

The purpose of the study was to investigate whether the student's responses were differentiated after the implementation of the teaching intervention. Analysis of the responses showed that students' behavioral approach in the programming was improved after the completion of the intervention. The difference in students' performance between the two tests was statistically significant. The second and third purpose of the research was to investigate whether their knowledge of programming and STEM concepts was also improved. To this end, the responses of the students to the other two criteria were compared before and after the completion of the intervention. The correlated t-test was used to compare the means of the distributions. The difference in students' performance was also found to be statistically significant (see Table 1). But there was no statistically significant relationship between students' performance, gender and/or the age.

*Table 1. Averages (M) and standard deviations (sd) of students' performance and parametric t-test results for dependent samples*

| | | M | sd | t-test |
|---|---|---|---|---|
| Behavioral approach to programming | Pretest | 10.48 | 1.31 | -6.35, $p < .05$ (df = 23) |
| | Posttest | 11.65 | 1.50 | |
| Programming knowledge | Pretest | 11.70 | 1.74 | -6.00, $p < .05$ (df = 23) |
| | Posttest | 12.57 | 1.53 | |
| STEM Knowledge | Pretest | 9.96 | 2.03 | -6.55, $p < .05$ (df = 57) |
| | Posttest | 10.87 | 1.77 | |

## Study Limitation

The present study is associated with several limitations. The main source of bias for this study could be the fact that the author was the person who designed this study and collected the data, as well as the teacher who implemented the course. Furthermore, the relevant data of this study were collected from only one secondary school in Greece. To generalize the findings, further studies should be carried out across different schools and/or geographical regions in Greece. Moreover, this intervention study was conducted only once and over a relatively short period, due to time constraints, and so participants might have limited exposure to this project-based experience. Data on intervention implementation and outcomes need to be collected, entered, checked and analyzed over a longer time frame. Finally, AI2 was adopted as the programming learning environment in this study. The results which were released by this study could be different if another environment was used (e.g., ALICE).

## Discussion of the Results

The fact that, at the end of the process, students improved their knowledge of programming and STEM concepts, as well as their attitude in programming, should be a positive outcome. It is widely known that student's involvement in an educational activity does not always lead to changes in behavior and active participation and of course, does not automatically translate into true knowledge transfer. Furthermore, the related literature reveal that many students treat programming as a boring and unattractive activity (Blikstein, 2018). We, can, therefore, argue that this teaching intervention based on the combination of programming, CT and STEM concepts using the App Inventor programming environment through authentic learning activities has helped students to significantly improve their behavioral approach as well as their knowledge and performance in both programming and STEM concepts.

## CONCLUSION

21st Century skills are related to a wide range of skills such as learning, innovation, information, media and technology (Qian & Clark, 2016). Students to be successful, they need to have a wide range of these skills. STEM-oriented education is thought to provide students with these necessary skills by generalizing the content of knowledge in real contexts. However, to achieve this, STEM courses require appropriate teaching and learning techniques that will help students to solve problems or make discoveries (Papadakis, 2018b). A combination of STEM and programming in school projects will enable students to create solid and functional representations for CT and STEM concepts and structures through an interesting and rich stimulating environment in which problems are explored in an exciting, creative and enjoyable way (Resnick & Silverman, 2005).

It has already noted that the empirical data from the teaching intervention are not generalizable due to the small sample size based on convenience, non-random sampling, as well as their origin from a single, public high school in Crete, Greece. But, since there is insufficient research in Greece on the educational benefits of teaching programming and STEM concepts using the App Inventor programming environment through authentic learning activities, this research enriches this topic.

In conclusion, the combined use of educational packages, free resources and programming environments in collaborative problem-solving activities, can support the creation of student-centered learning environments, and to make the STEM and programming courses an easy and attractive activity in students' eyes.

## COMPETING INTERESTS

The author declares that he has no competing interests.

## FUNDING

The author received no financial support for the research, authorship, and/or publication of this article.

## ACKNOWLEDGMENT

The idea for this educational practice was given by a project that is freely distributable in the App Inventor project collection at the following address: http://ai2.appinventor.mit.edu/?locale=en#4768766558863360. The author would like to thank all the students for their participation and assistance during this study.

## REFERENCES

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin Inroads*, *39*(2), 32–36. doi:10.1145/1272848.1272879

Bers, M. U. (2017). The Seymour test. *International Journal of Child-Computer Interaction*, *14*(100), 10–14. doi:10.1016/j.ijcci.2017.06.004

Blikstein, P. (2018). *Pre-College Computer Science Education: a Survey of the Field*. Google LLC. Retrieved from https://goo.gl/gmS1Vm

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*.

Chen, P., & Huang, R. (2017). Design thinking in app inventor game design and development: a case study. In *Advanced Learning Technologies (ICALT), 2017 IEEE 17th International Conference on*, (pp.139–141). IEEE. 10.1109/ICALT.2017.161

Cheng, G. (2019). Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. *Computers in Human Behavior*, *92*, 361–372. doi:10.1016/j.chb.2018.11.043

Chiu, C. F. (2014). Use of problem-solving approach to teach scratch programming for adult novice programmers. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, (pp.710–711). ACM. 10.1145/2538862.2544284

Cinganotto, L. (2017). *When content and learning join up, great things can happen*. Retrieved from https://www.schooleducationgateway.eu/en/pub/viewpoints/experts/when-content-and-learning-join.htm

Cohen, L., Manion, L., & Morrison, K. (2002). *Research methods in education*. Routledge. doi:10.4324/9780203224342

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational Thinking: a Guide for Teachers. *Computing at Schools*. Retrieved from https://community.computingatschool.org.uk/resources/2324/single

Cussó-Calabuig, R., Farran, X. C., & Bosch-Capblanch, X. (2018). Effects of intensive use of computers in secondary school on gender differences in attitudes towards ICT: A systematic review. *Education and Information Technologies*, *23*(5), 2111–2139. doi:10.100710639-018-9706-6

Dewey, R. A. (2018). *Psychology: An Introduction.* Retrieved from http://www.intropsych.com/ch01_psychology_and_science/independent_dependent_and_subject_variables.html

European Schoolnet. (2014). *Computing our future – Priorities, school curricula and initiatives across Europe*. Retrieved from https://goo.gl/WAjaCs

Eurostat (2019a). *Persons employed as ICT specialists and total employment, EU-28, 2006-2016 ICT*. Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php/File:Persons_employed_as_ICT_specialists_and_total_employment,_EU-28,_2006-2016_ICT17.png

Eurostat. (2019b). *ICT specialists - statistics on hard-to-fill vacancies in enterprises*. Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_specialists_-_statistics_on_hard-to-fill_vacancies_in_enterprises

Eurostat. (2019c). *ICT specialists are predominantly male*. Retrieved from https://ec.europa.eu/eurostat/en/web/products-eurostat-news/-/DDN-20190513-1

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning*, *32*(6), 576–593. doi:10.1111/jcal.12155

Fincher, S., Cooper, S., Kölling, M., & Maloney, J. (2010). Comparing alice, greenfoot & scratch. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp.192–193). 10.1145/1734263.1734327

Gray, J., Abelson, H., Wolber, D., & Friend, M. (2012). Teaching CS principles with app inventor. In *Proceedings of the 50th Annual Southeast Regional Conference* (pp.405–406). ACM. 10.1145/2184512.2184628

Guzdial, M. (2004). Programming environments for novices. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 127–154). Abingdon, UK: Taylor & Francis.

Guzdial, M. (2018). *Confusion over the Forms of Programming Problems: Mathematics/Physics versus CS.* Retrieved from https://computinged.wordpress.com/2018/02/26/confusion-over-the-formsof-programming-problems-mathematics-physics-versus-cs/

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. doi:10.1016/j.compedu.2018.07.004

Jordaan, D. B. (2018). Board games in the computer science class to improve students' knowledge of the Python programming language. In *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)* (pp.1–5). IEEE. 10.1109/ICONIC.2018.8601207

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, *37*(2), 83–137. doi:10.1145/1089733.1089734

Kleinschmager, S., & Hanenberg, S. (2011). How to Rate Programming Skills in Programming Experiments? A Preliminary, Exploratory, Study Based on University Marks, Pretests, and Self-Estimation. In *Proc. ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools* (pp. 15–24). Portland, OR: ACM.

Loyd, B. H., & Gressard, C. (1984). Reliability and factoral validity of computer attitude scale. *Educational and Psychological Measurement*, *44*(2), 501–505. doi:10.1177/0013164484442033

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *Transaction on Computing Education, 10*(4), 16:1–16:15.

MIT App Inventor. (2019). *MIT App Inventor*. Retrieved from http://appinventor.mit.edu/

Morrison, J., & Bartlett, R. V. (2009). STEM as a curriculum: An experiential approach. *Education Week*, *28*(23), 28–31.

Orfanakis, V., & Papadakis, St. (2014). A new programming environment for teaching programming. A first acquaintance with enchanting. *Proceedings of the 2nd international virtual Scientific Conference - Scieconf 2014, EDIS,* 268–273.

Orfanakis, V., & Papadakis, St. (2016). Teaching basic programming concepts to novice programmers in Secondary Education using Twitter, Python, Arduino and a coffee machine. In *Proceedings of the Hellenic Conference on Innovating STEM Education (HISTEM)*. University of Athens.

Organization for Economic Cooperation and Development (OECD). (2013). *OECD Skills Outlook 2013*. Paris: OECD Publishing. doi:10.1787/9789264204256-

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia: Social and Behavioral Sciences*, *191*, 1479–1482. doi:10.1016/j.sbspro.2015.04.224

Papadakis, St. (2018a). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, St. (2018b). Is pair programming more effective than solo programming for secondary education novice programmers? A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, St., & Kalogiannakis, M. (2017). Using Gamification for Supporting an Introductory Programming Course. The Case of ClassCraft in a Secondary Education Classroom. In *Interactivity, Game Creation, Design, Learning, and Innovation* (pp. 366–375). Cham: Springer.

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In Fardoun, H.M. & Gallud, J.A. (Ed.), *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*, (pp.1–7). ACM.

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2016). Using scratch and app inventor for teaching introductory programming in secondary education. A case study. *International Journal of Technology Enhanced Learning*, *8*(3–4), 217–233. doi:10.1504/IJTEL.2016.082317

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *Int. J. Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, *99*, 415–427. doi:10.1016/j.chb.2019.01.008

Qian, M., & Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, *63*, 50–58. doi:10.1016/j.chb.2016.05.023

Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education*, *18*(1), 1–24. doi:10.1145/3077618

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children* (pp. 117-122). 10.1145/1109540.1109556

Sangiuliano, M. (2015). *Role models leading girls into careers in science*. Retrieved from https://www.schooleducationgateway.eu/en/pub/viewpoints/experts/role_models_leading_girls_into.htm

SFIA Foundation. (2017). *The Skills Framework for the Information Age – SFIA*. Retrieved from https://www.sfia-online.org/en/reference-guide

Shi, N., Cui, W., Zhang, P., & Sun, X. (2018). Evaluating the effectiveness roles of variables in the novice programmers learning. *Journal of Educational Computing Research*, *56*(2), 181–201. doi:10.1177/0735633117707312

Vuorikari, R. (2015). *Becoming digitally competent: A task for the 21st-century citizen*. Retrieved from https://www.schooleducationgateway.eu/en/pub/viewpoints/experts/riina_vuorikari_-_becoming_dig.htm

Wang, J. (2017). Is the US education system ready for CS for all? *Communications of the ACM*, *60*(8), 26–28. doi:10.1145/3108928

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (ITiCSE'14)*, (pp.39–44). ACM.

Weintrop, D. (2015). Comparing text-based, blocks-based, and hybrid blocks/text programming tools. In *Proceedings of the Eleventh annual International Conference on International Computing Education Research*, (pp.283–284). 10.1145/2787622.2787752

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children*, (pp.199–208). 10.1145/2771839.2771860

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wolber, D. (2011). App Inventor and real-world motivation. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*, (pp.601–606). ACM. 10.1145/1953163.1953329

Zapušek, M., & Rugelj, J. (2013). Learning programming with serious games. *EAI Endorsed Trans. Game Based Learn*, *13*(1), e6. doi:10.4108/trans.gbl.01-06.2013.e6

## ADDITIONAL READING

Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York, NY: Teachers College Press.

Kalogiannakis, M., & Papadakis, St. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-service Teachers to Promote STEM Activities in Preschool Classrooms. In St. Papadakis and M. Kalogiannakis (Eds), Mobile Learning Applications in Early Childhood Education, USA-PA: IGI Global (under publication). doi:10.4018/978-1-7998-1486-3.ch005

Papadakis, St. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *Int. J. Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, St. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-primary School Age, In St. Papadakis and M. Kalogiannakis (Eds), Mobile Learning Applications in Early Childhood Education, USA-PA: IGI Global (under publication).

Papadakis, St., & Kalogiannakis, M. (2018). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed), Early Childhood Education, 1-19, i-first chapter, UK-London: IntechOpen Limited.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Technology Enhanced Learning*, *10*(3), 235–250.

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In St. Papadakis and M. Kalogiannakis (Eds), Mobile Learning Applications in Early Childhood Education, USA-PA: IGI Global (under publication).

Papadakis, St. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

## KEY TERMS AND DEFINITIONS

**App:** An app is a software application that works, generally, on a mobile technology such as a smartphone, tablet, or other similar device.

**Block-Based Coding Language:** A programming language that uses graphic elements as a means of providing visual cues to the user as to how and where commands may be used.

**ICT (Information and Communication Technology):** A term that covers all technical means used to handle information and aid communication, including software and hardware.

**Mobile Device:** The mobile device is a small-sized, portable computer that typically has touch-screen features.

**Novice Programmer:** A computer programmer who is not experienced at programming.

**Smartphone:** A smartphone is a term used to describe a category of mobile devices with computer-like functionality.

**STEM:** The term STEM (science, technology, engineering, and mathematics) is an acronym used by those relevant to the educational method concerning the fields of science, technology, engineering, and mathematics.

**Text-Based Languages:** A programming language that does not involve graphical elements (blocks) as a main part of its programming language, but instead is mostly oriented around text.

# Chapter 3
# Investigation of Variables Related to Computational Thinking Self-Efficacy Level in Middle School Students:
## Are Demographic Variables, Academic Success, or Programming-Related Variables More Important?

**Hatice Yildiz Durak**
https://orcid.org/0000-0002-5689-1805
*Bartin University, Turkey*

**Mustafa Saritepeci**
https://orcid.org/0000-0002-6984-0652
*Necmettin Erbakan University, Turkey*

**Ahmet Topçu**
https://orcid.org/0000-0001-6981-9098
*MNE, Turkey*

**Aykut Durak**
https://orcid.org/0000-0002-5689-1805
*Bartin University, Turkey*



## ABSTRACT

*This study examined the predictiveness of demographic and academic variables and the variables which are in relation with programming on computational thinking (CT) self-efficacy of middle school students who received and who did not receive programming education. Relational screening model was utilized*

DOI: 10.4018/978-1-7998-4576-8.ch003

*in this study. One-hundred ninety-nine middle school students from 5th and 6th grades in Turkey composed the participants of the research. As the result of the research, it was found that CT self-efficacy level is low. Furthermore, programming experiences of the students are approximately two years. The most important predictor of CT self-efficacy of the students who received programming education is demographic variables. Predictive variables' relative order of importance on CT self-efficacy of the students who received programming education are gender, utilized programming tool, math class grade point average, and attitude towards programming.*

## INTRODUCTION

Computational thinking (CT) is one of the often-mentioned concepts today. This concept does not have an agreed universal description (Lockwood & Mooney, 2017; Moreno-León, Román-González, & Robles, 2018; Yildiz-Durak & Saritepeci, 2018). Accordingly, it is observed that the scale tools necessary for CT's scopes, development and scaling have a diverse range (Zhang & Nouri, 2019). While Papert (1980) explained CT within context of math education, he emphasized application of children's procedural thinking by way of computer programming. Wing (2006) argued that for everyone CT is a skill which should be developed, and CT is needed to be regarded as a basic skill not only in computer programming processes, but also in every action involving analytical ability of human. According to Durak (2020), content of CT is not limited to the problem-solving processes in human minds. CT includes all of computing processes (Hsu, Chang, & Hung, 2018).

Development of CT skills via programming has an important place, particularly at K-12, for enabling of 21st century skills (Wong & Cheung, 2018). Even at some resources CT is handled as the skill of 21st century. However, it is important to detect at which learning level and with which variables CT is in relation with. Thus, according to Wing (2006), contribution of the knowledge taught to children during the education given at early ages to problem-solving process, how to apply CT skills and how to make logical analysis should be taught as well. Children's comprehension of CT skills is utilized for turning a problem into a easy-to-understand problem by expressing it with their own description, for solving problems with a mood similar to computer scientist when we encounter IT problems, for designing systems and turning problem's solution process into a thinking system which can be understood by people (ISTE (International Society for Technology in Education) & CSTA(Computer Science Teachers Association), 2011; Wing, 2006; Durak, 2020). Therefore, by solving the problems in different fields via CT skills, students can be enabled to be individuals who are able to solve the problems they will encounter in information society through current technologies, academically successful and life-long learners (Gülbahar, Kert, & Kalelioğlu, 2019).

In the studies examining the skills in relation with CT, it was stated that CT includes such skills as abstraction to solve the problems algorithmically, reflective thinking and creative thinking (Aho, 2012; Barr & Stephenson, 2011; Brennan & Resnick, 2012; García-Peñalvo, 2017; Grover & Pea, 2013; Riley & Hunt, 2014). On the other hand, CT programming activities were often examined side by side. This status may cause CT to be regarded as the programming. According to Kukul (2018), handling of CT in problem-solving processes with almost same descriptions and processes caused the false notion that computational thinking skills will also increase in direct proportion when programming training is given. Qualls and Sherrell (2010) emphasized the necessity that programming, and CT are separately handled. CT is the solution development to the problem by using computer science and programming concepts, management of abstraction and transferring process. Within this context, it is thought to be necessary that contribution of programming training to development of CT skill at the individuals who received and who did not receive programming training is evaluated. Therefore, in this study, computational self-efficacy levels of middle school students who received and who did not receive programming training were examined. Furthermore, predictiveness of various variables on examination of the variables in relation with CT self-efficacy was researched as well.

## BACKGROUND

### What is CT?

CT is a thinking process which expresses active use of information and communication technologies' concepts in solution of complex problems (Shute, Sun, & Asbell-Clarke, 2017). According to Wing (2006), CT as a basic skill everyone should have is a way of problem-solving, system designing, automation processes and understanding human behaviors by using basic concepts of computer sciences. According to CSTA and ISTE (2011), CT as a problem-solving process is a process which includes tools of information and communication technologies and use of them in a solution oriented way; regulation, analysis and modelling of the data; finding optimum solution and generalization of the solution. Besides the given descriptions, it can be said that CT is often handled by way of problem-solving skills and programming in the studies about enabling CT in educational environments. Barr and Stephenson (2011) emphasized that CT at K-12 is a skill which is in relation with problem-solving skills and with self-confidence and determination displayed upon encountering specific problems. However, Brennan and Resnick (2012), based on their Scratch block grounding programming studies, emphasized that three essential components take role in description of CT as data processing concepts, data processing applications and data processing viewpoints. In explanation of data processing concepts; syntactic, semantic and schematic information (e.g. variables, loops (repeat, do...while)) which are commonly used in programming, conditions (e.g. If...else) and operators (arithmetic and comparison) were utilized.

As data processing applications in thinking and application processes, it was explained as the use of strategical information (e.g. testing and error debugging, re-use and abstraction) which is used in order to solve the programming problem. Data processing viewpoints, however, were expressed as comprehension of individual, social and technological relations about programming in their environments. These descriptions and components demonstrate that programming training has an important role in enabling CT skills. However, CT should not be regarded only as a skill which is gained by programming of the technological tool which was used in solution. Thus, in regard to enabling CT skill, importance of IT activities in which programming training activities are not available and they include learning by creating diversity was also emphasized in various studies (Kordaki & Kakavas, 2017; Sarıtepeci, 2020).

## CT Education at K-12

Today CT is regarded as a key skill for all individuals in 21st century and curriculum development endeavors have been made (Yadav, Hong, & Stephenson, 2016). This status is observed as an economic and technological requirement for future employment factors and it is promoted within this context (Chen et al., 2017).

Though CT is a concept emerging in recent years, at first in 1980's Papert (1980) argued that students need to adopt these skills via Logo programming. Papert (1980) had the idea that students will be able to adopt high level thinking skills through Logo. After such endeavour of Papert, the idea of using programming training for purpose of enabling students to adopt high level thinking skills at K-12 level or for purpose of developing these skills has not been accepted for a long time (Lye & Koh, 2014). But, thanks to emphasis of Wing (2006, 2008) on the fact that CT is a skill needed to be adopted by everyone and common acceptance of this opinion, in recent years the emphasis on education of CT sciences and particularly the emphasis on programming training for development of CT skills increase day by day. Even Papert's (1980) idea of using environments having low level information requirement like Logo within content of programming training for education and development of high level skills was not accepted in those years, today this idea is observed to receiving large acceptance via visual programming languages at K-12 like Scratch, Alice, Kodu, Code.org. The basic motivation behind common use of such visual programming tools in CT education (Hsu, Chang & Hung, 2018; Marcelino et al. 2018; Sarıtepeci, 2019) is the fact that they enable focusing on cognitive processes which will support development of CT skills by decreasing unnecessary syntactic and cognitive load (Lye & Koh, 2014).

## Role of Programming Training in Developing Computational Thinking Self-Efficacy at K-12

According to Lye and Koh (2014), programming contributes to developing of CT which involves such high-level thinking skills like problem-solving, abstraction by way of supporting students to use concepts of CT science. In the similar way; Kazimoglu, Kiernan, Bacon and MacKinnon (2011) indicate that one of the basic aims of computer education is adoption of CT skill and continuous emphasis of this. At this point, the question needed to be asked is how does programming training develop CT.

Seow, Looi, How, Wadhwa and Wu (2019) remark that many countries which accepted importance of CT skills integrated programming courses into K-12 in order to support development of these skills. Thus, it is thought that applications which will develop students' IT skills like programming and problem solving, data science, algorithmic thinking, engineering design process, systematic thinking and abstraction will be simply performed. The most important point in this is the fact that CT skills are not a programming skill and a well-organized programming training is an important tool in development of CT skill (García-Peñalvo & Mendes 2017). Through a well-designed programming training, students have the opportunity for using and developing various skills like algorithmic thinking, abstraction, debugging, logical thinking, problem-solving and creativity which are in close relation with CT (Buitrago Flórez et al., 2017; National Research Council, 2010).

Programming with CT is not the same concepts (Papadakis, Kalogiannakis, Zaranis, 2016). However, according to Papadakis (2019) programming is a popular tool for teaching CT. Because programming has been emphasized in many studies that it is effective to develop high level thinking skills, make CT concepts concrete and integrate CT into the learning process (Papadakis, Kalogiannakis, Orfanakis,

& Zaranis, 2017). In addition, early teaching of programming provides an important opportunity for students to develop and use CT or higher-level thinking skills (Durak, 2020; Papadakis, Kalogiannakis, Orfanakis, & Zaranis, 2016; Yildiz Durak, 2018). In the supporting way, it is a frequently emphasized status in the literature that programming training is an effective tool for the education and development of CT skills (Boechler et al., 2014; Koorsse, Cilliers & Calitz, 2015; Piteira & Costa, 2012; Sarıtepeci & Yildiz Durak, 2017; Sarıtepeci, 2020).

## Role of Demographic Variables, Academic Success and Variables in Relation with Programming in Development of Computational Thinking

In the last several years important number of studies were conducted on determination of the elements being effective for development of CT and on efficient application of CT education in the literature. In general, these studies include determination of the variables in relation with CT and examination of factors and statuses which are thought to being/to be effective in education or development of this skill. However, in the new studies expansion of nomological network is regarded as important in determination of the elements which will be taken into consideration in CT education. Within this context, in this study the relation between CT level and demographic variables (gender, age, educational level); the relation between academic success (success in maths, success in science, success in information technologies) and variables related with programming training (attitude towards programming, programming experience, utilized programming tool) were examined.

One of the variables whose effect was examined in education of CT skill is gender. Basic opinion lying behind the idea that gender variable may be effective for CT education is the hypothesis that this skill is in direct relation with computer sciences and the fact that computer science education has an important field in enabling of this skill (Yildiz Durak & Sarıtepeci, 2018). Determinations on the fact that girls compared to boys have a negative attitude towards computer science (Buitrago Flórez et al., 2017; Stein & Nickerson, 2004) supports this status. While some part of the studies in the literature is of the supporting value that gender is a determinant factor in education of CT skill (Dagienė, Pelikis & Stupurienė, 2015; Román-González, Pérez-González & Jiménez-Fernández, 2017), in various studies it was demonstrated that boys and girls displayed similar characteristics in terms of CT levels (Atmatzidou, & Demetriadis, 2016; Sarıtepeci, 2017; Taylor & Baek, 2019; Yildiz Durak & Sarıtepeci, 2018). Effects of gender is observed to display variance in education of CT skill and development of this skill. Therefore, in order to understand the relation between gender factor and CT skill at K12 level, related nomological network is needed to be expended. Within this context, in this study we attach importance to examination of gender factor's effect on CT levels of the two groups who participated and who did not participate to programming training.

In this study, another variable whose effect in CT education and development at K12 level was handled is age. There is limited level of studies which handle the relation between gender and CT at K12 level in the literature. In one of these studies, Atmatzidou and Demetriadis (2016) remarked that age of participants has limited effect on development of CT skill, however, actual important thing in development of this skill is participation to sufficient amount of education activity. In another study, Çetinkaya (2019) in his study conducted with participants from 5th grade level in age group between 9.5 and 11.5 years reached to the result that CT level differentiates based on age. The basic separation between these two studies may result from the fact that in the study of Çetinkaya (2019) participants who are at concrete operations step which is a critically important period and who just begin to take part in abstract operations step are placed at same educational level and are exposed to the same application. Participants over the age of 11 years who start to adopt abstract thinking are more possible to perform better compared to younger age groups in the activities targeting abstract CT education. In the supporting way, it can be said that cognitive development level is important in development of IT skill (Román-González et al., 2017). However, there are different studies which handle effect of educational level which is indirectly related with age on CT (Korkmaz, Çakır, Özden, Oluk & Sarıoğlu, 2015; Yildiz Durak & Sarıtepeci, 2018). In two studies conducted at different levels, Korkmaz et al. (2015) and Yildiz Durak and Sarıtepeci (2018) demonstrated that educational level is in a reverse direction relation with CT. Korkmaz et al. (2015) relates this status with the fact that incompetence of maintaining the sustainability of educational activities on development of CT skill in education institutions. When handled within this context, related researches in which sustainability of CT education activities are achieved or which are about the statuses not including such activities are thought to be beneficial in understanding of educational level's effect. In this study, within this context, which factors are determinant for CT under the conditions that receiving and not receiving programming training which is the most preferred thing in CT education at 5th and 6th grade levels was tried to be determined.

Activities on programming training are most commonly used opinions which are utilized in education of the skills related with CT (García-Peñalvo & Mendes, 2017; Koorsse, Cilliers, & Calitz, 2015; Sarıtepeci & Yildiz Durak, 2017; Sarıtepeci, 2020). When the literature was examined, there are many number of studies on the fact that programming is effective on development of CT and related skills and on reflection of CT levels (Akram et al., 2019; Grover & Pea, 2013; Román-González et al., 2017; Taylor & Baek, 2019; Sarıtepeci & Yildiz Durak, 2017). Though there are criticizing opinions on the fact that programming training activities are handled as the sole option in enabling CT skill (Sarıtepeci, 2019; Voogt, Fisser, Good, Mishra, & Yadav, 2015), it receives a general acceptance that programming training is a strong tool for enabling skills related with CT. From this point forth, whether there are differences in predictiveness of elementary school students' statuses of receiving programming training on CT levels was examined.

## Importance of the Study

It is an important issue to detect with which variables CT, involving many important skills, is in relation at K-12 level. As a result of literature examinations, the result was reached in some related researches that programming is important in CT education (García-Peñalvo, 2017). In some studies, it was emphasized that CT is in relation with demographic and academic variables (Yildiz Durak & Saritepeci, 2018). Yet, it is thought that there is the need in the literature for an up-to-date study which examines these variables in an unified way at middle school level. Additionally, it is also necessary to detect how the current status is among the students who received and who did not receive programming training. Thus, programming training and development process of CT may be regarded as the same procedure steps.

## Aim of the Study

In this study examination of predictiveness of demographic, academic variables and the variables in relation with programming on CT self-efficacy level of middle school students who received and who did not receive programming training was aimed. In line with this aim, following research questions were composed:

1. How are computational thinking self-efficacy, current statuses about programming and academic success levels of the students who received and who did not receive programming training?
2. Do demographic, academic variables of the students who received programming training and their current statuses about programming predict their scores of computational thinking self-efficacy meaningfully?
3. Do demographic, academic variables of the students who did not receive programming training and their current statuses about programming predict their scores of computational thinking self-efficacy meaningfully?

## Methods

In this study which aims to detect the relation between CT self-efficacy level of the students who received and who did not receive programming training and demographic, academic variables and the variables in relation with programming; relational screening model was utilized. While screening model is used in order to detect the current status and tendency, relational model aims to detect the availability and the direction of the relation between the variables (Karasar, 2005). Since it aims to detect the current status about CT self-efficacy and researches the relation between this status and demographic, academic variables and the variables in relation with programming, this study is a research in the model of relational screening.

## Study Group

The participants of the research are composed of 199 students from 5th and 6th grades during spring semester 2018-2019 academic year at a middle school in a city centre in Turkey. Furthermore, the reason behind preferring 5th and 6th grades is the fact that Information Technologies and Software classes are compulsory at these grades. The participants were selected in accordance with convenience sampling method. This method enables the application to be achieved through the participants who are suitable for the target and within easy reach. Among the participants, 50.3% is male and 49.7% is female. Majority (67.3%) of the students are 11 years old. While the percentage of the students who received programming, training is 48.2 (n=96), percentage of the students who did not receive programming training is 51.8 (n=103).

## Data Collection Tools

In the research, Self-Description Form and a scale were utilized as data collection tool.

Self-Description Form: This form was developed by the researchers. Expert opinions were consulted for development of the form. This form is composed of 3 parts. At the first part, there are 3 items about demographic information, at the second part there are 4 items about academic variables and at the last part there are 3 items about programming.

Computational Thinking Self-Efficacy Scale: This scale was developed by Gülbahar, Kert and Kalelioğlu (2019) in order to scale self-efficacy perception of the students from 10-14 age group about computational thinking skills. In particular, this scale is expected to contribute to practitioners in order to better guide students with high self-confidence in programming in their future education. It is composed of totally 36 items and 5 factors. In this scale, which is composed of five factors, sub-scopes are as “Algorithm Designing Efficacy, Problem-solving Efficacy, Data Processing Efficacy, Basic Programming Efficacy, Self-confidence Efficacy". In this study, Cronbach alpha reliability coefficient calculated for the scale is 0.940.

## Data Collection and Analysis

Data were collected online between March-June in 2019 year from middle school students. This research is a screening research to reveal the current situation. As a matter of fact, there is no application related to programming teaching. All secondary schools in Turkey, 5 and 6 are compulsory courses in Information Technology and Software class. The curriculum of this course has been examined. This course is 2 hours a week. In this curriculum, under the "Problem Solving and Programming" unit, "1. Problem-Solving Concepts and Approaches, 2. Programming" titles. Throughout the academic year, this unit was included in the 5th grade for 36 hours with 27 goals. This unit accounts for half of the total 72 hours allocated for the course. Similarly, the same unit was included in the 6th grade. In the 6th grade, 36 hours of the 72-hour lesson is reserved for this unit. Goals and teaching hours for teaching programming standard for all of Turkey, though, are variations on each teacher's practice. A scan was made to see these differences and describe the current situation. Students filled the data collection tool at computing classes.

For the analysis of the collected data, SPSS program was benefited from. Before the analysis; normality, kurtosis, skewness coefficients and homogeneity of the data were examined. Descriptive analyses and linear hierarchical regression analysis were utilized for analysis of the data collected through data collection tools in the research.

In the regression analysis, dependent and independent variables need to be continuous variables scaled by minimum interval scale and to display normal distribution (Büyüköztürk, 2005). In the current study, categorical variables like gender, classroom level, used programming tools were turned into artificial variables which is called "dummy" variable. Discrete variables which were coded as dummy variable were listed as follows. Gender variable has two categories as "female" and "male". "male" category was coded as "0" and turned into dummy variable. For the classroom variable, "5th grade" category was coded as "0". At the "Utilized programming tool" variable, "Scratch" category's values of "1" were coded as "0" and turned into dummy variable. Before the analysis, missing and extreme values were examined. Later, the data were controlled whether to meet linear multiple hierarchical regression analysis hypotheses. Based on the values, the relation between predictive variables and dependent variables is linear and the scores display normal distribution. Furthermore, "multiple connectedness" hypothesis was tested among the predictive variables. It is expected that the VIF value related with independent variables is lower than10 and the tolerance value is higher than 10.; the correlation values between the variables are not higher than .80 level (Field, 2009). Calculated VIF values vary between 1.10 and 2.99. When the VIF values were examined, it can be said that multiple connectedness problem was not available. Additionally, for this reason, the relation between dependent and independent variables was examined through Pearson correlation analysis. It was observed that the relation between variables varied between 0.014 and 0.579 and there was not multiple connectedness problem between the variables.

## Findings

Within frame of research questions, related findings were listed.

*Table 1. Findings on CT Self-efficacy of Middle School Students who Receive and who did not Receive Programming training, Their Statuses about Programming and Their Academic Success Levels*

| | | | | Students who Received Programming Training* | | | Students who did not Receive Programming training | | |
|---|---|---|---|---|---|---|---|---|---|
| | Number of Items | Minimum score | Maximum score | M | M/k | SD | M | M/k | SD |
| Computational Thinking Self-efficacy | 36 | 36.00 | 108.00 | 68.87 | 1.91 | 13.76 | 60.82 | 1.68 | 17.64 |
| *Algorithm Designing Efficacy* | 9 | 9.00 | 27.00 | 16.96 | 1.88 | 3.69 | 14.94 | 1.66 | 4.51 |
| *Problem-Solving Efficacy* | 10 | 10.00 | 30.00 | 18.92 | 1.89 | 5.05 | 16.76 | 1.68 | 5.38 |
| *Data Processing Efficacy* | 7 | 7.00 | 21.00 | 12.34 | 1.76 | 3.70 | 11.51 | 1.64 | 3.90 |
| *Basic Programming Efficacy* | 5 | 5.00 | 15.00 | 9.40 | 1.88 | 2.75 | 8.40 | 1.68 | 2.97 |
| *Self-confidence Efficacy* | 5 | 5.00 | 15.00 | 11.25 | 2.25 | 3.22 | 9.22 | 1.84 | 3.01 |

* Received Programming training: Scratch, Code.org, Kodable, KoduGameLab, mBlock, Robotic Programming, AppInvertor, Alice. Course hours per week: 2

In line with the first research question; descriptive findings on CT self-efficacy of the students, their statuses about programming and their academic success levels were demonstrated at Table 1.

According Table 1, CT self-efficacy levels of the students who received and who did not receive programming training are low. When CT self-efficacy of the students who received and who did not receive programming training was observed, CT self-efficacy scores of the students who received

*Table 2. Findings on Academic Success Levels of Middle School Students who Received and who did not Receive Programming Training*

| | | | | Students who Received Programming training | | | Students who did not Receive Programming training | | |
|---|---|---|---|---|---|---|---|---|---|
| | Number of Items | Minimum score | Maximum score | M | M/k | SD | M | M/k | SD |
| General Academic Average | - | 2 | 5 | 4.63 | - | 0.60 | 4.48 | - | 0.67 |
| Information Technologies Class grade point average | - | 2 | 5 | 4.60 | - | 0.67 | 4.20 | - | 0.90 |
| Math Class grade point average | - | 2 | 5 | 4.47 | - | 0.77 | 4.26 | - | 1.02 |
| Physical Science Class grade point average | - | 1 | 5 | 4.47 | - | 0.61 | 4.32 | - | 0.87 |

programming training were observed to be higher. When CT self-efficacy sub-scopes of the students who received programming training were observed, data processing efficacy had the lowest (M=1.76, SD=3.70); self-confidence efficacy had the highest (M=2.25, SD=3.22) general academic average. When CT self-efficacy sub-scopes of the students who did not receive programming training were also observed, data processing efficacy had the lowest (M=1.64, SD=3.90); self-confidence efficacy had the highest (M=1.84, SD=3.01) general academic average

According to Table 2, general academic average is higher for the students who received programming training compared to the students who did not receive. Similar status is available also for Information Technologies, math, Physical science classes. However, general academic averages were observed to range from 4.20 to 4.63 out of 5. In short, academic success levels of the participants can be said to be high.

According to Table 3, programming experience of the students is 2.67 years in average. While score of attitudes towards programming is 7.69 for the students who received programming training, score of attitudes towards programming is 4.98 for the students who did not receive programming training.

*Table 3. Findings on Academic Success Levels of Middle School Students who Received and who did not Receive Programming training*

| | | | Students who Received Programming Training | | | Students who did not Receive Programming Training | | |
|---|---|---|---|---|---|---|---|---|
| | **Minimum score** | **Maximum score** | **M** | **M/k** | **SD** | **M** | **M/k** | **SD** |
| Programming Experience | 0 | 9 | 2.67 | - | 2.12 | 0.00 | - | 0.00 |
| Attitude towards Programming | 0 | 10 | 7.69 | - | 2.65 | 4.98 | - | 3.00 |

According to Table 4, among the students 103 stated that they did not receive programming training, however 3 indicated that they did not used a programming tool during programming training. It was observed that among the programming tools used by the students reportedly the most frequently (%24.1) used tool was Scratch.

In line with the second research question of the research, results of linear hierarchical regression analysis which was conducted in order to detect the predictiveness of demographic, academic variables

*Table 4. Findings on Programming Environments used by Middle School Students*

| | | f | % |
|---|---|---|---|
| **Used Programming Tools** | I did not used programming tool* | 106 | 53.3 |
| | Scratch | 48 | 24.1 |
| | Code.org | 26 | 13.1 |
| | Other** | 19 | 9.5 |

* Individuals who did not receive programming training or have not used a programming tool in programming training yet.

**Other: Kodable (f=3; %1.5), Kodu GameLab (f=1; %0.5), mBlock (f=7; %3.5), Robotic Programming (f=6; % 3), App Invertor (f=1; %0.5), Alice (f=1; %0.5)

*Table 5. Linear Hierarchical Regression Analysis Results- Students who Received Programming training*

| | | Computational Thinking Self-efficacy | | |
|---|---|---|---|---|
| | | Students who Received Programming Training (n=96) | | |
| **Model** | **Predictor** | **β** | **t** | **p** |
| **1** | Gender | .335** | 3.337 | .001 |
| | Age | .023 | .188 | .852 |
| | Educational level | .094 | .788 | .433 |
| | | **R=.357, $R^2$ =.127, F(3,92)=4.470, Sig.=.006** | | |
| **2** | General Academic Average | -.180 | -1.190 | .237 |
| | Information Technologies Class grade point average | .060 | .450 | .654 |
| | Math Class grade point average | .284* | 2.027 | .046 |
| | Physical Science Class grade point average | -.193 | -1.559 | .123 |
| | | **R=.422, $R^2$ =.178, F(7, 88)=2.728 Sig.=.013** | | |
| **3** | Attitude towards Programming | -.209* | -2.091 | .040 |
| | Programming Experience | .107 | 1.043 | .300 |
| | Used Programming Tool | .287* | 2.813 | .006 |
| | | **R=.522, $R^2$ =.273, F(10,85)=3.184, Sig.=.002** | | |

*p< .05

and the variables related with programming on computational thinking self-efficacy of the students who received programming training were presented at Table 5.

As it was observed at Table 5, with regard to CT self-efficacy 3 steps were added to multiple hierarchical regression analysis for per groups who received and who did not receive programming training. From the total variance of CT self-efficacy scores of the students who received programming training; demographic variables explain 12.7%, academic variables explain 5.1% furthermore, variables related with programming explains 9.5%. Model 1, Model 2 and Model 3 ($p<0.05$) are meaningful. Model 1 ($R=0.357$, $R^2=0.127$, $p<.05$) is among the most important predictors of CT self-efficacy of the students who received programming training. Model 3 (together with previous 2 other models) explains 27.3% of CT self-efficacy of the students who received programming training. Based on regression coefficient, predictor variables' relative order of importance on CT self-efficacy of the students who received programming training is as gender, used programming tool, math class grade point average and the attitude towards programming.

In line with the third research question of the research, results of the linear hierarchical regression analysis conducted in order to detect the predictiveness of demographic, academic variables and the

*Table 6. Linear Hierarchical Regression Analysis Results - Students who did not Receive Programming Training*

| | | Computational Thinking Self-efficacy | | |
|---|---|---|---|---|
| | | students who did not receive programming training (n=103) | | |
| **Model** | **Predictor** | **β** | **t** | **p** |
| **1** | Gender | -.015 | -.147 | .884 |
| | Age | -.044 | -.389 | .698 |
| | Educational level | .026 | .224 | .823 |
| | | **R=.043, $R^2$ =.002, F(3, 99)=0.062, Sig.=0.980** | | |
| **2** | General Academic Average | -.082 | -.490 | .625 |
| | Information Technologies Class grade point average | .004 | .027 | .979 |
| | Math Class grade point average | -.088 | -.572 | .568 |
| | Physical Science Class grade point average | -.281 | -1.845 | .068 |
| | | **R=0.397, $R^2$ =0.158, F(7, 95)=2.547 Sig.=0.019** | | |
| **3** | Attitude towards Programming | -.118 | -1.211 | .229 |
| | Programming Experience | - | - | - |
| | Used Programming Tool | - | - | - |
| | | **R=.413, $R^2$ =.171, F(8, 94)=2.423, Sig.=.020** | | |

*p< .05

variables related with programming on computational thinking self-efficacy of the students who did not receive programming training were presented at Table 6.

As it was observed at Table 6, from the total variance of CT self-efficacy scores of the students who did not receive programming training; demographic variables explain 0.2%, academic variables explain 15.6% furthermore, variables related with programming explains 1.3%. Model 2 and Model 3 ($p<0.05$) are meaningful. Model 2 (R=0.397, R2 =0.158, $p<.05$) is among the most important predictors of CT self-efficacy of the students who did receive programming training. Model 3 (together with previous 2 other models) explains 17.1% of CT self-efficacy of the students who received programming training. Consequently, with regard to CT self-efficacy out of 6 (3+3) models which were created separately for per groups who received and who did not receive programming training 5 models were meaningful.

## Result, Discussion and Suggestions

Aim of this study is examination of predictiveness of demographic, academic variables and variables related with programming on CT self-efficacy of the students who received programming training in this study. In different studies in the literature, effects of programming training on CT self-efficacy were handled. Yet, in the literature any study handling the factors affecting CT self-efficacy of K12 students who received and who did not receive programming training was not reached. Within this context, the factors which predict CT self-efficacy levels of middle school students who received and who did not receive programming training were examined. Within frame of the literature, 3 models were created as per group which received, and which did not receive programming training and these models were tested.

## Demographic Variables and CT Self -Efficacy Level

According to the results of the study, Model 1 in which the relation between demographic characteristics of middle school students who received programming education and CT self-efficacy level is meaningful. Model 1 is the most important predictor of CT self-efficacy of the group which received programming training. Relative effects of demographic characteristics on CT self-efficacy are gender, educational level and age. Model 1 which examines predictive effects of demographic characteristics of the students who did not receive programming training on CT self-efficacy level is not meaningful.

Gender is the predictive variable having the highest relative effect on CT self-efficacy levels of the students who received programming training. Accordingly, CT self-efficacy level of girl participants is higher than boys. However, relative effect of gender on CT self-efficacy of the students who did not receive programming training is not meaningful. Though there are different studies in literature revealing the fact that gender is an effective variable on CT (Dagienė, Pelikis, & Stupurienė, 2015; Román-González, Pérez-González & Jiménez-Fernández, 2017), there are also studies which reveal that CT is not an important factor in terms of CT (Alsancak Sırakaya, 2019; Atmatzidou & Demetriadis, 2016; Korkmaz et al., 2015; Papavlasopoulou, Sharma, & Giannakos, 2019; Taylor & Baek,2019; Yildiz Durak & Sarıtepeci, 2018). In another study displaying similar characteristics with the results gained from the group which received programming training conducted by Sarıtepeci (2017), algorithmic thinking and critical thinking levels of girl participants in CT subscales are meaningfully higher than boys. However, expectation from a group which received programming training is that CT self-efficacy levels of boy students are higher compared to girls or display similar characteristics. This study at this point differentiates form the literature. Prefer of educations which includes different encouraging experiences on CS or programming training in order to direct girls to computer science in recent years and to decrease gender discrimination which is generally thought to be available and decrease of which is supported via different studies (Kelleher, Pausch, &Kiesler, 2007) may have caused a change for support of girl students. Additionally, the fact that effect of gender on programming training and related variables stayed at limited level at K6 level (Kalelioğlu, 2015) may have enabled differentiation of CT self-efficacy levels of girl participants who received programming training at 5th and 6th grades through encouraging educational processes compared to boys.

In both modelling studies, relative effects of variables of educational level and age on CT self-efficacy level are at limited level. According to this, the result was reached that variables of educational level and gender are not meaningful predictors on CT level. Román-González etal. (2017) in their study found that there is a positive direction relation between classroom level and CT skills. However, in their studies Román-González etal. (2017) made examinations in two-paired groups as 5&6, 7&8, and 9&10. When it is handled with this regard, it can be said that Román-González et al. (2017) predicted that there is not a difference in CT levels in terms of educational levels between 5th and 6th grades. In some studies, however, it was reported that there is a negative direction relation with educational level and CT self-efficacy level (Yildiz Durak & Sarıtepeci, 2018; Korkmaz et al., 2015). In both these two researches, wider ranges compared to this study were handled in terms of educational levels. Additionally, in both studies, not maintaining the continuity of CT training and In CS training, experiencing interruptions at different steps were indicated as the main reason of the reverse direction relation of classroom level and CT level (Yildiz Durak & Sarıtepeci, 2018; Korkmaz et al., 2015).

## Academic Variables and CT Self-Efficacy Level

Based on the results gained through hierarchical regression analysis, the relation between CT levels of the students who received programming training and academic variables is meaningful. Contribution of academic variables which were included to hierarchical regression at Model 2 to explained variance is 5.1%. Relative effect of academic variables on CT self-efficacy is listed as math class grade point average, science class grade point average, generalized grade point average and information technologies class grade point average. Similarly, also in the group which did not receive programming training it was determined that there is a meaningful relation between academic variables and CT self-efficacy level. Contribution of academic variables in the group which did not receive programming training to explained variance is 15.6% and with this ratio it is most explanatory step of the model. In this group relative effect of academic variables on CT self-efficacy is listed as science class, moth class, general and information technologies class respectively.

Math class grade point average is the predictor variable which mostly comes to the fore front among the academic variables related to CT self-efficacy level in the group having programming training. Math class grade point average is one of important predictors of CT self-efficacy level. According to this, depending on the increase of math success of students, it was determined that CT self-efficacy level increased. In the literature, there are studies revealing the fact that effects of math class success and attitude towards math class on CT skill are important (Yildiz-Durak & Sarıtepeci, 2018). However, based on the findings of hierarchical regression relative effects of other variables on CT self-efficacy are at limited level and it was determined that these are not meaningful predictors. Moreover, in the group which did not receive programming training, even Model 2 through which academic variables were included to hierarchical regression is meaningful, relative effect of each one of academic variables on CT self-efficacy level is not meaningful.

In both modelling study, the result was reached that grade point average of information technologies class is not an important predictor of CT. Education and development of CT skill is handled within frame of education of computer science. In the supporting way, Buitrago Flórez et al. (2017) states that CT consists use of basic concepts related with CS in solution of any problems. Within this context, emergence of such a result can be related with the fact that content of IT class in terms of utilized method and technique does not have a structure which will support CT skill directly or indirectly, or with the fact that IT class is not proceeded in a way which will enable this. Additionally, the result can be related with the entity of a negative direction but not meaningful relation between science class success level and CT. In contrast, science class is expected to have a structure that assist the development of CT-related competencies (NRC, 2012; Yıldız Durak & Sarıtepeci, 2018). Accordingly, it can be deduced that the participants did not have a learning experience suitable for the nature of the science class.

## Variables Related with Programming and CT Self-Efficacy Level

According to the results of the study, it was determined that there is a meaningful relation between the variables related with programming which was included to hierarchical regression analysis at Model 3 in the group which received programming training and CT self-efficacy level. Variables related with programming explain 9.5% of total variance. Order of importance of the variables related with programming on CT self-efficacy levels of participants is as utilized programming tool, attitude towards programming and programming experience. In the group which did not receive programming training, Model 3 through which variables related with programming was included to the analysis is meaningful. Model 3 explains 1.3% of total variance. In the group which did not receive programming training, attitude towards programming takes part in Model 3.

In the group which received programming training, utilized programming tool is the variable related with programming whose relative effect on CT self-efficacy is highest. According to this, CT self-efficacy levels of the students in whose programming activities Scratch was utilized are higher compared to the students received education with other programming tools. This status can be related with the facts that Scratch has simple use compared to other visual programming languages, it does not need Internet connection, it operates in computers with low-level hardware and it consists big part of the characteristics of other frequently preferred programming languages. In supporting way, in the literature in some studies it was reported that utilization of Scratch in programming training is effective on students' acquiring positive experiences about programming and developing positive attitudes (Smith & Neumann, 2014). In addition, Scratch was frequently tackled in various researches (For example: Holt, 2011; Marcelino et al., 2018; Oluk & Korkmaz, 2016; Saritepeci, 2020) as an effective tool in teaching and developing CT-related competencies.

According to the results of applied hierarchical regression, it was determined that attitude towards programming on CT self-efficacy level was statistically meaningful predictor in received programming training group. There was a significant inverse relation between attitude towards programming and CT self-efficacy level in this group. The main reason of this issue can be shown as the lack of a structure that includes attractive activities that support the active learner engagement and creativity of programming teaching processes. In support of this, there was that the relative effect of programming experience on CT skill level was limited in the hierarchical regression model. In the literature, it is emphasized that, in some programming trainings, the activities are superficial and intensive use of demonstration method has a limited effect on CT teaching and development of these skills (Lye & Koh 2014; Saritepeci, 2020).

In the group which did not receive programming training there is not a meaningful relation between attitude towards programming and CT. In both modelling studies there was a negative direction relation between CT self-efficacy and attitude toward programming. In the group which did not receive programming training, it is within scope of the expectation that there is a preconception against programming training and as a result of this entity of a negative attitude. Because in general there is an acceptance that it is hard to understand and maintain programming training (Yukselturk & Altiok, 2017).

## Limitations and Recommendations for Practice and Further Research

There are some limitations within frame of this study. In the current study, even statistically there is not a meaningful difference between students' general academic averages, yet there is difference as arithmetic average.

Students' prior knowledge about previous programming is limited only with the year they received education. Content and learning level were not taken into consideration. By examining their experiences in programming training in depth in future studies, the model developed in this study can be extended.

It was found that the programming tool utilized in programming process was one of the important predictors on CT self-efficacy of the students. How and in which way the programming tools contribute to the development of CT self-efficacy can be examined via experimental studies.

Math class grade point average is one of the important predictors on CT self-efficacy. There is the need for future studies on how and in which way math subjects can be integrated to programming activities and applications.

## ACKNOWLEDGMENT

This study was not supported by any financially supported organization.

## REFERENCES

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835. doi:10.1093/comjnl/bxs074

Akram, B., Min, W., Wiebe, E., Mott, B., Boyer, K. E., & Lester, J. (2019). Assessing Middle School Students' Computational Thinking Through Programming Trajectory Analysis. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1269-1269). ACM. 10.1145/3287324.3293798

Alsancak Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Turkish Journal of Social Research/Turkiye Sosyal Arastirmalar Dergisi, 23*(2).

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661–670. doi:10.1016/j.robot.2015.10.008

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. doi:10.1145/1929887.1929905

Boechler, P., Artym, C., Dejong, E., Carbonaro, M., & Stroulia, E. (2014). Computational thinking, code complexity, and prior experience in a videogame-building assignment. In *2014 IEEE 14th International Conference on Advanced Learning Technologies* (pp. 396-398). IEEE. 10.1109/ICALT.2014.118

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the annual meeting of the American Educational Research Association, Vancouver, British Columbia, Canada. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834–860. doi:10.3102/0034654317710096

Büyüköztürk, Ş. (2005). *Handbook of data analysis for social sciences*. Ankara: PegemA.

Çetinkaya, H. N. (2019). *Bilişim teknolojileri ve yazılım dersindeki etkinliklerin bilgi işlemsel düşünme ve bazı değişkenler açısından incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi. İnönü Üniversitesi, Eğitim Bilimleri Enstitüsü.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162–175. doi:10.1016/j.compedu.2017.03.001

Computer Science Teachers Association, & International Society for Technology in Education. (2011). *Computational Thinking: Leadership Toolkit* (1st ed.) Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf

Dagienė, V., Pelikis, E., & Stupurienė, G. (2015). Introducing Computational Thinking through a Contest on Informatics: Problem-solving and Gender Issues. *Informacijos Mokslai*. *Information Sciences*, 73.

Durak, H. Y. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology*. *Knowledge and Learning*, *25*(1), 179–195. doi:10.100710758-018-9391-y

Field, A. P. (2009). *Discovering statistics using SPSS* (3rd ed.). London: Sage.

García-Peñalvo, F. J. (2017, October). Computational thinking issues. In *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality* (p. 1). ACM.

García-Peñalvo, F. J., & Mendes, A. J. (2017). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, *80*, 407–411. doi:10.1016/j.chb.2017.12.005

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği: Geçerlik ve Güvenirlik Çalışması. *Turkish Journal of Computer and Mathematics Education*, *10*(1), 1–29.

Holt, L. (2011). *Creating Digital Stories with Scratch to Promote Computational Thinking*. Paper presented at the Society for Information Technology & Teacher Education International Conference.

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. doi:10.1016/j.compedu.2018.07.004

Karasar, N. (2005). *Scientific research method*. Ankara: Nobel Publishing.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2011). Understanding computational thinking before programming: Developing guidelines for the design of games to learn introductory programming through game-play. *International Journal of Game-Based Learning*, *1*(3), 30–52. doi:10.4018/ijgbl.2011070103

Kelleher, C., Pausch, R., Pausch, R., & Kiesler, S. (2007, April). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). ACM. 10.1145/1240624.1240844

Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, *82*, 162–178. doi:10.1016/j.compedu.2014.11.020

Kordaki, M., & Kakavas, P. (2017). Digital Storytelling as an effective framework for the development of computational thinking skills. *EDULEARN2017*, 3-5.

Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., & Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, *34*(2), 68–87.

Kukul, V. (2018). *The effect of different structured processes on students' computational thinking skills, self-efficacy levels and programming achievements in programming teaching* (Unpublished PhD Thesis). Gazi University, Turkey

Lockwood, J., & Mooney, A. (2017). *Computational thinking in education: Where does it fit? A systematic literary review*. Retrieved from https://arxiv.org/abs/1703.07659v3

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. doi:10.1016/j.chb.2014.09.012

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational Thinking and scratch at distance. *Computers in Human Behavior*, *80*, 470–477. doi:10.1016/j.chb.2017.09.025

Moreno-León, J., Román-González, M., & Robles, G. (2018, April). On computational thinking as a universal skill: A review of the latest research on this ability. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1684-1689). IEEE. 10.1109/EDUCON.2018.8363437

National Research Council (NRC). (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.

NRC. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

Oluk, A., & Korkmaz, Ö. (2016). Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables. *Online Submission*, *8*(11), 1–7.

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *Int. J. Teaching and Case Studies*, *10*(1), 31–52.

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2016). Using Scratch and App Inventor for teaching introductory programming in Secondary Education. A case study. *International Journal of Technology Enhanced Learning*, *1*(1), 1. doi:10.1504/IJTEL.2016.10001503

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, St., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in Preschool Education. A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187–202. doi:10.1504/IJMLO.2016.077867

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Piteira, M., & Costa, C. (2012, June). Computer programming and novice programmers. In *Proceedings of the Workshop on Information Systems and Design of Communication* (pp. 51-53). ACM.

Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, *25*(5), 66–71.

Riley, D., & Hunt, K. A. (2014). *Computational thinking for the modern problem solver*. Chapman and Hall/CRC. doi:10.1201/b16688

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. doi:10.1016/j.chb.2016.08.047

Sarıtepeci, M. (2017). *Analysis of computational thinking skill level in secondary education in terms of various variables*. Paper presented at the 5th International Instructional Technologies & Teacher Education Symposium–ITTES, İzmir, Turkey.

Saritepeci, M. (2020). Developing Computational Thinking Skills of High School Students: Design-Based Learning Activities and Programming Tasks. *The Asia-Pacific Education Researcher*, *29*(1), 35–54. doi:10.100740299-019-00480-2

Sarıtepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational Research and Practice*, 490-501.

Seow, P., Looi, C. K., How, M. L., Wadhwa, B., & Wu, L. K. (2019). Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore. In *Computational Thinking Education* (pp. 345–361). Singapore: Springer. doi:10.1007/978-981-13-6528-7_19

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. doi:10.1016/j.edurev.2017.09.003

Smith, C. P., & Neumann, M. D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, *21*(3), 185–188. doi:10.5951/teacchilmath.21.3.0185

Stein, C., & Nickerson, K. (2004). Botball robotics and gender differences in middle school teams. *Proceedings American society for engineering education annual conference*. http://dpm.kipr.org/papers/asee04-gender.pdf

Taylor, K., & Baek, Y. (2019). Grouping matters in computational robotic activities. *Computers in Human Behavior*, *93*, 99–105. doi:10.1016/j.chb.2018.12.010

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.

Wong, G. K. W., & Cheung, H. Y. (2018). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 1–13.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. doi:10.100711528-016-0087-7

Yildiz Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, *34*(6), 740–752. doi:10.1111/jcal.12281

Yildiz Durak, H., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191–202. doi:10.1016/j.compedu.2017.09.004

Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, *48*(3), 789–801. doi:10.1111/bjet.12453

Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. doi:10.1016/j.compedu.2019.103607

## ADDITIONAL READING


Durak, H. Y. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology. Knowledge and Learning*, *25*(1), 179–195. doi:10.100710758-018-9391-y

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği: Geçerlik ve Güvenirlik Çalışması. *Turkish Journal of Computer and Mathematics Education*, *10*(1), 1–29.

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2016). Using Scratch and App Inventor for teaching introductory programming in Secondary Education. A case study. *International Journal of Technology Enhanced Learning*, *1*(1), 1. doi:10.1504/IJTEL.2016.10001503

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, St., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in Preschool Education. A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187–202. doi:10.1504/IJMLO.2016.077867

Saritepeci, M. (2020). Developing Computational Thinking Skills of High School Students: Design-Based Learning Activities and Programming Tasks. *The Asia-Pacific Education Researcher*, *29*(1), 35–54. doi:10.100740299-019-00480-2

Sarıtepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational research and practice*, 490-501.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. doi:10.100711528-016-0087-7

Yildiz Durak, H., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191–202. doi:10.1016/j.compedu.2017.09.004

## KEY TERMS AND DEFINITIONS

**Algorithm Designing Efficacy:** It is the ability to use algorithms by making logical inquiries knowing for which purpose the algorithms are used.

**Basic Programming Efficacy:** It is the ability to know and apply the basic concepts and stages of the programming process.

**Computational Thinking:** This concept is a thinking process which expresses active use of information and communication technologies' concepts in solution of complex problems.

**Computational Thinking Self-Efficacy:** Within the scope of this study, this concept is considered as a competence that includes the dimensions of "Algorithm Designing Efficacy, Problem-solving Efficacy, Data Processing Efficacy, Basic Programming Efficacy, Self-confidence Efficacy."

**Data Processing Efficacy:** Being aware of what the data is, the ability to use their knowledge and skills on data types, data presentation and the transformation of digital data into different forms in the problem-solving process.

**Problem-Solving Efficacy:** It is the adequacy of performing a problem-solving process in a logical context using experiences.

**Self-Confidence Efficacy:** In the problem-solving process, it is the awareness of noticing mistakes, choosing the best way, and prioritizing work/processes.

# Chapter 4

# Automating the Assessment of Algorithms and Programming Concepts in App Inventor Projects in Middle School

**Nathalia da Cruz Alves**
https://orcid.org/0000-0002-3523-9199
*Federal University of Santa Catarina, Brazil*

**Christiane Gresse von Wangenheim**
*Federal University of Santa Catarina, Brazil*

**Jean C. R. Hauck**
https://orcid.org/0000-0001-6550-9092
*Federal University of Santa Catarina, Brazil*

**Adriano F. Borgatto**
https://orcid.org/0000-0001-6280-2525
*Federal University of Santa Catarina, Brazil*

## ABSTRACT

*As computer science education makes its way into schools, diverse initiatives worldwide promote computer science education in K-12, often focusing on teaching algorithms and programming with block-based programming languages such as Scratch or App Inventor. However, alternatives to assess the learning of computer science concepts on this educational stage are still scarce. This chapter presents an automated rubric for assessing algorithms and programming concepts of App Inventor projects at middle school level. The assessment is based on a rubric proposed in alignment with the K-12 Computer Science Framework with satisfactory reliability and validity. The rubric has been automated through a web-based system that allows assessing App Inventor projects through static code analysis. As a result, it can support computer science education in practice providing feedback to students and teachers.*

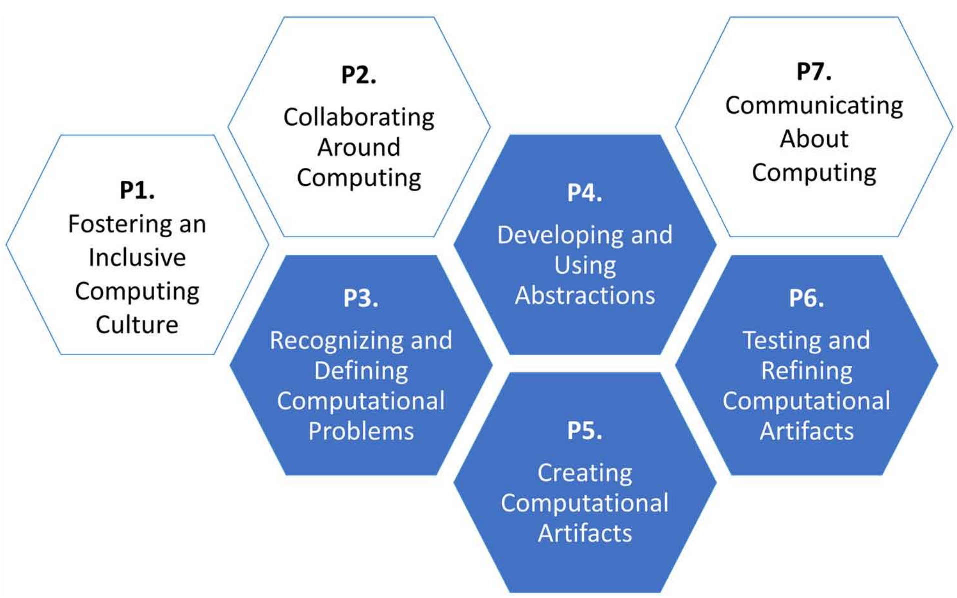
DOI: 10.4018/978-1-7998-4576-8.ch004

## INTRODUCTION

Computer science is relevant to most activities nowadays. Regardless of a professional's area of expertise, it is essential to know the fundamentals and basic principles of computer science. However, in general, people are not as well informed as they should be demonstrating a knowledge gap in computer science on all levels (CSTA, 2016). In order to popularize computer science for all, several initiatives have emerged around the world (Bocconi, Chioccariello, Dettori, Ferrari, & Engelhardt, 2016). And, as the inclusion of computer science in K-12 is considered one of the most effective approaches, several efforts have been made to develop guidelines and curricula for K-12 computer science education. One of the most prominent is the K-12 Computer Science Framework developed by the Computer Science Teachers Association (CSTA, 2016). The CSTA K-12 framework serves as a high-level guide that can be used for developing custom computer science curricula for three educational levels for ages ranging from kindergarten through high school. Middle school typically consists of grades 5-6 to 9-10, ranging from age 10-11 to 14-15 years.

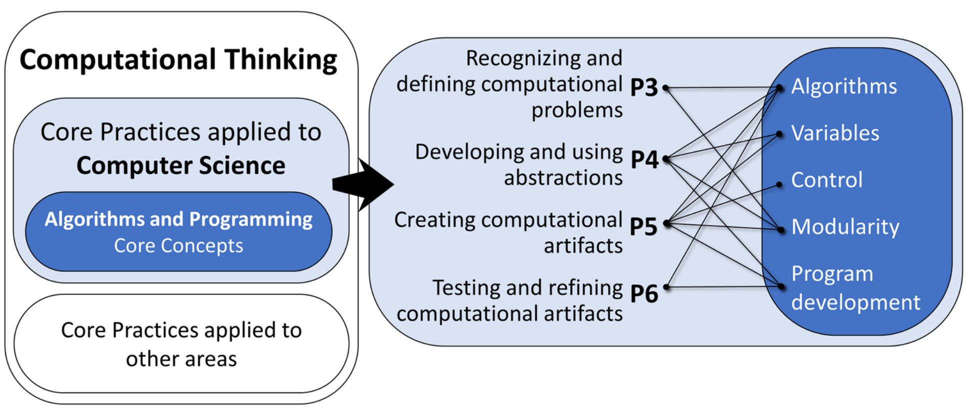
The CSTA K-12 framework defines a set of core practices that refer to behaviors that computer science literate students use to fully engage with basic computer science concepts (CSTA, 2016). These practices defined in the framework also embrace computational thinking by practices 3, 4, 5, and 6 (Figure 1). Computational thinking refers to the thinking processes involved in creating algorithmic solutions, or step-by-steps, that can be performed by a computer (Wing, 2006). It is a relevant universal competence, not only reserved for computer scientists, representing a way in which humans solve problems (Wing, 2006). Complementary, core practices 1, 2, and 7 (Figure 1) represent general computer science practices that complement computational thinking.

*Figure 1. Computational thinking core practices in the K-12 Computer Science Framework (CSTA, 2016)*



According to the CSTA K-12 framework (CSTA, 2016), computational thinking core practices can be applied to areas other than computer science. However, computer science offers unique opportunities to develop computational thinking (CSTA, 2016). Thus, in the context of this chapter, the connection between practices of computational thinking and computer science algorithms and programming concepts is explored, as proposed by the K-12 Computer Science Standards (CSTA, 2017) in the K-12 Computer Science Framework (CSTA, 2016).

*Figure 2. Computational thinking core practices and their connection with computer science algorithms and programming concepts (CSTA, 2016; CSTA, 2017)*

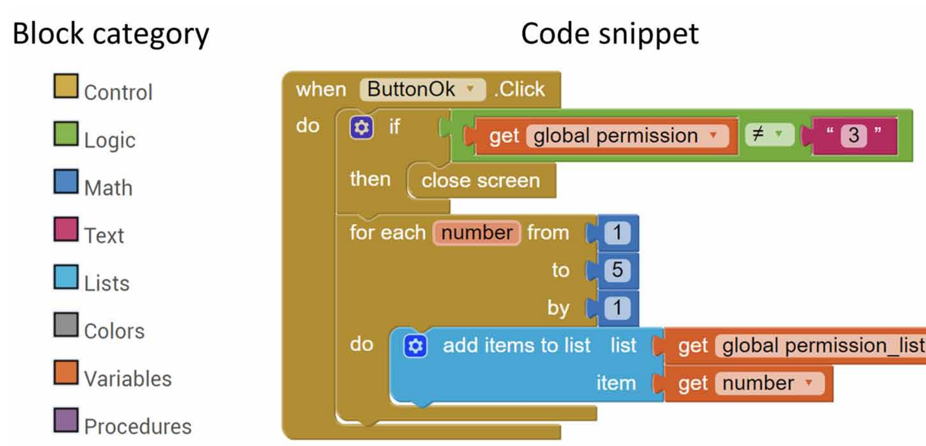



Algorithms and programming are used to control all computing systems, enabling people to solve problems and communicate with the world in new ways. Basic algorithms and programming concepts include algorithms, variables, control, modularity, and program development (Figure 2). Algorithms can be implemented using a programming language and are designed to be performed by humans and computers. Variables are used for storing and manipulating data from computer programs. Control structures are used to specify the order in which instructions are executed within an algorithm or program (e.g., using loops and conditionals). Modularity is used for dividing complex tasks into simpler tasks and combining them to create something complex, such as creating procedures and functions. Program development represents the software engineering process that is repeated until the criteria are satisfied (CSTA, 2016). Other guidelines and curricula, such as Computing at School (CAS, 2015) or the Australian Curriculum, Assessment and Reporting Authority (ACARA, 2015), cover similar core concepts and practices.

There are several ways to develop computational thinking in K-12 such as workshops, summer camps, courses, etc.) (Grover & Pea, 2013; Lye & Koh, 2014). These instructional units often focus on algorithm and programming concepts, adopting mostly practical activities. These include well-structured activities, such as part of the Hour of Code initiative (CODE, 2015) or Code Combat (CodeCombat, 2016). Such activities typically have a complete description of what the student should do, and the expected result of the activity is the same for everyone. On the other hand, ill-structured learning activities focus on creating original computational artifacts, such as games, animations, and applications (Brennan & Resnick, 2012; Seiter & Foreman, 2013). Such activities have several possible solutions, and the choice on which way to reach a solution is up to the student (Reed, 2002).

Often, programming activities in K-12 are performed using visual block-based programming environments (Lye & Koh, 2014). Visual block-based programming environments allow programming via graphical blocks rather than textual commands (Figure 3). These environments resemble programming blocks to pieces of a puzzle (Weintrop & Wilensky, 2015), where each block has a color that corresponds to the color of its category (Figure 3). Most block-based programming environments reduce syntactic errors since the graphic format of each block does not allow the nesting of blocks that do not satisfy the language syntax (Weintrop & Wilensky, 2015). They contain command blocks that are related to algorithmic and programming concepts, such as loops, variables, conditionals, etc.

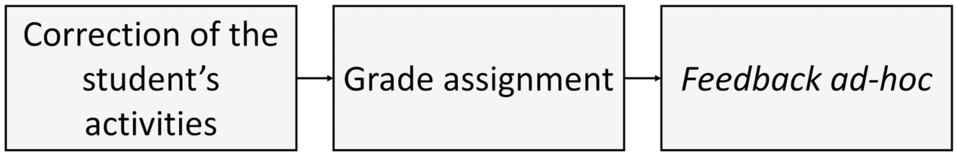
*Figure 3. Example categories and code in a block-based programming environment*




There exist diverse visual block-based programming environments, such as App Inventor (https://appinventor.mit.edu/), Scratch (https://scratch.mit.edu/), and BYOB Snap (https://snap.berkeley.edu/). These block-based programming environments are indicated for introducing algorithms and programming concepts in primary and secondary education (Papadakis et al., 2017). They allow creating not only simple programs but also very complex programs such as 2D and 3D games, mobile applications, and animations.

As part of the learning cycle, it is crucial to assess students' projects to provide feedback to them and the teacher. Feedback is important for the student to know his/her performance as well as to the teacher to improve retention and knowledge transfer. This includes formative assessment, which is the process of collecting data on student learning during the instructional unit in order to revise the instructional process to make it more effective (Branch, 2009), as well as, summative assessment, which is the process of data collection at the end of the educational process, typically allocating a grade to determine the degree of achievement of learning objectives (Branch, 2009). A grade may refer to a quantitative or qualitative value, e.g., concepts such as 'A' to 'E' may be assigned, or values within a defined range, such as 0 to 10, or percentages from 0 to 100. By manually assessing ill-structured activities, the teacher typically performs the assessment of the main topics addressed in the instructional unit. Then, the teacher assigns a grade individually and provides instructional feedback depending on the issues observed in the student project (Figure 4).

*Figure 4. Manual assessment process of ill-structured learning activities*




General criteria based on the learning objectives are used in order to synthesize student learning with evidence (Branch, 2009). However, there is no consensus in the literature on which measurement criteria should be common for assessing ill-structured learning activities (Ala-Mutka & Jarvinen, 2004; Brennan & Resnick, 2012; Grover, Cooper, & Pea, 2014). Typically, correct functionality, good design, and programming style are desirable features. Yet, the definitions and importance of these characteristics in the assessment vary widely. Different teachers emphasize different aspects based on their personal experience and the course objectives (Ala-Mutka & Jarvinen, 2004; Fitzgerald, Hanks, Lister, McCauley, & Murphy, 2013).

To carry out a uniform and consistent assessment of ill-structured learning activities, rubrics are frequently used (Stegeman, Barendsen, & Smetsers, 2016). Rubrics assist in defining more transparent and effective assessment metrics, allowing a more precise detail of the assessment process and, at the same time, facilitate the diagnosis of specific problems within the teaching-learning process. A rubric consists of a set of items (or criteria) to identify the characteristic to be measured, a rating scale with several performance levels that can be achieved and a scoring system (qualitative or quantitative) related to each performance level, and verbal descriptors that detail verbatim for each performance level. Using a rubric, the teacher can calculate grades and provide feedback indicating the performance level reflected by the specific score(s) of an assessment.

However, a manual assessment process, especially with larger classes, is time-consuming and repetitive. Thus, the automation of assessment can bring several benefits, including also the consistency between assessments as they are all done equally and quickly. Considering also practical constraints and a trend towards open and massive online courses, the development and improvement of automated solutions is required in order to enable fast, real-time assessment and feedback for students can further improve students' learning process as well as reduce the workload of teachers.

Although there exist some rubrics, mostly focusing on the assessment of Scratch projects (Alves, Gresse von Wangenheim, & Hauck, 2019), there is still a lack of automated support for the assessment of App Inventor projects and evaluation of the proposed approaches. Thus, this chapter presents an automated rubric for assessing algorithms and programming concepts related to computational thinking practices based on the source code of App Inventor projects created as a result of ill-structured learning activities as part of a long-term research on assessing the learning of computing competencies in K-12. Earlier results presenting the CodeMaster rubric have been published in (Alves, Gresse von Wangenheim, Hauck, & Borgatto, 2020). This chapter focus on detailing the development and the assessment process of the CodeMaster rubric, in alignment with computational thinking core practices and algorithms and programming concepts as defined in the K-12 Computer Science Framework (CSTA, 2016). In addition, novel evaluation results are presented, including an analysis of the correlations between pairs of criteria (or items), analyzing in-depth the relationship between each defined criteria. The CodeMaster rubric is available online (http://apps.computacaonaescola.ufsc.br:8080/) in order to facilitate the assessment process as part of computer science education with App Inventor.

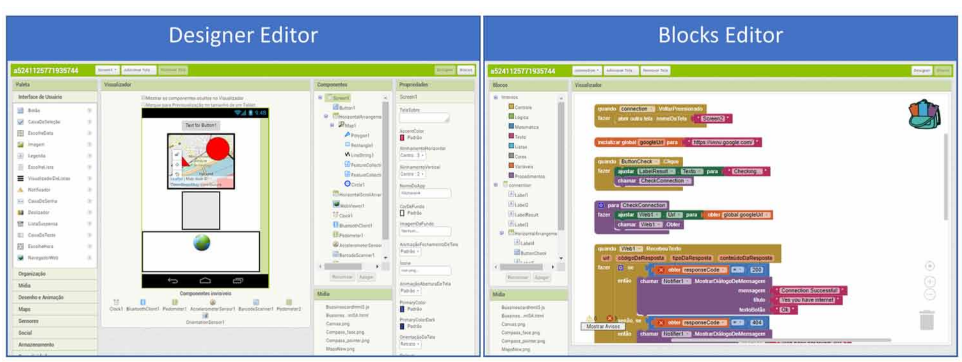
## BACKGROUND

### App Inventor

Considering the wide use of smartphones worldwide (NewZoo, 2020), a programming environment that allows anyone without advanced knowledge in mobile application development to create their application can contribute significantly to popularize computing competencies. In this context, App Inventor is a block-based programming environment for building mobile applications for Android devices (Figure 5). Compared to other similar novice programming environments, e.g., Scratch and Alice, App Inventor allows students to engage more with algorithms and programming concepts (Papadakis, 2019) and requires fewer technological resources (Papadakis and Orfanakis, 2018).

App Inventor uses commands related to mobile algorithms and programming concepts, including commands that use mobile device features. It is an open-source project that was initially created by Google and is currently maintained by the Massachusetts Institute of Technology (MIT). App Inventor provides an online programming environment with a block-based command editor (Figure 5). The current version is App Inventor 2.0, the older version App Inventor Classic was taken out of production in 2015. Using a mobile device or emulator running the Android operating system and the App Inventor Companion application, the applications being developed can be tested in real-time. App Inventor is available in several languages, including English, Spanish, French, Chinese, and Portuguese.

*Figure 5. App Inventor programming environment*




With App Inventor, a project of a mobile application can be created in two stages. First, user interface components are defined in the Designer area (Figure 5), where it is also possible to specify non-visual components, such as sensors, social and media components, that access features from the mobile device or other applications. In a second stage, in the Blocks area, the application behavior is programmed by connecting visual blocks (Figure 5). Command blocks represent events, conditions, or actions for a specific mobile application component, such as button pressed, taking a picture with a camera, etc. (Table 1). They can also represent algorithmic and programming concepts such as variables, loops, etc. (Turbak, Mustafaraj, Svanberg, & Dawson, 2017).

*Table 1. Designer area components and commands in the App Inventor Blocks area*

| Designer component | Description |
|---|---|
| User Interface | All visible components of the application to create a visual interface like buttons, images, text boxes, etc. |
| Layout | Components that assist in the organization of visible user interface components. |
| Media | All media components like cameras, media players, translators, etc. |
| Drawing and Animation | Components that allow the user to draw and view animations |
| Maps | Components that allow insertion of maps with markers. |
| Sensors | Components that interact with mobile device sensors such as accelerometer, barcode scanner, gyroscope, location, pedometer, etc. |
| Social | Components that allow the application to communicate with other social applications like email, phone calls, text messages, twitter, etc. |
| Storage | Components that allow the creation of different types of databases to store data. |
| Connectivity | Components that allow connection to other devices or applications via Bluetooth, WEB or Application Programming Interface (API). |
| LEGO® MINDSTORMS® | Components that provide a high-level interface for the robot LEGO MINDSTORMS |
| Experimental | Components available that are not stable yet (alpha test) |
| Extensions | An option that allows adding external libraries with additional programming blocks. |
| **Programming block** | **Description** |
| Control | Commands responsible for controlling loops, conditionals, and iterations in lists. |
| Logic | Commands responsible for logical operations on Boolean variables and operands. |
| Math | Commands responsible for mathematical operations on variables, including basic operations such as addition, subtraction, multiplication, and division, as well as more complex operations such as sine, cosine, and so on. |
| Text | Commands that handle text manipulation, including commands for string concatenation, string division, and so on. |
| Lists | Commands responsible for creating and manipulating list data. |
| Colors | Commands responsible for setting the color of application components. |
| Variables | Commands responsible for creating and manipulating variables. |
| Procedures | Commands responsible for defining procedures and functions. |
| Events | Commands that handle events triggered by the Designer components. |
| Get and Set | Commands that change the attributes of the Designer components. |

## Related Work

Considering the importance of support for the assessment of programming activities in order to expand the teaching of computer science in K-12, an analysis of the state-of-the-art was performed following the procedure defined by Petersen, Feldt, Mujtaba, and Mattsson (2008). The analysis of the state-of-the-art aims at identifying existing approaches (methods, models, rubrics, frameworks) to assess algorithms and programming concepts in the context of K-12. Detailed results of the analysis are presented in Alves, Gresse von Wangenheim, & Hauck (2019).

## Review Protocol

The main objective of this systematic mapping study was to identify approaches for assessing programming activities based on code created with block-based visual programming languages. Therefore, studies assessing programming competencies based on other artifacts, such as tests, interviews, etc. were excluded. In order to enable the analysis of relevant topics in this context, studies that do not present substantial information regarding which and how programming concepts related to computational thinking are assessed were excluded. Studies that do not present significant details if and how instructional feedback and assessment are provided were also excluded. The search included studies published between 1997 and 2018.

## Results

Only a few approaches were found (Alves, Gresse von Wangenheim, & Hauck 2019). The existing approaches are used for formative and summative assessment. They are designed for different types of programming activities, including well-structured activities with a correct or a single solution, as well as ill-structured activities within the context of project-based learning. The goal of most approaches is to support the assessment process. However, some are also intended for students to monitor and guide their learning progress. Examples are Dr. Scratch (Moreno-Leon & Robles, 2015), Ninja Code Village (Ota, Morimoto, and Kato, 2016), CodeMaster 1.0 (Gresse von Wangenheim et al., 2018) and CTP (Koh, Basawapatna, Nickerson, & Repenning, 2014). Few approaches provide hints or suggestions (other than a grade) to guide the learning process in a personalized way. Approaches typically provide grades as feedback based on code analysis. Two approaches (Moreno-León & Robles, 2015; Gresse von Wangenheim et al., 2018) use gamification elements, presenting the assessment result in a playful way using badges.

Only a few approaches are automated through a static and dynamic analysis of the code created by the students. An advantage of static analysis approaches is that there is no need of pre-defined solutions in advance as they measure certain code qualities. Thus, static analysis provides a solution for assessing ill-structured activities in project-based learning contexts. However, the lack of a predefined solution for such ill-structured activities limits the analysis of specific characteristics, not allowing, for example, to validate the correctness of the code. Dynamic analysis approaches, on the other hand, can be applied for the assessment of well-structured activities. However, a disadvantage is that the internal code structure is not analyzed. Therefore, a program can be considered correct (when generating the expected output) even when the expected programming commands are not used, and the solution is poorly designed, for example, when repeating commands instead of using loops.

Among the automated approaches, some do not provide a graphical user interface displaying the results only via the terminal that runs the script. This can difficult their adoption due to the lack of technical knowledge of non-computer people. Another factor that can difficult their widespread adoption is their availability in English only, with only a few exceptions available in multiple languages. Another disadvantage is that the existing tools are provided as standalone solutions not integrated into programming environments or course management systems in order to facilitate their adoption in the educational context.

By focusing on the assessment through the analysis of the code created by the students, the approaches infer the assessment of computational thinking practices by analyzing if algorithms and programming concepts are present in the code. This illustrates the strong emphasis on the analysis of programming-related skills by most approaches. Additional elements, such as usability, code organization, documentation, aesthetics, or creativity, are rarely assessed, mostly by manual approaches only. This subjective assessment, less found in automated approaches based solely on code analysis, also indicates the need for alternative assessment methods. Alternative methods, such as interviews or observations, can provide a broader assessment, especially when it comes to computational thinking practices and perspectives (Brennan & Resnick, 2012). In this regard, approaches based on code analysis can be considered as one way of assessing computational thinking based on a performance-based assessment of the algorithms and programming concepts used in the code, rather than being the sole means of assessment and should be used together with other types of assessment. Thus, one of the aims of automating performance-based assessment is to free the teachers' time, allowing them to concentrate on other complementary methods for assessing more subjective, unique, and original criteria.

### Existing Solutions for App Inventor

Currently assessment approaches specifically focusing on App Inventor such as the mobile computational thinking rubric (Sherman & Martin, 2015), Quizly (Maiorana, Giordano, & Morelli, 2015), and CodeMaster 1.0 (Gresse von Wangenheim et al., 2018) are very scarce. Furthermore, as Quizly analyzes well-structured activities with correct solutions known in advance (Maiorana, Giordano, & Morelli, 2015), only mobile computational thinking rubric (Sherman & Martin, 2015) and CodeMaster 1.0 (Gresse von Wangenheim et al., 2018) provide support for assessing ill-structured activities. Yet, another disadvantage of these approaches for ill-structured activities is that the mobile computational thinking rubric (Sherman & Martin, 2015) has not been automated, supporting only the manual assessment of the mobile applications developed by the students. None of the approaches presents an evaluation of reliability and validity. Other shortcomings of the existing approaches for App Inventor are the lack of a definition of the educational level they address. Therefore, there is a need not only for automated assessment approaches for assessing ill-structured activities with App Inventor, but also for the conceptual definition of computational thinking assessment strategies and criteria according to the respective educational level in alignment with well-established curricula and frameworks. In this context, this chapter presents and analyzes an automated approach for the assessment of App inventor projects created as a result of ill-structured learning activities. The criteria for the assessment are derived from learning objectives defined by the CSTA standard (CSTA, 2017) and identified in the state-of-the-art.

## RESEARCH METHOD

Following a multimethod approach (Saunders, Lewis, & Thornhill, 2009), this research is conducted in several phases as a part of our long-term research on the assessment of computing education in K-12:

1. Design of the CodeMaster rubric. Following the instructional design model ADDIE (Branch, 2009), the rubric development approach proposed by Moskal and Leydens (2000) and the Goal/Question/Metric approach (Basili, Caldiera, & Rombach, 1994), the CodeMaster rubric has been developed by defining assessment items with respect to learning objectives in alignment with the K-12 Computer Science Framework (CSTA, 2016). The face validity of the rubric has been evalu-

ated through an expert panel following the procedure proposed by Beecham, Hall, Britton, Cottee, & Rainer (2005).

2. Automation of the CodeMaster rubric. Automated support for assessment with the CodeMaster rubric has been developed following an iterative and incremental process (Larman & Basili, 2003), including requirements analysis, modeling, software construction, and testing.
3. Evaluation of the CodeMaster rubric. The CodeMaster rubric has been evaluated through a case study (Yin, 2017). The objective of the study is to analyze the CodeMaster rubric concerning its reliability and construct validity in the context of computer science education. Data has been collected on a large-scale through the automated assessment of 88,608 App Inventor projects from the App Inventor Gallery (http://ai2.appinventor.mit.edu). The reliability of the rubric has been statistically analyzed using Cronbach's alpha coefficient (Cronbach, 1951), and the correlation between pair of items was evaluated.

## DEFINITION OF THE ASSESSMENT RUBRIC

Following the instructional design model ADDIE (Branch, 2009), in a first step, the context is analyzed. Considering the importance of teaching computer science in K-12 and the progress made in this field (CSTA, 2016), the assessment rubric is defined within the context of K-12 computer science education, on the educational level of middle school focusing on grade 3 through grade 10, for students age 8 to 16 years (Figure 6).

*Figure 6. Educational context of the rubric in accordance to the CSTA (2016) classification of educational levels*


Exploring the connection between computational thinking practices and algorithms and programming concepts in ill-structured learning activities, the rubric is systematically developed following the Goal/Question/Metric approach (Basili, Caldiera, & Rombach, 1994). Using the App Inventor project created by the student, a performance-based assessment rubric is defined to be used for the assessment. The rubric is defined in a top-down manner so that each level is detailed from the most abstract element to the most concrete until all the measurements are explicitly defined. Thus, the rubric is described on three levels:

(1) **conceptual level**: the objective of the rubric is to assess algorithms and programming concepts related to computational thinking practices in the source code of App Inventor projects created as a result of ill-structured learning activity on the middle school level.

(2) **operational level**: the characterization of the assessment is based on the definition of concepts and practices of the K–12 Computer Science Framework (CSTA, 2016) and the results of the review of the state-of-the-art (Alves, Gresse von Wangenheim, & Hauck, 2019). Table 2 presents the analysis questions (column 1) that are derived from the learning objectives of algorithms and programming concepts related to computational thinking practices. The learning objectives related to each question are defined based on existing literature (Brennan & Resnick, 2012; Gresse von Wangenheim et al., 2018; Sherman & Martin, 2015) and the K-12 Computer Science Framework (column 2).

(3) quantitative level: each metric is associated with an item of a rubric for the automatic assessment of the source code of the App Inventor project (Table 3). For each item, performance levels are defined using ordinal scales. Performance levels are defined based on the observable presence of computer science algorithms and programming concepts for each item, ranging from "item is not (or minimally) present" to a progressive usage of the item based on the learning objective defined in Table 2. Depending on the specific performance levels for each of the items, the scales range from dichotomous to 4-point ordinal scales (Table 3).

Considering the items defined in Table 3, a grade is calculated based on the Classical Test Theory, as the sum of the individual item scores (DeVellis, 2003). The final grade value is adjusted to the scoring scale [0, 10].

An expert panel reviewed the rubric regarding its correctness, completeness, and consistency. The expert panel was composed of a multidisciplinary group of 8 researchers with experience in computer science and education. Overall, the experts considered the rubric to be correct and complete. The only issues detected were related to consistency, suggesting minimal rewording.

## AUTOMATING THE ASSESSMENT

The rubric has been implemented using as input the source code of an App Inventor project. It allows to automatically perform an assessment through a static code analysis using the process of tokenization. Based on the table of tokens, it is possible to extract from the source code the performance level of each item of the rubric. Then, for each item, a score is assigned. A final grade consisting of the sum of the scores is calculated, and a "ninja" badge is used to represent the final grade (Figure 7).

The CodeMaster tool presents the assessment results to students and teachers. Instructional feedback is shown to the student for each item assessed, indicating the score received per item (Figure 8). The student can consult the rubric for further details. The feedback can motivate and help the students to improve their performance levels throughout the learning activity.

In addition, the teacher can use the tool to assess a set of App Inventor projects at once for a whole class (Figure 9). The teacher also can access the rubric and see explanations and suggestions on how the students can improve their performance.

The tool has been implemented in Java with JSP (Java Server Pages), JavaScript, HTML5, and CSS3 and is available publicly (http://apps.computacaonaescola.ufsc.br:8080/).

*Table 2. Decomposition of the operational and quantitative levels*

| Analysis Question | Learning objectives | | Measurement |
|---|---|---|---|
| | Description | Source | Metrics |
| PA1. What is the level of performance in algorithms regarding the practices of computational thinking? | LO1: Incorporate operators (arithmetic, relational, boolean) into programs. | Brennan and Resnick, 2012; Gresse von Wangenheim et al., 2018; Grover, Basu, and Schank, 2018 | App Inventor project: Check if arithmetic, boolean, and relational operators are used in the source code. |
| PA2. What is the level of performance in data representation concerning the practices of computational thinking? | LO2: Create programs with variables to store, modify data, and perform operations on their values. | CSTA, 2017: 1B-AP-09 | App Inventor project: Check if variables are created, or its values are modified in the source code.<br>App Inventor project: Check if strings are created, or its values are modified in the source code. |
| | LO3: Create variables with a clear definition without generic or default names. | CSTA, 2017: 2-AP-11 | App Inventor project: Check if variable names are changed from the default in the source code. |
| | LO4: Use lists to simplify solutions by avoiding the use of simple variables repeatedly. | CSTA, 2017: 3A-AP-14 | App Inventor project: Check if lists are used in the source code. |
| | LO5: Incorporate data persistence into programs. | Sherman, Martin, Baldwin, and DeFilippo, 2014; Sherman and Martin, 2015 | App Inventor project: Check if Data Persistence components blocks are used in the source code. |
| PA3. What is the level of control performance concerning computational thinking practices? | LO6: Create and develop programs that include events. | CSTA, 2017: 1B-AP-10 | App Inventor project: Check if events are used in the source code. |
| | LO7: Create and develop programs that include loops. | CSTA, 2017: 1BAP-10; 2-AP-12 | App Inventor project: Check if loops are used in the source code. |
| | LO8: Create and develop programs that include conditionals. | CSTA, 2017: 1B-AP-10; 2-AP-12 | App Inventor project: Check if conditionals are used in the source code. |
| | LO9: Incorporate synchronization into programs. | Brennan and Resnick, 2012; Gresse von Wangenheim et al., 2018 | App Inventor project: Check if sync blocks are used in the source code. |
| PA4. What is the level of modularity performance concerning computational thinking practices? | LO10: Break down problems into subproblems to facilitate program development and review. | CSTA, 2017: 1B-AP-11; 2-AP-13 | App Inventor project: Check if procedures and functions are created in the source code. |
| | LO11: Create procedures to organize code and make it easier to reuse. | CSTA, 2017: 2- AP-14 | App Inventor project: Check if procedures and functions are created in the source code. |
| PA5. What is the level of program development performance concerning the practices of computational thinking? | LO12: Incorporate existing code, media, library into original programs, and give recognition. | CSTA, 2017: 2-AP-16 | App Inventor project: Check if libraries have been added to the source code. |
| PA6. What is the level of performance concerning mobile computational thinking? | LO13: Incorporating components from App Inventor into programs. | Sherman et al., 2014; Sherman and Martin, 2015; Gresse von Wangenheim et al., 2018 | App Inventor project: Check if App Inventor specific components are included and are used in the source code. |

*Table 3. CodeMaster 2.0 rubric for assessing App Inventor projects (Alves, Gresse von Wangenheim, Hauck, & Borgatto, 2020)*

| | Item | Performance Level | | | | Learning Objective |
|---|---|---|---|---|---|---|
| | | **0 points** | **1 point** | **2 points** | **3 points** | |
| **Algorithms & programming** | 1. Operators | No operator blocks are used. | Arithmetic operator blocks are used. | Relational operator blocks are used. | Boolean operator blocks are used. | LO1 |
| | 2. Variables | No use of variables. | Modification or use of predefined variables. | Creation and operation with variables. | - | LO2 |
| | 3. Strings | No use of strings. | Use of creating string block to change design elements texts. | Creation and operation with strings. | - | LO2 |
| | 4. Naming | Few or no names are changed from their defaults. | 10 to 25% of the names are changed from their defaults. | 26 to 75% of the names are changed from their defaults. | More than 75% of the names are changed from their defaults. | LO3 |
| | 5. Lists | No lists are used. | One single-dimensional list is used. | More than one single-dimensional list is used. | Lists of tuples are used. | LO4 |
| | 6. Data persistence | Data are stored only in variables or UI component properties and do not persist when the application is closed. | Data is stored in files. | The local database is used. | A web database is used. | LO5 |
| | 7. Events | No type of event handler is used. | One type of event handler is used. | Two or three types of event handlers are used. | More than three types of event handlers are used. | LO6 |
| | 8. Loops | No use of loops. | Simple loops are used. | ‘For each’ loops with simple variables are used. | ’For each’ loops with list items are used. | LO7 |
| | 9. Conditional | No use of conditionals. | Uses ‘if’ structure. | Uses one ‘if then else’ structure. | Uses more than one ‘if then else’ structure. | LO8 |
| | 10. Synchronization | No use of the timer for synchronization. | Use of timer for synchronization. | - | - | LO9 |
| | 11. Procedural Abstraction | No use of procedures. | One procedure is defined and called. | More than one procedure defined. | There are procedures for code organization and re-use. | LO10, LO11 |

*continues on following page*

*Table 3. Continued*

| | Item | Performance Level | | | | Learning Objective |
|---|---|---|---|---|---|---|
| | | **0 points** | **1 point** | **2 points** | **3 points** | |
| **Mobile algorithms & programming** | 12. Sensors | No use of sensors. | One type of sensor is used. | Two types of sensors are used. | More than two types of sensors are used. | LO13 |
| | 13. Extensions | No use of extensions. | Uses at least blocks from 1 extension. | | | LO12 |
| | 14. Drawing and Animation | No use of drawing and animation components. | Uses a canvas component. | Uses ball component. | Uses image sprite component. | LO13 |
| | 15. Maps | No use of maps. | Use of a map block | Use of map markers blocks. | - | LO13 |
| | 16. Screens | Single screen with visual components, whose state is not changed programmatically. | Single screen with visual components, whose state is changed programmatically. | Three screens with visual components of which at least one is programmed to change state. | Four screens with visual components of which at least two are programmed to change state. | LO13 |

*Figure 7. Assessment process of the source code of App Inventor projects*

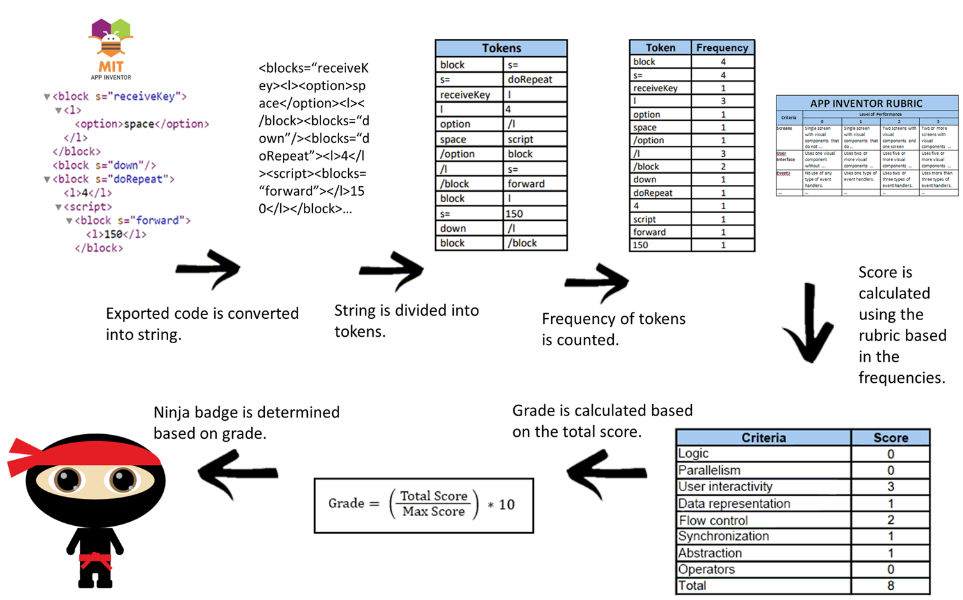


*Figure 8. Screen presenting the assessment of the source code of an App Inventor project to the student*

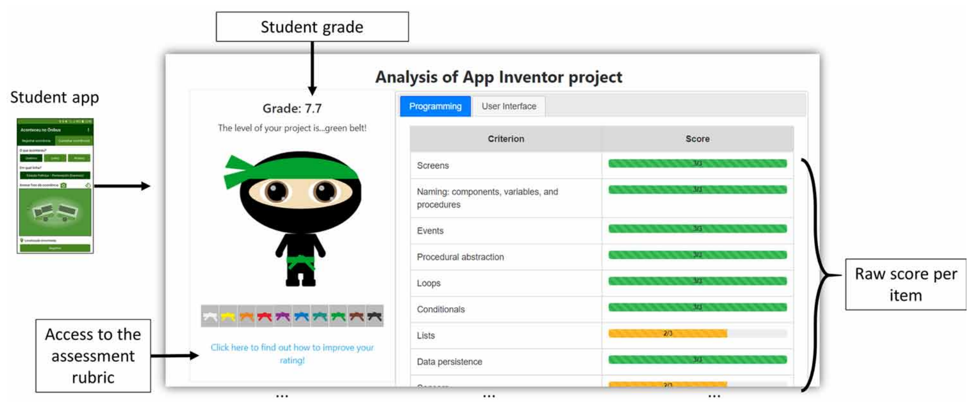
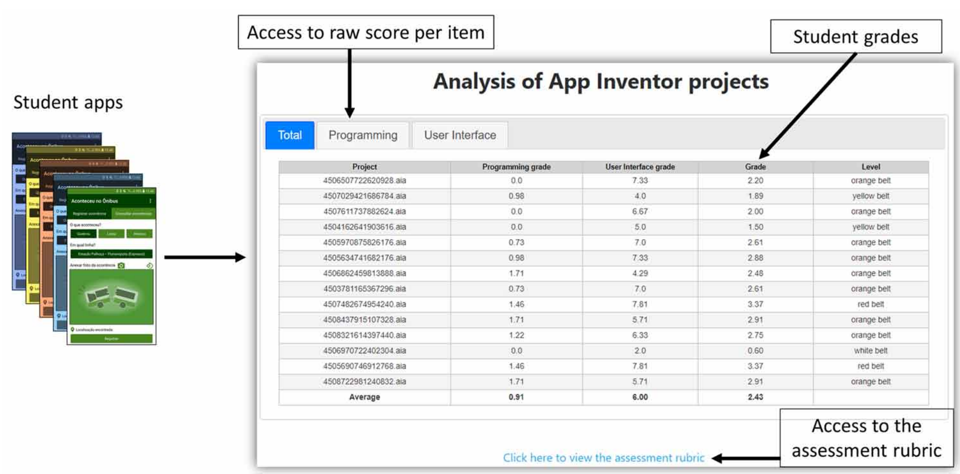



*Figure 9. Screen presenting the assessments of a set of source code of App Inventor projects to the teacher*




## EVALUATION OF THE RUBRIC

The quality of the rubric has been evaluated in terms of reliability and validity in order to obtain evidence of its internal consistency. Cronbach's alpha coefficient (1951) was used to analyze reliability, and polychoric correlations between pairs of items were used to analyze if there is evidence of convergent validity (Olsson, 1979).

### Sample

In order to obtain a large sample size, the source code of 88,812 App Inventor projects (available in 2018 at the App Inventor Gallery) was assessed using the CodeMaster tool. The assessment results were stored in a relational database and then exported to a CSV file. The collected data were grouped in a single table to analyze the rubric. Table 4 presents the scoring frequency per item for all App Inventor projects source code assessed with the CodeMaster tool.

*Table 4. Score frequency per item for all App Inventor projects source code assessed with CodeMaster tool*

| Item | 0 points | 1 point | 2 points | 3 points |
|---|---|---|---|---|
| 1. Operators | 40003 | 7659 | 8678 | 32472 |
| 2. Variables | 19616 | 22109 | 47087 | - |
| 3. Strings | 30241 | 36118 | 22453 | - |
| 4. Naming | 34905 | 9073 | 36656 | 8178 |
| 5. Lists | 72228 | 6154 | 10166 | 264 |
| 6. Data persistence | 79307 | 857 | 7490 | 1158 |
| 7. Events | 7798 | 10171 | 10470 | 60373 |
| 8. Loops | 83546 | 1066 | 1456 | 2744 |
| 9. Conditional | 51938 | 12189 | 15960 | 8725 |
| 10. Synchronization | 68574 | 20238 | - | - |
| 11. Procedural Abstraction | 71501 | 2166 | 2344 | 12801 |
| 12. Sensors | 58481 | 27351 | 2703 | 277 |
| 13. Extensions | 88812 | **0** | - | - |
| 14. Drawing and Animation | 61929 | 7957 | 3130 | 15796 |
| 15. Maps | 88740 | **36** | **36** | - |
| 16. Screens | 6855 | 54211 | 4216 | 23530 |

For dichotomous items with only two performance levels defined, e.g., item 10 and item 13, score frequencies are reported only on the defined performance levels. In the same way, 3-point items, e.g., item 2 and item 3, score frequencies are reported only for the three performance levels defined (Table 4). Some items present low frequencies in some scores, such as item 15 for scores 1 and 2 and item 13 for score 1 (marked in bold in Table 4). The low frequency of scores for item 15 can be explained by the fact that in 2018 this was a new feature added in App Inventor. Thus, for many of the older App Inventor projects present in the sample, this component was not yet available to use at the time the project was created. In addition, item 13 does not have any App Inventor project with a score above 0 because the App Inventor Gallery does not allow publishing App Inventor projects with extensions. Considering the null frequency of item 13, this item was excluded from the analysis. Item 15 (Maps) was analyzed, albeit its low frequency in scores above 0.

## Reliability

The internal consistency of CodeMaster was measured using Cronbach's alpha coefficient (1951). Cronbach's alpha values below 0.7 are considered unacceptable, and any value greater than 0.7 indicates a good internal consistency of the instrument. Analyzing all items of the CodeMaster rubric, excluding item 13 (extension), an acceptable value of Cronbach's alpha ($\alpha = 0.84$) was obtained.

## Validity

To obtain evidence of the convergent validity of the items in CodeMaster, item intercorrelations were calculated (DeVellis, 2003). For convergent validity, the items of the same sub-dimension are expected to have a higher correlation (Trochim & Donnelly, 2008). To analyze the intercorrelations between the items of the same sub-dimension, the polychoric correlation was used, being the most appropriate for observed ordinal variables (Olsson, 1979). The matrix presented in Table 5 shows the correlation coefficient, indicating the degree of correlation between two ordinal items (pairs of items). According to Cohen (1998), a correlation between items is considered satisfactory if the correlation coefficient is greater than 0.29, indicating that there is a medium or high correlation between items. Good correlations are marked in bold in Table 5, without item 13 (Extensions) excluded from the analysis due to the sample not contain a single project with extension.

### Correlations of Algorithms & Programming Items

Items related to algorithms and programming, as defined by CSTA (2016), are items 1 to 11 (Table 5). Item 1 (Operators), presents a high correlation with almost all items of the rubric, this indicated that operators, such as Boolean and relational, are strongly correlated with all the concepts of algorithms and programming. In this regard, the more significant correlations of item 1 are with item 2 (correlation of 0,769) and item 9 (correlation of 0,757). These correlations are indeed expected, as in programming, typically operators are used with variables (item 2) and inside conditions of if-then-else clauses (item 9).

Items related to data representation (items 2, 3, 4, and 5) also have a good correlation with almost all other items, since few values below 0.29 were detected. In this set, the most significant correlation is between item 2 (variables) and item 11 (procedural abstraction). Here again, this is expected, as increasing the level of procedural abstraction, e.g., creating more functions and procedures in App Inventor, typically requires creating more variables alongside to obtain desired results.

Regarding items more related to control concepts, almost all items (6, 7, 8, 9, and 10) demonstrate a good correlation. However, the pair of items 7 and 8 (Events and Loops) presents a value slightly below 0.29. This low correlation can be explained by the fact that App Inventor, being an event-oriented programming environment, favors the use of events over the use of loops (Turbak, Sherman, Martin, Wolber, & Pokress, 2014). Also, one other pair of items 5 and 10 (Lists and Sync) present a value way below 0.29, which may be due to the fact the lists are not needed in order to sync some action.

### Correlations of Mobile Algorithms & Programming Items

Items 12 to 16 are related to mobile algorithms and programming (Table 5). Item 13 (extensions) is excluded from the analysis as the sample does not contain App Inventor projects with extensions. The remaining items present low correlations. Only one pair (items 12 and 14) has a correlation above 0.29. The observation that almost all items do not correlate with values above 0.29, can be explained by the fact that mobile applications typically have specific functionality, and, thus, do not require several features at once. For example, any mobile application does not necessarily need GPS, Bluetooth, Twitter, camera, etc.

Regarding negative correlations, there is a strong negative correlation between item 14 (drawing and animation) and item 15 (maps). In this regard, there is not a conceptual explanation, other than that, as

*Table 5. Correlations per pair of items for all App Inventor projects assessed with CodeMaster*

| | Item | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Algorithms & Programming (A&P)** | 1. Operators | 1,0 | | | | | | | | | | | | | | | |
| | 2. Variables | **0,769** | 1,0 | | | | | | | | | | | | | | |
| | 3. Strings | **0,510** | **0,541** | 1,0 | | | | | | | | | | | | | |
| | 4. Naming | **0,584** | **0,578** | **0,464** | 1,0 | | | | | | | | | | | | |
| | 5. Lists | **0,390** | **0,553** | **0,498** | **0,372** | 1,0 | | | | | | | | | | | |
| | 6. Data persistence | **0,502** | **0,528** | **0,636** | **0,323** | **0,446** | 1,0 | | | | | | | | | | |
| | 7. Events | **0,717** | **0,687** | **0,618** | **0,634** | **0,335** | **0,591** | 1,0 | | | | | | | | | |
| | 8. Loops | **0,455** | **0,550** | **0,488** | **0,395** | **0,696** | **0,427** | 0,257 | 1,0 | | | | | | | | |
| | 9. Conditional | **0,757** | **0,641** | **0,570** | **0,508** | **0,473** | **0,468** | **0,589** | **0,476** | 1,0 | | | | | | | |
| | 10. Synch. | **0,700** | **0,609** | **0,431** | **0,482** | 0,214 | 0,289 | **0,671** | **0,291** | **0,529** | 1,0 | | | | | | |
| | 11. Proc. Abst. | **0,638** | **0,747** | **0,603** | **0,639** | **0,428** | **0,331** | **0,614** | **0,583** | **0,610** | **0,659** | 1,0 | | | | | |
| **Mobile A&P** | 12. Sensors | **0,503** | **0,517** | **0,351** | **0,296** | 0,193 | 0,278 | **0,483** | 0,209 | **0,375** | **0,906** | **0,486** | 1,0 | | | | |
| | 13. Ext. (excluded) | - | - | - | - | - | - | - | - | - | - | - | - | - | - | | |
| | 14. Draw. & Ani. | **0,500** | **0,662** | 0,093 | **0,399** | -0,033 | -0,046 | **0,501** | 0,081 | 0,227 | **0,634** | **0,559** | **0,472** | - | 1,0 | | |
| | 15. Maps | 0,112 | 0,140 | 0,161 | 0,051 | 0,138 | 0,267 | 0,106 | 0,121 | 0,068 | 0,014 | 0,027 | 0,238 | - | -0,334 | 1,0 | |
| | 16. Screens | **0,364** | **0,337** | **0,533** | 0,257 | 0,289 | **0,439** | **0,754** | 0,233 | **0,354** | 0,274 | 0,195 | 0,203 | - | 0,099 | 0,260 | 1,0 |

only 72 projects were assessed with performance level above zero (see Table 4) this could have led to noise in the computation of item 15 correlations. As the sample used in this study contains older App Inventor projects (before 2018) that do not contain map components, the lack of sufficient data for item 15 may have caused a poor representation in the sample.

Due to the lack of large-scale evaluations of other assessment approaches related to App Inventor, no direct comparisons of these results are possible. However, it can be assumed, that these results can be transferred to similar items in other approaches for App Inventor, such as the mobile computational thinking rubric (Sherman et al., 2014; Sherman and Martin, 2015), Quizly (Maiorana, Giordano, & Morelli, 2015), and CodeMaster 1.0 (Gresse von Wangenheim et al., 2018). Yet, comparisons to evaluations related to the assessment of other types of programming languages, such as Scratch, have to consider the different characteristics of the block-based programming environments such as Dr. Scratch (Moreno-Leon & Robles, 2015) and Ninja Code Village (Ota, Morimoto, and Kato, 2016). This is due to the fact that some App Inventor characteristics are not present in other block-based programming environments, e.g., expressing an iterative process as an event that performs a single step of the iteration every time it is triggered instead of using loops (Turbak et al., 2014) and vice versa.

### Threats to Validity

Potential threats of the evaluation were identified, and mitigation strategies were applied to minimize their impact. In order to mitigate threats related to the case study design, a systematic methodology was defined and documented using the GQM approach (Basili, Caldiera, & Rombach, 1994). Another risk is related to the quality of data gathered in a single sample in terms of data standardization. As the study is limited solely to assessments using CodeMaster, this risk is minimized as all analyzes were performed automatically using the same rubric. Another issue concerns the grouping of data from different contexts. App Inventor projects come from a variety of contexts in the App Inventor community, from all over the world, and no additional information about the background of App Inventor projects creators in the App Inventor Gallery is available. However, as the objective is to analyze the validity of the rubric in a context-independent manner, this is not considered a problem here. In terms of external validity, a threat to the possibility of generalizing the results is related to the sample size and the diversity of data used for the evaluation. The analysis is based on data collected from the App Inventor Gallery, involving a sample of 88,812 App Inventor community projects. This is a satisfactory sample size, allowing significant results to be generated.

## FUTURE RESEARCH DIRECTIONS

The development of the CodeMaster rubric and tool represents a first step in the direction of operationalizing the assessment of App Inventor projects. The results of this analysis can be used in order to improve curriculum guidelines and for the design of instructional units. Based on the results of this research, to re-conduct the evaluation of the rubric with an updated dataset is intended in order to represent data in a more balanced way, including the newer components of the App Inventor environment. Using statistical analysis of the assessment data to define the sequencing and difficulty of concepts using, e.g., Item Response Theory (Samejima, 1969) can also allow to analyze the difficulty of items (Alves, Gresse von Wangenheim, Hauck, Borgatto, & Andrade, 2020).

Another research direction is the consideration of concepts and practices beyond algorithms and programming. This includes cross-cutting concepts such as human-computer interaction, including the design of user interfaces, important for the usability and aesthetics of mobile applications. In order to support the assessment of the user interface design of apps, based on its compliance with design theory and guidelines, an automated rubric for the assessment of the visual design of App Inventor projects can complement the assessment of App Inventor projects (Solecki, Porto, Alves, Gresse von Wangenheim, Hauck, & Borgatto, 2020). In addition, as recently machine learning approaches have shown great promise to assess visual aesthetics (Solecki, Porto, Alves, Gresse von Wangenheim, Hauck, & Borgatto, 2020), deep learning approaches to automatically quantify the visual aesthetics of Android mobile user interfaces adopting a regression-based approach can help to provide a holistic assessment.

Furthermore, as computing education is assumed to also impact on other important 21st century skills such as creativity, critical thinking or innovative problem solving (Romero, Lepage & Lille, 2017), it also important to define assessment models that in an adequate way assess these skills (Lucas, 2016). Although there exist several definitions and models proposed to assess skills such as creativity, there is still a lack of reliable and up-to-date assessment methods in practice, especially in context of computing education (Susnea & Vasiliu, 2016) and assessment based on software artifacts created by the students (Bennett, Koh, & Repenning, 2013).

Existing research also points out the need for completing automated performance-based assessment through alternative methods such as interviews, observation as well as self-assessment or peer reviews (Brennan & Resnick, 2012). Therefore, further research should also be directed to the study on how a combination of different methods can enable learners to place new ideas into a broader context and understand their own learning.

## CONCLUSION

This article presents an approach for automating the assessment of algorithms and programming concepts related to computational thinking practices in the source code of App Inventor projects created as a result of ill-structured learning activities. Results of an evaluation of the rubric demonstrate satisfactory reliability (Cronbach's alpha $\alpha$=.84). Algorithms and programming items (criteria) have a high correlation between each pair of items. However, mobile algorithms and programming are sparser, as they typically are not used together, and the correlations between the pair of items are lower. In this respect, the results of our research contribute to a better understanding of the measurement of algorithm & programming concepts based on App Inventor projects. Furthermore, by providing online automation of the CodeMaster rubric, the results are expected to support the teaching of algorithms and programming in practice. The CodeMaster assessment rubric and the tool can be used by students to obtain immediate feedback on their specific project throughout the learning process or by teachers in order to assess and grade all programming projects of a whole class. It can also be used by instructional designers to characterize and create reference/example projects as well as to identify improvement opportunities for instructional units.

## ACKNOWLEDGMENT

We would like to thank all researchers from the MIT App Inventor team, who provided support for the access to the App Inventor Gallery. This research was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) [Finance Code 001]; and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) [302149/2016-3].

## REFERENCES

ACARA. (2015). *Australian Curriculum, Assessment and Reporting Authority*. Retrieved from https://www.acara.edu.au/

Ala-Mutka, K., & Jarvinen, H.-M. (2004). Assessment process for programming assignments. In *Proceedings of IEEE International Conference on Advanced Learning Technologies* (pp. 181-185). Washington, DC: IEEE. 10.1109/ICALT.2004.1357399

Alves, N. da C., Gresse von Wangenheim, C., & Hauck, J. C. R. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, *18*(1), 17–39. doi:10.15388/infedu.2019.02

Alves, N. da C., Gresse von Wangenheim, C., Hauck, J. C. R., & Borgatto, A. F. (2020). A Large-scale Evaluation of a Rubric for the Automatic Assessment of Algorithms and Programming Concepts. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 556–562). Portland, OR: ACM. 10.1145/3328778.3366840

Alves, N. da C., Gresse von Wangenheim, C., Hauck, J. C. R., Borgatto, A. F., & de Andrade, D. F. (2020). An Item Response Theory Analysis of the Sequencing of Algorithms & Programming Concepts. *Proceedings of the International Conference on Computational Thinking Education*.

App Inventor. (2019). *MIT App Inventor*. Retrieved from https://appinventor.mit.edu/

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric approach. In J. J. Marciniak (Ed.), *Encyclopedia of Software Engineering* (pp. 528–532). New York: Wiley.

Beecham, S., Hall, T., Britton, C., Cottee, M., & Rainer, A. (2005). Using an expert panel to validate a requirements process improvement model. *Journal of Systems and Software*, *76*(3), 251–275. doi:10.1016/j.jss.2004.06.004

Bennett, V. E., Koh, K. H., & Repenning, A. (2013). Computing creativity: Divergence in computational thinking. In *Proceedings of the 44th ACM technical symposium on computer science education* (pp. 359-364). Denver, CO: ACM. 10.1145/2445196.2445302

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice. Technical report, European Union Scientific and Technical Research Reports. EUR 28295 EN*. Publications Office of the European Union.

Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York: Springer. doi:10.1007/978-0-387-09506-6

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the Annual Meeting of the American Educational Research Association*.

CAS. (2015). *Computing at School*. Retrieved from https://www.computingatschool.org.uk/

CODE. (2015). *Hour of Code*. Retrieved from https://hourofcode.com

CODECOMBAT. (2016). *Code Combat*. Retrieved from https://codecombat.com

Cohen, J. (1998). *Statistical Power Analysis for the Behavioral Sciences*. New York: Routledge Academic.

Computer Science Teachers Association. (2016). *K-12 Computer Science Framework*. Retrieved from https://k12cs.org/

Computer Science Teachers Association. (2017). *CSTA K-12 Computer Science Standards*. Revised 2017. Retrieved from http://www.csteachers.org/standards

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*(3), 297–334. doi:10.1007/BF02310555

DeVellis, R. F. (2003). *Scale development: theory and applications*. Thousand Oaks: SAGE Publications.

Fitzgerald, S., Hanks, B., Lister, R., McCauley, R., & Murphy, L. (2013). What are we thinking when we grade programs? In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 471-476). Denver, CO: ACM. 10.1145/2445196.2445339

Gresse von Wangenheim, C., Hauck, J. C. R., Demetrio, M. F., Pelle, R., Alves, N. da C., Barbosa, H., & Azevedo, L. F. (2018). CodeMaster – automatic assessment and grading of app inventor and Snap! programs. *Informatics in Education*, *17*(1), 117–150. doi:10.15388/infedu.2018.08

Grover, S., Basu, S., & Schank, P. (2018). What We Can Learn About Student Learning From Open-Ended Programming Projects in Middle School Computer Science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 999-1004). Baltimore, MD. ACM. 10.1145/3159450.3159522

Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 57–62). Uppsala, Sweden. ACM.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Koh, K. H., Basawapatna, A., Nickerson, H., & Repenning, A. (2014). Real Time Assessment of Computational Thinking. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 49-52). Melbourne, Australia. IEEE.

Larman, C., & Basili, V. (2003). Iterative and incremental developments: A brief history. *IEEE Computer*, *36*(6), 47–56. doi:10.1109/MC.2003.1204375

Lucas, B. (2016). A Five-Dimensional Model of Creativity and its Assessment in Schools. *Applied Measurement in Education*, *29*(4), 278–290. doi:10.1080/08957347.2016.1209206

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*(C), 51–61. doi:10.1016/j.chb.2014.09.012

Maiorana, F., Giordano, D., & Morelli, R. (2015). Quizly: A live coding assessment platform for App Inventor. *Proceeding of IEEE Blocks and Beyond Workshop*, 25-30. 10.1109/BLOCKS.2015.7368995

Moreno-León, J., & Robles, G. (2015). Dr. Scratch: A web tool to automatically evaluate Scratch projects. In *Proceedings of the 10th Workshop in Primary and Secondary Computing Education* (pp. 132–133). London, UK: ACM. 10.1145/2818314.2818338

Moskal, B. M., & Leydens, J. A. (2000). Scoring Rubric Development: Validity and Reliability. *Practical Assessment, Research & Evaluation*, *7*(10).

NewZoo. (2020). *Global Mobile Market Report*. Retrieved from https://newzoo.com/products/reports/global-mobile-market-report/

Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, *44*(4), 443–460. doi:10.1007/BF02296207

Ota, G., Morimoto, Y., & Kato, H. (2016). Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 238-239). Cambridge, UK: IEEE. 10.1109/VLHCC.2016.7739695

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning*, *10*(1-2), 44–72. doi:10.1504/IJTEL.2018.088333

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering* (pp. 68-77). BCS Learning & Development.

Reed, D. (2002). The use of ill-defined problems for developing problem-solving and empirical skills in CS1. *Journal of Computing Sciences in Colleges*, *18*(1), 121–133.

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(42), 1–15. doi:10.118641239-017-0080-z

Samejima, F. A. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometric Monograph*, *34*(4, Pt.2), 17. doi:10.1007/BF03372160

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students* (5th ed.). Harlow: Prentice Hall.

Scratch. (2019). Retrieved from https://scratch.mit.edu/

Seiter, L., & Foreman, B. (2013). Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (pp. 59-66). La Jolla, CA: ACM. 10.1145/2493394.2493403

Sherman, M., & Martin, F. (2015). The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, *30*(6), 53–59.

Sherman, M., Martin, F., Baldwin, L., & DeFilippo, J. (2014). *App Inventor Project Rubric – Computational Thinking through Mobile Computing.* Retrieved from https://nsfmobilect.files.wordpress.com/2014/09/mobile-ct-rubric-forapp-inventor-2014-09-01.pdf

Snap, B. Y. O. B. (2019). *Snap! Build Your Own Blocks*. Retrieved from https://snap.berkeley.edu/

Solecki, I., Porto, J., & Alves, N. da C. (2020). Automated Assessment of the Visual Design of Android Apps Developed with App Inventor. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 51–57). Portland, OR: ACM. 10.1145/3328778.3366868

Stegeman, M., Barendsen, E., & Smetsers, S. (2016). Designing a rubric for feedback on code quality in programming courses. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 160-164). Koli, Finland: ACM. 10.1145/2999541.2999555

Susnea, I., & Vasiliu, G. (2016). A Fuzzy Logic Software Tool and a New Scale for the Assessment of Creativity. *International Journal of Computers, Communications & Control*, *11*(3), 441–449. doi:10.15837/ijccc.2016.3.2192

Trochim, W. M., & Donnelly, J. P. (2008). *Research methods knowledge base* (3rd ed.). Mason, OH: Atomic Dog Publishing.

Turbak, F., Mustafaraj, F., Svanberg, M., & Dawson, M. (2017). Identifying and analyzing original projects in an open-ended blocks programming environment. In *Proceedings of the 23rd International Conference on Distributed Multimedia Systems, Visual Languages and Sentient Systems* (pp. 110-118). Pittsburgh, PA: KSI Research Inc. and Knowledge Systems Institute Graduate School. 10.18293/DMS2017-021

Turbak, F., Sherman, M., Martin, F., Wolber, D., & Pokress, S. C. (2014). Events First Programming in App Inventor. *Journal of Computing Sciences in Colleges*, *29*(6), 81–89.

Weintrop, D., & Wilensky, U. (2015). Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 101-110). Omaha, NE: ACM. 10.1145/2787622.2787721

Wing, J. (2006). Computational Thinking. *Communications of the ACM*, *49*(3), 33–36. doi:10.1145/1118178.1118215

Yin, R. K. (2017). *Case Study Research: Design and Methods* (6th ed.). Thousand Oaks: SAGE Publications.

## ADDITIONAL READING

Adeshola, I., & Mohammed Abubakar, A. (2020) Assessment of Higher Order Thinking Skills: Digital Assessment Techniques. In Assessment, Testing, and Measurement Strategies in Global Higher Education (pp. 153-168). IGI Global.

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing Computational Thinking Patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 245-250). Dallas, TX, USA. ACM.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, *58*(1), 240–249. doi:10.1016/j.compedu.2011.08.006

Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., ... Waite, R. (2013). Assessment of Computer Science Learning in a Scratch-Based Outreach Program. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 371-376). Denver, Colorado, USA. 10.1145/2445196.2445304

Funke, A., Geldreich, K., & Hubwieser, P. (2017). Analysis of Scratch Projects of an Introductory Programming Course for Primary School Students. In *Proceeding of IEEE Global Engineering Education Conference* (pp. 1229-1236). Athens, Greece. IEEE. 10.1109/EDUCON.2017.7943005

Grover, S., Bienkowski, M., Niekrasz, J., & Hauswirth, M. (2016). Assessing problem-solving process at scale. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale* (pp. 245-248). Edinburgh, Scotland. ACM. 10.1145/2876034.2893425

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Journal Computer Science Education*, *25*(2), 199–237. doi:10.1080/08993408.2015.1033142

Hadad, R., & Lawless, K. A. (2015). Assessing Computational Thinking. In *Encyclopedia of Information Science and Technology* (3rd ed., pp. 1568–1578). IGI Global. doi:10.4018/978-1-4666-5888-2.ch150

Johnson, D. E. (2016). ITCH: Individual Testing of Computer Homework for Scratch Assignments. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 223-227). Memphis, Tennessee, USA. ACM. 10.1145/2839509.2844600

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Moreno-León, J., Román-González, M., Harteveld, C., & Robles, G. (2017). On the Automatic Assessment of Computational Thinking Skills: A Comparison with Human Experts. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2788-2795). Denver, Colorado, USA. ACM. 10.1145/3027063.3053216

Özçınar, H. (2018). A Brief Discussion on Incentives and Barriers to Computational Thinking Education. In *Teaching Computational Thinking in Primary Education* (pp. 1–17). IGI Global. doi:10.4018/978-1-5225-3200-2.ch001

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. In Early Childhood Development: Concepts, Methodologies, Tools, and Applications (pp. 797-819). IGI Global. doi:10.4018/978-1-5225-7507-8.ch039

Rich, P., & Browning, S. F. (2020). Using Dr. Scratch as a Formative Feedback Tool to Assess Computational Thinking. In *Handbook of Research on Integrating Computer Science and Computational Thinking in K-12 Education* (pp. 220–242). IGI Global. doi:10.4018/978-1-7998-1479-5.ch012

Xie, B., & Abelson, H. (2016). Skill progression in MIT App Inventor. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 213-217). Cambridge, UK. IEEE.

## KEY TERMS AND DEFINITIONS

**Algorithms and Programming:** A subdimension of App Inventor projects assessed by the CodeMaster rubric. This subdimension is assessed based on the K-12 Computer Science Standards (CSTA, 2017), which decomposes algorithms and programming into five subconcepts: algorithms, variables, control, modularity, and program development.

**Automated Assessment:** An assessment that is done using software that assesses several factors and characteristics. Typically, this software implements an assessment model based on a rubric containing the characteristics to be analyzed.

**Ill-Structured Learning Activity:** A learning activity that has several possible solutions, and the judgment on which path to reach a solution is up to the student. Solutions can be considered partially correct or incorrect.

**K-12 Computer Science Education:** Computer Science Education for the educational stage comprising Kindergarten to High School. Includes methods, tools, models, rubrics, etc. developed specifically for the education of students aged 5 to 17 years.

**Mobile Algorithms and Programming:** A subdimension of App Inventor projects assessed by the CodeMaster rubric. This subdimension is assessed based on the rubric proposed by Sherman et al. (2014) and Sherman and Martin (2015), which are related to App Inventor mobile components.

**Rubric:** A matrix that details each assessment item in rows and performance levels in columns using a rating scale. A rubric can facilitate the diagnosis of specific problems within the teaching-learning process.

**Visual Block-Based Programming Environment:** Programming environments that allow programming via graphical rather than textual command blocks and to execute these programs. Examples are App Inventor (2019), Scratch (2019), and Snap (2019).

# Section 2
# Teaching and Learning Computational Thinking

# Chapter 5
# Assessing Algorithmic Thinking Skills in Early Childhood Education:
## Evaluation in Physical and Natural Science Courses

**Kalliopi Kanaki**
*University of Crete, Greece*

**Michail Kalogiannakis**
*University of Crete, Greece*

**Dimitrios Stamovlasis**
https://orcid.org/0000-0003-0808-9065
*Aristotle University of Thessaloniki, Greece*

## ABSTRACT

*This chapter presents part of a wider project aimed at developing computational thinking assessment instruments for first and second grade primary school students. The applicability of the specific proposed tool, which concerns merely the algorithmic thinking (AT), was tested within the Environmental Study course (ESc). The main pillar of the work is the computational environment PhysGramming. The assessment of AT was based on mental tasks involving puzzles which require AT abilities. The AT test comprised of four puzzles with 4, 6, 9, and 12 pieces respectively, and the puzzle-solving performance was measured at the nominal level (success/failure). Latent class analysis (LCA), a robust multivariate method for categorical data, was implemented, which distinguished two clusters/latent classes corresponding to two distinct levels of AT. Moreover, LCA with covariates, such as gender, grade, achievement in ESc, and the use of plan revealed the association of the above variables with the AT skill-levels. Finally, the results and their implications for theory and practice are discussed.*

DOI: 10.4018/978-1-7998-4576-8.ch005

## INTRODUCTION

In the modern digital age, cultivating computational thinking (CT) is considered important at all levels of compulsory education. It is expected that, by the second half of the 21st century, it will be a fundamental skill, like writing, reading and arithmetic (Wing, 2006). Fundamental aspects of CT are skills such as collection, organisation and analysis of data, algorithmic thinking (AT) and abstraction (Barr et al., 2011; Barr & Stephenson, 2011). The work presented in this chapter concerns the assessment of AT skills in early childhood education, within the context of physical and natural science courses. The assessment tool constructed for the needs of the present work was applied in the classroom attuned to the ethical guidelines of educational research by the first author in the city of Heraklion, Crete, from February to June, 2019. The research sample constituted of 435 first and second grade primary school students. To establish the credibility of the research, a computer scientist with long experience in teaching also participated as an external observer. Moreover, the class teachers were present during the research. The work presented in this chapter is part of a wider project on CT that focuses primarily on developing a tool for evaluating the CT skills of first and second grade primary school students. The main pillar for constructing the assessment tool is the digital platform PhysGramming - Physical Science Programming - that was created for the purposes of the above-mentioned project. PhysGramming is a computational environment that follows the basic principles of the theory of constructivism and game based learning. Constructivism is a general educational philosophy that faces learning as an active process and not as a passive one (Mattar, 2018). Game-based learning exploits the engaging and motivating characteristics of games, digital or not, in order to deliver specified learning goals, outcomes and experiences (Akman & Özgül, 2015; de Freitas, 2006). PhysGramming enables young students to design their own digital games, being vigorous creators instead of passive consumers of digital technology (Kanaki & Kalogiannakis, 2018). It proposes a hybrid format of visual and text-based programming techniques, with emphasis on object-orientation. However, no direct reference to programming concepts and practices is made. The assessment tool was implemented with first and second grade primary school students, within an Environmental Study course (ESc) setting. The research focused on three key points: (a) checking the applicability of the new tool, including validity issues, (b) evaluating the levels of students' fundamental CT skills at the first stages of schooling, (c) investigating potential association between the levels of basic CT skills and the understanding of the ESc-content, along with other individual differences, such as gender and grade.

Carrying out the project of constructing a CT assessment tool for early grades is driven by the fact that the issue of evaluating CT continues to be a research challenge, despite the efforts made in the relevant field (Grover, 2015). A review of the international literature shows that the assessment of CT is still in the early stages and does not cover its entire spectrum, nor all the age groups (Bocconi et al., 2016).

## BACKGROUND

In this section, the authors present a literature review on theoretical issues that elucidate relevant concepts on CT and its dimensions, and support the construction of the proposed assessment tool. Specifically, the review elaborates issues on AT and on the integration of studying physical and natural sciences into compulsory education, as the research was carried out within the context of the ESc. Other scientific areas that are fundamental for this research and have been the subject of the literature review are game-based learning, computer programming and student assessment.

## Computational Thinking

Nowadays, CT is considered an essential asset for all productive members of a modern society (Wing, 2006). Its cultivation has already been recognised as an important learning objective in compulsory education (K-12), and has attracted the attention of researchers, educators and policy makers around the world (Barr et al., 2011; Barr & Stephenson, 2011; Grover, 2015). CT has also emerged as a key means of studying a multitude of disciplines (Grover & Pea, 2013). Furthermore, Next Generation Science Standards (NGSS, 2013) have included CT, along with mathematics, in the list of the recommended science course practices in K-12. Contemporary literature suggests introducing CT even in the earliest stages of education, starting from kindergarten (Angeli et al., 2016; Kalogiannakis & Kanaki, 2020; Kanaki & Kalogiannakis, 2018; Kazakoff et al., 2013; Papadakis et al., 2016; Sullivan & Bers, 2016; Sung et al., 2017). The international literature points out that basic aspects or dimensions of CT are collection, organisation and analysis of data, AT and abstraction (Barr et al., 2011; Barr & Stephenson, 2011). The above constitute the basic elements to be considered for CT evaluation, development and cultivation (Barr & Stephenson, 2011; Grover & Pea, 2013; Kanaki & Kalogiannakis, 2019).

Having recognised the cultivation of CT as an important goal of compulsory education, a number of techniques for introducing it to various grades in K-12, have been proposed within the context of several scientific fields (Angeli & Valanides, 2020; Barr et al., 2011; Bell & Lodi, 2019; Dlab et al., 2019; Fessakis et al., 2013; Hsu et al., 2018; Kanaki & Kalogiannakis, 2018; Mannila et al., 2014; Papadakis, 2020; Sung et al., 2017; Voogt et al., 2015). These techniques could be classified into three major categories (Sapounidis et. al., 2015; Wohl et al., 2015): (a) those which are based on computer programming, through developmentally appropriate programming environments, and/or the use of digital games, (b) the unplugged ones, that "do not require the use of digital devices or any kind of specific hardware" (Brackmann et al., 2017) and (c) the techniques based on robotics.

## Algorithmic Thinking

In the literature, there are many definitions for the notion of algorithm. The authors' approach fosters the following definition: "An algorithm is the method of solving a problem and consists of precisely defined steps" (Futschek, 2006), with the addition that these steps have to be finite and realisable in finite time.

AT is often considered as one of the most important skills that can be cultivated by someone trained in computer science (Futschek, 2006; Hromkovič et al., 2016). Prerequisites for understanding and/or building an algorithm are abilities such as: (a) analysing the data of a problem, (b) accurately expressing a problem, (c) finding the key actions needed to solve a problem, (d) using these actions to construct a proper algorithm, (e) identifing not only the simple cases of a problem, but also the more complicated ones, (f) optimising an algorithm.

AT is also closely linked to some other fundamental skills such as abstraction, reasoning, structured thinking, creativity, problem-solving ability (Futschek & Moschitz, 2010), identifying the components of a problem, top-down planning, repetition, organising key data, generalising and customising ability, finding optimal solutions (Cooper et al., 2000).

Beyond its importance AT is often considered by students as a difficult and unattractive process. Thus, the development of effective teaching approaches is by far needed (Futschek & Moschitz, 2010). Particularly in the early stages of cultivating AT, simple methods should be adopted, focusing on: (a) tasks related to everyday problems, (b) problems that do not require prior programming knowledge, (c) methods of algorithm description that provide physical representation, (d) basic activities that trainees are familiar with in their everyday lives, (e) feedback provision (Futschek, 2006; Futschek & Moschitz, 2010). Furthermore, there is a need for systems that: (a) run the algorithms, (b) allow the learners to experiment with algorithms, (c) offer immediate learning experiences and (d) have the flexibility to run various algorithms (Futschek, 2006; Futschek & Moschitz, 2010).

One method of developing AT that is suitable for students in the early stages of compulsory education suggests that students themselves run the algorithms, just as a processor would (Futschek & Moschitz, 2010). More precisely, the learners play the algorithms, turning themselves into intelligent processors. The core idea of this method is that the students understand the algorithms better, when they play the algorithms themselves. For implementing this educational method, the students work in groups, aiming to communicate their ideas and learn from each other. The process, due to its strong playful and teamwork character, is especially appealing, captures the students' attention and ensures their active involvement in relevant instructional activities (Futschek & Moschitz, 2010). A notable feature is that this technique gives students the opportunity not just to play predefined algorithms, but also to invent their own ones. Therefore, when a problem is given, students try to analyse it, become creative by proposing solutions, formulate the algorithm, implement it by playing it, and finally present it to their classmates so that they can draw conclusions about its effectiveness (Futschek & Moschitz, 2010).

Another method, suitable for young children, involves using tangible objects to solve a problem. The algorithm can be illustrated either by using tangible objects provided or through a simulation application in Scratch/BYOB. The learning scenario is called Tim the Train and is appropriate for young children, starting at the age of five. Tim and Train includes a variety of problems that must be solved. These problems have a playful character and their difficulty has several levels (Futschek & Moschitz, 2011). The beginning can be done with simple serial problems. Depending on each child's ability to solve problems, they might deal only with the easy ones, or attempt to solve more difficult problems, that even include a repetition structure. Displaying solutions does not require computer programming knowledge and it is achieved using command-line cards (Futschek & Moschitz, 2011).

For older students, the methods proposed include programming languages such as Python, Logo (Hromkovič et al., 2016) and Alice (Cooper et al., 2000).

## Physical and Natural Sciences

Modern societies are strongly technology and science oriented. Thus, constructing a skilled workforce emerges as a compelling need at the present time. This fact makes scientific literacy a lifelong goal (Clements & Sarama, 2016; Flores, 2015; Watters et al., 2001).

The beginning of developing scientific literacy in the early childhood years is of major importance in order to take advantage of the innate interest and excitement of children when dealing with scientific issues (Eshach & Fried, 2005; Patrick & Mantzicopoulos, 2015; Trundle, 2010). Science fascinates children and their engagement with it begins effortlessly from the first months of their lives. From babyhood, they show a remarkable sensitivity to issues that adults would call scientific (Clements & Sarama, 2016). For example, babies from the age of three or four months understand that objects need support in order to prevent them from falling (Clements & Sarama, 2016). From the very first year of their lives, they realise that inanimate objects cannot move on their own without physical force (Clements & Sarama, 2016). Prior to entering compulsory education, children have already acquired knowledge of scientific fields, understand cause-and-effect relationship, and have developed some reasoning skills (Piasta et al., 2015).

The systematic study of science should be considered an appropriate and important educational goal, given the inherent interest of children in science. Since it is in the nature of children to think scientifically, the systematic learning of scientific topics does not compel them. The scientific questions that children ask - such as “why” questions - show that engaging with science is natural, while their interest in scientific issues comes easily (Bell & Clair, 2015; Clements & Sarama, 2016; Patrick & Mantzicopoulos, 2015).

Systematic engagement with science encourages children to observe, ask questions, hypothesise, design and execute experiments, measure, process and explain data, formulate theories and models (Schleicher et al., 2009). In addition, they gain knowledge about the environment and the environmental problems, cultivate environmental sensitivity and develop skills in solving relevant problems (Ardoin et al., 2018). Within the scope of a gaining broad scientific knowledge, children are educated on issues related to their geological environment and informed about the natural phenomena associated with it (Kalogiannakis et al., 2010).

For example, in Greece - where there are 39 active and inactive volcanoes, with seismicity ranked first in Europe and sixth in the world - the education system must provide students with knowledge regarding volcanoes, earthquakes and natural disasters. Consequently, it makes sense if Greek students are educated about the preparedness and management of these natural phenomena (Juhadi et al., 2017; Kalogiannakis et al., 2010). Within an educational context of this kind, future citizens’ attitudes and perceptions are formed, which leads to demythologizing the natural phenomena that are part of their lives. Hence, they are prepared to deal with the natural phenomena effectively and calmly, both collectively and personally (Kalogiannakis et al., 2010).

In international literature, one can find additional claims that support the idea of engaging young children with science: (a) Children have fun while observing and thinking about nature. (b) Students’ contact with science fosters a positive attitude towards it.

Early contact with investigating physical phenomena leads to a better understanding of the scientific concepts that will be studied later in the context of formal learning. (c) Students develop self-esteem and self-confidence, which helps them succeed in more advanced science courses in the future. (d) The use of a language enriched with scientific terms by young children has a positive impact on the understanding of the respective scientific concepts. (e) Children can understand scientific concepts and make scientific arguments. (f) Dealing with science is an effective means of developing scientific thinking (Clements & Sarama, 2016; DeJarnette, 2012; Eshach & Fried, 2005; Schleicher et al., 2009).

It is notable that Next Generation Science Standards (NGSS, 2013) place particular emphasis on the systematic study of science, thereby triggering renewed interest in upgrading science and technology education. According to these standards, one of the goals of science education is to draw students’ attention to identifying problems, dilemmas and real-life needs, and then seek their solutions.

In order to fulfill the vision of building a skilled workforce, it is important for the educational systems to adopt instructional approaches that provide today’s young students creative opportunities to collaborate with their peers to explore, ask questions, gain knowledge, analyse and evaluate information, generate new ideas, create and test models and, ultimately, draw conclusions. The problem is that, without designing and implementing appropriate educational activities, primary school students may never be exposed to educational environments that will allow them to develop such skills (Flores, 2015).

## Game-Based Learning

Game-based learning is popular in primary education, since it is associated with highly engaging experiences that foster knowledge acquisition and content understanding (Lester, 2014). In fact, playing games is a considerably attractive activity for people of all ages (Mishra et al., 2016). In addition, playing games affects an individual's behavior (Mishra et al., 2016; van der Kooij et al., 2015). Modern playful interactive environments are sources of fun and, at the same time, they provide experiences that enhance learning (Mishra et al., 2016; Sun et al., 2015).

While playing games, children are engaged in interesting and enjoyable activities (Bakhsh, 2016; Butler, 2015; Kinzie & Joseph, 2008; Prensky, 2001). The main objective is to achieve a goal, by following specific rules (Bakhsh, 2016; Prensky, 2001). Nowadays, the idea that games distract students from learning is outdated (Kinzie & Joseph, 2008; Rieber, 1996). Rather, they are seen as tools for cultivating children's social behavior, as well as an important part of their culture and lives (Kinzie & Joseph, 2008; Lancy & Grove, 2017).

As they play, children develop strategic thinking, while in the case of group games, they practice their cooperative skills. In addition, in game-based learning environments, the innate disposition for learning is activated, the research spirit is cultivated, and the students more easily accept the concepts introduced and the knowledge they are offered (Akman & Özgül, 2015; Dickey, 2006; Kinzie & Joseph, 2008; Mohammed et al., 2017; Sung & Hwang, 2013).

Incorporating games into the educational process can contribute to the development of children's exploratory attitude and enhance their interest in learning. In addition, it not only encourages the active involvement of students in the educational process, but also offers interactive learning experiences. Finally, educational approaches that support game-based learning are more attractive to students than traditional techniques and contribute to knowledge acquisition and retention (Chang et al., 2012; Sung & Hwang, 2013).

Nowadays, computers are not large and expensive systems anymore and, thus, they are accessible to the public. The emergence and gradual spread of personal computers has encouraged the creation of digital games, designed to deepen understanding of scientific concepts. Digital educational environments have also been implemented in order to help people acquire scientific knowledge. In addition, the ease of access to the Internet, as well as the rapid development of smart mobile devices, have changed daily lives, thus affecting digital games and educational environments (Huh et al., 2016). Educational digital games are no longer available only at home or at school, but anywhere and anytime, even during a trip (Dillon et al., 2006). The change in the digital gaming culture has led many companies and educational institutes to develop educational games for smart mobile devices as well as digital environments that enable students to create their own digital mobile games (Hirsh-Pasek at al., 2015; Huh et al., 2016; Oberhuber et al., 2017; Papadakis & Kalogiannakis, 2017).

The way teachers think of games as learning tools reflects their educational mindset and philosophy. There are two dominant perceptions regarding the introduction of games into the educational process. According to the first approach, games can be used as educational tools. These games are designed to set educational goals and, at the same time, entertain children (Kafai, 2006; Oblinger, 2010). According to the second approach, the use of ready-made educational games gives its place to the construction of games by the students themselves. In this case, students are transformed from passive users to active creators of digital technology (Kafai, 2006; Prensky, 2008).

## Computer Programming

Although cultivating computer programming skills is not a prerequisite for enhancing CT and its components, it is considered a very effective mean for their development (Voogt et al., 2015). In the modern digital age, computer programming is no longer viewed as a specialised activity, suitable only for developers (Resnick et al., 2009). On the contrary, it has been introduced into compulsory education, even in its early stages (Flannery et al., 2013; Voogt et al., 2015).

Incorporating programming into compulsory education dates back to the 1960s, when the LOGO programming language was introduced as a teaching tool for mathematics (Feurzeig et al., 2011; Pardamean & Suparyanto, 2015). By the early 1980s, however, LOGO had failed to establish itself in the educational arena, perhaps because of the incompatibility between its constructivist character and the behavioral instructional culture of the time (Agalianos et al., 2001).

In recent years, a renewed interest in learning computer programming in compulsory education is observed, with emphasis placed on its potential learning outcomes (Grover & Pea, 2013; Kafai & Burke, 2013; Lye & Koh, 2014). Some of the relevant learning objectives are the ability to think more systematically (Kafai & Burke, 2013) and the development of mathematical and scientific knowledge (Sengupta et al., 2013). Nevertheless, it is worth mentioning that many of the contemporary programming languages and programming environments, such as Scratch and Alice, have been designed in accordance with the principles of LOGO (Utting et al., 2010).

# ASSESSING ALGORITHMIC THINKING

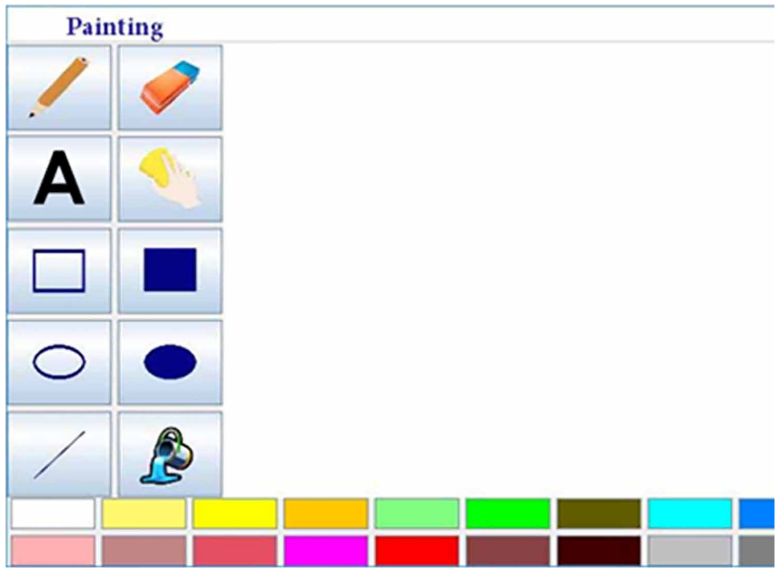
## PhysGramming

PhysGramming is a computational environment compatible with all the operating systems. It was programmed with Java (Arnold et al., 2015), in the Eclipse programming environment (Wiegand, 2004) and it is also designed to run on smart mobile devices, so as to provide the advantages of mobile learning (Zhu et al., 2016).

It provides a writing and an audio mode. Audio mode is proposed for students who have not developed writing skills yet. Through PhysGramming, students exercise their ability to assign values to entity attributes, under the umbrella of a thematic unit. Students can choose the entities they will study. The entities are visualised through images. The teacher can embed the images into PhysGramming, while preparing the lesson. In case of outdoor learning activities, students can work with their own images i.e., entities' photographs taken by students using their smart mobile devices. Alternatively, students can deploy a painting application embedded into PhysGramming, in order to paint their own pictures (Figure 1).

In order to illustrate the functionality of PhysGramming, as well as the methodology for integrating it into the educational process, let's focus on the study of the animals and especially on their eating habits. Before the students use PhysGramming, the teacher must present the content of the lesson to the students. The example of animals is indicative, since PhysGramming is implemented to allow the study of any group of entities, living or non-living. In any case, the teacher chooses which characteristics of the entities will be studied, but the students themselves define the values of the attributes. For example, let's assume that the subject of the lesson is botany and more precisely plants, such as trees, shrubs and

*Figure 1. Painting application*




herbaceous perennials. The teacher asks the students to specify the name of the plants under study, as well as if they lose all of their leaves for part of the year. The students have to determine which plants are deciduous and which are evergreen. These values are entered into PhysGramming by the students.

Returning to the example of the animals and their eating habits, at first, the student/user of PhysGramming must select and/or paint pictures of the animals. For each one of the selected/painted images, command lines appear on the screen (Figure 2). The next step is to determine the value of the attribute "NAME" for each animal depicted in the pictures. The child can add as many lines as they wish and delete some of the lines they have already added. As shown in Figure 2, the authors propose a hybrid schema of visual and text-based programming techniques, emphasizing object-orientation. However, there is no direct reference to programming concepts and practices.

After identifying the names of the animals, the student/user of PhysGramming has to identify their eating habits (Figure 3). The possible values of the attribute "NUTRITION HABIT" are: herbivorous, carnivorous and omnivorous. The child gives these values as an entry into PhysGramming, through a form especially designed for this purpose.

## Puzzles

After the procedure of assigning entities' attributes is completed, PhysGramming automatically creates puzzles, matching games and group games. Now, the students can choose which games they want to play (Figure 4). These games are unique, just like the students' paintings and photographs, as well as the entities each student selects to study.

PhysGramming creates puzzles for each animal image that the student has selected or painted. The user can choose the picture of the puzzle and its pieces: 4, 6, 9 or 12 (Figure 5).

*Figure 2. Assigning values to the attribute "NAME"*

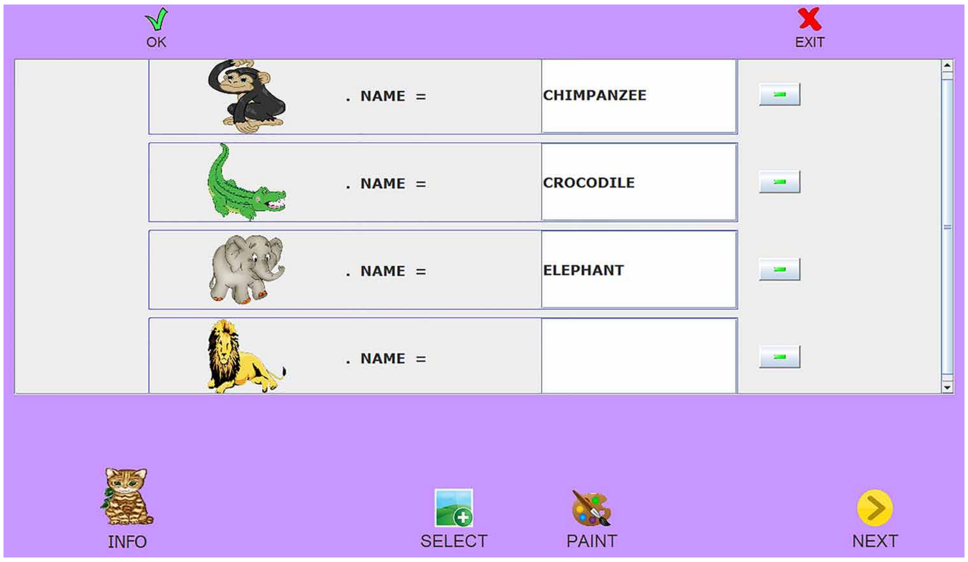
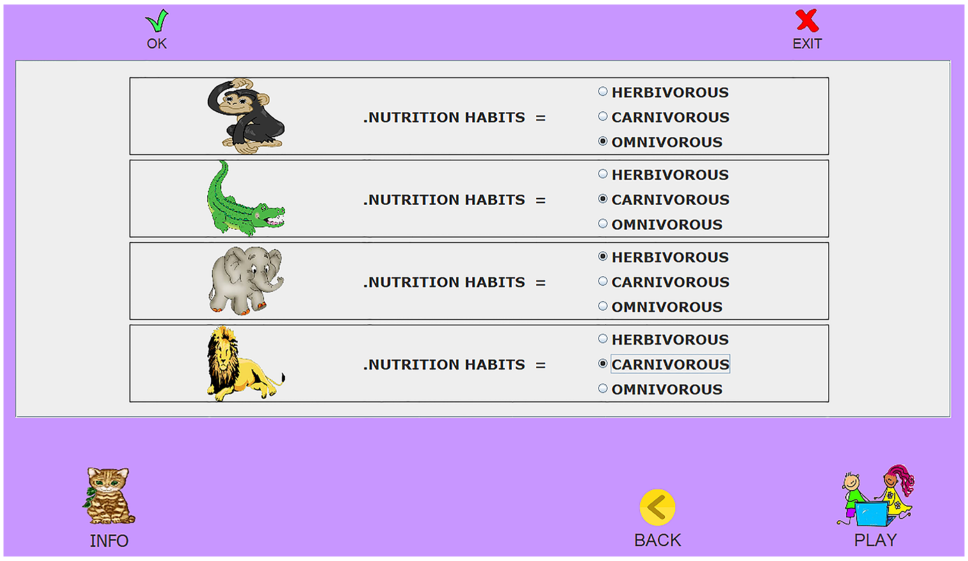


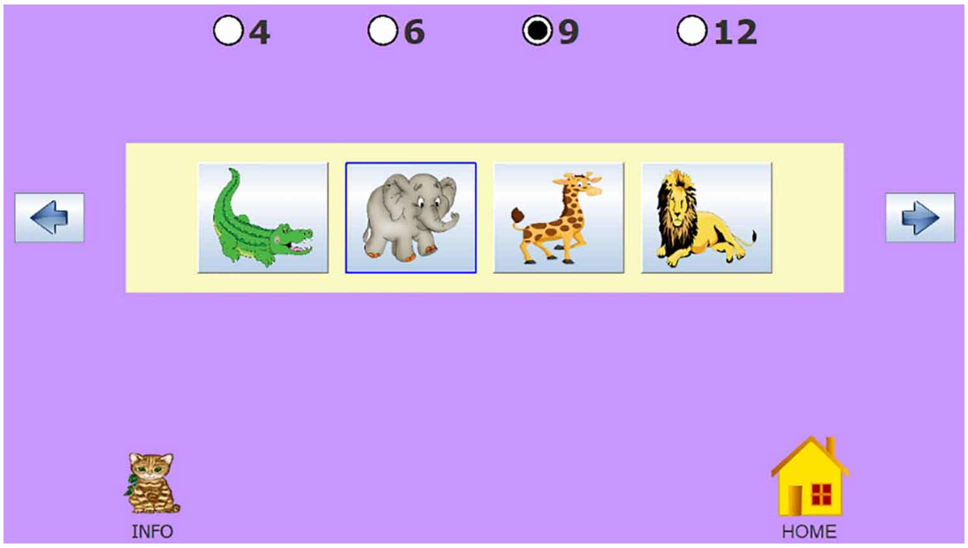

*Figure 3. Assigning values to the attribute "NUTRITION HABITS"*




Each puzzle is a grid, the dimensions of which depend on the number of the puzzle pieces (Figure 6). The grid contains mixed pieces of an animal's image. But, a cell of the grid is empty i.e., a part of the picture is missing. Each piece can be moved horizontally, vertically or diagonally, provided that a neighboring cell is empty. The piece to be moved will occupy the empty cell and the previous cell occupied by the piece moved will be empty. The user must reposition the pieces of the image until the puzzle is solved. The difficulty of solving the puzzles increases as the number of their pieces increase.

*Figure 4. Selecting game to play*


*Figure 5. Selecting the image and the puzzle's pieces*




When students solve a puzzle, a fireworks animation appears on the screen (Figure 7), accompanied by an audio reward. The inclusion of this audiovisual media reward aims to enhance the attractiveness and friendliness of PhysGramming, in view of the fact that these factors contribute to the effectiveness of an educational software in order for students/users to achieve the established learning goals (McManis & Gunnewig, 2012). Immediate reward encourages young children to increase their efforts to attain the learning objectives (Shoukry et al., 2015) and improves their prospects to remain focused on what they are doing (Peijnenborgh et al., 2016; Richter et al., 2015). In addition, immediate reward fosters friendly competition among students (Montola et al., 2009) and empowers their self-confidence (Bleumers et al., 2012) and their acceptance by third parties (Richter et al., 2015).

Figure 6. A nine-piece elephant puzzle

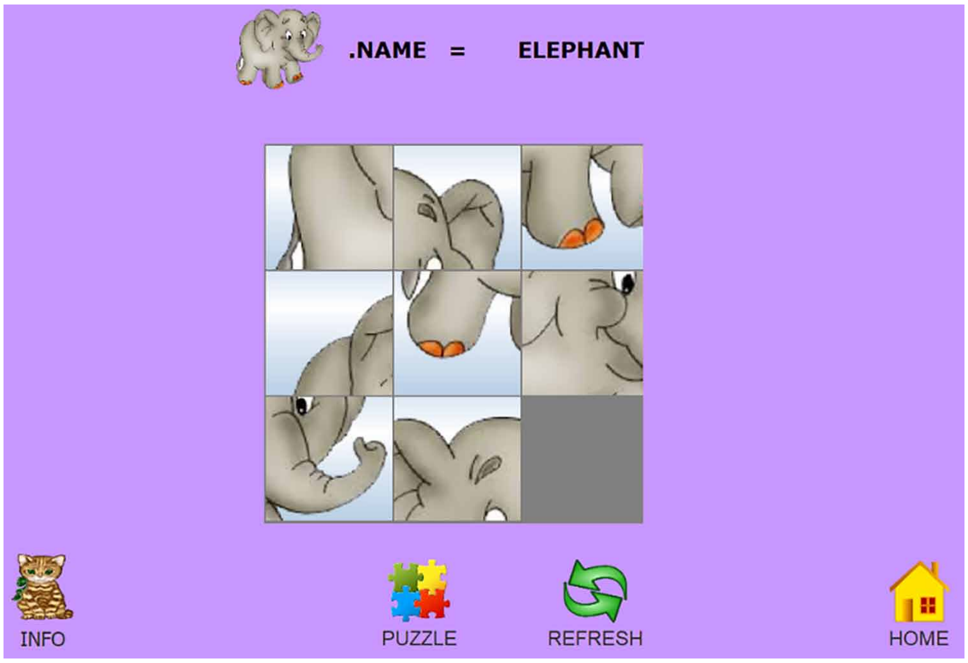
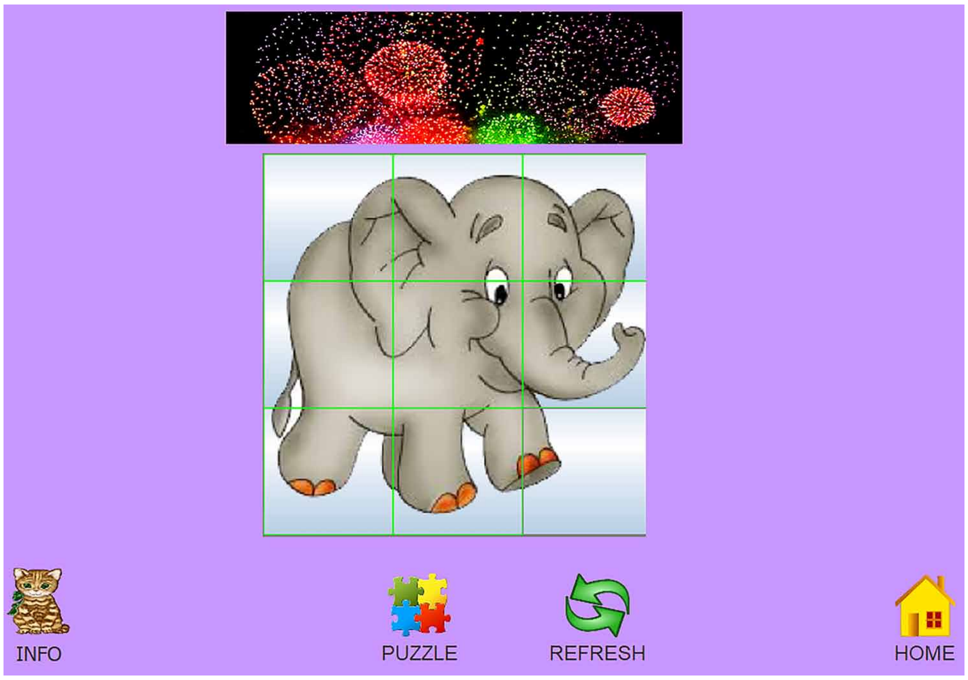



Figure 7. Rewarding students with fireworks animation



## Theoretical Remarks on the Validity Issue

Some mental tasks such as jigsaw puzzles are popular games providing amusement or enjoyment. However, besides the entertaining aspects, solving puzzles is related to skill acquisition, which is useful for everyday life and for scientific and professional activities. The scientists often face the challenge of solving puzzles of various kinds, while working in interdisciplinary fields, such as: biology, archeology, image processing, resolving voice communication security issues, e.g. in military and businesses, through coding and decoding applications (Paikin & Tal, 2015; Pomeranz et al., 2011; Zhao et al., 2007). This fact justifies the efforts which appeared in the international literature focused on finding algorithms for solving jigsaw puzzles (Gallagher, 2012; Paikin & Tal, 2015; Pomeranz et al., 2011; Wolfson et al., 1988; Zhao et al., 2007). On the other hand, algorithmic puzzles, that is, puzzles encompassing clearly defined procedures for reaching the solution, could serve teaching purposes and be a perfect mean to familiarise students to major concepts and methods of algorithmic problem solving. In the educational context, being involved with puzzles allows students' minds to think about algorithms at a more abstract level and assists them to understand the development of potential pathways to the solutions (Levitin & Levitin, 2011).

Therefore, the link between puzzle solving and algorithmic ability is by far established, so that the former can serve as an assessment procedure for the latter. An individual's ability to solve puzzles encompasses the criterion for evaluating his/her AT. The above elaboration about the theoretical association between AT and puzzle solving supports the validity of the developed instrument.

## Description of the Implemented Tools and Related Procedures

At a quantitative level, the researchers test whether children solve puzzles and which kind of puzzles they solve (4, 6, 9, 12-piece puzzles), by examining PhysGramming's log files. Log files provide information on the number of puzzles solved, their kind, and the moves the students have made to solve them. Corresponding information is recorded about students' failed attempts. In the Appendix, a log file is presented (Appendix – Figure 16), where the reader sees in detail all the attempts a student made to solve the puzzles - successful or not. Recording unsuccessful attempts is essential. Indeed, if a student fails to solve a puzzle, recording unsuccessful attempts proves that they have at least tried, but failed. If no effort is recorded, then, for some reason, the student did not attempt to solve the puzzles and therefore they should be excluded from the sample. One reason a student might not get involved in solving puzzles might be that they spent their entire available time painting the entities they wanted to deal with and, therefore, they did not have enough time to play. Indeed, while conducting the research, there was a student who did not have enough time to solve the puzzles they had created. The student was excluded from the sample, as the authors had no information to draw conclusions about their AT.

The information of each attempt is separated from the next one's by dotted lines. In each block of information, the first line informs the reader about the picture of the puzzle the student attempted to solve. Next to the name of the picture, the reader can see the number of the puzzle pieces. For example, in the fourth block it is recorded that the student attempted to solve a 9-piece chicken puzzle. Each time the student manages to solve a puzzle, the phrase "Fireworks !!" is recorded in the corresponding block of information. In the appendix's log file, in the fourth block the reader sees that the student managed to solve the puzzle. On the contrary, in the last block, where the student tried to solve the 12-piece dog puzzle, the phrase “Fireworks !!” is not included. Therefore, it is concluded that the student failed to solve this particular puzzle.

On the last line of each block, the reader sees how many moves the student made in trying to solve the puzzle. For example, in the fourth block 61 moves are recorded. In the appendix’s log file, the reader sees that the student made nine puzzle-solving attempts, seven of which were successful. In particular, they managed to solve two 4-piece puzzles, two 6-piece puzzles, two 9-piece puzzles and one 12-piece puzzle. This particular information is recorded in the student’s spreadsheet, which contains information on the use of the assessment tool (Figure 8). At this point, the reason for recording the number of moves that the student made should be explained. The original image of each puzzle results from the random placement of its pieces. In the case of the 4-piece puzzles, it is not negligible that the pieces are positioned in the right place from the beginning. In such a case, the student does not need to make any moves and, consequently, solving this particular puzzle should not be included in the successful attempts. The more pieces the puzzles have, the more likely they are not randomly correctly positioned. However, the authors considered it was appropriate to cover this case and, thus, they recorded all the moves made in all the kinds of puzzles, regardless of the number of their pieces. It should be noted that during the research the authors identified some cases of 4-piece puzzles that were correctly positioned from the beginning. However, such a case was not identified of puzzles with more pieces.

*Figure 8. Recording puzzle solving attempts data*

| | A | B | C | D |
|---|---|---|---|---|
| 66 | | | | |
| 67 | PUZZLE | | | |
| 68 | PIECES | TRIED | SOLVED | |
| 69 | 4 | YES | 2 | |
| 70 | 6 | YES | 2 | |
| 71 | 9 | YES | 2 | |
| 72 | 12 | YES | 1 | |
| 73 | | | | |

At a qualitative level, the authors conducted personal interviews in which they recorded the students’ work-plan to ensure that the puzzles were not solved by chance.

A spreadsheet was developed as a template for the indexation and the subsequent processing of the research data. An individual spreadsheet based on the template was constructed for each student, where the research data for that student were recorded. To provide the reader with a complete picture of how the research data was processed, the following paragraphs will include, where appropriate, sections of a random student’s spreadsheet that relate to the variable being examined.

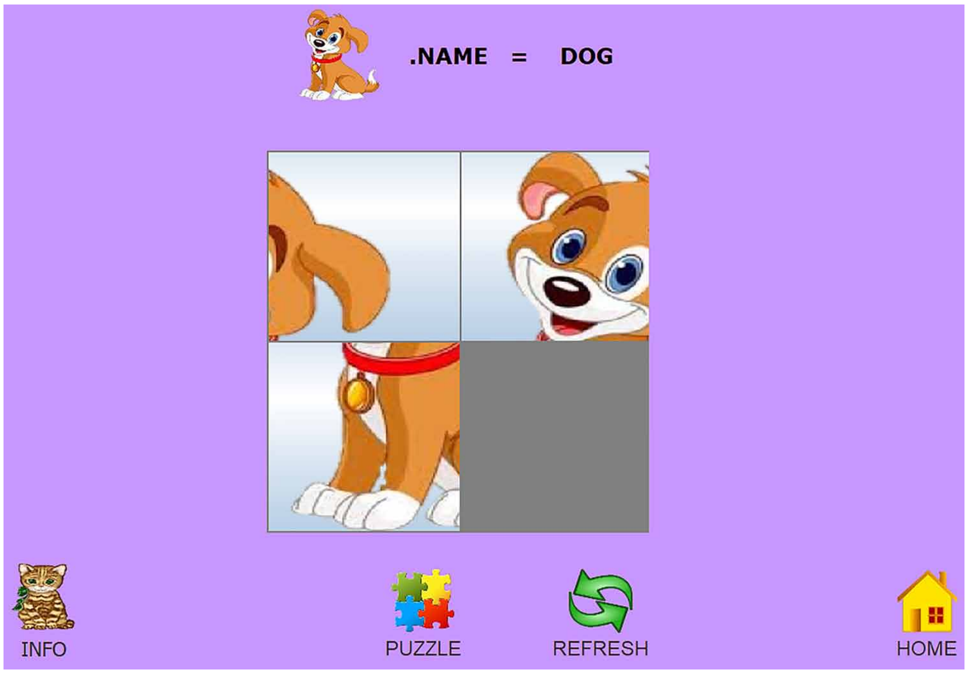
## Algorithmic Thinking Levels

This section presents the results obtained in trying to identify the levels of AT of first and second grade students of primary school. Therefore, the research question attempted to be answered was: "What are the levels of AT of students in the early stages of schooling?"

To answer this question, the students involved in the research were asked to solve the dog puzzles (Figures 9-12). As already mentioned, PhysGramming produces four kinds of puzzles (with 4, 6, 9 and 12 pieces), for each image the student has selected, painted and/or photographed. So, as part of the research process, the students were asked to solve the 4-piece dog puzzle first, then the 6-piece puzzle, and so on.

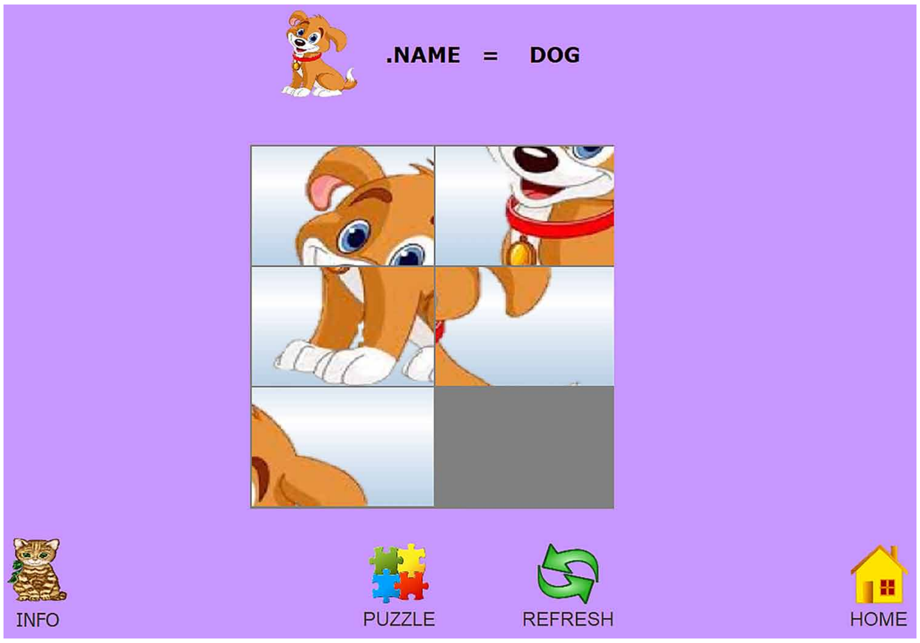
The decision to test all students in the puzzle of a particular animal emerged as a result of trying to eliminate potential extrinsic factors that would alter the impact of research conditions. The authors considered that the picture of the puzzles could be an extrinsic factor, since the puzzles of some images might be more difficult to solve than the puzzles of other images. Therefore, it was decided that the students would be assessed on the levels of their AT by solving puzzles that depicted the same animal. When the dog puzzles were all solved, the students were free to solve puzzles that depicted other animals.

*Figure 9. 4-piece dog puzzle*




In Figures 9, 10, 11 and 12, the reader sees the dog puzzles. PhysGramming is programmed to initially leave the lower right cell of the puzzle's grid empty. As already mentioned, the student can only move pieces that are neighboring an empty cell. Therefore, in the case of 4-piece puzzles (Figure 9), all the pieces can be moved. This fact makes solving these kind of puzzles quite easy.

*Figure 10. 6-piece dog puzzle*




In the case of 6-piece puzzles (Figure 10), if the reader assigns numbers to the rows of the table from top to bottom, they realise that the pieces in the first line cannot be moved unless one of the two pieces in the second line is moved first. The piece to be moved will leave its previous position empty. Thus, the empty cell will no longer be on the third line, but on the second one, and, therefore, it will be neighboring the first line pieces, which can now be moved. It is concluded that the difficulty of solving the 6-piece puzzle is greater than that of the 4-piece puzzle.

In the case of 9-piece puzzles (Figure 11), the difficulty of solving increases, since the pieces of the first line and the first column cannot initially be moved.

In the case of the 12-piece puzzles (Figure 12), which are the most difficult, the pieces in the first and second line, and the first column pieces cannot initially be moved.

Randomly moving the puzzle pieces will not help the students solve them, except perhaps in the case of the 4-piece puzzles. For all other cases, constructing a work-plan is necessary for solving a puzzle. The more pieces the puzzles have, the more complex the work-plan of solving the puzzle becomes. Therefore, trying to solve it becomes more difficult and demands higher levels of AT. The authors, therefore, argue that the more pieces the puzzles have, the higher the levels of AT are needed for them to be solved.

## Quantitative Assessment

A key variable of this research is the maximum number of puzzle pieces that each student managed to solve. The authors consider that the puzzle with the maximum number of pieces that the students have managed to solve is the key to determining the level of their AT (Table 1). The scale the authors propose has five levels. The levels of AT are defined according to the maximum number of puzzle pieces the student has solved. The authors have assigned a code number to each one of the levels.

*Figure 11. 9-piece dog puzzle*

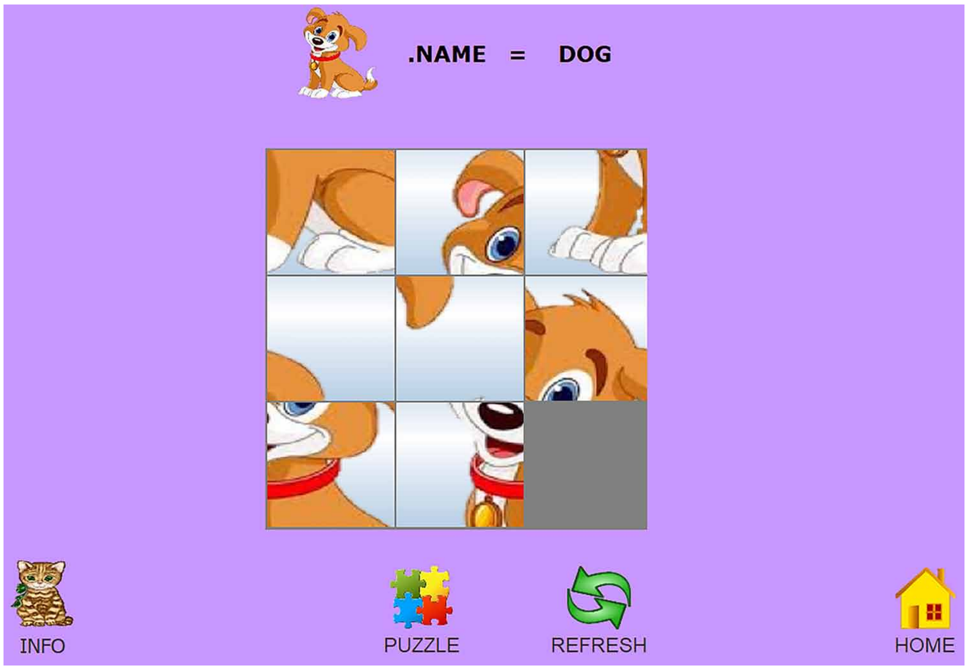



*Figure 12. 12-piece dog puzzle*

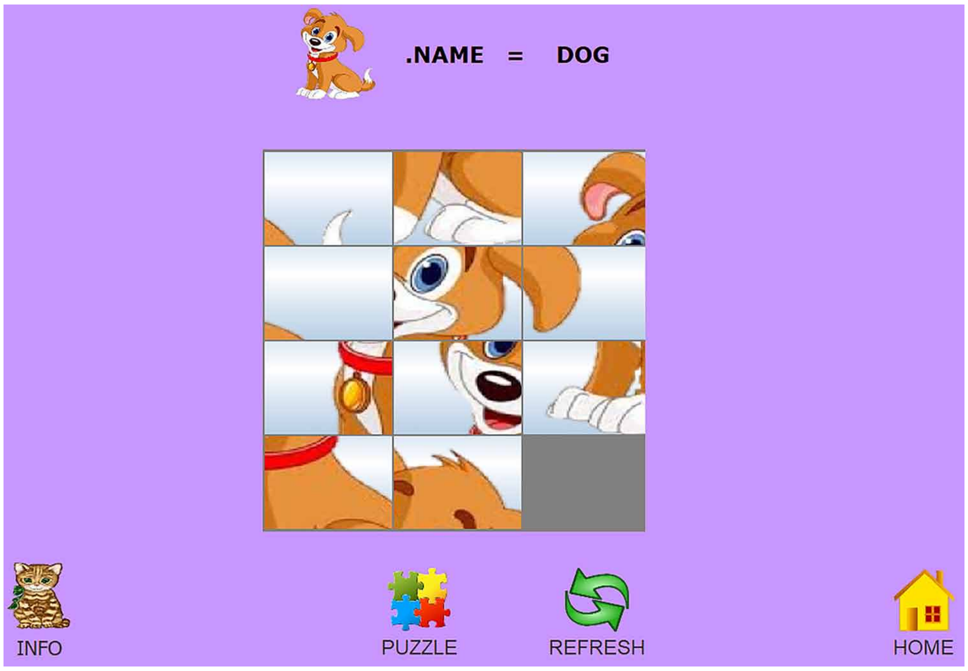


*Table 1. AT levels*

| Code number | AT levels | Maximum number of puzzle pieces |
|---|---|---|
| 0 | Not diagnosed | No puzzles solved |
| 1 | Basic | 4 |
| 2 | Medium | 6 |
| 3 | Sufficient | 9 |
| 4 | Excellent | 12 |

When defining the levels of AT, the authors considered the case of not managing to solve puzzles. In such a case, the relevant log file contains only unsuccessful attempts. The case in which a student has not solved a puzzle is not a simple case. On the contrary, it needs further investigation. The student may not have solved any puzzles because they simply do not like the puzzles and do not want to seriously try to solve them, or because they have not understood the instructions given and they do not know what to do, or, finally, because of low AT. Therefore, the authors cannot arbitrarily conclude that the students have low AT. In this case, the authors say that the student's AT levels are not diagnosed.

Another issue that needs to be addressed is the reason why the authors planned PhysGramming to produce only four kinds of puzzles and not more, e.g. 16-piece puzzles, 20-piece puzzles, etc. If PhysGramming was programmed to produce more kinds of puzzles, a larger scale to assess the levels of AT would be available. However, the authors decided to stick to the four kinds of puzzles because of the time constraints of applying the assessment tool in the classroom.

As already mentioned, each student had one instructional hour to choose which animals to deal with, define their names and their eating habits, and then play with the games that PhysGramming automatically constructed. If PhysGramming produced more kinds of puzzles, the researchers wouldn't know if students did not manage to solve them all because they couldn't solve them or just because they didn't catch up. Furthermore, adding another puzzle, e.g. with 16 pieces, would be an unnecessary exaggeration, since the 12-piece puzzles are already very difficult to solve, and so solving them in the limited time students have, demonstrates their excellent levels of AT.

## Qualitative Assessment

At a qualitative level, the parameter that tested the students' AT levels was their ability to present a work-plan for solving the puzzles. Checking this parameter is of great importance. The students' ability to express the work-plan they devised proves that they did indeed employ their AT and did not solve the puzzles by chance.

Each student was asked, within the context of a personal interview, how they would solve a particular puzzle. More specifically, using a computer, the researchers showed each student a 6-piece dog puzzle and asked: "Where would you start from to solve this dog puzzle?" In most cases, the students answered that they would start from a certain part of the picture, e.g. the head. Some students went a step further and described the sequence of their actions, e.g. "I'll start from the head and I'll continue with the body."

In some cases, the students thought that the researchers were asking them to solve the puzzle. Thus, they attempted to grab the mouse and start to solve it. In those cases, the researchers interrupted them, explaining to them that they didn't want to solve the puzzle, but to tell them in words how they would solve it. Some students immediately understood what the researchers were asking them and immediately replied. Others, however, showed the researchers a piece of the puzzle saying: "This piece must go there, then there ..." That is, they were trying to solve it verbally. In some cases, the researchers had to rephrase the question: "What part of the dog would you start with to solve the puzzle?" There were very few students that still could not understand what the researchers were asking. In those cases, the first author stood up and said to the student; "Look at me. If I were a puzzle, where would you start from to solve the puzzle?" In this way, the students were unhooked from the PhysGramming puzzles and presented a plan. Then, the researchers went back to the dog puzzle, re-asking the original question.

## Testing the Relationship Between Computational Thinking and the Levels of the Content Understanding of the Environmental Study Course

This section presents the results obtained when attempting to identify the levels of the content understanding of the ESc, in relation to the levels of AT of first and second grade primary school students. Thus, the research question that attempted to be answered was: "How do the levels of AT of first and second grade primary school students relate to the levels of the content understanding of the ESc?" The variables used for examining the sample of this study are: (a) the students' understanding of the ESc-content, and (b) the levels of their AT.

According to the original design of the assessment tool, the content understanding would be assessed by studying the relevant worksheet (Appendix – Figure 17. Worksheet 1). The researchers recorded each student's answers in the student's spreadsheets and compared them with the correct ones. The wrong answers were identified and the total number of errors were calculated (Figure 13).

*Figure 13. Recording content understanding data*

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 7 | | | | | | | |
| 8 | CONTENT UNDERSTANDING | | | | | | |
| 9 | ANIMALS | RESPONSE | CORRECT RESPONSE | | MISTAKES | 1 | |
| 10 | CAT | CARNIVOROUS | CARNIVOROUS | | | | |
| 11 | COW | HERBIVOROUS | HERBIVOROUS | | | | |
| 12 | PIG | OMNIVOROUS | OMNIVOROUS | | | | |
| 13 | SHEEP | HERBIVOROUS | HERBIVOROUS | | | | |
| 14 | HORSE | HERBIVOROUS | HERBIVOROUS | | | | |
| 15 | DOG | CARNIVOROUS | CARNIVOROUS | | | | |
| 16 | CHICKEN | OMNIVOROUS | OMNIVOROUS | | | | |
| 17 | RABBIT | HERBIVOROUS | HERBIVOROUS | | | | |
| 18 | LION | CARNIVOROUS | CARNIVOROUS | | | | |
| 19 | PIGEON | OMNIVOROUS | HERBIVOROUS | | | | |
| 20 | | | | | | | |

When applying the assessment tool in the classroom, the researchers paid extra attention not to allow the students to copy from each other. However, the researchers have to admit that there is always the possibility of someone escaping their attention and copying without realising it. The researchers exclude the possibility that someone could copy the values of all the 12 entities of the worksheet. Nevertheless, even if some students were able to copy the attribute value of even one entity, it would damage the accuracy of the results.

*Figure 14. Recording data of identifying mistakes*

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 20 | | | | | | | |
| 21 | ERROR - FINDING | | | | MISTAKES | 1 | |
| 22 | STATEMENT | RESPONSES | CORRECT RESPONSE | | | | |
| 23 | CAT IS CARNIVOROUS | RIGHT | RIGHT | | CONSINSTENT - ONE MISTAKE | | |
| 24 | COW IS CARNIVOROUS | WRONG | WRONG | | | | |
| 25 | PIG IS OMNIVOROUS | RIGHT | RIGHT | | EXCELLENT | | |
| 26 | SHEEP IS HERBIVOROUS | RIGHT | RIGHT | | | | |
| 27 | HORSE IS CARNIVOROUS | WRONG | WRONG | | | | |
| 28 | DOG IS CARNIVOROUS | RIGHT | RIGHT | | | | |
| 29 | CHICKEN IS OMNIVOROUS | RIGHT | RIGHT | | | | |
| 30 | RABBIT IS HERBIVOROUS | RIGHT | RIGHT | | | | |
| 31 | LION IS HERBIVOROUS | WRONG | WRONG | | | | |
| 32 | PIGEON IS HERBIVOROUS | WRONG | RIGHT | | | | |
| 33 | | | | | | | |

In addition, students who did not understand the content were likely to give random answers to one or more questions. These cases would damage the accuracy of the results as well.

Aiming at eliminating factors that would impair the accuracy of the research results, the authors considered that it would be more appropriate to study the “Content understanding” Worksheets (Appendix – Figure 17. Worksheet 1) in combination with the “Finding out mistakes” Worksheets (Appendix - Figure 18. Worksheet 2). In Figure 14, the reader sees the part of a random student’s translated spreadsheet that presents their answers in the second worksheet.

Given that the same entities were used in the construction of the two worksheets, the authors test the consistency of the students in reflecting their beliefs in the two worksheets.

Since the number of possible different values that would result from combining the results of the two worksheets would be considerably large, the authors grouped the data and selected a representative value for each group. The representative values the authors came up with were based on the grading scale of the fifth and sixth grade of primary school, as set out in the Ministerial Decision Φ.7Α/ΦΜ/212191/Δ1. This decision states: “In the fifth and sixth grades of primary school, descriptive assessment is suggested in conjunction with a grading scale that is verbal and numerical, as follows: Excellent (9-10), Very good (7-8), Good (5-6), Approximately good (‹5)”. Although the research relates to first and second grade primary school students, the authors borrow this grading scale, provided that for the first and second students the same Ministerial Decision proposes only descriptive evaluation.

Since each worksheet examines the eating habits of 10 animals, the authors thought that the students had a good understanding of the content if they gave nine or 10 correct answers, they understood the content very well if they gave seven or eight correct answers, etc. However, given the fact that the students’ performance was assessed in completing both worksheets, the authors also took into account the consistency in completing them (Table 2).

*Table 2. Content understanding levels*

| Grade scale | Number of mistakes | Details |
|---|---|---|
| Excellent | 0 - 1 | No mistake.<br>1 mistake in common.<br>1 mistake on only one worksheet. |
| Very good | 2 - 3 | 2 - 3 mistakes in common.<br>2 - 3 mistakes on a spreadsheet and less common mistakes on the other.<br>1 different error on each worksheet. |
| Good | 4 – 5 | 4 - 5 mistakes in common.<br>4 - 5 mistakes on a spreadsheet and less common mistakes on the other.<br>2 - 3 different errors on each worksheet. |
| Approximately good | < 5 | All the other cases |

# EMPIRICAL RESEARCH

## Rationale and Research Questions

Having explained and analysed the functionality of PhysGramming along with the proposed assessment tool for AT, this section presents an empirical investigation, which was carried out in order to demonstrate the applicability of the proposed method. The main goal of this empirical study is to investigate students' algorithmic skill levels at age 6-8 and to associate them with a number of independent variables, such as, grade, gender, students' achievement in the related ESc. In addition, a connection between algorithmic skills and a potential plan followed, if any.

## Methodology

### Sample and Procedure

The participants (N = 435, 48.5% female) were primary school students in first grade (50.1%) and second grade (49.9%). Data were collected in the school's Information Technology laboratory, by the first author and an assistant researcher – observer. Children sat in pairs at the workstations. Before using PhysGramming, the researchers gave the students brief instructions about its functionality. In all stages of this enquiry, a well-defined and robust ethical framework was followed (Cohen et al., 2002).

### Instruments

Accessing students' algorithmic skills was achieved via the instrument designed for the purpose of the present research, the description of which was provided in a previous section.

Other independent variables measured were the plan, the plan details and the ESc-content understanding. The plan is a categorical variable expressing the existence of work-plan for solving the puzzles. The plan details variable provides information about the student's plan and specifically describes the part of the image from which the student started to solve the puzzle. The ESc-content understanding variable measures the ESc achievement using a four-point grade scale (Table 2). In addition, gender and grade were implemented as covariates associated with AT-skills.

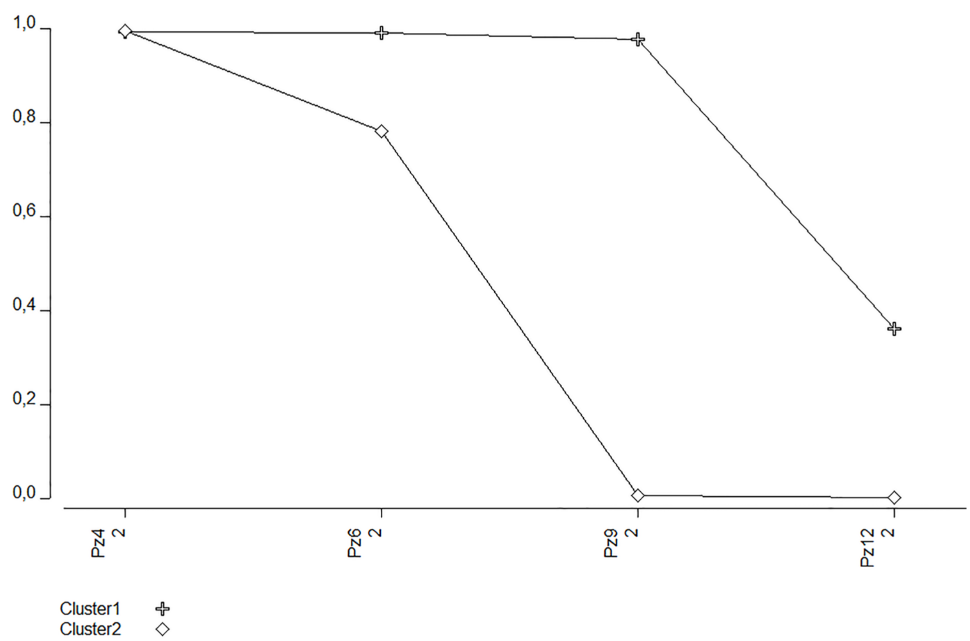
## Statistical Analysis- Latent Class Analysis

Given that all the variables were measured at the nominal level, Latent Class Analysis (LCA), was implemented. LCA is an advanced multivariate method designed for categorical data, which identifies clusters or latent classes, that is, groups of students sharing similar response patterns (Clogg, 1995). LCA is a robust multivariate statistical method applied efficiently to various fields in educational research, e.g. in students' mental model research (Stamovlasis et. al., 2013), in studies on self-regulated learning strategies of talented students (Gonida et. al., 2018), in investigation of students' profiles on graphical and tangible robot programming (Sapounidis et. al., 2018), to mention a few. The classification, which uses Bayesian statistics, is based on the conditional probabilities (CP), that is the probability of providing specific responses (e.g. right /wrong), given that the student belongs to a certain cluster or latent class. The number of clusters is decided with a series of indices, such as the number of parameters, entropy-$R^2$, degrees of freedom and the Bayesian Information Criterion (BIC). In addition, the resulted cluster-memberships could be associated with covariates, external, independent variables. A lucid and detailed illustration of LCA applied to educational research could be found elsewhere (Stamovlasis et. al., 2018).

## Results

LCA was applied with the scores (success/fail) of Pz4, Pz6, Pz9 and Pz12 (4, 6, 9 and 12-piece puzzles, respectively) as input.

*Figure 15. Conditional probabilities of the two clusters/latent classes representing the levels of AT skills*



LCA lead to a two-cluster solution (df = 6, classification-error = 0.0109, BIC = 1250.75, AIC = 1213.04, Npar = 9, entropy $R^2 = 0.92$, $G^2(2) = 14.02$, $\chi^2(2) = 157.31$) as the best parsimonious model with the lower BIC values.

Cluster 1 (51.49%) includes students having high probability of success in all puzzles, Pz4, Pz6 and Pz9, and with a low probability of success in puzzles Pz12. Cluster 2 (48.51%) includes students having high probability of success in puzzles Pz4 and Pz6, which however fail in puzzles Pz9 and Pz12 (Figure 15).

## The Effect of Covariates

The effect of covariates is depicted in Table 3. Gender is related to students' achievement in the algorithmic tasks. Cluster 1, which includes students with the highest AT skills, is positively associated with boys (b = 0.094, $p < 0.05$, one tail) and negatively associated with girls (b = - 0.094, $p < 0.05$, one tail). The statistical association, however, is marginal.

*Table 3. The association of cluster membership with the covariates. Coefficients, standard deviation, z-values, Wald and p-values.*

| | Cluster1 | SD | z-value | Cluster2 | SD | z-value | Wald |
|---|---|---|---|---|---|---|---|
| **Gender** | | | | | | | |
| Boy | 0.094 | 0.052 | 1.79 | -0.094 | 0.052 | -1.79 | 3.20* |
| Girl | -0.094 | 0.052 | -1.79 | 0.094 | 0.052 | 1.79 | |
| **Grade** | | | | | | | |
| First | -0.118 | 0.053 | -22.15 | 0.118 | 0.053 | 22.15 | 4.91** |
| Second | 0.118 | 0.053 | 22.15 | -0.118 | 0.053 | -22.15 | |
| **ESc** | | | | | | | |
| Aprox. Good | -0.248 | 0.095 | -26.07 | 0.248 | 0.095 | 26.07 | 13.69*** |
| Good | -0.139 | 0.093 | -14.93 | 0.139 | 0.093 | 14.93 | |
| Very Good | 0.160 | 0.086 | 1.87 | -0.160 | 0.086 | -1.87 | |
| Excellent | 0.227 | 0.093 | 24.48 | -0.227 | 0.093 | -24.45 | |
| **Plan** | | | | | | | |
| without plan | -0.627 | 0.199 | -31.53 | 0.627 | 0.199 | 31.53 | 9.94*** |
| with plan | 0.627 | 0.199 | 31.53 | -0.627 | 0.199 | -31.53 | |
| * $p < 0.05$ (one tail), ** $p < 0.05$, *** $p < 0.01$ | | | | | | | |

There is also an effect of Grade. Cluster 1, which includes students with higher algorithmic skills, is positively associated (b=0.118, $p<0.05$) with Grade B. That is, most probably the high achievers belong to Grade B.

Students were allocated into two groups: The first included those students which did not follow any plan and the second included those students which followed a plan.

Cluster 1 is positively associated with the second group (b = 0.627, $p < 0.01$), and negatively associated with the first group (b = -0.627, $p < 0.01$). That is, students with the highest algorithmic skills most probably followed some kind of plan. On the contrary, Cluster 2 is negatively associated with the second group ($b = -0.627$, $p < 0.01$) and positively associated with the first group ($b = +0.627$, $p < 0.01$). Students with low algorithmic skills most probably are those who followed no plan.

Finally, the algorithmic skills were associated with students' ESc-content understanding. Cluster 1 is positively associated with the excellent level ($b = 0.227$, $p < 0.01$) and negatively associated with the approximately good level ($b = -0.248$, $p < 0.01$). The opposite holds for Cluster 2. Conclusively, the algorithmic skills in CT are statistically associated with students' performance in the ESc-content understanding test.

## DISCUSSION AND CONCLUSION

Interesting conclusions emerge from the results of the present research concerning the students' levels of AT in the first and second grade and the association of these AT levels with other individual differences. Note that since the implemented method LCA is a psychometric method the two ensuing clusters represent two distinct levels of performance and they are direct measures of the AT derived from four categorical observable variables.

Half of the sample belongs to Cluster 1, (51.49%) which includes students with a high probability of success in all puzzles, Pz4, Pz6 and Pz9, and with a low probability of success in puzzle Pz12. The other half of the sample, Cluster 2, (48.51%) includes students with a high probability of success in puzzle Pz4 and Pz6 but fail in puzzles Pz9 and Pz12 (Figure 15).

The ensued variability indicates that for this age the instrument has satisfactory discriminative ability. The analysis of the covariates showed an effect of gender and grade. That is, boys outperform girls and pupils in second grade outperform pupils in first grade in AT skills. The later effect is interpreted by the anticipated developmental maturity gained by elder students.

LCA performed with the resulted levels of AT as independent variable and students' ESc-content understanding as dependent variable indicated that AT skill level is associated with the course achievement, demonstrating that the CT skills, as might be anticipated, are predictors of academic performance.

An additional aspect that was brought into light by the present analysis is the effect of an existing work-plan for solving puzzles. Students who had a work-plan had higher probabilities of success in all puzzles. This can be interpreted when considering the nature of the task involved in puzzle solving, where the application of a step-by-step controllable process, that is, by definition, an algorithm, is required. Many students might be conscious about what strategy to follow and make it explicit. This denotes a familiarity with these procedures; nevertheless, it seems that it favors success for the majority of students.

Also, an interesting finding is that a few students who solved at least one 9-piece puzzle, did not present a work-plan. Excluding the possibility that they are unreliable or missing-value recordings, these cases suggest that the mental processes might proceed via a holistic rather than analytical mode, where no specific predetermined plan is operating (Kozhevnikov, 2007). These cases deserve special focus and might be investigated in future research.

Recapitulating, the present endeavor is an original work aimed at the development of an assessment tool for CT levels of pupils in the first and second grade of primary school. In this attempt, the novel PhysGramming was proved to be by far useful, since it meets the needs of modern students who, as digital natives, are not passive users of digital technology, but they prefer to be actively involved.

The present work aims to inform educators, researchers and educational policy makers who believe that, nowadays, the systematic introduction of CT in compulsory education is a compelling need. In addition, it adds to the field of CT cultivation, by suggesting solutions on assessment issues and providing valid means that allow the initial evaluation of students' CT levels, which in turn will help the design of targeted teaching interventions for its development. The proposed tool can also be used to facilitate assessing the effectiveness of related teaching interventions.

The impact of the promising results is limited of course by the opportunity sample used and the fact that this is a first attempt to address the questions under study. Future research will attempt to replicate the findings and to further improve and extend the proposed tool. Moreover, it would be interesting to investigate the choice of the type of plan used while solving the puzzles. Note that the majority of the students started from the head, a finding that needs interpretation regarding the cause of such choice, e.g. if it is due to a semiotic role of the 'head' as a starting point or if it is a convenient random choice. It is also interesting to examine the association of computational skills with other disciplines, such as language learning.

Modern societies' need to cultivate students' CT in all levels of compulsory education has drawn the attention of educators, researchers and educational policy makers and triggered relevant inquiries. Responding to this request, with the present chapter the authors make an initiative for a novel contribution and conclusively they re-emphasise that an indispensable prerequisite for the effective cultivation of CT is a coherent theoretical framework providing clear definitions along with the development of assessment tools that are developmentally appropriate for the target groups.

## REFERENCES

Agalianos, A., Noss, R., & Whitty, G. (2001). Logo in mainstream schools: The struggle over the soul of an educational innovation. *British Journal of Sociology of Education*, *22*(4), 479–500. doi:10.1080/01425690120094449

Akman, B., & Özgül, S. G. (2015). Role of play in teaching science in the early childhood years. In *Research in early childhood science education* (pp. 237–258). Dordrecht: Springer. doi:10.1007/978-94-017-9505-0_11

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, 105954. doi:10.1016/j.chb.2019.03.018

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47–57.

Ardoin, N. M., Bowers, A. W., Roth, N. W., & Holthuis, N. (2018). Environmental education and K-12 student outcomes: A review and analysis of research. *The Journal of Environmental Education*, *49*(1), 1–17. doi:10.1080/00958964.2017.1366155

Bakhsh, S. A. (2016). Using Games as a Tool in Teaching Vocabulary to Young Learners. *English Language Teaching*, *9*(7), 120–128. doi:10.5539/elt.v9n7p120

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning and Leading with Technology*, *38*(6), 20–23.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. doi:10.1145/1929887.1929905

Bell, T., & Lodi, M. (2019). *Constructing computational thinking without using computers.* Retrieved March 8, 2020, from https://hal.inria.fr/hal-02378761/document

Bleumers, L., All, A., Mariën, I., Schurmans, D., Van Looy, J., Jacobs, A., ... de Grove, F. (2012). *State of play of digital games for empowerment and inclusion: A review of the literature and empirical cases.* European Commission.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint Research Centre (Seville site).

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 65-72). 10.1145/3137065.3137069

Butler, Y. G. (2015). The use of computer games as foreign language learning tasks for digital natives. *System*, *54*, 91–102. doi:10.1016/j.system.2014.10.010

Chang, K. E., Wu, L. J., Weng, S. E., & Sung, Y. T. (2012). Embedding game-based problem-solving phase into problem-posing system for mathematics learning. *Computers & Education*, *58*(2), 775–786. doi:10.1016/j.compedu.2011.10.002

Clements, D. H., & Sarama, J. (2016). Math, science, and technology in the early grades. *The Future of Children*, *26*(2), 75–94. doi:10.1353/foc.2016.0013

Clogg, C. C. (1995). Latent class models. In G. Arminger, C. C. Clogg, & M. E. Sobel (Eds.), *Handbook of Statistical Modeling for the Social and Behavioral Sciences* (pp. 311–359). New York, NY: Plenum. doi:10.1007/978-1-4899-1292-3_6

Cohen, L., Manion, L., & Morrison, K. (2002). *Research methods in education*. London: Routledge. doi:10.4324/9780203224342

Cooper, S., Dann, W., & Pausch, R. (2000). Developing algorithmic thinking with Alice. In The proceedings of ISECON (Vol. 17, pp. 506-539). Academic Press.

de Freitas, S. (2006). *Learning in immersive worlds: A review of game-based learning*. Retrieved January 5, 2020, from https://researchrepository.murdoch.edu.au/id/eprint/35774/1/gamingreport_v3.pdf

DeJarnette, N. (2012). America's children: Providing early exposure to STEM (science, technology, engineering and math) initiatives. *Education*, *133*(1), 77–84.

Dillon, J., Rickinson, M., Teamey, K., Morris, M., Choi, M. Y., Sanders, D., & Benefield, P. (2006). The value of outdoor learning: Evidence from research in the UK and elsewhere. *The School Science Review*, *87*(320), 107–113.

Dlab, M. H., Hoić-Božić, N., Anđelić, M., & Botički, I. (2019). Digital games and tools for development of computational thinking in primary school. In *Proceedings of the International Conference on Management, Economics & Social Science-ICMESS* (pp. 1-7). Academic Press.

Eshach, H., & Fried, M. N. (2005). Should science be taught in early childhood? *Journal of Science Education and Technology*, *14*(3), 315–336. doi:10.100710956-005-7198-9

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97. doi:10.1016/j.compedu.2012.11.016

Feurzeig, W., Papert, S. A., & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, *19*(5), 487–501. doi:10.1080/10494820903520040

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: Support for early childhood learning through computer programming. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 1-10). 10.1145/2485760.2485785

Flores, I. M. (2015). Developing Preservice Teachers' Self-Efficacy through Field-Based Science Teaching Practice with Elementary Students. *Research in Higher Education*, *§§§*, 27.

Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. In *Lecture Notes in Computer Science 4226* (pp. 159–168). Springer; doi:10.1007/11915355_15

Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. *Proceedings of the 2010 Constructionist Approaches to Creative Learning, Thinking and Education: Lessons for the 21st Century (Constructionism 2010)*, 1-10.

Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 155-164). Springer. 10.1007/978-3-642-24722-4_14

Gallagher, A. C. (2012). Jigsaw puzzles with pieces of unknown orientation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 382-389). IEEE. 10.1109/CVPR.2012.6247699

Gonida, E., Karabenick, S., Stamovlasis, D., Metallidou, P., & Greece, C. T. Y. (2018). Help seeking as a self-regulated learning strategy and achievement goals: The case of academically talented adolescents. *High Ability Studies*, *30*(1-2), 147–166. doi:10.1080/13598139.2018.1535244

Grover, S. (2015). "Systems of Assessments" for deeper learning of computational thinking in K-12. In *Proceedings of the 2015 annual meeting of the American educational research association* (pp. 15-20). Academic Press.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 *A Review of the State of the Field. Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting education in "educational" apps: Lessons from the science of learning. *Psychological Science in the Public Interest*, *16*(1), 3–34. doi:10.1177/1529100615569721 PMID:25985468

Hromkovič, J., Kohn, T., Komm, D., & Serafini, G. (2016). Examples of algorithmic thinking in programming education. *Olympiads in Informatics*, *10*(1-2), 111–124. doi:10.15388/ioi.2016.08

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. doi:10.1016/j.compedu.2018.07.004

Huh, J. H., Kim, H. B., & Seo, K. (2016). A design of smart-based education gamification platform using mobile devices. *Advanced Science and Technology Letters*, *127*, 85–90. doi:10.14257/astl.2016.127.18

Juhadi, J., Nugraha, S. B., & Banowati, E. (2017). Disaster Education Model for Early Childhood. *Sumatra Journal of Disaster*. *Geography and Geography Education*, *1*(2), 89–93.

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture*, *1*(1), 36–40. doi:10.1177/1555412005281767

Kafai, Y. B., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, *95*(1), 61–65. doi:10.1177/003172171309500111

Kalogiannakis, M., & Kanaki, K. (2020). Introducing Computational Thinking Unplugged in Early Childhood Education Within the Context of Physical and Natural Science Courses: A Pilot Study in Greece. In J. Keengwe & P. Wachira (Eds.), *Handbook of Research on Integrating Computer Science and Computational Thinking in K-12 Education* (pp. 164–190). Hershey, PA: IGI Global. doi:10.4018/978-1-7998-1479-5.ch010

Kalogiannakis, M., Rekoumi, C., Antipa, E., & Poullou, V. (2010). Preschool education and geology within the scope of environmental education: the case of a teaching intervention at kindergarten. In *Proceedings of the 3rd World Conference on Science and Technology Education (ICASE 2010), Innovation in Science and Technology Education: Research, Policy, Practice, Tartu-Estonia* (*Vol. 28*, pp. 159-163). Academic Press.

Kanaki, K., & Kalogianakis, M. (2019). Assessing Computational Thinking Skills at First Stages of Schooling. In *Proceedings of the 2019 3rd International Conference on Education and E-Learning* (pp. 135-139). New York: ACM. doi:10.1145/3371647.3371651

Kanaki, K., & Kalogiannakis, M. (2018). Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses. *Education and Information Technologies*, *23*(6), 2673–2698. doi:10.100710639-018-9736-0

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255. doi:10.100710643-012-0554-5

Kinzie, M. B., & Joseph, D. R. (2008). Gender differences in game activity preferences of middle school children: Implications for educational game design. *Educational Technology Research and Development*, *56*(5-6), 643–663. doi:10.100711423-007-9076-z

Kozhevnikov, M. (2007). Cognitive styles in the context of modern psychology: Toward an integrated framework of cognitive style. *Psychological Bulletin*, *133*(3), 464–481. doi:10.1037/0033-2909.133.3.464 PMID:17469987

Lester, J. C., Spires, H. A., Nietfeld, J. L., Minogue, J., Mott, B. W., & Lobene, E. V. (2014). Designing game-based learning environments for elementary science education: A narrative-centered learning perspective. *Information Sciences*, *264*, 4–18. doi:10.1016/j.ins.2013.09.005

Levitin, A., & Levitin, M. (2011). *Algorithmic puzzles*. New York: Oxford University Press; doi:10.1016/j.ins.2013.09.005

Livingstone, S., & Locatelli, E. (2012). Ethical dilemmas in qualitative research with youth on/offline. *International Journal of Learning and Media*, *4*(2), 67–75. doi:10.1162/IJLM_a_00096

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. doi:10.1016/j.chb.2014.09.012

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation and technology in computer science education conference* (pp. 1-29). New York: ACM. doi:10.1145/2713609.2713610

Mattar, J. (2018). Constructivism and connectivism in education technology: Active, situated, authentic, experiential, and anchored learning. *RIED. Revista Iberoamericana de Educación a Distancia*, *21*(2), 201–217. doi:10.5944/ried.21.2.20055

McManis, L. D., & Gunnewig, S. B. (2012). Finding the education in educational technology with early learners. *YC Young Children*, *67*(3), 14.

Mishra, J., Anguera, J. A., & Gazzaley, A. (2016). Video games for neuro-cognitive optimization. *Neuron*, *90*(2), 214–218. doi:10.1016/j.neuron.2016.04.010 PMID:27100194

Montola, M., Nummenmaa, T., Lucero, A., Boberg, M., & Korhonen, H. (2009). Applying game achievement systems to enhance user experience in a photo sharing service. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era* (pp. 94-97). ACM. 10.1145/1621841.1621859

NGSS Lead States. (2013). *Next Generation Science Standards: For states, by states*. Washington, DC: The National Academies Press; doi:10.17226/18290

Oberhuber, S., Kothe, T., Schneegass, S., & Alt, F. (2017). Augmented Games: Exploring Design Opportunities in AR Settings With Children. In *Proceedings of the 2017 Conference on Interaction Design and Children* (pp. 371-377). ACM. 10.1145/3078072.3079734

Oblinger, D. G. (2010). The next generation of educational engagement. *Journal of Interactive Media in Education*, *2004*(1), 10. doi:10.5334/2004-8-oblinger

Paikin, G., & Tal, A. (2015). Solving multiple square jigsaw puzzles with missing pieces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4832-4839). 10.1109/CVPR.2015.7299116

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). IGI Global. doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2017). Mobile educational applications for children: What educators and parents need to know. *International Journal of Mobile Learning and Organisation*, *11*(3), 256–277. doi:10.1504/IJMLO.2017.085338

Papadakis, St., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in Preschool Education. A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187–202. doi:10.1504/IJMLO.2016.077867

Pardamean, B., & Suparyanto, T. (2015). Improving problem-solving skills through Logo programming language. *Stanisław Juszczyk*, 52.

Patrick, H., & Mantzicopoulos, P. (2015). Young children's motivation for learning science. In *Research in early childhood science education* (pp. 7–34). Dordrecht: Springer. doi:10.1007/978-94-017-9505-0_2

Peijnenborgh, J. C., Hurks, P. P., Aldenkamp, A. P., van der Spek, E. D., Rauterberg, M. G., Vles, J. S., & Hendriksen, J. G. (2016). A study on the validity of a computer-based game to assess cognitive processes, reward mechanisms, and time perception in children aged 4-8 years. *JMIR Serious Games, 4*(2), e15. doi:10.2196/games.5997

Pomeranz, D., Shemesh, M., & Ben-Shahar, O. (2011). A fully automated greedy square jigsaw puzzle solver. In CVPR 2011 (pp. 9-16). IEEE. doi:10.1109/CVPR.2011.5995331

Prensky, M. (2001). Fun, play and games: What makes games engaging. *Digital Game-Based Learning, 5*(1), 5-31.

Prensky, M. (2008). Students as designers and creators of educational computer games: Who else? *British Journal of Educational Technology*, *39*(6), 1004–1019. doi:10.1111/j.1467-8535.2008.00823_2.x

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. B. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. doi:10.1145/1592761.1592779

Richter, G., Raban, D. R., & Rafaeli, S. (2015). Studying gamification: the effect of rewards and incentives on motivation. In *Gamification in education and business* (pp. 21–46). Cham: Springer. doi:10.1007/978-3-319-10208-5_2

Sapounidis, T., Demetriadis, S., & Stamelos, I. (2015). Evaluating children performance with graphical and tangible robot programming tools. *Personal and Ubiquitous Computing*, *19*(1), 225–237. doi:10.100700779-014-0774-3

Sapounidis, T., Papadopoulos, P., Demetriadis, S., & Stamovlasis, D. (2019). Tangible and Graphical Programming with Experienced Children: A Mixed Methods Analysis. International. *Journal of Child-Computer Interaction*, *19*, 67–78. doi:10.1016/j.ijcci.2018.12.001

Sapounidis, T., Stamovlasis, D., & Dimitriadis, S. (2018). Latent Class Modeling of Children's Preferences Profiles on Tangible and Graphical Robot Programming. *IEEE Transactions on Education*, *62*(2), 127–133. doi:10.1109/TE.2018.2876363

Schleicher, A., Zimmer, K., Evans, J., & Clements, N. (2009). *PISA 2009 Assessment Framework: Key Competencies in Reading, Mathematics and Science.* OECD Publishing. doi:10.1787/9789264062658-en

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351–380. doi:10.100710639-012-9240-x

Shoukry, L., Sturm, C., & Galal-Edeen, G. H. (2015). Pre-MEGa: a proposed framework for the design and evaluation of preschoolers' mobile educational games. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering* (pp. 385–390). Cham: Springer. doi:10.1007/978-3-319-06773-5_52

Stamovlasis, D., Papageorgiou, G., & Tsitsipis, G. (2013). The coherent versus fragmented knowledge hypotheses for the structure of matter: An investigation with a robust statistical methodology. *Chemistry Education Research and Practice*, *14*(4), 485–495. doi:10.1039/C3RP00042G

Stamovlasis, D., Papageorgiou, G., Tsitsipis, G., Tsikalas, T., & Vaiopoulou, J. (2018). Illustration of Step-Wise Latent Class Modelling with Covariates and Taxometric Analysis in Research Probing Children's Mental Models in Learning Sciences. *Frontiers in Psychology*, *9*, 532. doi:10.3389/fpsyg.2018.00532 PMID:29713300

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20. doi:10.100710798-015-9304-5

Sun, C. T., Ye, S. H., & Wang, Y. J. (2015). Effects of commercial video games on cognitive elaboration of physical concepts. *Computers & Education*, *88*, 169–181. doi:10.1016/j.compedu.2015.05.002

Sung, H. Y., & Hwang, G. J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & Education*, *63*, 43–51. doi:10.1016/j.compedu.2012.11.019

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education, *Technology. Knowledge and Learning*, *22*(3), 443–463. doi:10.100710758-017-9328-x

Trundle, K. C. (2010). Teaching science during the early childhood years. *Best practices and research base*.

Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch--a discussion. *ACM Transactions on Computing Education*, *10*(4), 17. doi:10.1145/1868358.1868364

van der Kooij, K., Hoogendoorn, E., Spijkerman, R., & Visch, V. (2015). Validation of games for behavioral change: Connecting the playful and serious. *International Journal of Serious Games*, *2*(3), 63–75. doi:10.17083/ijsg.v2i3.75

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

Watters, J. J., Diezmann, C. M., Grieshaber, S. J., & Davis, J. M. (2001). Enhancing science education for young children: A contemporary initiative. *Australasian Journal of Early Childhood*, *26*(2), 1–7. doi:10.1177/183693910102600202

Wiegand, J. (2004). Eclipse: A platform for integrating development tools. *IBM Systems Journal*, *43*(2), 371–383. doi:10.1147j.432.0371

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 55-60). ACM. 10.1145/2818314.2818340

Wolfson, H., Schonberg, E., Kalvin, A., & Lamdan, Y. (1988). Solving jigsaw puzzles by computer. *Annals of Operations Research*, *12*(1), 51–64. doi:10.1007/BF02186360

Zhao, Y. X., Su, M. C., Chou, Z. L., & Lee, J. (2007). A puzzle solver and its application in speech descrambling. In *WSEAS International Conference on Computer Engineering and Applications* (pp. 171-176). Academic Press.

Zhu, Z. T., Yu, M. H., & Riezebos, P. (2016). A research framework of smart education. *Smart Learning Environments, 3*(4), 1-17. doi:10.118640561-016-0026-2

## ADDITIONAL READING

Alves, N. D. C., Von Wangenheim, C. G., & Hauck, J. C. (2019). Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. *Informatics in Education*, *18*(1), 17–39. doi:10.15388/infedu.2019.02

Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, *4*(4), 355–369. doi:10.100740692-017-0090-9

Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. In *Emerging research, practice, and policy on computational thinking* (pp. 269–288). Cham: Springer; doi:10.1007/978-3-319-52691-1_17

Kafai, Y. B., Proctor, C., & Lui, D. A. (2019). Framing computational thinking for computational literacies in K-12 education. In *Weizenbaum Conference* (p. 6). DEU.

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Kiss, G., & Arki, Z. (2017). The influence of game-based programming education on the algorithmic thinking. *Procedia: Social and Behavioral Sciences*, *237*, 613–617. doi:10.1016/j.sbspro.2017.02.020

Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education*, *127*, 178–189. doi:10.1016/j.compedu.2018.08.026

Martins-Pacheco, L. H., von Wangenheim, C. A. G., & da Cruz Alves, N. (2019). Assessment of Computational Thinking in K-12 Context: Educational Practices, Limits and Possibilities-A Systematic Mapping Study. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (Vol. 1, pp. 292-303). 10.5220/0007738102920303

Mezak, J., & Papak, P. P. (2018, May). Learning scenarios and encouraging algorithmic thinking. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 0760-0765). IEEE. 10.23919/MIPRO.2018.8400141

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, *11*(1), 1–17. doi:10.1080/20004508.2019.1627844

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papanastasiou, G., Drigas, A., & Skianis, C. (2017). Serious games in preschool and primary education: Benefits and impacts on curriculum course syllabus. [iJET]. *International Journal of Emerging Technologies in Learning*, *12*(01), 44–56. doi:10.3991/ijet.v12i01.6065

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational Thinking Education* (pp. 79–98). Singapore: Springer; doi:10.1007/978-981-13-6528-7_6

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3–31. doi:10.1177/0735633117743918

Yadav, A., Larimore, R., Rich, K., & Schwarz, C. (2019). Integrating computational thinking in elementary classrooms: Introducing a toolkit to support teachers. In *Society for Information Technology & Teacher Education International Conference* (pp. 347-350). Association for the Advancement of Computing in Education (AACE).

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. doi:10.1016/j.compedu.2019.103607

## KEY TERMS AND DEFINITIONS

**Computational Thinking:** A process by which fundamental principles of computer science are employed in order to solve problems, design systems and understand human behavior.

**Early Childhood Education:** Formal and informal education provided to children from birth up to the age of eight.

**Game-Based Learning:** Learning derived from the use of games.

**Mobile Learning:** Education or training facilitated be portable devices such as smartphones and/or tablets.

**Natural Science:** The systematic study of living systems.

**Physical Science:** The systematic study of non-living systems.

## APPENDIX

*Figure 16. Log File*

```
logPuzzle.txt - Notepad
File  Edit  Format  View  Help
dog.png - 4

Fireworks!!
Clicks = 22

----------------------------
chicken.png - 6

Clicks = 40

Clicks = 0

----------------------------
cat.png - 6

Fireworks!!
Clicks = 19

----------------------------
chicken.png - 9

Fireworks!!
Clicks = 61

----------------------------
horse.png - 12

Fireworks!!
Clicks = 165

----------------------------
horse.png - 4

Fireworks!!
Clicks = 6

----------------------------
chicken.png - 6

Fireworks!!
Clicks = 20

----------------------------
cat.png - 9

Fireworks!!
Clicks = 180

----------------------------
dog.png - 12

Clicks = 243
```

*Figure 17. Worksheet 1 – Content understanding*

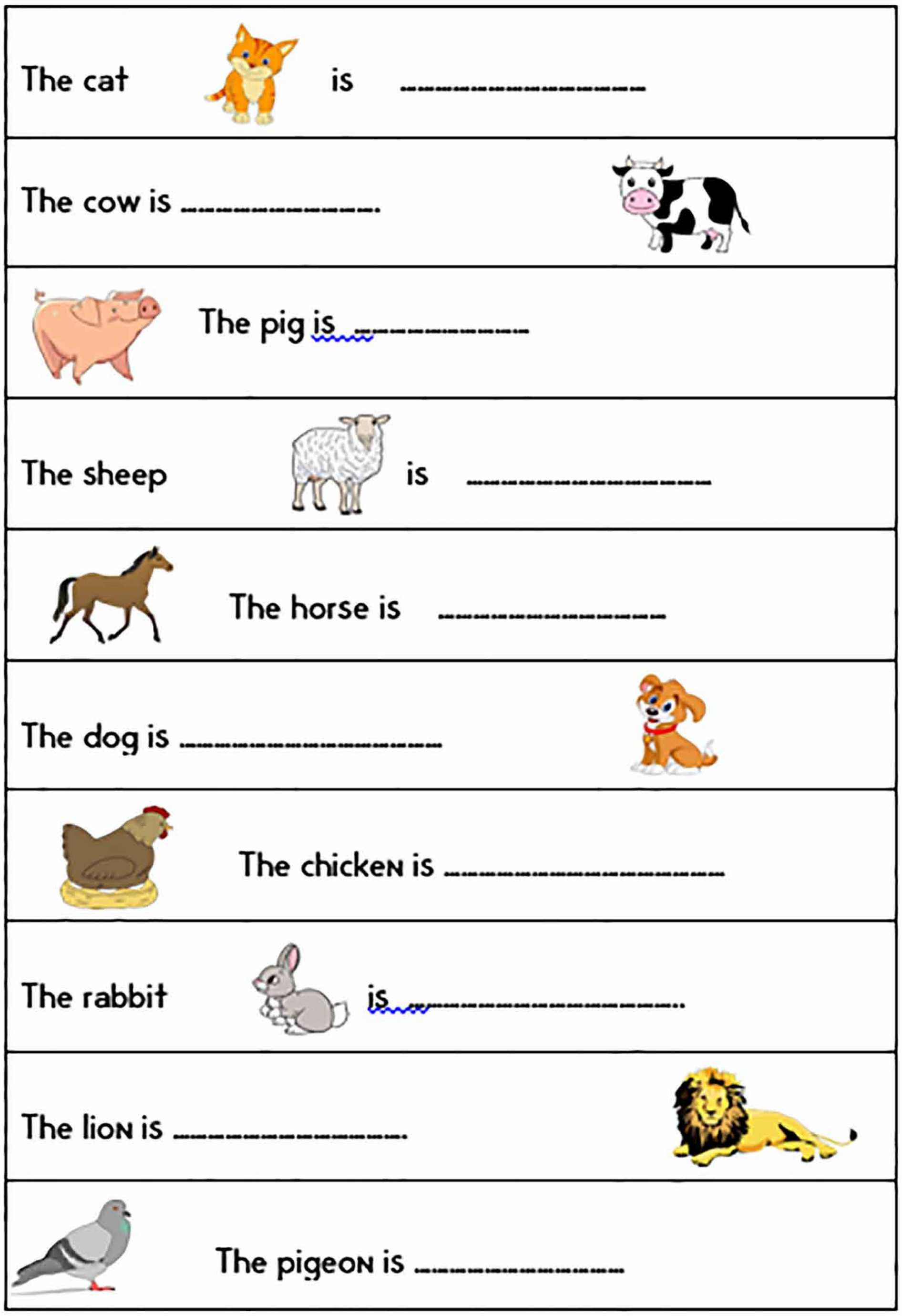
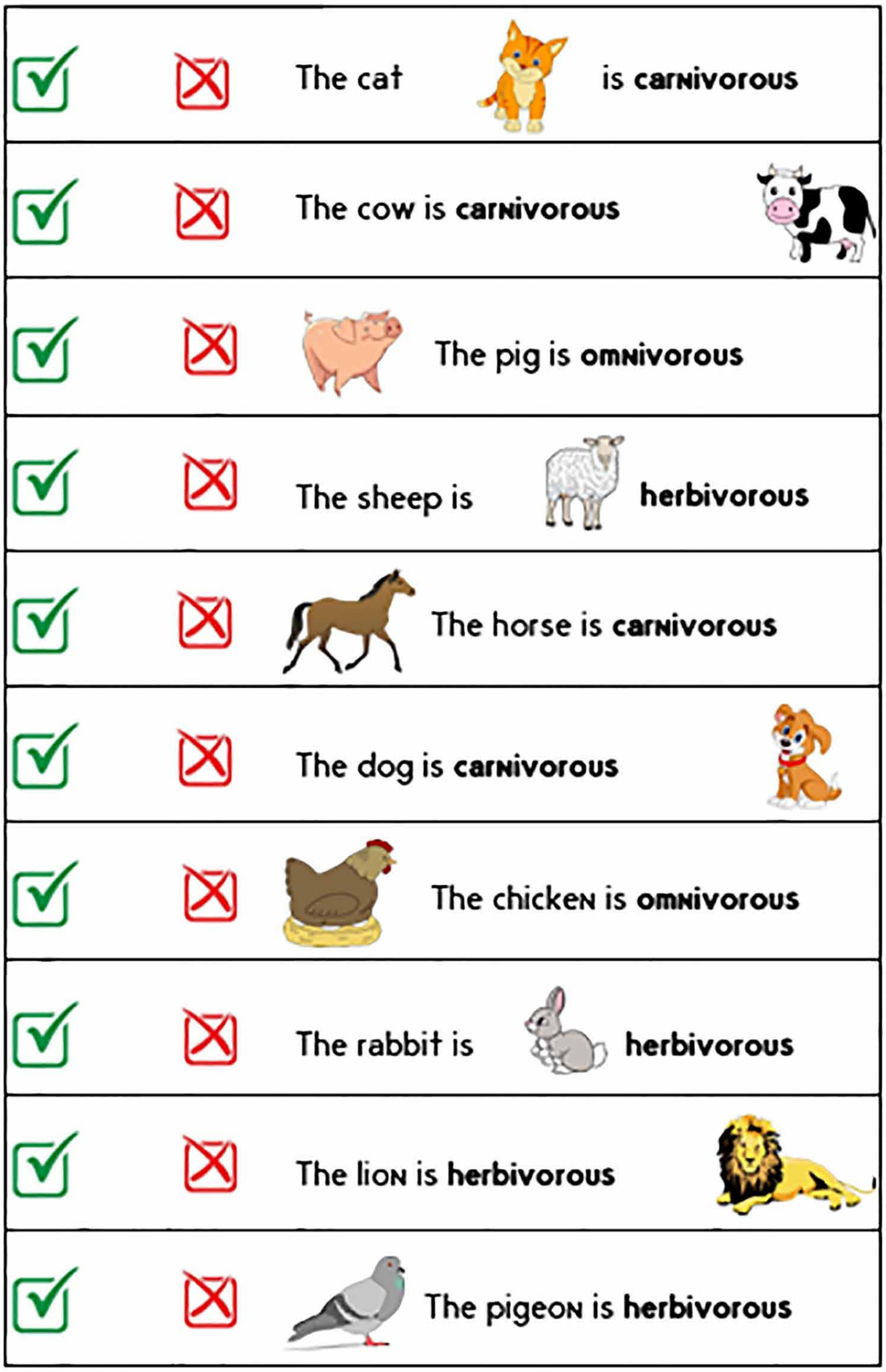


*Figure 18. Worksheet 2 – Finding out mistakes*



# Chapter 6
# Computational Thinking: Activities

**Konstantinos V. Zacharis**
*5th General Lyceum of Karditsa, Greece & University of Thessaly, Greece*

**Antonios D. Niros**
https://orcid.org/0000-0002-9207-4124
*Experimental High School of Mytilene of the University of the Aegean, Greece*

## ABSTRACT

*Computational thinking is a novel problem-solving approach that enhances the interpolation of digital technologies with human ideas. It does not replace the emphasis on creativity, logical, and critical thinking, but rather highlights these skills by proposing ways to organize, modify, and formulate a problem so that it can be resolved by computers. In this work, exemplary computer thinking activities are proposed, which require modeling, problem-solving, planning, and optimization skills. Science-based learning, technology, engineering, art, mathematics, as well as modeling, simulation, programming, and robotics enhance and support computational thought.*

## INTRODUCTION

The human environment today is largely transformed to a digital ecosystem within which hardware and software objects are developed and interacted. The main objective of educational innovation is to enlighten the new generation of students with the characteristics of this environment along with a set of necessary skills and abilities in order to be able to meet the requirements of the demanding future.

The term "Computational Thinking" (CT), despite the fact that there is no generally accepted definition for it (and therefore no way to identify it with sufficient scientific precision at present), is used to declare and manifest all those features of an IT scientist's thinking when trying to face a problem, e.g. splitting the problem into simpler parts, identifying patterns, categorizing and modeling, algorithm design, expanding, linking and reusing code fragments, optimizing end-to-end solutions, and so on. All of these elements lead to a narrative, appropriate to describe the construction of efficient solutions and their subsequent automation by being translated directly into a programming language.

DOI: 10.4018/978-1-7998-4576-8.ch006

The teacher will easily find out that he/she is already using many of the CT skills subconsciously when dealing with problem solving in any scientific discipline, e.g. in mathematics (abstraction, pattern recognition), in chemistry-biology (simulation), in literature (text analysis, categorization, word etymology), etc. This is quite reasonable because these skills comprise the digital equivalent of cognitive skills, which should be part of the mental armor of every professional educator and every citizen as well, in order to be able to effectively copy with modern challenges. For this reason, one can argue that CT skills have a self-imposed intersecting implementation in almost every systematic study.

## BACKGROUND

Introduction of CT into the curricula of primary and secondary education should be a priority for policymakers in every country, for the forthcoming period. The necessity of this action is documented by the fact that the CT as a high-level skill transversely crosses all scientific fields of the curriculum, because it makes the student able to discover patterns, trends or norms within scattered/complex data logs, demonstrates an inductive/deductive reasoning, retrieves and analyzes logically information modules, models and simulates real world problems.

CT is a problem-solving approach that enhances the integration of digital technologies with human ideas. It does not replace the emphasis on creativity, logical or critical analysis. Instead, it highlights these skills through the description of ways and procedures for organizing and solving problems, that can be supported by the monotony of computer operations. In essence, it sets the foundations for the development of Artificial Intelligence.

The multifaceted set of concepts and skills, which are included in the CT methods, can be cultivated and taught in the classroom. It is a direction towards problem solving that is used by computer scientists. Many people confuse it with the specifics of computer operations. It is not actually related to this but merely to the way humans think when trying to solve a problem. Thus, modern students, equipped with computational devices of all kinds and using their logical thinking and imagination, are in a position to face problems that previous generations would not even have imagined.

By grouping the set of basic CT skills (see Figure 1), anyone usually refers to:

1. Analysis - decomposition
2. Generalization - abstraction
3. Pattern recognition
4. Automation - algorithm design
5. Modelling - Simulation
6. Optimization

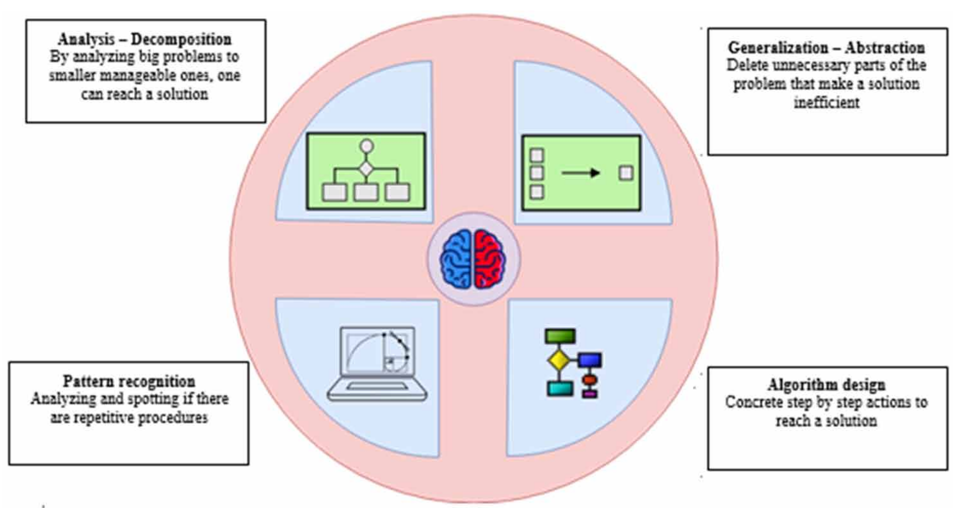
Properties 1st and 2nd are in some sense opposite. The 1st shows the movement from the general idea to the individual modules and is one of the most common techniques of reducing the complexity of a problem, often referred to as "divide and conquer". The 2nd shows the inverse action moving up to multiple hierarchy levels and is also known as "dynamic programming". The 3rd property has its origin in human vision, for instance, a hunter's eye is highly trained to recognize its prey among the tree foliage. It is a property that we use it frequently in our everyday lives and at the same time, a scientific concept with enormous impact across all disciplines. The 4th property is a central subject of Information

Science, which is already being taught extensively in Secondary Education. The 5th property concerns with the construction of a model (that is the blueprint of a system on a smaller scale). Finally, the 6th one is actually an iterative process of approximating better and more efficient (usually in terms of time response) solutions to a problem, something which is required by all software programs today.

## Course Design

In this section, we propose four activities which are representative of the scope and dynamics of the CT in all scientific fields. The reader can access a complete library of curriculum plans embodying the CT approach on the "Google_for_Education" website, among many other sources. He/she can also find ready-made programs written in a modern programming language that accompany these lessons. It is at his/her discretion to evaluate the material and adapt it to the preferable way it should be taught in the classroom. Finally, he/she will find out that all teaching methodology modules can be translated to be CT compatible, as shown in Figure 1.

*Figure 1. CT model*

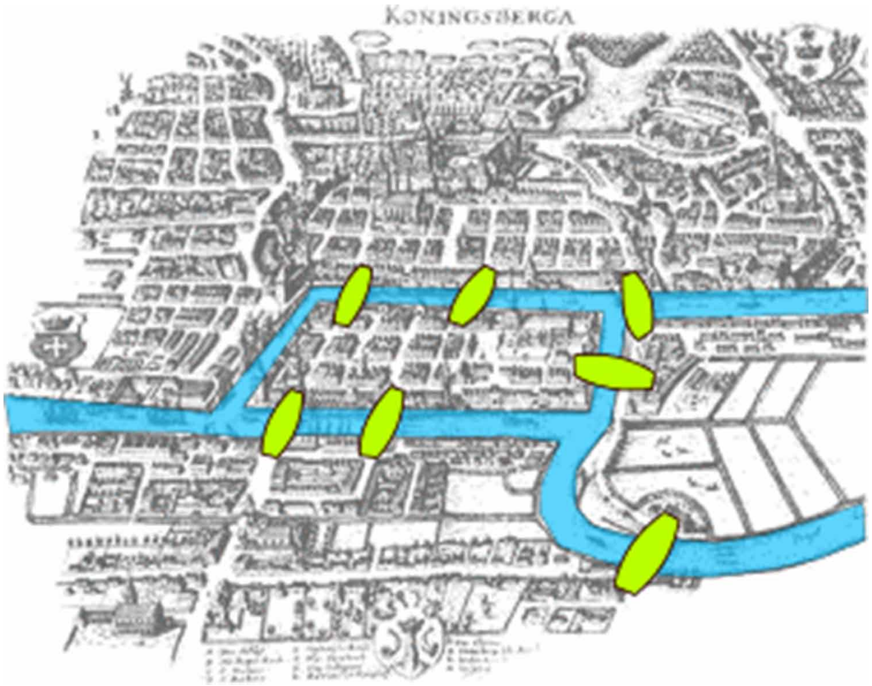



## Sample 1st: A Trip to Königsberg

To efficiently implement this activity, you need to travel to the historical city of Köningsberg, East Prussia (now there is the Russian city of Kaliningrad). The city is built on the banks of river Pregel that includes two big islands, which are connected between them and with the main land through seven bridges (Fig. 2).

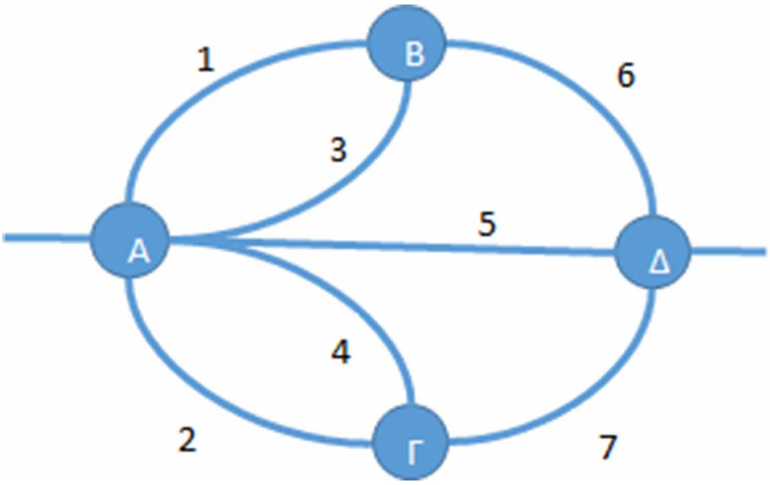
You decide to take a walk in the old city. On every bridge, you stop a take a panoramic picture with your cell phone. But once you enter a bridge, you are not allowed to go back to the opposite direction. Also, it does not matter which is your starting point for this tour. Your target is to propose a route that crosses every bridge once and only.

*Figure 2. City map*




This problem actually refers to defining a "hamiltonian" path in this particular city. It is actually an old problem, one of the most important in graph theory, that was solved by the famous mathematician Euler in 1735 (by proving that no such path exists in Köningsberg) that set the foundations for an entire scientific branch, theory of Informatics. Another version of this problem is the famous TSP (Travelling Salesman Problem).

*Figure 3. The city of Köningsberg with its 7 bridges*



Try to verify Euler's proposition that no single route exists that includes all bridges and crosses them just once. Using the city map, create an equivalent graph (Fig. 3) where every node represents a piece of land and every edge one bridge. Write down all possible routes that traverse this particular graph by crossing as many bridges as you can. How many different routes can you devise? Can you find one that cross all bridges only once?

## Sample 2nd: Secret Code

Suppose you want to deliver a message and the only way you can do that is to make a post in some social medium or electronic wall, which many other people can watch. What could you and your friend do in order to be sure that, although many people see your message, you are actually the two that can understand it? Could you better encrypt it? An encrypted message is one that is encoded in a way that is very difficult for a third person to understand. In this sample you are asked to create a secret coding and then to make encrypted messages, which will be decrypted only inside your team.

Let's start by a simple encoding activity based on patterns. Every letter of the alphabet corresponds to a number. In order to "crack" such arithmetic codes, you should guess the position of the letters in the alphabet. Let's say the word "DREAM" is encoded as "FTGCO", can you find out how? Can you code e.g. the word "STUDENT", based on the scheme of Fig. 4? How will you do that?

Try to discover what is the pattern behind the initial word ("DREAM") and the encoded one. In which position is the letter D? In 4th position obviously. And the encoded letter F is in 6th. That is a shift left by 2 position. Check if this rule applied to the other letters of the message too! Now you are ready to encrypt and decrypt any simple message within your team!

*Figure 4. Encryption code*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

Divide the class into teams. Ask from each team first to devise a secret code (based on simple rule) and then to write and encoded message to a piece of paper, something like "We will meet in the city center", encrypted by each different rule, e.g. using the rule of "shift two" above gives something like "Yg yknn oggv kp vjg ekva efpvgt". Ask from the other teams to guess what the message is. Discuss on the possible decryption methods.

## Sample 3rd: A Hacker is Born

You will devise an encryption code according to the logic of the previous activity and try to apply it to the encoding and decoding of your own messages. Three groups of four students are needed to perform this activity. First, the people in the first group will have to discuss and agree on the rule. People in the second and third groups should not listen to this discussion. The rule at the beginning should be simple, e.g. reverse the alphabet list and place it below the original. This will create the following one-to-one match (see fig. 5):

*Figure 5. Code for message encoding*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Z | Y | X | W | V | U | T | S | R | Q | P | O | N | M | L | K | J | I | H | G | F | E | D | C | B | A |

Now you will try to encrypt a simple message, e.g. the phrase "WE WILL MEET IN THE SQUARE". If you use the "-" character to separate letters and "_" for the space between words, then your message will look like:

[4-22_3-18-15-15_14-22-22-7_18-13_7-19-22_8-10-6-26-9-22]

Write the encrypted message on a piece of paper and ask the second group to decrypt the message and the third group (who does not know the rule) to guess its contents. Repeat the same procedure with a second simple phrase e.g. "WE WILL COME TO STUDY INFORMATICS". Check their answers. Discuss the third group's attempt to decipher the two messages.

You can repeat the same process by devising a more complex rule, e.g. "The alphabet is divided into two groups, the vowels and the consonants. The vowels and the consonants are arranged in reverse order creating the following 1-1 correspondence (see fig. 6). "

*Figure 6. Code for message encoding*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Y | U | O | I | E | A | Z | X | W | V | T | S | R | Q | P | N | M | L | K | J | H | G | F | D | C | B |

Compare the encrypted messages again. What do you observe? Are there any numbers that are repeated more often than others? Discuss in class what those numbers might be.

Imagine now that another group is posting an encrypted message, without knowing the rule it used. How could you decrypt it? What patterns would you look for when considering the frequency and location of integers?

## Sample 4th: Creating Games

Prepare the students to tackle a large-scale programming project, such as making a computer game, using decomposition. Even for a relatively simple game, the project should typically be decomposed as follows: planning, design, algorithms, coding, animation, graphics, sound, debugging and sharing. A project like this would lend itself to a collaborative, team-based approach, with development planned over a number of weeks.

The software "Greenfoot", built on Java programming language, is chosen in order the students to construct a computer game. We present seven important activities for designing and building this computer game (a monkey that tries to eat bananas while avoiding snakes).

### Activity 1

A monkey must know at all times how many bananas has eaten and what the score is, which is growing by eating bananas. State the appropriate fields in the Monkey class. In the first level, each banana gives the same score, but by expanding the game, the score given by a banana may be different, so it is suggested to indicate two different integer fields: one for the number of bananas that has eaten and one for the score.

Next, set the class constructor and provide the appropriate initial values. Remember that the constructor is the only case of a method that has no retrieved form at all and has the same name (even capital and lower case characters) with the class:

```
public Monkey()
{
// field initialization
}
```

After completing your code and no syntax errors occur, return to the main window, right-click on the monkey in the game world, and select the Inspect mode from the popup menu. A box will appear with the status of the object, i.e. a list of all object fields and current values.

### Activity 2

Set the Monkey class to checkKeypress() to control the monkey's movement. By pressing the left or right arrow keys on the keyboard, the monkey will turn counter-clockwise or clockwise, respectively, by a certain number of degrees. Try different values to achieve a smooth rotation.

Remember to call this method using the act method.

### Activity 3

Specify the *lookForBanana()* method in the Monkey class to control the detection of a banana. If there is a banana in the current position, the monkey will eat it and its power (score) will increase by 50 points. Remember to call this method using the act method.

### Activity 4

Set the *lookForSnake()* method in the Monkey class to control the monkey's impact on a poisonous snake. In the event of an impact, the game will end - all characters will remain stationary. Remember to call this method using the act method.

### Activity 5

Extend the **Monkey** class's l*ookForSnake()* method so that in the event of the game finish the "Game Over" message and the final score appear as shown in the figure below.

Note: you can use the ScoreBoard class you are getting ready to create an object of this class by passing the appropriate values to the constructor and adding that object to the current world / level of the game.

### Activity 6

Set the **Snake** class to a *turnAtEdge* method that will check if the snake has reached the limits of the world, so it will rotate it in a small number of degrees to keep moving inside the world and not remain "trapped" within the limits of the world. Remember to call this method using the act method.

### Activity 7

Set the Snake class to randomTurn which will turn left or right in a number of degrees the snake to some of its moves. Remember to call this method using the act method.

**Note**: You can use the static method *getRandomNumber()* of the Greenfoot class, which produces random numbers in the desired interval. For example, the *Greenfoot.getRandomNumber(100)* will return an integer number between 0 and 99.

**Caution!!** The *getRandomNumber()* method, like all other Greenfoot class methods, has been declared as static. This means that the method belongs to the class itself and we do not need to first create an object of the class and then invoke the method for that object. A static method is called writing the class name, a dot, and finally the method we want to call.

After completing these activities, you have implemented the first version of your game! You can always modify its features, in order to make your game to more interesting!

## Sample 5th: Collaborative Project

Organize for students to carry out a collaborative project online, for example through developing a multi-page website. Students could choose a broad topic of interest like e-safety, decompose it into smaller parts and then work collaboratively to develop pages for their website, exploring each individual sub-topic. The process of writing these pages can be further decomposed, through planning, research, drafting, reviewing and publishing phases.

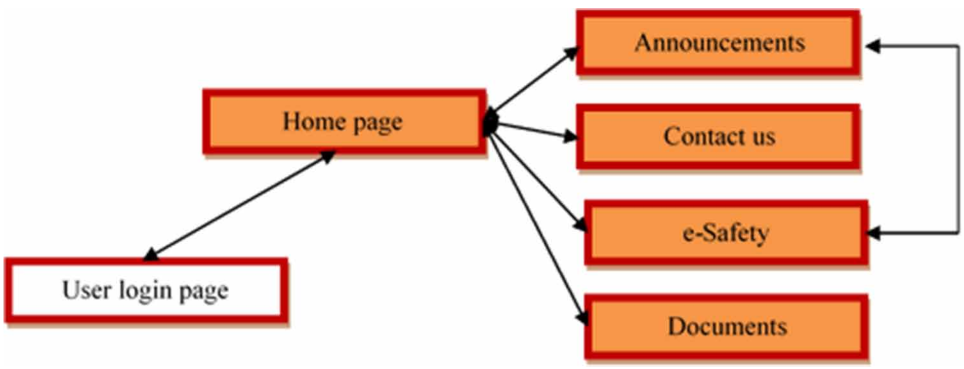
The goal is to create a dynamic website that can support online lessons about e-safety, while offering a set of functions and tools both for tutors and also for students and visitors of the site. So, there is an actual need to support users with different roles, in order to distinguish each one by his/her own capacity like administrator, author or visitor (e.g. administrators should be granted full accessibility while the simple users and visitors could be more restricted). All users should also be certified, i.e. they must be registered within the database and they should provide their email and password in order to enter the site (that is, to "make login"). So the web site will consist of the following web pages:

- **Home page (index)**
- **Announcements**

- **Contact us**
- **e-Safety**
- **Documents and**
- **User login (login)**

Before the above web pages appear, both the teacher and the students, should initially create the user login page for each registered user to enter its own personal login information. Therefore, a total of six individual web pages will appear. In the following Figure 7, we can see the structure of this web site:

*Figure 7. The structure of the website*




## Sample 6th: Writing a Story

In this activity (from the Google for Education), the students act as test writers, co-authoring the chapters of a single story. Each member writes a chapter independently, which begins and ends with some fixed "historical" points. After all the members have written the individual chapters assigned to them, the team collaborates and makes adjustments for each chapter (decomposition) so that the whole story is ultimately rational and coherent. The process of merging funds involves pattern recognition, abstraction, and generalization.

Students analyze the process and discuss whether it could be more effective and efficient (algorithmic design). To execute the activity requires as many teams as the chapters of history. Each student writes their own chapter without consulting or sharing information with other students. Each chapter must advance the story from the previous historical point to the next historical point. Thus, the first chapter begins with historical point 1 and ends with historical point 2. Chapter 2 begins with historical point 2 and ends with historical point 3 and so on. The length of each chapter can be determined based on the skill level of each group of students.

In the next phase of the activity, students will combine their separate chapters into a single story. The chapters can be radically different, requiring students to disassemble historical points, viewpoints and overall history and identify patterns in chapters to create a coherent and comprehensive work.

Students will take turns reading the individual chapters of their story and then work together to make the story logical and coherent. When reviewing chapters, try to keep each chapter as unique as possible. The ultimate goal is to make the whole story logical by combining every chapter, not rewriting everybody's chapters.

Students discuss what happened with the creation of their stories and how the work could be organized to make the problems that arose less likely through algorithm design, pattern recognition, abstraction and generalization.

Discuss the following questions with the students:

- What was the most difficult part of "debugging" in your story? Must the entire funds be adjusted or could you be able to reconcile the funds with only minor changes?
- What should you change in your story to facilitate the debugging process? Would you make your story simpler and more sensible, or more ridiculous and fantastic? Basically, how do you write the story so that it contains the smallest number of "errors"?
- How do you change the rules of this activity to ensure that a minimum number of "errors" are created? Suppose all chapters must be written independently by different students at the same time.
- Error recognition "is the" pattern recognition "skill of the PC. Can you outline some common error patterns that could be helpful for other writers to use to trace their story? This is related to pattern recognition, abstraction and generalization.

## NOVICE PROGRAMMING ENVIRONMENTS

The Novice Programming Environments (NPEs) adapt a programming approach that replaces syntax code and complicated text commands with components and graphics, that can be handled through drag-and-drop operations (Krul, 2012). This approach lessens the cognitive burden required in syntactic code writing, which is especially important for small age students, while it allows the user to focus on the semantic handling of the problem. Thus, the user is allowed to experiment with different component parts or code-blocks, by simply connecting, moving or discarding graphical objects. The two most popular NPEs, which we are shorty displaying, are Scratch and App Inventor (Papadakis et al., 2014). Both have been successfully introduced, since the last decade, in the curricula of primary and secondary level education informatics course, by the Greek Ministry of Education (Government Gazette, 2014).

### Scratch

Scratch was created by the Lifelong Kindergarten research group of the MIT Media Lab (http://scratch.mit.edu) and is regarded as the premier environment for introducing children to programming (Guzdial, 2004). Maloney and Resnick, members of the development team of Scratch, had stated that the development of Scratch, wanted to lower the ceiling programming level, in order for children to start programming earlier (Siegle, 2009). These researchers consider learning programming like writing. So, it is quite appropriate for children to start with simple forms of expression and gradually learn more sophisticated ways to express themselves over time. Using Scratch students are converted from consumers to producers of media, creating their own interactive stories and games, which can be freely redistributed via the Internet (Resnick, 2008).

Scratch encourages and facilitates the development of programs using a mixture of multimedia elements such as graphics, audio, video etc., in order to create new projects (Roy et al., 2012). Besides, the name Scratch suggests the idea of mixing, modification and mimics, like techniques of dj's, who play with their vinyl records in order to create new sounds. Scratch has been named as the YouTube of interactive media, because daily, 'scratchers' from around the world, upload many hundreds of novel stuff on the project website, the source code of which remains freeware. Within three years the number of projects upload at the official website exploded to more than two million (Federici, 2011). At the same time, the software was translated into more than fifty languages worldwide.

For the needs of programmers, Scratch provides ready characters, scenes and multimedia elements. Additional information can be created by the user with the help of the built-in painting tool, while it may also use external sources (Ford, 2008). In Scratch each object (sprite) can have one or more scenarios (scripts) connected to it. The scripts add properties to objects, allowing them to act in whatever way the user wishes within one's work. The scenarios are generated by joining blocks which are organized by category, such as control, traffic etc. A minimum use of the keyboard is required by the user, as the blocks move in the scene using the mouse and are joined together like Lego bricks. Blocks are connected only when their syntax is reasonable, thus completely eliminating syntax errors and greatly simplifying to the overall application development. Scratch supports the use of variables, selection and repetition structures and conditional object-oriented and parallel programming.

Scratch presents some weaknesses as it is pointed out by Harvey (2010). It does not apply the concept of recursion and its support for data structures, classes and inheritance is weak. However, these weaknesses are intentional, according to Scratch creators, in order to deliberately act as a deterrent for expert programmers and attract mostly young children. The University of Berkeley also has developed an extension called BYOB (Build Your Own Blocks), which addresses many of the shortcomings of Scratch. Figure 8 shows a simple screenshot of Scratch.

## App Inventor

App Inventor is a free web development environment that uses tiles in order to create applications for smart mobile devices running operating system Android (Wilson, 2012). App Inventor was announced as a pilot project of Google Labs in late 2009 and continued to develop until the end of 2011. The head of the development team was Professor of MIT Harold Abelson (2010). In early 2012 he transferred the work at the Center for Mobile Learning of MIT (Mobile Learning Center) for public use, as open source software. Within two years since its adoption from MIT, App Inventor has attracted more than two million registered users, many thousands of whom are active on a weekly basis. Before the advent of this tool, creating an Android application was a difficult and specialized process, because of the many requirements, such as good knowledge of Java and need of familiarity with professional software development tools (e.g. Eclipse, Android SDK, etc.). The App Inventor instead adopts the successful example of the use of visual programming with tiles (e.g. Scratch), adjusted to the programming smart mobile devices, mostly smartphones (Gray et al., 2012). The New York Times has called App Inventor as 'Do-it-yourself App Creation Software (Papadakis et al., 2014) The development environment supports all three popular operating systems. Figures 9 and 10 comparatively demonstrate the difference in creating a portable audio player using Java and App Inventor at the same time. One does not need to be expert to find out how easier things are done with that tool!

*Figure 8. Scratch screenshot*

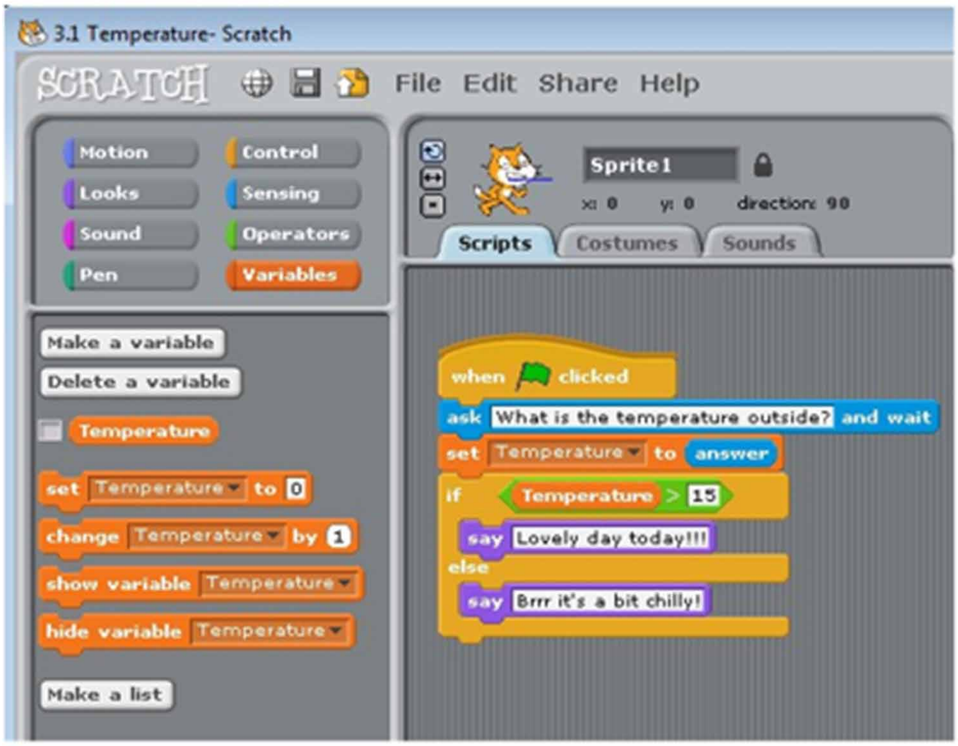



*Figure 9. Example Java code*

```
import java.net.URL;
import javax.swing.*;
import javax.sound.sampled.*;
public class PlaySound {
    public static void main(String[] args) throws Exception {
        URL url = new URL(
            "http://users.sch.gr/vorfan/media/testsound.wav");
        Clip clip = AudioSystem.getClip();
        // getAudioInputStream() also accepts a File or InputStream
        AudioInputStream ais = AudioSystem.
            getAudioInputStream( url );
        clip.open(ais);
        clip.loop(Clip.LOOP_CONTINUOUSLY);
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                // A GUI element to prevent the Clip's daemon Thread
                // from terminating at the end of the main()
                JOptionPane.showMessageDialog(null, "Press close to
stop!");
            }
        });
    }
}
```

*Figure 10. Example App Inventor code*


Combining the increased incentive for users to create portable applications with the advantages of using a block programming environment, App Inventor could be used for teaching basic control structures (MIT, 2013). Liu et al. (2013) indicate that App Inventor is extensively used in primary and secondary level education in U.S., while at the same time several universities adjust their curricula by introducing it for teaching and learning programming concepts (http://mobile-csp.org/). As Loukides (2010) notes, the purpose of App Inventor is to allow people who normally would not ever be able to develop applications they would wish for, to create them easily without becoming necessarily expert programmers. The overall aim of developing AI is to allow users of devices with Android software to evolve from consumers to creators (Papadakis et al., 2016; 2017). App Inventor consists of two main parts which allow users to build their applications a) Designer: this is a website where the user selects his components to implement and adjusts the properties of each one and b) Author (blocks editor): this is a feasible java window where the user assembles the code tiles and adjusts their behavior.

The App Inventor helps the user to directly observe the modifications of the application in the Android device. Alternatively, there is a possibility to use the integrated simulator. This is a complete virtual appliance, with the only negative point that is relatively slow compared to a portable device. The final product may be packed in the Android application package (format apk) or distributed in its web store. Although the last two years of development many features of Android are constantly incorporated in App Inventor (e.g. using GPS) there are still several limitations. For example, the inability to create full object-oriented programs, to access sources on the internet without the use of an external API (e.g. Facebook, YouTube, Amazon) or to create files with program size bigger than 5 MB. The MIT team is working on new releases of this tool which will incorporate new capabilities and features.

## SOLUTIONS AND RECOMMENDATIONS

With this increasing attention for teaching CT concepts, already from the early stages of formal education, a lot of successful examples have been accumulated inside the education system in various countries (Papadakis 2018a; 2018b). Despite the fact that the demand for horizontal integration of CT approach in the curricula of primary and secondary education is elevated, it is not quite clear how this will be achieved. Blueprints of formal policies and practices rarely come up in a more or less ad-hoc based operation in everyday teaching in classrooms.

Computing involves not only programming and practicing computing skills but also recursive thinking, model matching, and compositional reasoning. These will enhance algorithmic thinking, parallel thinking, and practical thinking as well. As a result, through CT, the learner develops problem-solving processes and dispositions. These are very important to the advancement of computer systems and functions, especially analysis and design as well as justifications to problems and solutions (Togyer & Wing, 2017).

In the previous section, samples of course designs incorporating basic CT skills where presented. The idea is to create repositories of good practices (including any other open resources) in the area of CT skills, such that every teacher/student can have free access to their content. This would be very helpful for the advancement of CT to a subject of equal, if not greater importance than reading, writing and math problem solving skills. Yet, care must be taken to ensure that CT is not confounded with programming languages or other technology subjects in general.

Abelson (2012) further points out that for CT to grow beyond re-constructible computational media, it needs to be complemented by computational values which promote open access to all educational community. This has paved the way to the popularity of open-source movements. Consequently, CT's objective of building individual capability and creativity (Weintrop et al., 2016) and the open-source movements have also influenced many professions and computer science school curricula also in developing countries.

So, future studies should highlight both CT components, necessary throughout school subjects and also CT strengths-weaknesses and differences-similarities inside modern curricula. It is also important to know how to assess and evaluate CT scenaria. Automated assessment tools could be developed, supported by state-of-the-art methods like "learning analytics". Also, organization of events like "Hour of code", "Informatics competitions" or other afterschool programs would be very helpful to that direction.

## FUTURE RESEARCH DIRECTIONS

Today CT, incorporating a cross-disciplinary set of mental skills, has become an essential literacy for most compulsory education learners in many different countries. These skills, like algorithmic thinking, logical reasoning, systematic information gathering and organizing, are mapped to various school grades through newly designed curricula. The educators' attempt is to implement those curricula in order to pinpoint the thematic coherence that transcends all formal scholar knowledge types through pedagogical approaches that use inquiry-based and collaborative learning as primary tools.

It is quite self-evident that, through sharpening problem-solving skills in various ways, CT gives young students better understanding and necessary knowledge of an increasingly digitized society in the perspective of the 4$^{th}$ industrial revolution. Thus, a lot of efforts are being done or expected to be done, by public educational systems, to further adopt this approach. The underlying philosophy of such systems is for the students to develop not just computer literacy but computational fluency, not just content competence but thinking skills leverage, not just following procedures but actively design their own learning paths. And this is quite an ambitious target and a promising research area.

At the same time, teacher perceptions, along with their skills, have to be appropriately aligned with these expectations. Teachers should engage in advancing their didactic ICT approaches by focusing their pre-service and professional development on those areas. They should keep up with technology advancement and understanding, feel more comfortable with computer infrastructure and various technology artifacts and review their stand on technology impact. The way they will rethink their strategy of incorporating ICT in everyday practice and become more effective, regardless of their discipline, reveals an opportunity for systematic future research.

## CONCLUSION

The diffusion of CT within school courses was the main reason of writing this work. An effort to explain its basic characteristics and develop corresponding models of sample activities was made. It would be interesting for teachers to study these examples and try to incorporate them into their daily practice. At this point, a simple reflection report could help summing up their experience gained from practical implementation. This is strongly recommended.

When they will do that, they will discover that CT is not just another set of soft skills but merely an invisible thread that traverses all subjects of the curriculum and repetitively unfolds in multiple hidden patterns within the educational process. Such an innovative approach sets solid foundations and prepares them for any changes in technologies that will arise in the future.

Generalizing beyond capabilities and skills, CT is compliant with the upcoming developments' prospective, which will allow better human understanding and deploy wider discussion about the power and boundaries of human intelligence. This will accelerate the process of transition to an artificial intelligence society, increasing the complexity of the ubiquitous digital ecosystem with which humanity is irreversibly intertwined.

## REFERENCES

Abelson, H. (2010). *Is App Inventor based on Scratch?* [Online forum comment]. Retrieved November 25, 2013, from https://groups.google.com/forum/#!topic/App Inventor/LvhZz8NZZ5g

Abelson, H. (2012). From computational thinking to computational values. In *Special Interest Group on Computer Science Education* (pp. 239–240). SIGCSE.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1-7), 107-117.

Cormen, T. H. (2009). *Introduction to algorithms*. MIT Press. cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Euclid, & Joyce, D. E. (1996). *Euclid's elements*. Clark University, Department of Mathematics and Computer Science.

Federici, S. (2011). A minimal, extensible, drag-and-drop implementation of the C programming language. In *Proceedings of the 2011 conference on Information technology education (SIGITE '11)*. New York: ACM. 10.1145/2047594.2047646

Ford, J. L. (2008). *Scratch Programming for Teens*. Boston, MA: Course Technology Press.

Google for Education. (n.d.). Retrieved from https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-materials

Government Gazette (2014). *Curriculum for Informatics courses in High School*. Greek Ministry of Education, FEK 932/14-4-2014.

Gray, J., Abelson, H., Wolber, D., & Friend, M. (2012). Teaching CS principles with app inventor. In *Proceedings of the 50th Annual Southeast Regional Conference (ACM-SE'12)*. ACM. 10.1145/2184512.2184628

Greenfoot. (n.d.). Retrieved from https://greenfoot.org/download

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School, 19*.

Guzdial, M. (2004). Programming environments for novices. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 127–154). London: Taylor & Francis.

Harvey, B., & Monig, J. (2010). Bringing 'No Ceiling' to Scratch: Can One Language Serve Kids and Computer Scientists? Proceedings of Constructionism 2010.

Knuth, D. (1968). *The art of computer programming 1: Fundamental algorithms 2: Seminumerical algorithms 3: Sorting and searching*. Addison-Wesley.

Krul, K. Y. (2012). *Teaching Control Structures Using App Inventor*. Master thesis. Retrieved from https://dspace.library.uu.nl/bitstream/handle/1874/254527/researchpaper-kevinkrul-25-6.pdf?sequence=1

Liu, J., Lin, C.-H., Potter, P., Philip, E., Zebulun, H., Barnett, D.-B., & Singleton, M. (2013). Going mobile with app inventor for android: a one-week computing workshop for K-12 teachers. *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13)*, 433-438. 10.1145/2445196.2445324

Loukides, M. (2010). *App Inventor and the culture wars*. Retrieved November 26, 2019, from http://radar.oreilly.com/2010/07/culture-wars.html

Massachusetts Institute of Technology. (2013). *What is App Inventor*? Retrieved December 01, 2013, from http://AppInventor.mit.edu/explore/content/what-app-inventor.html

Papadakis, St. (2018a). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, St. (2018b). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In *Proceedings of the 2014 workshop on interaction design in educational environments* (pp. 1-7). 10.1145/2643604.2643613

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2016). Using scratch and app inventor for teaching introductory programming in secondary education. A case study. *International Journal of Technology Enhanced Learning*, *8*(3–4), 217–233. doi:10.1504/IJTEL.2016.082317

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Resnick, M. (2008). Sowing the Seeds for a More Creative Society. *Learning and Leading with Technology*, *35*(4), 18–22.

Roy, K., Rousse, W. C., & DeMeritt, D. B. (2012). Comparing the mobile novice programming environments: App Inventor for Android vs. GameSalad. In *Proceedings of the 2012 IEEE Frontiers in Education Conference (FIE '12)*. Washington, DC: IEEE Computer Society. 10.1109/FIE.2012.6462363

Siegle, D. (2009). Developing Student Programming and Problem-Solving Skills with Visual Basic. *Gifted Child Today, 32*(4), 24-29. https://doi.acm.org/10.1145/2157136.21572

Togyer, J., & Wing, M. J. (2017). *Research notebook: Computational thinking–what and why?* Academic Press.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728.

Weintrop, D., Holbert, N. S., Horn, M., & Wilensky, U. (2016). Computational thinking in constructionist video games. *International Journal of Game-Based Learning*, *6*(1), 1–17. doi:10.4018/IJGBL.2016010101

Wilson, A., & Moffat, D. 2010. Evaluating scratch to introduce younger schoolchildren to programming. J. Lawrance, R. Bellamy (Eds.), *Proceedings of the 22nd annual workshop of the psychology of programming interest group – PPIG2010*. Universidad Carlos III de Madrid.

Wing, J. (2010). *Research Notebook: Computational Thinking - What and Why? The Link*. Pittsburgh, PA: Carnegie Mellon. Retrieved from http://www.cs.cmu.edu/ ~CompThink/resources/TheLinkWing.pdf

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. doi:10.100711528-016-0087-7

## ADDITIONAL READING

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978–988. doi:10.1016/j.compedu.2011.10.006

Campbell, C., Speldewinde, C., Howitt, C., & MacDonald, A. (2018). STEM practice in the early years. Creative Education Journal Special Edition Preschool Education Research, 9(1), 11-25.

Demetriou, K., & Nikiforidou, Z. (2019). The relational space of educational technology: Early childhood students' views. *Global Studies of Childhood*, *9*(4), 290–305. doi:10.1177/2043610619881458

DePryck, K. (2016). From computational thinking to coding and back. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 27-29). ACM. 10.1145/3012430.3012492

Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2010* (pp. 4006-4014). Chesapeake, VA: AACE.

Instefjord, E. J., & Munthe, E. (2017). Educating digitally competent teachers: A study of integration of professional digital competence in teacher education. *Teaching and Teacher Education*, *67*, 37–45. doi:10.1016/j.tate.2017.05.016

Jimoyiannis, A., & Komis, V. (2007). Examining Teacher's beliefs about ICT in Education. *Teacher Development*, *11*(2), 149–173. doi:10.1080/13664530701414779

Kalogiannakis, M., Nirgianaki, G.-M., & Papadakis, St. (2018). Teaching magnetism to preschool children: The effectiveness of picture story reading. *Early Childhood Education Journal*, *46*(5), 535–546. doi:10.100710643-017-0884-4

Kalogiannakis, M., & Papadakis, St. (2017). Combining mobile technologies in environmental education: A Greek case study. *International Journal of Mobile Learning and Organisation*, *11*(2), 108–130. doi:10.1504/IJMLO.2017.084272

Kalogiannakis, M., & Papadakis, St. (2018). An Evaluation of Greek Educational Android Apps for Pre-Schoolers. In Finlayson, O., McLoughlin, E., Erduran, S., & Childs, P. (Eds.), Electronic Proceedings of the ESERA 2017 Conference. *Research, Practice and Collaboration in Science Education*, Part 4/Strand 4 (co-ed. Kalle Juuti & Eleni A. Kyza, Digital Resources for Science Teaching and Learning), (pp. 593-603). Dublin, Ireland: Dublin City University.

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital Student Conference Platform Implementation: The case study of the "Research Project" course. *Open Education*, *12*(2), 5–23.

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, St., Kalogiannakis, M., Sifaki, E., & Vidakis, N. (2018). Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University. *EAI (European Alliance for Innovation) Endorsed Transactions on Creative Technologies, 5*(16), e1, 1-12.

Papadakis, S. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

Psycharis, S. (2015). The Impact of Computational Experiment and Formative Assessment in Inquiry Based Teaching and Learning Approach in STEM Education. *Journal of Science Education and Technology*, *25*(2), 316–326. doi:10.100710956-015-9595-z

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

## KEY TERMS AND DEFINITIONS

**Block-Based Coding Language:** A programming language that uses graphic elements as a means of providing visual cues to the user as to how and where commands may be used.

**Computational Thinking:** A novel problem-solving approach that enhances the interpolation of digital technologies with human ideas.

**Educational Robotics:** Is an interdisciplinary learning environment based on the use of robots and electronic components to enhance the development of skills and competencies in students.

**Programming:** The process of planning, scheduling, or performing of a program for a device (such as a computer).

**STEM:** The term STEM (science, technology, engineering, and mathematics) is an acronym used by those relevant to the educational method concerning the fields of science, technology, engineering, and mathematics.

**Text-Based Language:** A programming language that does not involve graphical elements (blocks) as a main part of its programming language, but instead is mostly oriented around text.

# Chapter 7
# Digital Game-Based Learning and Computational Thinking in P-12 Education:
## A Systematic Literature Review on Playing Games for Learning Programming

**Anastasios Theodoropoulos**
https://orcid.org/0000-0002-0143-4228
*University of Peloponnese, Greece*

**Georgios Lepouras**
https://orcid.org/0000-0001-6094-3308
*University of Peloponnese, Greece*

## ABSTRACT

*The objective of this chapter is to explore the evolution and opportunities of the emerging field of digital games for programming learning, the challenges and tensions that they present, and how educators may be able to collectively advance this work to benefit student learning. This work summarizes previous empirical evidence concerning the positive impacts and outcomes of digital games in computing education, or even impacts that do not let games to spread. Hence, a systematic literature review is carried out in this context to provide a comprehensive overview of works carried out towards incorporating digital games in order to acquire CT skills or learn basic programming concepts within P12 education. The chapter discusses on the range of indicators and measures used in the 44 selected studies, together with methodological limitations and recommendations for further work in this area.*



DOI: 10.4018/978-1-7998-4576-8.ch007

## INTRODUCTION

Because of the explosive growth of Computational Thinking (CT), many educational programs are formulated around essential CT skills. Though, several curriculum designers that aim to implement CT education into the school settings use programming tools and environments to expose students to coding. However, problem-solving skills are not something that can be developed by every student over a short period of time or over the same conditions. It requires students to experience and reflect on the consequences of their actions. This takes time, practice and effort that sometimes school's time and place do not permit. At the same time, learners continue to face difficulties in CT education, even though they use fluently computers and one could indicate that studying programming would be easier today (Buitrago Flórez et al., 2017). Extended research conducted over the past years show that these difficulties are still present and that students seem to be even less interested in programming (Grover & Basu, 2017). To this end, come the digital or else called computer games. Games that can reinvent CS in schools by motivating and educating all students, including girls and underrepresented populations, to acquire CT skills.

Positively, there are plenty of works that deploy digital immersive technologies such as computer games to improve CT skills in several educational contexts. The most frequently occurring outcomes and impacts seem to be knowledge acquisition/content understanding and affective and motivational outcomes (Basawapatna et al., 2010; Combefis et al., 2016). Moreover, playing digital games to learn programming is linked to a range of perceptual, cognitive, behavioral, affective and motivational impacts and outcomes (Theodoropoulos et al., 2017). The topic seems to be increasingly interesting because of its effectiveness, and therefore there are studies that summarize previously published works within computing education. In a recent review study, Lindberg et al. (2019) present games that consume numerous programming topics in their gameplay, by searching common game stores. Authors preformed an investigation into K-12 programming curriculum standards from seven countries and suggested which age groups match the games found. Vahldick et al. (2014) evaluated 40 games which claim to support the development of competencies in introductory computer programming courses (many were used in higher education) and found that most of them aim to develop students' skills, through solving problems. In another study, Lockwood and Mooney (2017), conducted a literature review with common programming tools used in secondary education. The authors listed programming tools (programming languages environments such as Python and game-development tools such as Scratch) as the main tools used to teach CT. They do not, however, identify papers where students play games to develop CT within schools. Moreover, the CT education domain is still in its infancy and requires research for developing theories of the learning mechanisms occurring in computer games (Kazimoglu, Kiernan, Bacon, & MacKinnon, 2012).

Although for the field of CT education it is clear that playing digital games leads to a variety of positive outcomes and impacts, it is also acknowledged that the literature on games is fragmented and lacks coherence (Lee et al., 2011). This lack of organization is regarded as an obstacle to progress in understanding the effects of games, developing more effective games and proposing guidance about how best to use them in programming learning. Zhao and Shute (2019) point out that the impact of digital games on CT development has only evaluated to a small extent. It also should be taken into account, that CT education is particularly challenging for students underrepresented in the fields of computing and engineering, such as girls and other learners from nondominant groups (Eordanidis et al., 2017). For these students, programming learning methods and digital games have been used together in such a way that one benefits from another.

Screening through literature, it can be concluded that, to the best of our knowledge there are no systematic literature reviews available that investigate the intersection of playing digital games and to learn programming in order to acquire CT skills within educational settings. Consequently, bearing in mind the aforementioned ideas, our focus is to examine the state of current research on playing digital games for acquiring CT skills and point out gaps in existing literature. Hence, a systematic literature review is carried out in this context to provide a comprehensive overview of works carried out towards incorporating digital games in order to acquire CT skills or learn basic programming concepts. This chapter discusses on the range of indicators and measures used in the empirical research papers within P12 education, together with limitations and recommendations for further work. This work can participate in summarizing the existing evidence of the proposed topic, in order to identify gaps and hopefully provide a framework for future research in the field.

The remainder of this chapter is structured as follows. The background section offers a brief introduction to games and programming learning, a summarized view of CT and the obstacles to CT education. The research design section describes the procedure for conducting this review. The results and discussion sections present both a quantitative and a qualitative analysis of the selected studies. Finally, the conclusion lays out suggestions for future studies, based on current research gaps.

## BACKGROUND: RELATED WORK

Jeannette Wing shaped the term CT in 2006, and she argued that it involves concepts like "*problem solving, designing systems, and understanding human behavior*" (Wing, 2006). In addition, CT is correlated with analytical thinking sills, since it shares with mathematical thinking in the general way in which someone approaches solving a problem (Bers, 2010). It also shares with engineering thinking in the way in which someone approaches designing and evaluating a large, complex system that operates within the constraints of the real world (Bers, 2010). It finally shares with scientific thinking in the way in which someone approaches understanding computability, intelligence, and therefore the human mind behavior (Bers, 2010). Therefore, CT is considered a fundamental ability for the 21st century, for everyone and not just the computer scientists (Rotherham & Willingham, 2010).

Literature shows that computer programming enhance CT skills and that their cultivation is usually accomplished through learning programming (Kazimoglu, Kiernan, Bacon, & MacKinnon, 2012). Programming or else called coding, is the core characteristic in the field of Computer Science Education (CSE). Characteristically, Ershov, one of the early Russian pioneers in the theoretical CS field, stated that: "*Programming is the second literacy*" (Ershov, 1981). Teaching CS started with programming and that slogan has become a popular metaphor, which is used widely around the world. Moreover, programming is about the process of writing programs. Programs may have various looks like text or shapes, and they can have a variety of instructions that command a computer to behave in a certain way. That behaviour is the process and a program specifies the computational process which can be executed at some time (Guzdial, 2015). Programming and CT are very close and correlated terms, since programming is a key tool for supporting the cognitive tasks involved in CT but is also a demonstration of computational competencies as well. Programming exposes students to CT which involves problem-solving using CS concepts like abstraction and decomposition (Lye & Koh, 2014).

Instead, digital games exist long before CT was popularized and labelled as an essential skill. Moreover, games which were once disconnected from schools are now being adopted by teachers as a key teaching tool. This comes as no surprise because computer games contain interactive, engaging and immersive elements that have educational affordances (Frazer et al., 2014).

In order to acquire CT skills and improve programming capabilities with the game-based method, it is important to highlight the difference between creating (*game design*, e.g. Scratch) and playing games (*gameplay*, e.g. code.org games). Game design refers to assignments whereby students are given the task to design and create games to demonstrate the application of learnt technical concepts (Basawapatna et al., 2010). While gameplay refers to serious games incorporated into traditional lesson plans so that students learn concepts through playing (Kazimoglu, Kiernan, Bacon, & Mackinnon, 2012). Most of these games involve a scenario designed to cover a basic programming task and learn algorithmic thinking, whereas some games cover more advanced learning objectives. In both ways, the main idea is to shorten the time between theory and practice and merge abstract concepts with practical experiences and therefore to inspire students to learn (Vahldick et al., 2014).

Previous work by Wassila and Tahar (2012) organized digital games according to the CT skills that they aim to develop in the students. Authors suggest that the teachers can choose the game that fits into their classroom needs and goals. They compiled three main game genres for that purpose: action (e.g. maze, combat and platform), strategy (e.g. adventure) and hybrid games (e.g. real-time adventure and simulations). The game genres that suits most cases are puzzles, simulations and strategy games (Wassila & Tahar, 2012). In another study, Malliarakis et al. (2014) examined games in terms of the educational value that they bring for teaching programming. The authors resulted that games should provide students with clear educational goals and learning outputs, an immersive environment, interesting scenarios and tools that help them communicate and collaborate with their classmates.

Moreover, the fact that every student encompasses a totally different style of learning and process data has long been recognized by educationalists. Previous research has revealed games as useful tools to encourage STEM participation and learning and to support identity exploration (2014). Digital games can be the key, since they support transformation of game-players' knowledge and self through participation in the gaming activity that involves the person in a dynamic individual-environment interaction (Shah et al., 2017). From a situative perspective on learning, digital games support transformation of game-players' knowledge and self through participation in the gaming activity that involves the whole person in a dynamic individual-environment interaction (Shah et al., 2017). Students need to apply knowledge in a new situation while playing computer games to learn programming. Mental approaches have to be turned into coding (Winslow, 1996). With games the learners have to solve coding problems and develop essential skills such as strategies for abstraction, generalization, decomposition and problem-solving.

Next, we present our research design and findings to research if digital games can be used as a vehicle to train students to acquire CT skills in a fun and engaging environment.

## RESEARCH METHODOLOGY

The main purpose of this work is to perform a systematic literature review in order to identify, evaluate and interpret relevant researches in the field of CT education with digital games. The research methodology of this study is based on the recently revised guidelines of systematic literature review (SLR) (Kitchenham et al., 2015) which is among the most widely approved. According to SLR, we followed a quasi-gold standard based search refinement method for optimal search results: Initially, we identified the need for this study from background, then we defined the review protocol, then we selected conducted automated searches, then we assessed the quality of these studies, and finally we conducted the data extraction.

## Research Questions

Based on the main purpose of this work and in order to highlight the existing evidence, gaps and future path for the proposed areas, we have established the focus sub-areas through the following research questions:

**RQ1:** Can digital immersive technologies such as digital games be used as an effective method to teach students CT?

Rationale: Highlight the positive or negative impacts and outcomes of incorporating digital games in computing education.

**RQ2:** What cognitive, pedagogical, and social processes are involved in programming learning through playing digital games?

Rationale: Explore the learning processes occur most.

**RQ3**: Who are the learners, in terms of biological, social, cognitive, experiential, personal learning and affective characteristics, within playing digital games for learning programming?

Rationale: Identify learners' characteristics.

**RQ4**: Does the participation in programming gaming activities influence specific programming concepts or CT skills?

Rationale: Identify the CT elements/skills that take advantage within digital games to learn programming.

**RQ5**: Are there any other factors that affect the performance of learners/players through participation in programming gaming activities?

Rationale: Identify factors that affect the learning process through playing digital games to learn programming.

In the following, we describe the activities carried out in each phase of this systematic literature review.

## Data Collection and Search Strategy

For data collection the keywords for the search sting were chosen. Firstly, general terms were used with the aim of assuring that most of the relevant research papers were included in the study. The main search terms were "games" and "programming". In order to finalize the search string (*Table 1*) for our review process, we used the following steps described in (Kitchenham et al., 2015):

- Derive major terms from the questions by identifying the main concepts.
- Identify alternative spellings and synonyms for major terms.
- Check the keywords in any relevant papers.
- Boolean OR usage to add alternatives spellings and synonyms.
- Boolean AND usage to link the major terms.

In order to collect high-quality data, we searched for previous works in the following online bibliographic databases: Association for Computing Machinery Digital Library (ACM), Science Direct, ERIC and Springer Link. Moreover, we searched independently in key educational journals and conferences in Computing and Computer Science education domains, including: ACM Transactions on Computing Education (TOCE), Computers and Education (Elsevier), International Journal of Child–Computer Interaction (Elsevier), Conference on Computer Science Education (SIGCSE), Conference on Human Factors in Computing Systems (SIGCHI), ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE) and ACM Workshop in Primary and Secondary Computing Education (WiPSCE). These resources were chosen because they index the most CS papers about CT and programming learning in educational settings.

## Inclusion and Exclusion Criteria

The procedure of data extraction took place in three phases: (1) Initially, a primary analysis was carried out to collect the standard information and inclusion-exclusion criteria types of data. The search results were reviewed carefully based on paper's title, abstract, keywords and conclusions to identify

*Table 1. The search procedure executed in each database.*

| Database | Search terms protocol | Additional information |
|---|---|---|
| **ACM Digital Library** | (recordAbstract:(+(school k12)+(programming "computational thinking")+(teaching teach learning learn)+games))) | - search in the field "Abstract"<br>- search from years 2009 to 2019<br>- 194 initial results |
| **IEEE Xplore** | ((((((((((:teaching) OR:teach) OR:learning) OR:learn) AND:programming) OR:computational thinking) AND:games) AND:school) OR:k12) OR:p12) | - search in the field "Abstract"<br>- search from years 2009 to 2019<br>- 184 initial results |
| **ERIC** | ((teaching OR learning OR teach OR learn) AND (games) AND (programming OR "computational thinking") AND (school OR k12)) | - search in all fields<br>- full text available<br>- search from 2009 to 2019<br>- 51 initial results |
| **ScienceDirect** | pub-date > 2009 and title-abstr-key((teaching OR teach OR learning OR learn) AND (programming OR "computational thinking") AND (games) AND (school OR k12)) | - search in the fields "Title", "abstract" or "author-specified keywords"<br>- search from years 2009 to 2019<br>- 39 initial results |
| **Springer Link** | (games) AND ("computer science education" OR "computing education") AND (programming OR "computational thinking") AND (school OR k12) AND (teach OR learn OR teaching OR learning) | - where the title contains<br>- search for articles<br>- search from 2009 to 2019<br>- 26 initial results |
| **Wiley, Taylor & Francis** | "teaching OR learning" and "programming OR computational thinking" in Abstract and "games" in Abstract and "school OR k12" in Abstract | - search in the field "Abstract"<br>- search for articles<br>- search from 2009 to 2019<br>- 11 initial results |

relevant papers. (2) Then a more careful examination over each of the papers were performed to collect the research question addressing level, answers to research questions and specific digital games and programming learning related data. (3) Finally, in the third round of review, through a full-text analysis based on the inclusion and exclusion criteria as presented at *Table 2*, we identified papers which have contributed to the field positively.

*Table 2. Extracted data from previous studies.*

| Type | Data |
|---|---|
| Standard Information | Title, Authors, Publication year, Journal or Conference name, Publisher, Paper Type, Number of Citations, Average Citations per year, Date of Extraction |
| Inclusion and exclusion criteria | Described activities irrelevant with playing games<br>Did not combine games with programming learning or computational thinking acquiring<br>Did not focus on educational settings (p12)<br>Inadequate evaluation methods<br>Described activities irrelevant with playing games<br>Did not combine games with programming learning<br>Did not follow well-structured research methods<br>Insufficient data or unclear findings<br>Empirical research OR game presentation<br>Language,<br>Peer-reviewed |
| Research Questions | Extent to which each research question is addressed (RQ1, RQ2, RQ3, RQ4, RQ5) |
| Answers to Research Questions | Direction of the study, Usage of Games in Teaching, Usage of Games in Learning, Effects of Games, Benefits reported, Challenges reported |

## Quality Assessment

To assess the quality of the selected studies, we performed an evaluation process based on selected criteria adapted from Kitchenham et al. (2015). The process focused on the methods that the selected studies adopted as well as their scientific rigor and the criteria were:

1. How well was the research was conducted in terms of method and structure?
2. How well was the data gathering performed?
3. How clear and consistent have the relations between results, analysis and conclusions been?
4. How generalizable are the findings of the study and to what degree would be applicable across age groups, class, ethnicity and so on?

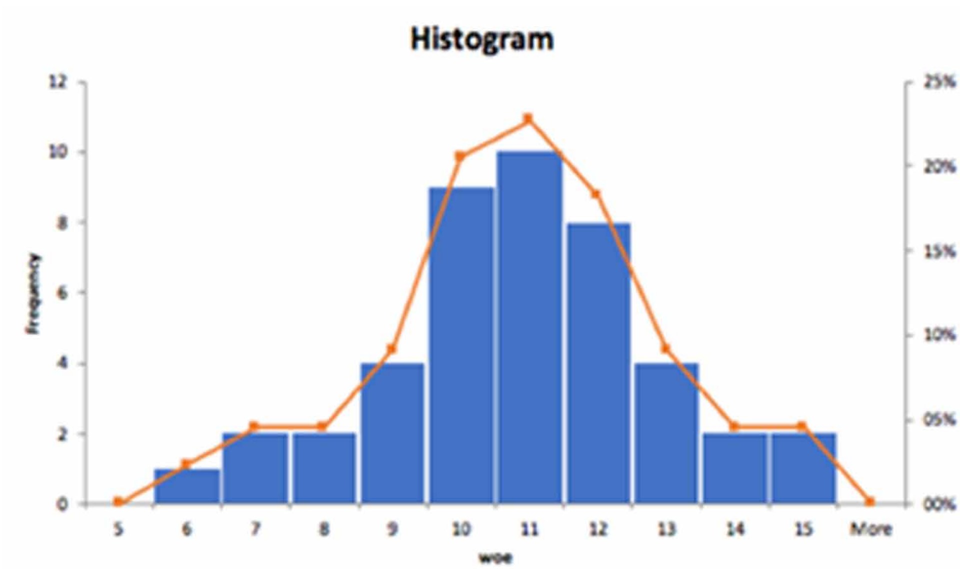
Moreover, the selected works were evaluated in the following that derived from the present study design:

5. Does the research design enable us to find relevant information to answer RQs of this study?

Two scholars carried out the evaluation separately, and each paper was assessed on a scale from 1 to 3 (low, moderate and high). In cases that the ratings ranged two points, a third researcher performed the evaluation process and finally the two biggest scores were adopted. The total weight of evidence (woe) for each publication was considered by adding scores on each of the five criteria. Scores varied from 5 to 15, with 5 being a low and 15 being a maximum.

The histogram in *Figure 1* shows the apparent gaps in the threshold scores adopted. Most studies received a good score, which is to be expected since they were published in peer-reviewed scientific conferences and journals. The mean rating for the selected papers was 10.86.

*Figure 1. Histogram of the quality assessment scores for included papers.*

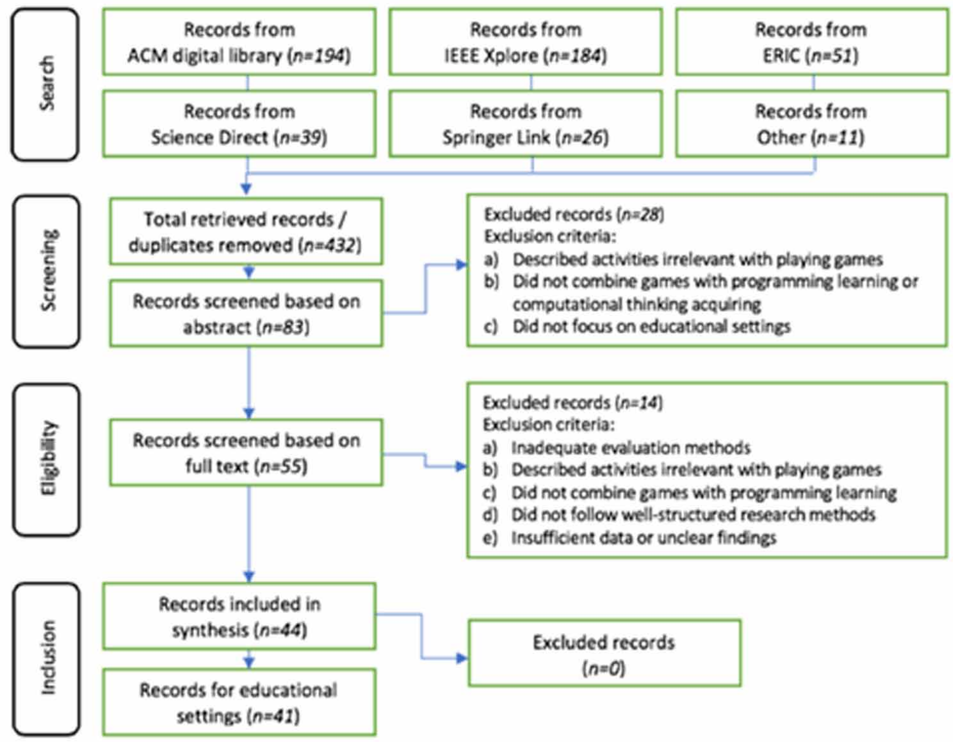



## RESULTS

From search in the digital databases/libraries presented in the previous section, 505 scientific publications were identified. The duplicates were then removed, and 432 work remained. After applying the first filter (screening), i.e. after reading of title, abstract and keywords, 83 papers were selected. Inter-rater reliability showed good to moderate agreement during the title ($\kappa = 31.7\%$), abstract ($\kappa = 44.3\%$) and full-text ($\kappa = 33.6\%$) screening stages. From the 83 papers of the first filter, 55 were selected after the application of the second one (eligibility) and finally 44 papers were approved to data extraction step. *Figure 1* shows some details of this process.

The selected works are listed in the *Reference Section* of this chapter [R01]-[R44]. Most works are published in conference proceedings *n=30* (73,17%), while only *n=14* are published in peer-reviewed journals (26,83%). *Figure 2* presents the detailed list of publication venues.

*Figure 2. Search process, indicating number of articles (note: a study might be excluded for several reasons).*




The analysis shows that there was a considerable increase in number of publications about this study after 2017 publications, versus an average of the previous years (*Figure 3*).

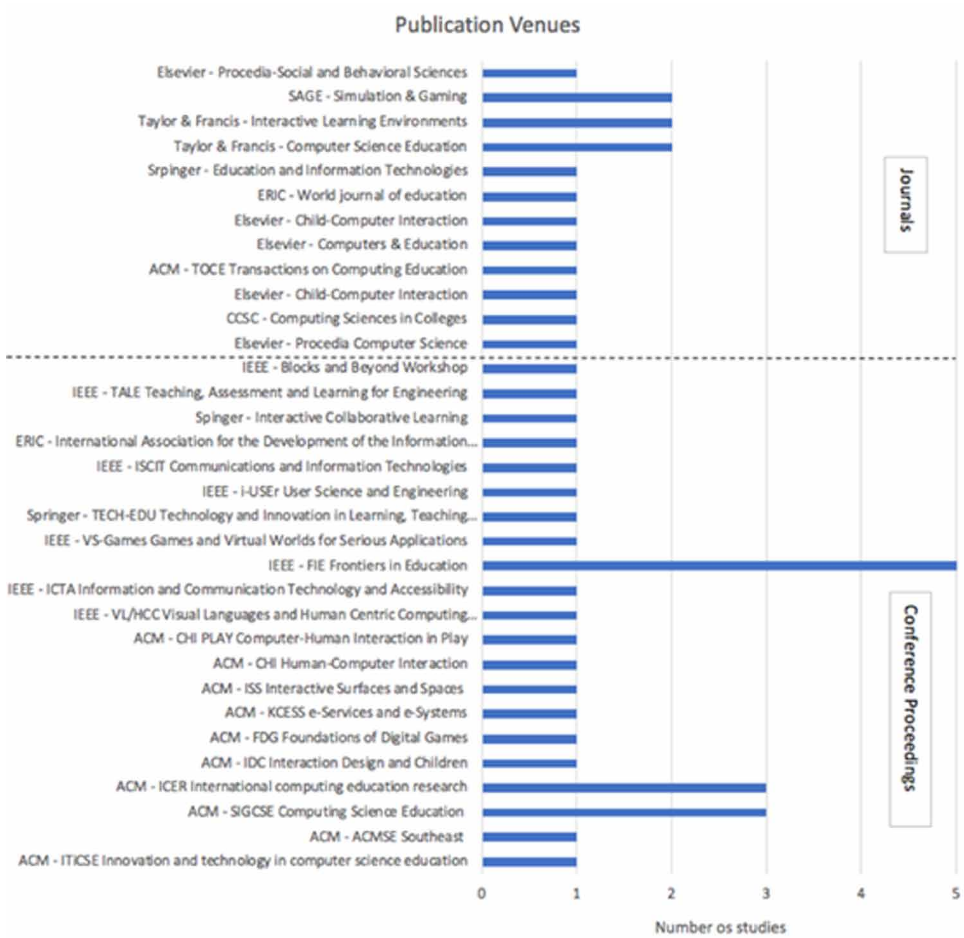
## Subject Area and Methods Used

In order to understand the role of games in CT education, it is very important to identify the scope of each study. Therefore, we outlined the subject area on which the papers focused and the instruments that were used. The largest number of studies (*n=38*) targeted to enhance CT, algorithmic thinking and basic programming skills. Moreover, most studies measured motivation and engagement of students through the gaming activities (*Table 3*).

Researchers used various instruments, usually in combination, to collect information and determine the outcome of the gaming workshops. In total, 22 studies used questionnaires, 9 log files, 5 observations and interviews and 3 studies were literature reviews on games about programming (*Table 4*).

Moreover, most studies used digital games for learning programming as part of CS courses at school settings. Some studies suggested that the current trends of learning through games can provide rich ground for developing more advanced programming experience (*n=7*).

*Figure 3. The list of publication venues.*




## Games Genres

Most of the selected papers relate to puzzle games (*n=32*). This was quite expected, given that such games are based on basic computational logic and pattern recognition and thus are easier to follow (Rogers, 2014). Other popular genres were simulations and strategy games. In addition, most studies investigated different games, which was also expected, given that most games are firstly presented through these works (*Figure 4*). The main coding activities that are possible through these games are:

- Write Code: the player has to write instructions, to accomplish a mission and so he creates a function or a program e.g. R10, R17.
- Program an Agent: the player types code and creates a program that represents the behavior of the intelligent agent, seeing him to react in real time e.g. R07, R14.
- Debug the Code: the player is given a program that contains a problem (bug) and he needs to find it and fix it e.g. R38, R42.

*Figure 4. The number of studies published from 2010 to 2019.*

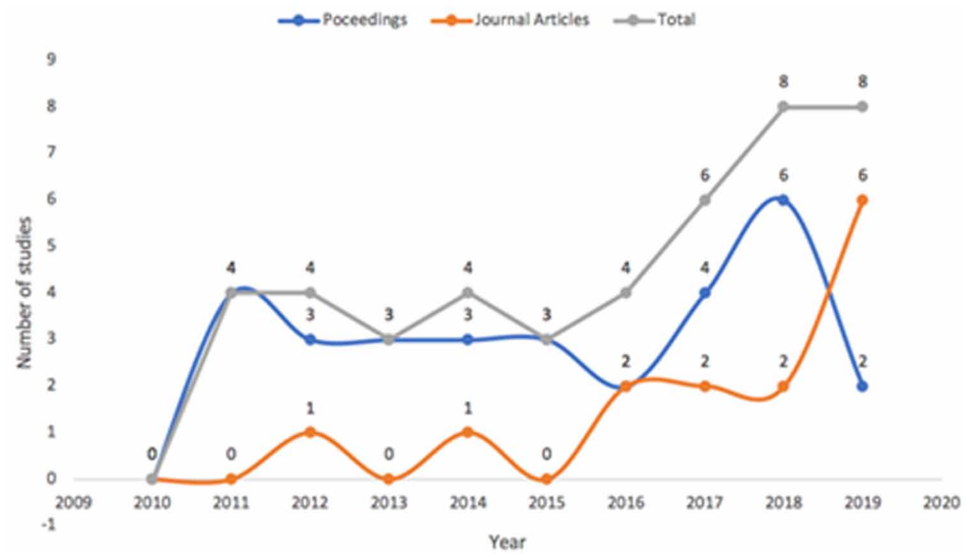


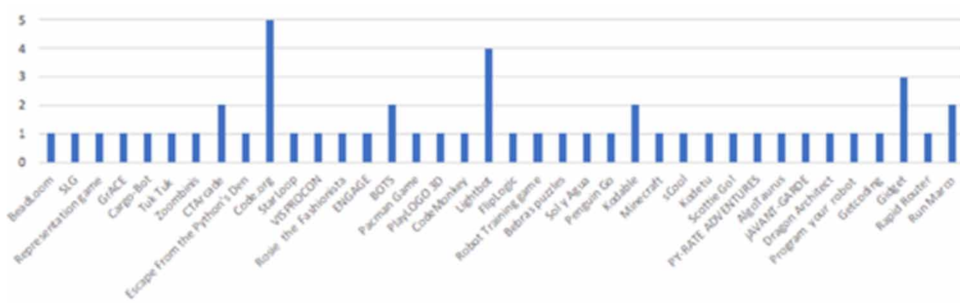

*Table 3. Measurements in the selected studies.*

| Subject area | References |
|---|---|
| Cognitive behavior, pedagogical aspects | R01, R02, R05, R07, R08, R11, R15, R19, R22, R25, R28, R30, R32, R34, R35, R36, R39, R40, R42, R42, R44 |
| User' s experience, satisfaction, enjoyment | R01, R04, R05, R10, R24, R25, R27, R28, R29, R31, R33, R34, R35, R36, R42 |
| Personality traits, confidence, efficacy | R02, R10, R13, R14, R15, R40, R42 |
| Motivation, engagement | R01, R03, R04, R06, R08, R12, R13, R14, R16, R17, R18, R19, R25, R26, R27, R31, R37, R41, R43 |
| Learning outcomes, performance | R01, R04, R05, R10, R15, R22, R23, R28, R29, R34, R35, R36, R42, R43 |
| Usability, playability, gameplay | R06, R09, R20, R21, R37, R38, R44 |
| Review studies, comparative analysis | R19, R20, R40, R41 |

*Table 4. Instruments used in the selected studies.*

| Instrument | References |
|---|---|
| Observations | R07, R12, R22, R35 |
| Interviews | R17, R26, R39, R41 |
| Literature review | R20, R41, R42 |
| Survey, questionnaires | R01, R03, R04, R05, R08, R10, R13, R14, R23, R24, R25, R27, R28, R29, R30, R31, R32, R33, R34, R36, R38, R42 |
| Game presentation, logs | R02, R03, R06, R09, R10, R11, R12, R15, R16, R18, R22, R23, R24, R32, R35, R37, R38, R42, R43, R44 |

*Figure 5. Digital games for learning coding.*




- Fill the Code: the player is given a program that contains a set of blocks/code and he has to fill some missing parts in order to complete a task e.g. R22, R43.

Moreover, considering the need to enhance the value of teaching or learning programming through digital games, previous work from Combefis et al. (2016) presents a taxonomy which includes four categories for serious games classification: Modality, Interaction style, Environment and Learning approach. Based on this classification, in the results of this study the most usual modality fields are visual and auditory, but sometimes it may be haptic like the StarLoop game (R11) a tangible interactive environment where the players use loops and procedures to solve puzzles. As for the interaction style, most games use combinations of styles like graphical direct manipulation, questions and answers, avatars etc. The basic feature that differentiates the learning environment in the games found by this review, is whether they use a block-based approach (e.g. Blockly) or a text-based programming language (e.g. Python). Most studies use the block-based approach (*n=31*), which was expected since there is no typing and syntax errors are eliminated which makes the game suitable for novice programming. The text-based approach (*n=10* studies), seems to be more suitable for advanced students, as they can code more complex structures.

## DISCUSSION

This section presents the answers to the research questions of this study, founded from the 44 extracted papers.

### Games to Teach and Learn CT, Positive and Negative Features (RQ1)

Playing digital games to teach and learn CT obviously results in a number of positive outcomes and effects (*Table 5*). This work identifies the most common subject areas for applying types of instructions after reviewing 44 studies (*Table 3*). Our literature review found that digital games are used widely in programming learning, as well as in CT education area. This result was anticipated, as digital games are one of the most prominent immersive technologies to which technological resources are applied. As the empirical works state, games is an effective method to teach students basic programming principles and thus acquire CT skills.

*Table 5. Positive and negative outcomes*

| Outcomes | References |
|---|---|
| Motivation and engagement | R01, R03, R04, R05, R06, R07, R08, R09, R10, R12, R13, R14, R16, R17, R18, R19, R24, R25, R26, R27, R28, R29, R31, R34, R35, R36, R37, R41, R42, R43, R44 |
| Direct feedback and learning gains | R01, R03, R04, R05, R07, R08, R10, R11, R12, R13, R14, R15, R16, R22, R23, R24, R25, R26, R28, R29, R30, R31, R32, R34, R35, R36, R42, R43, R44 |
| Ease of use (no prior experience or knowledge needed) | R01, R04, R05, R06, R09, R10, R20, R21, R24, R25, R27, R28, R29, R31, R33, R34, R35, R36, R37, R38, R42, R42, R44 |
| Assessment and control of performance | R01, R04, R05, R07, R10, R15, R22, R23, R28, R29, R34, R35, R36, R40, R42, R43, R44 |
| Scaffolded learning-gameplay | R02, R05, R08, R10, R28, R34, R35, R36 |
| Difficulty and usability | R06, R09, R20, R21, R37, R38, R44 |
| Collaboration | R04, R11, R19, R28, R29 |
| Creativity | R13, R17, R32 |

Though, the main advantage of the games approach (reported in *n=31* studies) is the motivation and the engagement that the media offers to learners (*Table 3*). For example, PY-RATE adventures (R34) is a 2d action platform game in which the player controls an avatar and has to surpass several obstacles, avoid or destroy enemies and complete all the levels. In the game PlayLOGO 3D (R18), students have to accomplish missions in a future contest for robot pilots which takes place at a constellation of Andromeda galaxy. As found through the games, teaching and learning CT incorporates numerous elements to motivate learners like the interactive scenario, the narrative and the players control.

Another positive game element that was revealed through numerous studies (*n=29*) is the direct feedback and the mastery in the form of ranking with process indicators (points, level ups, badges etc.). Gidget (R42), is a game where a little robot excuses himself for not being able to write code correctly to fulfill the missions. The players get immediate feedback which shows to have a noteworthy impact on motivation to program and learning success.

Moreover, the fun and scaffolded learning with challenges that increase can be very helpful for children to better articulate algorithmic thinking. CTArcade (R08) is a game that uses well-known games like Tic-Tac-Toe and players have to build a strategy through their gameplay. The scaffolded gameplay of this approach and the fund interface facilitated young students to better understand CT thinking patterns, which were tacitly present when playing the game on paper, but not clearly obvious to them until using the CTArcade game. Also, the social connection with other players, can provide positive outcomes (R28).

In addition, digital games can be used to teach and learn CT without prior programming experience, since they avoid the problem where students need a great deal of programming knowledge before they can start to develop their own programs, e.g. Gidget (R42) which incorporates the debugging games approach. There are also various choices of digital games to learn programming as presented at *Figure 4*, which do not require special hardware resources or materials and can therefore easily adopted to different educational contexts.

Instead, there are problematical impacts reported in some cases. Many studies explored learners' attitudes towards the use of games in CT education through questionnaires. GrACE (R04) was tested with middle school students and resulted that despite the enjoyment that most students agreed, in terms of difficulty most of them said that the game was frustrating, challenging, and hard. And since we talk about serious games, we want students to learn by paying. Assessing the results of gaming is a very important issue to address, which draws a lot of focus. Some games provide analytics (R03, R12, R22) to better understand the players' behavior, while in some studies the authors developed an assessment rubric to evaluate gaming performance (R10). In Gidget (R43), researchers found that including tests in their game led students to play more and complete extra levels, which indicates better engagement. However, as shown from most studies there are some limitations on the scoring since several aspects in the gameplay such as assessment and controlling the game accordingly are ignored. In some cases (R10, R17, R30), the game provides students with certificates of attained skills, though these are given outside the gaming context, which might not be able to encourage students to keep playing. Moreover, marking players' performances should include various factors such as of games genres, CT skills taught, programming concepts and gaming context. Games could incorporate mechanics for scoring performance comprising categories, like aim, implementation, integration, and primary assessment type (Shute & Ke, 2012).

As shown from most studies, ease of use, perceived usefulness, enjoyment, simplicity and clarity of intent are important factors that draw learners to use games. Game designers should take them into consideration seriously while developing serious games for CT education, and educators should also try to integrate these factors into teaching to increase learning performance and inspire students to engage in learning activities.

Finally, we cannot generalize the positive or negative elements from the serious games to learn programming, since results could change under different conditions e.g. age group, cultural background. So, what can be said is that digital games have much potential to assist in CT education. However, the gain in learning by students is not guaranteed just by the simple playing games, as there are several factors that can determine the outcome.

## Processes Involved (RQ2)

Learning and digital games have increasingly linked each other, and numerous models have been developed that identify the learning effects that playing games can take. Especially, in CT education mental approaches have to be turned into coding (Winslow, 1996). The 44 works presented here, provided insights for the following processes: learning outcomes (CT skills or programming knowledge acquired), cognitive outcomes (involving declarative, procedural and situational knowledge) and affective outcomes (attitudes or beliefs) (*Table 6*). The games' benefits involve enhancing mental skills. In the majority of the game's learners must solve coding problems in order to develop CT skills, while many studies presented (*n=14*) statistically significant learning differences in groups who used games in the educational process.

Actually, digital games have the potential to help students find many ways to learn. For example, the Robot Training game (R22), encourages learners to investigate and find multiple ways of tackling the problems. It has grading levels in such a way that exposes them to different CT concepts and as the story goes, they have to apply previous knowledge, without directly understanding the learning processes involved. In study (R25) researchers examined the attitudinal influences and cognitive results from a short intervention with students playing the game Penguin Go. The results showed that CT skills were initially improved significantly, but when the researchers added additional constraints to the programing puzzles, the students did not generate a significant impact on learning. In sCool game (R31) learners had problems to transfer the learned concepts into similar problems and they focused more on the gameplay

*Table 6. Processes involved*

| Processes | References |
|---|---|
| Cognitive behavior, pedagogical | R01, R02, R05, R07, R08, R11, R15, R19, R22, R25, R28, R30, R32, R34, R35, R36, R39, R40, R42, R44 |
| Learning outcomes, knowledge acquired | R01, R04, R05, R10, R15, R22, R23, R28, R29, R30, R34, R35, R36, R42, R43 |
| Personality traits, confidence, efficacy | R02, R10, R13, R14, R15, R40, R42 |
| Social interaction, collaboration | R04, R11, R19, R28, R29 |

instead of the learning content. The study showed that teachers need to have a wide overview about the students learning progress. Therefore, it seems necessary to examine further the way that CT concepts are introduced through games.

Modern reports of efficient learning recognize that many variables contribute to the task of performance or influence it. In study R15, the BOTS game was tested by applying three strategies: instructional text, worked examples and buggy code. Authors evaluated the different methods with middle school students and resulted that bugs demonstrated the lower completion time and solution code length in assessment puzzles. In R30 authors conducted a research with children aged 4-6, and games for code.org. They found increasement in the non-verbal cognitive abilities of most children but there was no statistically significant difference in their problem-solving skills through the games. To this end, Connolly et al. (2009) propose a model for serious games that incorporates motivational variables such as interest and effort, as well as the students' attitudes and preferences towards games in addition to the student performance consideration.

Another process is the social interaction which is useful about the mixing of children of different genders, classes, and intellectual levels in the same gaming environment. Socialization involves aspects of interaction between players (in a cooperative and/or competitive manner) while developing CT skills. Some studies reported positive effects on collaboration through games in order to solve puzzles (R04, R11, R29). In the game Program Your Robot (R19), players can discuss between them and form common the strategies to solve the challenges and therefore achieve a better score. An appropriate method towards collaboration and socialization processes, is the pair-programming approach, which for example is implemented in the Engage game (R14).

Moreover, some studies focus on elements of games that might help reduce student dropout rates in the CS domain (R13, R14). Study R14 suggests that secondary school level may be too late for significantly impacting CS attitudes. Digital games can even help to recruit more students in CS by maximizing the participation and motivate programming novices (R05).

Finally, there is another important process that goes beyond learning CT and coding. It is the concept of metacognition where children think about their learning. Games like Gidget (R42, R43, R44), may help students to explain their answers and immediate motivate them and learn. Designers of educational technologies like games should take into account how the human mind works and its cognitive processes. For example, working memory represents a major element affecting learning performance in serious games and may become overloaded if more than a few pieces of information are processed at the same time. The present review suggests adopting a broad range of outcomes on playing games relating to CT skills acquisition, and relate them to affective and motivational outcomes.

## Learners Characteristics (RQ3)

Collectively more than 1,000 students participated in the studies reviewed for this chapter. Apart from a few studies, the sample size of the participants was fewer than 50 participants, some workshops (*n=17*) had fewer than 20 participants, while a small number of cases consisted of more than 50 people (*n=8*).

Most studies (46%) focused on middle school students, followed by 41% focused on high school students, 39% focused on elementary school students, only 7% focused on early childhood education, while many of the studies included multiple school levels. Considering the participants' age (*Table 7*), the majority of the studies involved ages up to 14 years (*n=32*) and a smaller number (*n=9*) were conducted with ages over 14 years. Four works had participants of mixed ages ranging from 12 to 17 years old.

Over half of the studies (59%) included both female and male participants. More than a third of the studies (37%) did not give any gender details. Some studies (R03, R09, R14, R18) explored games in order to interest and engage more female youth in CS.

*Table 7. Age level of the participants in the selected studies.*

| Learners | References |
|---|---|
| Preschool | R06, R26, R30 |
| Elementary | R02, R07, R08, R13, R15, R18, R20, R21, R22, R23, R24, R27, R28, R29, R33, R35 |
| Middle school | R01, R02, R03, R04, R08, R10, R11, R13, R15, R16, R20, R25, R31, R32, R38, R42, R43, R44 |
| High school | R01, R05, R10, R12, R14, R16, R17, R20, R31, R36, R38, R39, R41, R42, R43, R44 |
| Postgraduate | R34 |

In terms of cultural background and diversity, the majority of the studies did not provide any information. Digital games promote learning for all and are an approach that "*doesn't draw boundaries between players and designers as participants of digital media culture but rather sees them as complementary to each other*" (Kafai & Burke, 2015).

## CT Concepts (RQ4)

The main implications derived from the analysis include that most games have been developed to cover programming concepts (such as variables, simple and nested if statements, loops, arrays and functions). Cargo-Bot (R05) covers the more complex concept of recursion. This fact was expected, since most games seem to successfully accomplish the educational goals, they have set with respect to the group of concepts they aim to teach. For each CT skill, we identified common elements found in the 44 selected works for learning programming:

*Abstraction*: this skill is main in games whose interaction occurs by state machine and is developed by definition of individual states/rules of the machine.

*Algorithmic Thinking*: most games include this skill by terms of a challenge that the player has to accomplish by following a step-by-step solution (algorithm). In most games the algorithm can be expressed as a sequence of instructions in the form of blocks, icons or text.

*Repetition*: most games provide the repetition structures with loops. Instead of repeating the same sequence of action multiple times, the player can use a loop. The games examined different kinds of loops: simple loops, conditional loops and nested loops.

*Decomposition*: many games adopt the divide and conquer strategy, where the player breaks the initial problem in smaller ones, less complex and easier to solve. For example, in the game Dragon Architect (R37), the player initially builds small structures (walls and towers), while later she/he builds more complex one like castles, by exploiting the small structures.

*Debugging*: this skill is used in in some games allowing the visualization of the solution execution proposed by the player, in order to analyze a behavior and compare it with the expected results. The learner develops the skill by analyzing, interpreting and fixing the code. Gidget (R44) and BOTS (R38) aim to teach programming using the debugging mechanism.

*Generalization*: in the selected studies this skill is developed by identification and generalization of patterns. In the game Zoombinis (R07), in order to solve a programming challenge player must use a common characteristic, determining that skill. In CTArcade (R08) the player must identify and generalize the right pattern to solve a puzzle. In some stages of games like Kodetu (R32) and LightBot (R29) the player needs identify specific movement patterns in order to navigate through a wide area.

*Table 8. CT and programming concepts.*

| Concept | References |
|---|---|
| Basic programming (e.g. loops and conditions) | R01, R02, R03, R04, R06, R07, R08, R09, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37, R38, R39, R42, R43, R44 |
| Algorithmic thinking, pattern recognition | R03, R06, R07, R08, R09, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37, R39, R42, R43, R44 |
| Procedures, subroutines | R04, R10, R11, R17, R19, R22, R26, R29 |
| Debugging | R38, R42, R43, R44 |
| Programming language | R02, R31, R34, R36 |
| Recursion | R05 |

Moreover, *Table 8* summarizes the CT and programming concepts found in the 44 studies.

However, as mentioned in the background section of this chapter, the goal of teaching programming is problem-solving transfer, i.e., users are expected to be able to apply what they have learned to solve problems that they have not been taught. The common entity of the 44 studies included in this SLR, is that students learn how to think differently about problems and try to solve them though games.

## Additional Factors (RQ5)

This section presents extra factors that affect the learning process through playing digital games to learn programming and were not included in the previous research questions. There appears that some studies correlate programming learning through games, with personality traits (R02, R10). The way a student acts in her/his daily life may affect learning programming through games. In study R10, introvert students performed better than extravert in games by code.org, which makes sense seeing that introverts generally prefer a quiet environment and playing a game on a computer and writing code fits with that preference. However, this highlights the need to access more factors when designing games for CT education.

Moreover, a game (R32) seeks the element of creativity among with CT skills. The study seeks differences between creativity and critical thinking levels of students based on different games. It was found that CT skills display a statistically significant difference based on the type of game that students play. If creativity and the acquisition of CT is better explained as a personal trait, then teachers and game designers should personalize learning experiences by creativity and try promoting this important skill.

Lastly, the element of feedback is very important and necessary in educational games. Good feedback can be implemented by providing guidance to players within the game and detecting patterns from game analytics (R38). However, most studies here inject CT into traditional game mechanics. Likewise, most games do not facilitate learning during the gameplay experience. They only give feedback after a stage end. If the game provides support while the learner is engaged in the task, it will help the student reflect about his/her errors in the context they happen. Study R39 reports that there are no games to adapt their interaction according to students' evolution or that support learning during the gameplay experience. Games with multiple modalities, adaptive or personalized, based on real world sensory data may be more successful (Combefis et al., 2016).

## THREADS TO VALIDITY

Although our research is based on the recent and widely used guidelines of SLR (Kitchenham et al., 2015), some issues may threaten the validity of the proposed methodology. First of all, there may exist studies that were not included in our selected papers. Indeed, the search process to the digital libraries may not have returned all relevant works, since each one has specific features. Moreover, we considered full studies only, while other types of works may be presented to the scientific databases. Then, it is possible that relevant studies stayed outside the choice of this study.

Furthermore, human factors may also influence the search strategy since the data extraction is a manual process and it is possible that relevant information has not been considered or misinterpreted. In order to minimize this bias, all the selected studies were read multiple times. Moreover, following the general recommendations of SLR (Kitchenham et al., 2015), we used both manual and automatic search, accompanied by a snowballing process, where two researchers reviewed the studies and assessed the quality and extracting data. The inter-rater reliability between the two scholars for the total scores (as presented at section (Quality Assessment) was .97, showing a very good agreement concerning the quality of the papers. Any problems that occurred in the evaluation process of the selected studies were addressed on a regular basis between two scholars, resolved through agreement, and the subsequent decisions were correctly reported, as suggested by (Belur et al., 2018).

Finally, the research questions of this study require answers that are not binary. To find answers, a categorization of terms was used by different initial studies. Nevertheless, to characterize games for CT education was a difficult process since previous works do not have classic boundaries and may some terms be used by different authors with the same meaning.

## CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This chapter examines the literature on playing digital games and specifically serious games, for acquiring basic CT skills or learning basic programming concepts, in regard to the potential positive impacts of gaming on users of school and preschool education, especially with respect to learning, skills enhancement, motivation and engagement. We explored the influence of games through previous empirical researches. For that purpose, we analyzed 44 peer-reviewed articles, selected from searches of the literature over the past ten years, covering a wide range of approaches in the area of learning through games. The goal is twofold: on one side, to provide an overview of the work that has been done in this area, and on the other side, to discuss characteristics of digital games to improve CT education.

The review indicates that digital games used in CT education provide positive effects, however, the effects are greatly dependent on the context in which the learning is being implemented, as well as on the users using it. They can be the vehicle to attract more students in the essential CT education. Though, for their effective application we should consider several factors, both when designing them and when using them and the area needs further research.

The present findings may be useful in creating educational programming games. They provide insight of how students can be creative problem-solvers through games. Moreover, this review may be useful to teachers that are considering using a game-based approach in their classrooms or policymakers that seek ways of implementing CT education into the school curriculum. It can also help further studies with digital games for academics and educators. Finally, it can enable curriculum designers with various interests to advance their knowledge base in this fresh yet important field. Digital games can reinvent CS in schools by motivating and educating all students including girls and underrepresented populations to acquire the essential CT skills. Yet, it is a challenge to see which other CS areas could benefit from incorporating the digital games approach.

## REFERENCES

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010). Using scalable game design to teach computer science from middle school to graduate school. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, 224–228. 10.1145/1822090.1822154

Belur, J., Tompson, L., Thornton, A., & Simon, M. (2018). Interrater reliability in systematic review methodology: Exploring variation in coder decision-making. *Sociological Methods & Research*.

Bers, M. U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, *12*(2), n2.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834–860. doi:10.3102/0034654317710096

Combefis, S., Beresnevičius, G., & Dagienė, V. (2016). Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, *10*(1), 39–60. doi:10.15388/ioi.2016.03

Connolly, T., Stansfield, M., & Hainey, T. (2009). Towards the development of a games-based learning evaluation framework. In *Games-based learning advancements for multi-sensory human computer interfaces: Techniques and effective practices* (pp. 251–273). IGI Global. doi:10.4018/978-1-60566-360-9.ch015

Eordanidis, S., Gee, E., & Carmichael, G. (2017). The effectiveness of pairing analog and digital games to teach computer science principles to female youth. *Journal of Computing Sciences in Colleges*, *32*(3), 12–19.

Ershov, A. P. (1981). *Programming, the second literacy*. Academic Press.

Frazer, A., Recio, A., Gilbert, L., & Wills, G. (2014). Profiling the educational value of computer games. *Interaction Design & Architecture (s) Journal-IxD&A, 19*, 1–19.

Grover, S., & Basu, S. (2017). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 267–272. 10.1145/3017680.3017723

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, *8*(6), 1–165. doi:10.2200/S00684ED1V01Y201511HCI033

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, *50*(4), 313–334. doi:10.1080/00461520.2015.1124022 PMID:27019536

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia: Social and Behavioral Sciences*, *47*, 1991–1999. doi:10.1016/j.sbspro.2012.06.938

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522–531. doi:10.1016/j.procs.2012.04.056

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews* (Vol. 4). CRC Press. doi:10.1201/b19467

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. doi:10.1145/1929887.1929902

Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, *50*(4), 1979–1995. doi:10.1111/bjet.12685

Lockwood, J., & Mooney, A. (2017). *Computational Thinking in Education: Where does it fit? A systematic literary review.* ArXiv Preprint ArXiv:1703.07659

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. doi:10.1016/j.chb.2014.09.012

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Educational games for teaching computer programming. In *Research on e-Learning and ICT in Education* (pp. 87–98). Springer.

Rogers, S. (2014). *Level Up! The guide to great video game design*. John Wiley & Sons.

Rotherham, A. J., & Willingham, D. T. (2010). 21st-Century" skills. *American Educator*, *17*(1), 17–20.

Shaffer, D. W. (2006). *How computer games help children learn*. Macmillan. doi:10.1057/9780230601994

Shah, M., Foster, A., & Barany, A. (2017). *Facilitating learning as identity change through game-based learning. In Game-Based Learning: Theory, Strategies and Performance Outcomes (pp. 257-278)*. New York, NY: Nova Publishers.

Shute, V. J., & Ke, F. (2012). Games, learning, and assessment. In *Assessment in game-based learning* (pp. 43–58). Springer. doi:10.1007/978-1-4614-3546-4_4

Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2017). How do different cognitive styles affect learning programming? Insights from a game-based approach in Greek schools. *ACM Transactions on Computing Education*, *17*(1), 3. doi:10.1145/2940330

Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 1–*7*.

Wassila, D., & Tahar, B. (2012). Using serious game to simplify algorithm learning. *International Conference on Education and E-Learning Innovations*, 1–5. 10.1109/ICEELI.2012.6360569

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Winslow, L. E. (1996). Programming pedagogy—A psychological overview. *ACM Sigcse Bulletin*, *28*(3), 17–22. doi:10.1145/234867.234872

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, *141*, 103633. doi:10.1016/j.compedu.2019.103633

## ADDITIONAL READING

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011, March). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 245-250).

Ch'ng, S. I., Low, Y. C., Lee, Y. L., Chia, W. C., & Yeong, L. S. (2019). Video Games: A Potential Vehicle for Teaching Computational Thinking. In *Computational Thinking Education* (pp. 247–260). Singapore: Springer. doi:10.1007/978-981-13-6528-7_14

Curzon, P., & McOwan, P. W. (2017). Power Of Computational Thinking. *The: Games, Magic And Puzzles To Help You Become A Computational Thinker/WSPC (Eu¬ rope).*

Grover, S. (2019, February). Thinking about Computational Thinking: Lessons from Education Research. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1283-1283). 10.1145/3287324.3293763

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522–531. doi:10.1016/j.procs.2012.04.056

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Educational games for teaching computer programming. In *Research on e-Learning and ICT in Education* (pp. 87–98). New York, NY: Springer.

Prensky, M. (2003). Digital game-based learning. *Computers in Entertainment*, *1*(1), 21–21. doi:10.1145/950566.950596

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. doi:10.1016/j.edurev.2017.09.003

Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014, October). A review of games designed to improve introductory computer programming competencies. In 2014 IEEE frontiers in education conference (FIE) proceedings (pp. 1-7). IEEE. doi:10.1109/FIE.2014.7044114

## KEY TERMS AND DEFINITIONS

**Computational Thinking Education:** The role of computational ideas as essential part of the problem-solving process and the 21st century learning.

**Computer Game:** A computer activity where user interacts with a machine in a playful way.

**Game-Based Learning:** A method to teach and learn with the use of gaming activities.

## APPENDIX

### Selected Papers

[R01] Boyce, A. K., Campbell, A., Pickford, S., Culler, D. & Barnes, T. Experimental evaluation of BeadLoom game: how adding game elements to an educational tool improves motivation and learning. in *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* 243–247 (ACM, 2011).

[R02] Kane, I., Pham, C.-T., Lewis, A. W. & Miller, V. Escape from the Python's Den: An Educational Game for Teaching Programming to Younger Students. in *Proceedings of the 2019 ACM Southeast Conference* 279–280 (ACM, 2019).

[R03] Eordanidis, S., Gee, E. & Carmichael, G. The effectiveness of pairing analog and digital games to teach computer science principles to female youth. *Journal of Computing Sciences in Colleges* **32**, 12–19 (2017).

[R04] Horn, B. *et al.* Design insights into the creation and evaluation of a computer science educational game. in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* 576–581 (ACM, 2016).

[R05] Tessler, J., Beth, B. & Lin, C. Using cargo-bot to provide contextualized learning of recursion. in *Proceedings of the ninth annual international ACM conference on International computing education research* 161–168 (ACM, 2013).

[R06] Koracharkornradt, C. Tuk Tuk: A Block-Based Programming Game. in *Proceedings of the 2017 Conference on Interaction Design and Children* 725–728 (ACM, 2017).

[R07] Rowe, E., Asbell-Clarke, J., Gasca, S. & Cunningham, K. Assessing implicit computational thinking in zoombinis gameplay. in *Proceedings of the 12th International Conference on the Foundations of Digital Games* 45 (ACM, 2017).

[R08] Lee, T. Y., Mauriello, M. L., Ahn, J. & Bederson, B. B. CTArcade: Computational thinking with games in school age children. *International Journal of Child-Computer Interaction* **2**, 26–33 (2014).

[R09] Zeid, A., Al-Mirza, G. & Al-Meshwah, L. SLG: an online educational simulation game to teach programming concepts. in *Proceedings of the Second Kuwait Conference on e-Services and e-Systems* 15 (ACM, 2011).

[R10] Theodoropoulos, A., Antoniou, A. & Lepouras, G. How do different cognitive styles affect learning programming? Insights from a game-based approach in Greek schools. *ACM Transactions on Computing Education (TOCE)* **17**, 3 (2017).

[R11] Marco, J., Bonillo, C. & Cerezo, E. A tangible interactive space odyssey to support children learning of computer programming. in *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* 300–305 (ACM, 2017).

[R12] Bromwich, K., Masoodian, M. & Rogers, B. Crossing the game threshold: A system for teaching basic programming constructs. in *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction* 56–63 (ACM, 2012).

[R13] AlSulaiman, S. & Horn, M. S. Peter the fashionista?: Computer programming games and gender oriented cultural forms. in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* 185–195 (ACM, 2015).

[R14] Buffum, P. S. *et al.* Empowering All Students: Closing the CS Confidence Gap with an In-School Initiative for Middle School Students. in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* 382–387 (ACM, 2016).

[R15] Zhi, R., Lytle, N. & Price, T. W. Exploring Instructional Support Design in an Educational Game for K-12 Computing Education. in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* 747–752 (ACM, 2018).

[R16] Khenissi, M. A., Essalmi, F. & Jemni, M. A learning version of Pacman game. in *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)* 1–3 (IEEE, 2013).

[R17] Silla, C. N. *et al.* Girls, Music and Computer Science. in *2018 IEEE Frontiers in Education Conference (FIE)* 1–6 (IEEE, 2018).

[R18] Paliokas, I., Arapidis, C. & Mpimpitsos, M. Playlogo 3d: A 3d interactive video game for early programming education: Let logo be a game. in *2011 Third International Conference on Games and Virtual Worlds for Serious Applications* 24–31 (IEEE, 2011).

[R19] Giannakoulas, A. & Xinogalos, S. Current Trends in On-line Games for Teaching Programming Concepts to Primary School Students. in *International Conference on Technology and Innovation in Learning, Teaching and Education* 62–78 (Springer, 2018).

[R20] Lehat, M. L., Mokhtar, R., Sokman, Y., Ismail, M. I. & Basir, N. M. Games: An approach to introduce computer programming for upper primary school students. in *2014 3rd International Conference on User Science and Engineering (i-USEr)* 85–90 (IEEE, 2014).

[R21] Beheshti, E. & Horn, M. S. Work in progress: Learning flow-of-control with FlipLogic: A game-based approach. in *2012 Frontiers in Education Conference Proceedings* 1–2 (IEEE, 2012).

[R22] Williams, C., Alafghani, E., Daley, A., Gregory, K. & Rydzewski, M. Teaching programming concepts to elementary students. in *2015 IEEE Frontiers in Education Conference (FIE)* 1–9 (IEEE, 2015).

[R23] Yuliana, I., Octavia, L. P., Sudarmilah, E. & Matahari, M. Introducing Computational Thinking Concept Learning in Building Cognitive Capacity and Character for Elementary Student. in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)* 549–554 (IEEE, 2019).

[R24] Akbar, M. *et al.* Sol y Agua: A Game-based Learning Platform to Engage Middle-school Students in STEM. in *2018 IEEE Frontiers in Education Conference (FIE)* 1–9 (IEEE, 2018).

[R25] Zhao, W. & Shute, V. J. Can playing a video game foster computational thinking skills? *Computers & Education* **141**, 103633 (2019).

[R26] Gomes, T. C. S., Falcão, T. P. & Tedesco, P. C. de A. R. Exploring an approach based on digital games for teaching programming concepts to young children. *International journal of child-computer interaction* **16**, 77–84 (2018).

[R27] Asad, K., Tibi, M. & Raiyn, J. Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World journal of education* **6**, 20–26 (2016).

[R28] Karsenti, T. & Bugmann, J. Exploring the Educational Potential of Minecraft: The Case of 118 Elementary-School Students. *International Association for Development of the Information Society* (2017).

[R29] Yallihep, M. & Kutlu, B. Mobile serious games: Effects on students' understanding of programming concepts and attitudes towards information technology. *Education and Information Technologies* 1–18 (2019).

[R30] Çiftci, S. & Bildiren, A. The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer Science Education* 1–19 (2019) doi:10.1080/08993408.2019.1696169.

[R31] Steinmaurer, A., Pirker, J. & Gütl, C. sCool-Game Based Learning in STEM Education: A Case Study in Secondary Education. in *International Conference on Interactive Collaborative Learning* 614–625 (Springer, 2018).

[R32] Hershkovitz, A. *et al.* Creativity in the acquisition of computational thinking. *Interactive Learning Environments* 1–17 (2019).

[R33] Vlahu-Gjorgievska, E., Videnovik, M. & Trajkovik, V. Computational Thinking and Coding Subject in Primary Schools: Methodological Approach Based on Alternative Cooperative and Individual Learning Cycles. in *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* 77–83 (IEEE, 2018).

[R34] Sideris, G. & Xinogalos, S. PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming With Python. *Simulation & Gaming* 1046878119872797 (2019).

[R35] Krajcsi, A., Csapodi, C. & Stettner, E. Algotaurus: an educational computer programming game for beginners. *Interactive Learning Environments* 1–14 (2019).

[R36] Galgouranas, S. & Xinogalos, S. jAVANT-GARDE: A cross-platform serious game for an introduction to programming with java. *Simulation & Gaming* **49**, 751–767 (2018).

[R37] Bauer, A., Butler, E. & Popović, Z. Approaches for teaching computational thinking strategies in an educational game: A position paper. in *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)* 121–123 (IEEE, 2015).

[R38] Liu, Z., Zhi, R., Hicks, A. & Barnes, T. Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education* **27**, 1–29 (2017).

[R39] Kazimoglu, C., Kiernan, M., Bacon, L. & Mackinnon, L. A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioral Sciences* **47**, 1991–1999 (2012).

[R40] Vahldick, A., Mendes, A. J. & Marcelino, M. J. A review of games designed to improve introductory computer programming competencies. in *2014 IEEE frontiers in education conference (FIE) proceedings* 1–7 (IEEE, 2014).

[R41] Kazimoglu, C., Kiernan, M., Bacon, L. & MacKinnon, L. Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science* **9**, 522–531 (2012).

[R42] Lee, M. J. & Ko, A. J. Personifying programming tool feedback improves novice programmers' learning. in *Proceedings of the seventh international workshop on Computing education research* 109–116 (ACM, 2011).

[R43] Lee, M. J., Ko, A. J. & Kwan, I. In-game assessments increase novice programmers' engagement and level completion speed. in *Proceedings of the ninth annual international ACM conference on International computing education research* 153–160 (ACM, 2013).

[R44] Lee, M. J. *et al.* Principles of a debugging-first puzzle game for computing education. in *2014 IEEE symposium on visual languages and human-centric computing (VL/HCC)* 57–64 (IEEE, 2014).

# Chapter 8
# The Role of Unplugged Coding Activity in Developing Computational Thinking in Ages 6-11

**Gaia Lombardi**
*Istituto Comprensivo Statale Via dei Salici, Legnano, Italy*

## ABSTRACT

*Coding is a spreading teaching methodology that is involving more students and teachers all over the world. But how can the practice of coding affect the development of computational thinking strategies in early years? The author, a primary school teacher, will investigate the Italian experience, believing that it may constitute an excellent field of study on the matter thanks to the enormous enthusiasm with which coding was received by the teachers, capable of renewing their teaching practices, particularly in primary school. This is a movement born from below, from the spontaneous participation of teachers, and which, in many cases, has been substantiated in what can be defined as unplugged activities, without the use of electronic technological tools.*

## INTRODUCTION

Coding and Computational Thinking are with no doubts the new keywords of teaching (Bocconi et el. 2016) but what is the true meaning of these words, and what are the differences of substance between the two terms? Computational Thinking is nowadays strongly argued to be a crucial skill for the 21st Century students, the so called "digital natives", -but, as a matter of fact, a generation of less or more good users of a pervasive digital technology (Bell & Roberts, 2016). However, let's start by pointing out that despite the fact that the role and significance of computing has increased in society and the economy (Wilson, Sudol, Stephenson, & Stehlik, 2010), student motivation to enroll in computing fields is in decline (Karakus, Uludag, Guler, Turner, & Ugur, 2012).

DOI: 10.4018/978-1-7998-4576-8.ch008

According to the studies of Jeannette Wing in 2006 (Wing, 2006), Computational Thinking can be defined as "the process of solving problems that allows to face and solve a problem in a procedural way" through a sequence of ordered steps that can be generalized to any similar situation (Brennan, & Resnick, 2012). The procedure thus identified takes the name of "algorithm", and its strength in its reproducibility through instructions so simple and unambiguous that they can be dictated to what is called an "ideal performer", human or machine it be.

From this point of view, Computational Thinking is a way of conceptualizing: thinking like a computer and describing a way of thinking at multiple levels of abstraction (del Olmo-Muñoz, Cózar-Gutiérrez, & González-Calero, 2020). In that sense, Computational Thinking can be defined also as a set of specific cognitive skills and problem-solving processes that allows the student to reformulate difficult problems into one, already known, that he can solve. To set an algorithm, the student also needs to then recursively, in a procedural way (Lee & Junoh, 2019; Kanaki & Kalogiannakis, 2018; Kalogiannakis & Kanaki, 2020).

Following from that, it appears with strong evidence that Computational Thinking is a skill everyone should possess, be it child, teenager or adult; not only what are defined as "millennials," who, both for generational and social reasons, could no longer disregard the use (and therefore the knowledge itself) of technological devices, but also their fathers, even the most obstinately "digital immigrants". Computational thinking is the "forma mentis" of the 21st century. In this social and cultural perspective, computational thinking is a universal basic skill, that should be even taught at school (Vidakis et al., 2019).

So, what is the relationship between computational thinking and coding? How are they interconnected and what role does each of them play in the school? Computational Thinking can be learned -and even mastered- from childhood, and the probably best way to do it is to develop it through the principles of coding. Coding is the training ground (or the playground...) in which to develop computational thinking skills. Through coding, in fact, children can find an immediate and tangible application to theoretical principles of Computational Thinking (Sands, Yadav, & Good, 2018).

From an educational point of view, the significant qualities of coding are that coding allows children to learn by doing; to collaborate in founding creative solutions to problems and to share their work; to have fun while practicing and to learn through fun activity; finally, coding is the main way to approach basic programming languages (Saxena, Lo, Hew, Hew, & Wong, 2020).

In this sense, coding can be considered not a mere school subject to study (and to add to an already very rich curriculum), but a didactic methodology in all respects (Kalogiannakis, 2008; Kalogiannakis & Papadakis, 2007; 2008).

Usually, when thinking about coding activities, the visual block programming carried out with Scratch, MIT's teaching platform, or with code.org immediately springs to mind; at a more advanced level, the use of Blocky, or the whole field of educational robotics and tinkering (Yadav, Hong & Stephenson, 2016). However, there is a huge amount of coding activities defined as unplugged, that is, which do not use digital tools, but which have the purpose of initiating or developing computational thinking in pupils of all ages. These activities, their description, and how they can be used in school are the subject of this Chapter.

## BACKGROUND: COGNITIVE THEORIES FROM PIAGET TO PAPERT

To understand the profound pedagogical, educational, and didactic meaning of the introduction of computational thinking and coding deeply and completely in the school, it is necessary to make a quick -and probably non-exhaustive- survey of the theoretical substrate of this choice (Papadakis & Kalogiannakis, 2010; 2018). Theoretical substrate that cannot but start from a description of the cognitive functioning of the child, in that we define with the adjective "computational" what is, in all the respects, a form of thought, therefore of mental organization in the face of the need to solve a problem (Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014).

First, it must be remembered that, between the ages of 3 and 11, children learn through their body, and this is as true as the age of the children involved is younger. For the Swiss psychologist and educationalist Piaget (Piaget & Cook, 1952) at this age the child is first in the preoperational stage (3-6 years) and then (6-11 years) in the concrete operational stage: children can think logically and even understand reversibility, but are limited to what they can physically manipulate. That is, children must touch, experiment, exercise all their senses and be involved with the whole body to acquire a lasting learning (Papadakis, Tousia & Polychronaki, 2018).

According to Piaget, from the of age 11 to 16 and onwards, children develop their abstract thought. Children are now able to think abstractly and utilize metacognition. It means that children can effectively access the conceptualization of the different subjects and of their specific abstract contents.

Going back to the concrete operational stage, is to remember how, for the teachers of foreign languages, Total Physical Response is the privileged way to access significantly (and having fun), the learning of language. Also, in the - sometimes hard- field of mathematics, it is now possible to detect how, with activities of movement and play, children can structure concepts such as numerosity, one-to-one correspondence, set of elements, spatiality, perimeter and area and geometric shapes.

In the first years of primary school, the exercise of the mind takes place through the exercise of the body, the finalized movement, the sensory experience. In this way the children keeps active all those that Gardner calls multiple intelligences (4); they can use all the experiential channels available to them and achieve a learning in which knowledge is interrelated and, guided by the specific abilities of the individual, begin to build a network of skills. In primary schools the keyword is interdisciplinarity, or the connection between the different fields of knowledge around the same object of learning.

In the 1970s Piaget's theories were collected and developed by the American mathematician Seymour Papert. It is in fact to Papert that we owe the first studies about Computational Thinking and coding, and the birth of Logo, the first useful environment to do programming aimed at cognitive development. Papert's work began in the 1960s at the MIT labs in Boston as a mathematician and computer scientist, but he soon became interested in the studies of Piaget. Papert's theories start from Piaget's ones, but go in some respects beyond from the constructivism of the Swiss pedagogist, Papert thus pass to an open "constructionism". Papert's constructionism is based precisely on the ability to build functional cognitive models and reuse them in similar contexts. It is what the programmer does when he tries to solve a computer problem by describing an algorithm that maybe reduces it to one or more problems whose solution is known.

According to Papert, constructionism is a word with two faces: one refers to the constructivist theory and to Piaget, which considers learning as a reconstruction and not as a mere transmission of knowledge (educationism); the other face extends the concept of manipulative materials by affirming that construction and therefore learning is more effective and mastered when it is not only mental, but is supported by real construction, by an activity such as the construction of a meaningful project. Papert calls this form of mental construction "concrete thinking". In Piaget's epistemological vision, this was only an "intermediate" stage, while in the approach proposed by Papert it becomes the protagonist of learning, a brand new kind of learning defined "syntonic" and based on three principles: continuity with the subject's previous experiences and knowledge; power to carry out personal meaningful projects; "cultural resonance" of the knowledge to be learned.

Papert's constructionism claims that learning is more efficient and profitable if it takes place through the production, by the learner, of concrete and real objects: what he calls "cognitive artifacts". The basic idea is that the mind needs to build objects and devices, to handle real materials in order to learn and generate ideas. In these attempts to represent the world around us, the learner proceeds by trial and error; learning develops through discussion, analysis, comparison, exposure and through the construction, disassembly and reconstruction of the cognitive artifacts.

In constructionism theory, the school is considered as a place of construction and not of knowledge transmission, and the computer becomes a learning tool that allows students to form their own knowledge and ideas in an active and participatory process. On the assumption that the child's mind is not a jar to be filled, Papert elaborates this simple question: "why don't we teach them to think, learn and play? (Papert, 1980)

"Some of the most crucial stages of mental development are based not on simply acquiring new skills, but on acquiring new administrative methods to use what is already known ... because a mind cannot grow much if it merely accumulates knowledge, but must also invent ways to make the most of the knowledge already possessed" (Papert, 1986).

Papert considers the computer as a tool to simulate real life and as a support for learning, even for very young children. On this idea, he realizes LOGO: a programming language understandable and usable even by elementary school children. LOGO is a tool that gives children -and anyone else- the full control over the computer, allowing them to obtain, based on mathematical and logical principles, quick and concrete results: drawings, music, poems.

In such a learning environment, the teacher naturally abandons his central role in the learning-teaching process and takes on that a promoter of the activities, in which children plan and learn by communicating and sharing their ideas. In the class, everyone learns from the other. In the interaction that is thus created, in which brainstorming and the exchange of ideas flow freely and continuously, some ideas may be proved valid, others a little less, of course, but each idea has the same dignity and can be discussed and tested by the learners. Children are independent and become responsible for their own learning; they can discover for themselves the knowledge they need. In Papert's theory, the computer then becomes a means of learning: not only a technological means, but an instrument to break down the wall between humanistic and scientific culture. The use of the computer as a learning tool has as its objective a form of active learning:

"The child should program the computer and not the computer program the child", Papert affirmed.

In this perspective, the computer is not merely a machine with which to process information, but a tool for building, learning, discovering and making mistakes. The error has nothing negative, but it is a constructive component of the learning process. Making mistakes means exploring in search of alternative solutions to the problem. The pupils learn by becoming aware of the mistakes. In such a vision of the learning process, the distinction between right and wrong loses its importance in terms of verification and evaluation, and the error is not considered as a block on the path, to be removed, corrected, and forgotten. Error correction is part of the understanding process and errors allow you to understand what happened and what went wrong. In this gradually approaching the goal proceeding by trial and error correction, the experience of programming (coding) proves to be an excellent tool for learning.

## RAFTING THE SITUATION IN THE ITALIAN SCHOOL

From a European Schoolnet survey (European Schoolnet, 2019) referring to the G20 countries, resulted that in 13 European countries information technology has been introduced or is about to be introduced in compulsory education. In 7 European countries, programming is already becoming an integral part of the school curriculum, for example Estonia, Greece and England.

Coding and computational thinking entered the Italian school in 2014 with the first projects held by the Ministry of Public Education through the site “Programma il futuro” (Program the Future), in partnership with Consorzio Cini. “Programma il futuro” partly resumed the visual block programming activities proposed by code.org and was devote to a basic alphabetization of the teachers, first, and of the students as a consequence; but the real didactic interest on this matter started with the MOOC Coding and Computational Thinking held by Professor Alessandro Bogliolo, Associate Professor of Information Processing Systems from the University of Urbino and Codeweek ambassador for Italy.

Starting from this first training, which met the interest of over 1500 Italian teachers of each school order, a vast learning and sharing community was set up and gave rise to initiatives such as the code-mooc.net site, where the teachers share and find resources for their lessons. The University of Urbino has taken up the need for continuous teacher training by proposing, on an annual basis, other MOOCs about algorithms, apps, culture and, from 2016, weeks of immersive Summer schools in which teachers can get to know each other, interact, expand their network of professional relationships. Italy has thus become the European country in which the largest participation in the EU Codeweek literacy week is counted, with thousands of events involving, in a festive atmosphere, pupils from kindergarten to primary and secondary schools, providing also open activities dedicated to the public.

The wave of enthusiasm shown by the teachers and the vitality of the projects and activities conceived, created and realized in schools of all kinds, all over the country, in the last five years lead to saying that coding was the electric shock that the Italian school was waiting for.

After at least two decades in which, due to the continuous solicitations by ministers who followed one another too quickly, there were no signs that proclamations, circulars, announcements, unfinished projects, coding awakened the desire to do and stimulated the creativity of the Italian teaching class, young not in age, but in soul.

The spread of coding practices in the Italian school is a movement of ideas really born and developed from below, from the autonomous initiatives of willing, creative teachers, eager to renew their teaching practice and to find the roots of their knowledge and their know how.

Coding soon became one of the many tools that Italian teachers collect in their toolboxes, perfectly combining with other innovative didactic suggestions such as the flipped classroom, the peer learning and, on the technological side, the massive introduction of IWB (interactive snowboard) into classes.

What importance to attribute to the Italian experience, and why make it an object of observation in a chapter of this volume? The Italian one is in fact a type of didactic experience which, as it was born from below, from the interest and participation of the teachers, has not yet been the subject of important scientific or academic studies, and the documentation of the courses and its effectiveness are still completely informal, mostly entrusted to the fulfillments pertinent to the end of the school year in the various institutes, or at least to articles, in specialized magazines or online; to conferences and sector events; to lectures and webinars that contribute to enriching the self-training course of the teachers themselves.

Talking about coding in Italian school today means entering a variegated world of good practices spontaneously (and courageously) shared by teachers who, in big cities as in small countries all around the Peninsula, add every day a piece to the great mosaic of freely shared activities.

This is particularly true with regard to the Primary School which, until the Eighties considered one of the best in the world for its teaching methods and its inclusion practices, was slipping into the repetitiveness of established schemes that left less and less autonomy to the pupils, in a trend widespread to the formula lectures + exercise + homework. Coding has entered into this repetitive mechanism unhinging from the foundations, restoring vitality to the desire to teach, restoring to the students, even in daily practice, their central role in the learning process.

Above all, coding immediately showed its potential in terms of a tool for real inclusion for students from the Countries (whose number has increased exponentially in recent years, as a consequence of the continuous immigration flows that have involved Italy and the whole of Europe) and of students with physical and mental disabilities, or other educational special needs, as well as for the many marginalized children suddenly finding themselves in a growing situation of poverty following the global economic crisis.

Another pivotal factor for the success of coding in the Italian school (and, let's remember, especially in primary school, with children between 6 and 11 years) was its multidisciplinary and its interdisciplinary nature, the possibility of being applied to all disciplines, not only the scientific ones, in transversal projects involving different fields of study. CLIL and coding is one of the most successful associations in teaching English at all school levels, often in connection with Science, Arts or Music contents.

Coding principles are often proposed through games, and this make coding especially suitable even for kindergarten. This is one -but not the most important- reason why the majority of coding activities for the age group 3-11 are proposed in an unplugged mode. It is not just a matter of mere issues about poor technological equipment: as already mentioned, the Italian Primary school (and in many classes the public kindergartens, too) has been the subject, in the last 10 years, of a massive infrastructural endowment including the introduction of IWB in every school, the start of classes 2.0 in experimental form, the introduction of the electronic register for evaluation and the administrative component. The choice to propose coding through unplugged activities and with a strong component of gaming responds to precise methodological reflections on learning in children from 3 to 11 years, and is grafted onto a tradition of active school, of research-action that has among the his noble founders the great Montessori methodological lesson.

## UNPLUGGED TOOLS FOR CODING: DESCRIPTION AND ROLE

In this part of the chapter we will present the most known unplugged activities, briefly describing each of them, exemplifying their main purpose, the possible implementations and the links with curricular disciplines. If coding is intended as a methodology and not as a curriculum content, in fact, it has a pivotal interdisciplinary value, as well as a transdisciplinary one, and its use is not the sole responsibility of the mathematics or technology teacher.

Unplugged activities show their usefulness in particular for younger pupils, or for this with special educational needs, but, although they are often proposed as a game, they help to understand, develop and convey very complex concepts both in computer science and mathematics, besides constituting an effective methodological substrate for the grafting of contents and concepts pertaining to all disciplinary fields.

In this context, therefore, we are allowed to broaden a little the field of study, from the group 6-11 years to a wider 3-14, above all in consideration of the fact that the main field of observation, the Italian school, considers Primary and First Level of Secondary education as a single school cycle, in which many institutions move on a vertical curriculum for the various disciplinary areas, in a perspective of continuity through the pupils' developing are.

Let's now enter the heart of the topic that interests this Chapter: what unplugged activities to propose for the different age groups and how to organize them?

First of all, it should be remembered that, under the definition of unplugged activities, very different methods of approaching coding are collected, from playful activities, to those structured in traditional lessons; from production of small artifacts, to storytelling; from the knowledge of binary calculation and hexadecimal code, to their use to realise spectacular drawings, pictures and decorations; to encryption and its secret codes. In fact, the definition "unplugged" involves the idea of approaching coding not necessary through the use of technological means such as PCs, tablets, IWBs, programmable objects; even if, often, the activities proposed in an unplugged way find their completion in the proper use of digital tools. The proposal of unplugged activities for the development of computational thinking does not derive, in most cases, from the lack of adequate technological equipment, or from the lack of familiarity with it, but from a precise methodological and didactic choice, based on a solid substrate theoretical already reported at the beginning of the chapter.

As the reader will see in the following paragraphs, the variety of unplugged proposals not only covers a very wide age spectrum, from 3 to 14/16 years, but has the unquestionable advantage of being able to be applied in a transversal way to all disciplines and all subjects of study, thus constituting a "background noise" in the planning of learning units. The extreme flexibility of unplugged coding also makes a very effective inclusion tool for pupils with special needs.

The purpose of this chapter consists precisely in describing some of these activities, thus making them available, through the reference links, to anyone wishing to use them, and showing their effectiveness in everyday school life.

## Pixel art

The so-called "pixel art" is certainly the most welcome and pleasant among the unplugged activities of starting coding, accessible to children already from the kindergarten age; but it is certainly the most controversial, as the roles of programmer and executor seem to be not clearly defined and, in most cases, children find themselves to be nothing more than mere executor of a code (be it given in colors, letters, or, in a spectacular 3d version, with paper cups and colored tissues). Where the approach to coding is a start to a way of thinking and problem solving, the beautiful drawings made with pixel art often translate, instead, into mere reproductions of images on squares. Heir to the learning skills for embroidery (in particular half stitch and cross stitch), pixel art starts from a grid on which children must reproduce a drawing of which a code is provided. The name pixel art refers, as evident, to the pixels (minimum units of information) that make up the digitized images. Sometimes the code is directly written in the grid squares, as in those images for aspiring painters to be colored by following the correspondence number / color; more often pupils are provided with a separate sheet in which the colors are indicated with a letter. If the teacher wishes to use the activity in a context of teaching a foreign language, the colors can be indicated with the first letter of the word in the target language. The images thus created can then be used on the occasion of holidays for greeting cards; or as illustrations of a storytelling; or, in the case of particularly elaborate images, to accompany history or geography lessons (in the higher classes of primary school it is used to make maps or maps consistent with the geography program). It is easy to understand how, proposed in this way, pixel art images do not have a significant cognitive value, rather assigning pupils the task of passive executors of a set of instructions. The real transition to a deep understanding of the computational application of this activity occurs when, at first, pupils are provided with the image of which they, individually or, more often, in pairs, must receive the code and write it syntactically correct; and finally, when the pupils themselves design a drawing, write the code and realize it, also proceeding with the eventual debugging during the work. This is an activity that lends itself to multiple levels of conceptualization, from the simple execution of a code to the approach to the hexadecimal code, fundamental for bringing pupils closer to binary calculation when they are not yet in possession of the notion of exponentiation.

## CodyRoby

The most of unplugged activities are playful activities such as board games, that make use of a grid often realized on the floor using adhesive tape, or on large sheets of paper, or on other supports on which children can move themselves, or move puppets, toys, objects, or even program small robots (but…bit sound we still consider using small programmable tools as an unplugged activity? As already underlined, very often unplugged and unplugged merge in a same project).

In this type of activity, the instructions for movement are normally given by oriented arrows. Very popular for its completeness and ease of use is the instruction set provided by the CodyRoby cards. More than a simple card game with pieces to move on a board, CodyRoby is a real instruction set made up of cards of different colors (green for move forward, red for turn right, yellow for turn left), carrying the picture of an oriented arrow; repetition cards (blue), function cards (purple) and other items to set targets. In its structure, CodyRoby resume the Scratch instruction blocks: as well as the programming blocks, in fact, the cards are shaped with joints that suggest the possibility of being concatenated in a sequence or inserted in blocks of repetition or conditional instructions, in an effectively unplugged version of an authentic programming environment.

The set of cards is used in individual challenges, in pairs or in groups, in which a player is entrusted with the role of Cody, the programmer who, appropriately arranging the cards in rows and reading them from left to right, writes and reads the program, while the other player assumes the role of Roby, the more or less ideal performer who moves on the grid to reach the goal identified by the game. On the next game turn, the parts exchange, and the programmer becomes the executor and vice versa. This exchange of roles allows children to understand how digital tools are not "thinking" tools, but skilled, rapid, precise executors of algorithms written by a human mind. The real problem solver is Cody; Roby merely shows the correctness of the solution found and puts it into practice. The exchange of roles also makes debugging and the comparison of solutions more effective: when Roby switches to the role of Cody he can, with his own programming path, test the correctness and economy of that previously identified by his partner and, if necessary, propose a more efficient alternative. A so designed game set helps children to develop their ability to think and act in a procedural way, proceeding by trial and error, self-correcting and comparing with their classmates. The use of the cards on a grid, in addition to providing children with a sense of familiarity with the methods of other board games, allows them to have a full view of the playing field and of all the moves that are made, eliciting their participation, exchange and, where necessary, debugging; acting with the whole body in the execution of the instructions (the memorization of the instructions is facilitate through the association with the color). Having a grid and an instruction set, as a regular format for the games, teachers can fill them with contents, and adapt them to the development of multiple skills, to the learning of specific concepts, to the memorization of words, to the representation of stories.

The "natural evolution" of CodyRoby is, as the reader can easily understand, the access to visual block programming platforms such as Scratch, code.org, blocky; however, in this chapter we will only mention them, referring to further studies (Papadakis, Kalogiannakis, Orfanakis, & Zaranis, 2014; 2016; 2017; 2019; Kanaki & Kalogiannakis, 2018, Kalogiannakis & Kanaki, 2020) to deepening their use and impact on teaching and learning, as the purpose of the chapter is to present activities that can be carried out in a totally unplugged way, without directly requesting the use of technological devices.

## CodyFeet and CodyColor

CodyColor and CodyFeet are an evolution, or a filiation, of CodyRoby. Both elaborated, at least initially, by Italian teachers during a Summer School of Coding and Computational Thinking held in 2018 by Professor Bogliolo of the University of Studies of Urbino, they also are games that involve the use of cards of different colors.

In the CodyFeet set, as the name suggests, instead of oriented arrows, the movement instructions are provided by small feet, an icon immediately comprehensible even by very young children. The color for "move forward" is grey, and the cards can be placed directly on the floor, forming a path gradually oriented by the association between the color and the direction of the feet represented on the paper. The cards are square in shape and the instruction to be performed is not a simple instruction, as it involves both rotation and forward movement. This movement mode results more natural to perform for younger children. CodyFeet therefore proposes itself as the first game with oriented cards to be offered to children from three years onwards, to help them develop lateralization and discrimination between left and right.

CodyColor conceptually follows CodyFeet: square tiles are grey (move forward), yellow (turn left) and red (turn right); they can be arranged on a grid, usually 5 x 5, randomly. The cards contain a reading instruction (the color code) and a movement instruction (rotation and movement in the adjacent square). There are various challenge and game levels: find the entry point in the grid that allows you to stay as long as possible in the path, taking as many steps as possible; find the exit point, indicated by a letter or number, given the entry point; walk a given number of steps trying not to intersect the paths of other players, and so on. Also, in this case, having a double movement instruction on the card (rotation and step forward) results more natural to children, who read the color and then move on.

Compared to the previous game, CodyColor is on a slightly higher level of abstraction: there are no longer the small feet drawn to indicate tearing and the consequent displacement, but the player must rely on the instruction given by the color.

CodyColor had a digital evolution in 2019 thanks to researchers from the University of Urbino who created a web game version in which it is possible to challenge each other from a distance. In this mode, CodyColor cannot be counted among the unplugged activities, but this does not reduce its game potential for the development of algorithmic thinking.

## Pen and Paper Algorithms

It may seem useless to open a paragraph dedicated to algorithms where we talk about unplugged coding. Coding is in itself an activity that involves the use of instructions, their concatenation in procedures, therefore an algorithmic thought. However, for an algorithm to be effective, it must be syntactically correct, generalizable, executable without further explanation from what we could define as the "ideal performer" (a subject who performs its task without imagination, creativity, or interpretations). Once an effective algorithm has been identified, it is forever, like diamonds - for this reason, an international petition now proposing to insert the algorithms among the intangible heritage of humanity protected by UNESCO.

Computational thinking finds its formalization in the realization of an algorithm: what is an algorithm but a procedure for solving a problem? The most immediate way to represent an algorithm in an unplugged way is to write it with pen and paper, inserting it in the diagram of a flowchart. Very useful are the flowcharts used, since the early years of primary school, to record the procedures for performing simple routine actions such as brushing your teeth or hands, getting dressed (see the Dress code game), turning the computer on and off. ... In addition to a clear mathematical and computer science value, the use of flowcharts as a structure to represent procedurality is proving to be a useful cognitive tool for children with difficulties in organizing thinking or working memory, such as autistic children.

## Binary Cards and Binary Bracelets: Spelling the Same Language as the Computer

Can the teaching of the fundamentals of binary code be defined as an unplugged coding activity? Surely yes, as it is the profound language of digital tools. If it is true that programming, with which coding is often confused in a simplistic way, certainly does not use binary code for the realization of programs, it is equally true that coding not only trains computational thinking, but moves the pupil from the position as a mere user of digital devices, to provide him with the tools, above all cognitive, to develop creative and proactive interactivity with them. Therefore, it is essential to know "what language" computers speak, how they "think". Only by deeply knowing the language, the children and young people of this generation defined as "digital native" will be able to develop an attitude of active use towards digital devices that are now indispensable for daily life, and they will be able to play the role not only of communication tools, gaming, entertainment, but becoming truly effective tools for reading reality and, therefore, learning.

Even with the binary code you can play games and make small artifacts, at least in primary school: examples are the "binary cards" and the binary bracelets. Binary cards are cards with a black back, and, on the front, black dots in numbers corresponding to the powers of 2. By turning the cards in an appropriate order, it is possible to "write" the numbers in decimal basis, then transforming them into binary code considering the black back as 0 and the front as 1. The activity is, as often happens, simpler to carry out than to explain, and for children it is easily understandable: they immediately recognize, in fact, the positional correspondence between the decimal system and the binary system, and very quickly they appear able to move between the two counting systems. Once the operation of the binary system is understood, children can practically operate using the binary code as a code for numbers and letters, manipulating various objects and materials, such as, for example, pearls of two different colors, with which to make bracelets in which binary "write" the initial of your name, or the name of a friend, or mother. incredibly, mothers of nine-year old seem to love these "binary bracelets"!

On the conceptual line between the knowledge and use of the binary code and the already mentioned pixel art there are the images created on a squared grid using the hexadecimal code or the Cartesian coordinates. Also in this case, once the code is understood, the true computational meaning of the activity does not lie in the execution of drawings according to given instructions, but in the pupil's ability to write independently (or, as in didactic activities, it is always preferable, in collaboration with others) the code to realize an image, so precise that it can be made executable by others.

## Secret Codes: Jumping into History and Mystery

Talking about codes also means talking about cryptography. The use, but above all the invention, of secret codes, is one of the unplugged coding activities that best combines the playful and manipulative aspect with the understanding of fundamental mathematical and computer science concepts. Cryptography constitutes, especially in the primary school years, a fun and effective basis to start talking about codes, and to start using binary code as a "computer language", offering teachers the opportunity to link content to coding history of communications and history of technology. Cryptography can also offer ideas for the creation of very simple products, such as wheels or rotors for the encryption and decryption of coded messages. Starting from simple cryptography activities, students can be led to reflect on the human need to communicate, and consequently on the very meaning of language, to discover that what we call digital has very distant roots that sink into the primary irrepressible instinct of man to symbolically represent, from the beginning of history, the world around him.

On the pure didactical side, in addition to pixel art, cryptography completes an operational framework on the concept of code and on the separation of the programmer/ performer roles that can be mastered by children from the higher grades of Primary School, and does it in a fun and complete way, without falling into the trivialization of the concept. As a matter of fact, trivialization is one of the main risks in proposing playing activities in school to approach or develop contents and concepts: according to the intention of making them easily practicable by everyone, the formal rigor necessary to guarantee the correctness of the contents is sometimes loss of sight.

## Bruno Munari's Binary Tree: How Coding Meets art

Born in Milan in 1907, Bruno Munari was one of the greatest protagonists of Italian art, graphics and design in the early 20th century. His multifaceted research on the theme of movement, light and the development of creativity and imagination in childhood has brought his books and ideas into the professional heritage of many Italian teachers. In his book "Disegnare un albero" ("Drawing a tree": perhaps the most famous and well-known in primary schools) (Munari, 1977), Munari suggests trying to draw the branching of a tree starting from a simple rule: each branch could be divided into two new branches; then each of the two branches could in turn be divided into two other branches and so on, in a binary structure that immediately leads to recognizing the powers of 2. The Munari tree is therefore a binary tree, and visually reproduces the tree, or graph, of the decisions. It is mathematics, of course, but it is also art: art within reach of a child, which the child can manipulate with different techniques (drawing, painting, collage), while developing his ability to see mathematical regularities. Building a graph with which to represent one's ability to make decisions is a concrete and tangible application of fully computational thinking.

## Dress Code

To respond to the request for coding activities for the Christmas period, the Dress Code of Santa Claus was developed from CodyRoby: a cartoon Santa Claus puppet is equipped with all his well recognizable clothing, from a warm underwear to black boots, and can be dressed like the old paper dolls with which children (at least until the 70s) used to play.

The coding element consists of the instructions to dress the Santa Claus following a correct procedure, from wool knickers to a scarf and hat. An activity of this type is very suitable for younger children, in the age of kindergarten (3-6 years old) to develop their ability to dress and undress independently, and naturally constitutes an element of strong inclusion for disabled children, in particular for the management of daily routines for children with more or less serious forms of autism or Asperger's syndrome.

# ASSESSMENT AND EVALUATION OF CODING ACTIVITY: A STILL OPEN QUESTION

Coding activities are fully part of what is defined as " situation task", in which skills, knowledge and competence are highlighted and interconnected. The "situation task", or task of reality is not, by its nature, a mere exercise, but the problematizing approach to a situation as close as possible to reality. Problem solving skills, lateral thinking, knowledge of the elements of reality and the ability to read them Marshall part of this kind of task.

How to verify and evaluate coding activities, then? Can it be included among the tasks that can be assessed through a summative assessment, expressed with a number or a letter or any other codicological way? Or is it preferable to adopt a formative evaluation, during work, with continuous feedback and appropriate scaffolding to support learning? Certainly the second, being the coding activity already self-assessing: proceeding by trial and error, the student is already able, during the course of the activity, to correct himself; he is able to do, more properly given the theme of the chapter, debugging. In this process the student is helped by his peers: coding activities often take place in pairs, with a constant feedback from the peer group. The pupil gradually learns, from the success or failure of his own path, to assess first and then evaluate himself in an informal way, arguing the steps of his work, the difficulties encountered, the paths traveled in an attempt to solve the problem and the procedure definitively identified as resolving. The evaluation process, therefore, already takes place during the activity, and can usefully be reported by the teacher in a discursive form or with an evaluation section in which both the skills and the disciplinary knowledge are observed. Similarly, to what happens in CLIL, the latter (knowledge relating to curricular contents) can, if necessary, be assessed with specific separate tests.

# CONCLUSION

At the moment, there are an increasing account of international academic studies uniquely dedicated to investigate the role that unplugged activities have on the development of computational thinking: there is in fact no doubt that they are widely used strategies of approaching to coding, especially in kindergarten and primary school (Papadakis 2016; 2018). Their power lies in being activities well suited to multidisciplinarity and interdisciplinarity; furthermore, they are often proposed in a playful form, with involvement of the whole body and all senses of the child; by their very nature, they do not require important IT equipment or special digital skills on the part of teachers; last byt not east, they are “highly creative” activities, collaborative but at the same time challenging, simple but not trivial and therefore fully inclusive.

To conclude, we cannot fail to recognize, in regard to having structurally introduced computational thinking and coding in the school curriculum, the merit of having entered a context of lively openness to innovative, motivating and increasingly individualized teaching, alongside other teaching methods such as flipped classroom, peer learning, open classes, the use of interactive digital tools. An all-round teaching innovation that provides the child (and not only) with a rich variety of cognitive tools to become independent in the structuring of his learning path, so that this can be effectively inclusive, in a perspective of long-life learning.

## REFERENCES

Bell, T., & Roberts, J. (2016). Computational thinking is more about humans than computers. *I SET, 1*, 3-7. Doi:10.18296et.0030

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). *Developing computational thinking in compulsory education.* European Commission, JRC Science for Policy Report.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)* (vol. 1, pp. 1-25). Academic Press.

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832. doi:10.1016/j.compedu.2020.103832

European Schoolnet. (2019). *Survey of Schools: ICT in education. Final report*. European Commission, DG Communications Networks, Content & Technology.

Kalogiannakis, M. (2008). From Learning to Use ICT to Use ICT for Learning: Technological Capabilities and Pedagogical Principles. In R. Kobayashi (Ed.), *New Educational Technology* (pp. 13–42). New York: Nova Publishers.

Kalogiannakis, M., & Papadakis, St. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In C. Constantinou, Z. Zacharias & M. Papaevripidou (Eds.), *Proceedings of the 8th International Conference On Computer Based Learning in Science*, (pp. 265-276). Academic Press.

Kalogiannakis, M., & Papadakis, St. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. In *Proceeding of the 5th WSEAS/IASME International Conference on Engineering Education* (pp. 305-311). Academic Press.

Karakus, M., Uludag, S., Guler, E., Turner, S. W., & Ugur, A. (2012, June). Teaching computing and programming fundamentals via App Inventor for Android. In *2012 International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1-8). IEEE. 10.1109/ITHET.2012.6246020

Lee, J., & Junoh, J. (2019). Implementing Unplugged Coding Activities in Early Childhood Classrooms. *Early Childhood Education Journal*, *47*(6), 709–716. doi:10.100710643-019-00967-z

Munari, B. (1977). *Disegnare um albero*. Zanichelli.

Papadakis, S. (2016). Creativity and innovation in European education. 10 years eTwinning. Past, present and the future. *International Journal of Technology Enhanced Learning*, *8*(3/4), 279–296. doi:10.1504/IJTEL.2016.082315

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers? A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, St., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In M. Kalogiannakis, D. Stavrou & P.G. Michaelidis (Eds.), *Proceedings of the 7th International Conference on Hands-on Science*, (pp. 235-240). Academic Press.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed.), *Early Childhood Education* (pp. 1–19). London, UK: IntechOpen Limited. doi:10.5772/intechopen.81714

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In *Proceedings of the 2014 workshop on interaction design in educational environments* (pp. 1-7). New York: ACM.

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. In *Early Childhood Development: Concepts, Methodologies, Tools, and Applications* (pp. 797–819). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-7507-8.ch039

Papadakis, S., Kalogiannakis, M., Zaranis, N., & Orfanakis, V. (2016). Using Scratch and App Inventor for teaching introductory programming in secondary education. A case study. *International Journal of Technology Enhanced Learning*, *8*(3-4), 217–233. doi:10.1504/IJTEL.2016.082317

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Papert, S. (1986). Constructionism: A new opportunity for elementary science education. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.

Papert, S. (1990). *Children, computers and powerful ideas*. New York: Basic Books.

Piaget, J., & Cook, M. (1952). *The origins of intelligence in children*. New York: International Universities Press. doi:10.1037/11494-000

Sands, P., Yadav, A., & Good, J. (2018). Computational Thinking in K-12: In-service teacher perceptions of computational thinking. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines* (pp. 151–164). New York: Springer. doi:10.1007/978-3-319-93566-9_8

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66. doi:10.100740299-019-00478-w

Vidakis, N., Barianos, A.-K., Trampas, A.-M., Papadakis, St., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In B. McLaren, R. Reilly, S. Zvacek, & J. Uhomoibhi (Eds), *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (vol. 1, pp. 36-43). 10.5220/0007810800360043

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on empty: The failure to teach K-12 computer science in the digital age*. New York: ACM.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding a 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. doi:10.100711528-016-0087-7

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), 1–16. doi:10.1145/2576872

## ADDITIONAL READING

Kalogiannakis, M., Nirgianaki, G.-M., & Papadakis, St. (2018). Teaching magnetism to preschool children: The effectiveness of picture story reading. *Early Childhood Education Journal*, *46*(5), 535–546. doi:10.100710643-017-0884-4

Kalogiannakis, M., & Papadakis, St. (2017). Combining mobile technologies in environmental education: A Greek case study. *International Journal of Mobile Learning and Organisation*, *11*(2), 108–130. doi:10.1504/IJMLO.2017.084272

Kalogiannakis, M., & Papadakis, St. (2018). An Evaluation of Greek Educational Android Apps for Pre-Schoolers. In Finlayson, O., McLoughlin, E., Erduran, S., & Childs, P. (Eds.), Electronic Proceedings of the ESERA 2017 Conference. *Research, Practice and Collaboration in Science Education*, Part 4/Strand 4 (co-ed. Kalle Juuti & Eleni A. Kyza, Digital Resources for Science Teaching and Learning), (pp. 593-603). Dublin, Ireland: Dublin City University.

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital Student Conference Platform Implementation: The case study of the "Research Project" course. *Open Education*, *12*(2), 5–23.

Papadakis, S. (2018). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, St., Kalogiannakis, M., Sifaki, E., & Vidakis, N. (2018). Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University. *EAI (European Alliance for Innovation) Endorsed Transactions on Creative Technologies, 5*(16), e1, 1-12.

Papadakis, S. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

## KEY TERMS AND DEFINITIONS

**Educational Robotics:** Is an interdisciplinary learning environment based on the use of robots and electronic components to enhance the development of skills and competencies in students.

**ICT (Information and Communication Technology):** A term that covers all technical means used to handle information and aid communication, including software and hardware.

**Programming:** Lines of code that are written in a certain language that demands a logic of reasoning from the developers.

**STEM:** The term STEM (science, technology, engineering, and mathematics) is an acronym used by those relevant to the educational method concerning the fields of science, technology, engineering, and mathematics.

# Chapter 9
# Teaching Computational Thinking Unplugged:
## A Review of Tools and Methodologies

**Emmanouil Poulakis**
https://orcid.org/0000-0001-7687-0617
*University of Thessaly, Greece*

**Panagiotis Politis**
https://orcid.org/0000-0002-5017-0505
*University of Thessaly, Greece*

## ABSTRACT

*This chapter focuses on the "unplugged" approach for teaching computational thinking (CT), that is, teaching without the use of computers or digital equipment. After a short discussion of the different definitions of CT, the chapter presents the most well-known tools and methodologies of unplugged philosophy, with a connection to CT concepts. The chapter also summarizes the main advantages of the unplugged approach to CT education and furthermore, the most important design principles of unplugged, kinaesthetic activities. A separate section is dedicated to blended approaches of plugged and unplugged activities and the evaluation of unplugged approaches. While more large-scale implementations are still required to fully evaluate the benefits of unplugged approaches to CT education, existing studies report positive findings, especially in relation to the use of unplugged approaches for CT education. The majority of these resources are available for use by educators free of charge on the internet, which makes them very useful as a CT teaching approach.*

## INTRODUCTION TO COMPUTATIONAL THINKING

This chapter presents a review of unplugged approaches (teaching without use of computers or digital equipment) in teaching Computational Thinking (CT) in K-12. It consists of six parts: a) quick review of CT definition attempts, b) a report of basic design instructions in order to create unplugged activi-

DOI: 10.4018/978-1-7998-4576-8.ch009

ties, c) a meticulous report of unplugged tools and methodologies, where apart from the description of each tool, a connection of the tool with CT concepts is attempted, when applicable, d) a report of blended research approaches, using both unplugged and plugged activities, e) a report of advantages of using unplugged teaching resources and of research articles about the efficiency and effectiveness of unplugged approaches in teaching CT, and finally f) a report of evaluation approaches of unplugged activities. In conclusion, this chapter states some outcomes and future research directions, summarizing the data presented in its main section.

CT has been attracting increasing attention since Wing's (2006) first attempt to conceptualize and analytically describe the term. Papert had used the term back to 1980 (Papert, 1993) to refer to the change of thinking processes in mathematics' education due to the use of computers in education, without giving a detailed definition, while Grover and Pea (2013) place the origins of the idea of CT even earlier, to Alan Perlis' references in 1960s. A widely accepted definition of CT has not still been formed, but several researchers have tried to review existing definition attempts, concluding to some basic common characteristics and concepts of CT.

In 2010 Cuny, Snyder and Wing (2010, cited in Wing, 2011) proposed a definition of CT: *Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent*. The International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) in USA published an Operational Definition (ISTE & CSTA, 2011) for K-12 education which characterizes CT as *a problem-solving process*, which includes characteristics and dispositions or attitudes, based on feedback by survey from nearly 700 Computer Science teachers, researchers and practitioners. Main characteristics are the formulation of problems in a way that computers or other tools can help to solve them, the logical organization and analysis of data, the representation of data through abstractions (models, simulations), the automation of solutions through algorithmic thinking, the identification, analysis and implementation of possible solutions (achieving the most efficient and effective one) and generalization, transfer of this problem solving process to a wide variety of problems. Dispositions and attitudes include confidence in dealing with complexity, persistence in working with difficult problems, tolerance for ambiguity, the ability to deal with open ended problems and the ability to communicate and work with others to achieve a common goal or solution.

Barr and Stephenson (2011) also report some core concepts in the context of capabilities, such as design of solutions, implementation of designs, debugging, modeling, simulations, systems analysis, reflecting on practice and communicating, using the vocabulary, recognition of abstractions, innovation, exploration, and creativity across disciplines, group problem solving and employment of diverse learning strategies. Moreover they report dispositions and pre-dispositions, such as confidence in dealing with complexity, persistence in working with difficult problems, the ability to handle ambiguity, the ability to deal with open-ended problems, setting aside differences to work with others to achieve a common goal or solution and knowing one's strengths and weaknesses when working with others. They also give some examples of core CT concepts and capabilities, such as data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, parallelization and simulation, in different teaching subjects.

In UK, The Royal Society published in 2012 a report about computing in education and presented another definition of CT (The Royal Society, 2012): "*Computational thinking is the process of recognising aspects of computation in the world that surrounds us, and applying tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes*". A cooperation of Harvard University and EDC's Center for Children and Technology created Computational Thinking with Scratch, where one more definition attempt of CT took place (Computational Thinking with Scratch, 2012), involving three key dimensions: concepts, practices and perspectives. Aho (2012) also concluded to another CT definition, similar to the one of Cuny, Snyder and Wing (2010, cited in Wing, 2011).

Selby and Woollard (2013) report on a general consent that a CT definition should include the concept of cognitive process and proceed with their own definition, using terms for which there is a consensus in the literature or using those terms that are well defined across disciplines: abstraction, decomposition, algorithmic thinking, evaluation and generalization. European Commission (2016) also reviews CT literature and gives a compact definition, while in another report of European Commission about educational policy and CT, an attempt of extracting common concepts and skills of CT definitions that recursively emerge from literature is made, concluding to (Bocconi et al., 2016): abstraction, algorithmic thinking, automation, decomposition, debugging and generalization.

## BACKGROUND

CS Education Research Group at University of Canterbury (New Zealand) has developed a project called CS Unplugged (CS stands for Computer Science), an effort that already counts more than 20 years of life and is probably the most famous unplugged toolbox connected with CS and CT.

CS Unplugged began as a methodology (i.e. a set of proposed activities) to help in teaching basic CS concepts without the use of computers, in an intriguing and engaging way for children (CS Unplugged, n.d.). Resources originated from academics choosing to do CS with children, rather than just talking about it, and so CS unplugged relates directly to CT, helping children think like computer scientists (Bell & Lodi, 2019). CS Unplugged is now updated and all tasks are connected with CT; without proposing a new CT definition, CS Unplugged uses six basic CT concepts, which are connected with each proposed activity (Bell & Lodi, 2019, University of Canterbury, n.d.): algorithmic thinking, abstraction, decomposition, generalization and patterns, evaluation and logic. Bell and Lodi (2019) propose these six CT thought processes, citing Selby and Woollard (2013).

Computing At School (CAS) is an initiative aiming to promote the teaching of CS at school (Computing At School, n.d.), which is further analyzed in next chapter. There seems to be some consent on these six concepts, as CAS has published a guide for teachers for CT (Computing At School, 2015) where same six concepts of CT are stated: logical reasoning, algorithmic thinking, decomposition, generalization (and patterns), abstraction (and representations) and evaluation. However, CAS also refers to techniques (reflecting, coding, designing, analyzing, applying) and approaches associated with CT (tinkering, creating, debugging, persevering, collaborating).

Bell et al. (2008; 2009) define the unplugged approach as "*exposing children to the great ideas of Computer Science without using computers*", using activities involving "*problem solving to achieve a goal, and in the process dealing with fundamental concepts from Computer Science*". Sentence and Csizmadia (2017) explain that constructivist principles support the strategies of using more kinaesthetic and active approaches to teaching in the CS classroom. The authors explain that in computing this is embodied in the unplugged approach, pointing out that CS Unplugged project introduced the term "*unplugged*" and the relevant style of teaching, originally referring to the use of activities to teach computer science concepts without the use of computers. Activities are kinaesthetic and stimulate an understanding of a concept in a very concrete and practical way. As Conde et al. (2017) state: "*Unplugged computing is the approach to teaching computing concepts using constructivist, often kinesthetic, activities away from computers*". Bell and Lodi (2019) point out that unplugged activities are not inherently constructionist, but share some intents with Papert's constructionism (teach abstract concepts with concrete, constructive and kinaesthetic activities), at the same time taking a constructivist approach, since students (re) construct knowledge for themselves.

Studies on CT literature already categorize unplugged as one of the approaches used in teaching CT (Kalelioglu, Gulbahar & Kukul, 2016), while other CT teaching approaches consist of using building-block programming, tangible programming, digital game creation through computer programming and educational robotics (Angeli & Jaipal-Jamani, 2018).

## TEACHING COMPUTATIONAL THINKING UNPLUGGED

Researchers and organizations have already created and proposed the use of various teaching resources and tools to develop CT in classroom. This chapter presents a review of existing unplugged tools and methodologies, and in particular the main design instructions for creating unplugged, kinaesthetic activities, the main advantages of teaching using the unplugged approach, ending with report of blended implementation and evaluation efforts.

### Creating Unplugged Teaching Material

Bell et al. (2008) use the term "*unplugged philosophy*", while evaluating if an activity fits in this philosophy. They shortly ask for simplicity (the rules can be explained quite quickly), engagement (the activity is attractive for children), and cooperation or competition (the children are motivated to work towards a goal, either as part of a team, or to try to find a better/faster solution than another group).

They also give instructions on how similar kinaesthetic activities should be designed:

- Focus on demonstrating CS concepts, rather than programming, as programming can be a bottleneck that prevents some students from ever finding out what the deeper concepts are.
- Make the activities kinaesthetic, generally on a large scale, involving team work.
- The activities should be fun and engaging, and not just busy work.
- The materials should be low cost.
- The material is released using a creative commons license, so that others can pass them on freely and make their own contributions.
- The activities aim to be gender neutral (or at least, attractive to girls), and tend to focus on cooperative approaches rather than individualistic ones.
- The activities often have a sense of story to capture interest and motivate children and the stories can be somewhat fantastical, as this appeals to children's imagination.
- Children are generally encouraged to discover answers for themselves (with Socratic style questioning or constructivist activities), since the purpose is not to teach the answer, but to "play" with the concepts.
- The activities should be reasonably error resilient, so that small errors on the part of a child or teacher don't ruin the whole point of the activity.

Moreover, Nishida et al. (2009) provide a CS Unplugged design pattern using the following elements:

- Pattern Name: Assign an appropriate name.
- Problem: Explanation of the problem that must be solved.
- Context: The context or situation in which the solution is applicable.
- Forces: The conditions in which this pattern is applicable.
- Solution: The strategy to solve the problem.
- Resulting context: Increase the breadth and quality of activities available.
- Rationale: Enhancing students' motivation, easy implementation, etc.

Rodriguez, Rader and Camp (2016) work on CS Unplugged activities, and propose some useful advice on designing and implementing activities:

- Use of a priming activity is useful to focus students' attention.
- Students need individual practice to fully grasp a concept.
- Instructions on worksheets must be kept to a minimum and edited carefully.
- Vocabulary on worksheets must be consistent with how topics are presented.
- Real world connections help to engage students' interest.

Finally, Waite et al. (2019) propose that unplugged activities are naturally constructivist in nature and can also be constructionist, an opinion that Bell and Lodi (2019) also express. Waite et al. (2019) proceed in their research applying Legitimation Code Theory (LCT) to help understand the effectiveness of teaching with unplugged methodology, and argue that effective unplugged activities should follow the pattern of semantic waves.

Designing unplugged activities obviously is not an easy task to accomplish. Sendurur (2019) reports limited competencies of pre-service CS teachers to prepare their own unplugged activities, concluding to a need of adaption of design and implementation of unplugged activities in teaching method courses of CS teacher training programs.

## Unplugged Tools and Methodologies

Related literature references CS Unplugged project and CS4FN magazine as main resources of unplugged activities (Sentence & Csizmadia, 2017). In this review the main inclusion criterion of unplugged toolboxes is that activities should be free of charge and can be used freely, usually under a creative commons license. However, an exception was made for the inclusion of some specific approaches, in order to provide a thorough review of modern unplugged methodologies. In this manner, CT contests which do not always publicize their tasks in order to reuse them, and some books that provide a new approach in unplugged CT teaching, were included.

This chapter is based on publications relevant to unplugged CT approach, mainly by searching in databases using the keywords "computational thinking" and "unplugged". The following online databases were used in this effort: ACM Digital Library, IEEE Xplore Digital Library, Elsevier Science Direct and Springer Link. Furthermore, relevant books available from Amazon online bookstore during the years 2018-2019 were reviewed and some of them were finally decided to be included in the review, as being important in order to compose an all-embracing presentation of unplugged methodologies. Finally, as emphasized in the conclusion of the chapter, not all CT teaching approaches have been thoroughly researched; so some more toolboxes came up from web searches and the authors' expertise in the field.

Bell and Lodi (2019) clarify that in literature some researchers use the term "unplugged" when referring specifically to CS Unplugged project, while other researchers use the same term with its broader meaning, referring to activities that do not require use of computer, which are not necessarily connected to CS Unplugged project. To avoid misunderstandings, they choose to refer to the resources on the CS Unplugged website using the full title "CS Unplugged", and the term "unplugged" when referring to the general concept of teaching CS without use of computers or digital devices. That is exactly what this chapter also follows.

## CS Unplugged

CS Unplugged project is arguably the first toolbox to come in mind to any CS educator hearing the word "unplugged", as it has already been present for more than 20 years as a free educational resource and has gained worldwide recognition through these years. Unplugged approach owns its name to Timothy Bell and the CS Education Research Group of University of Canterbury in New Zealand. Basic concepts of CS and as an outcome, basic concepts of CT, are taught through activities that neither demand use of computers, nor require knowledge of specific programming languages (CS Unplugged, n.d.).

The first publication of CS Unplugged activities took place at mid-90s, and was almost immediately formed into a book (Bell, Witten & Fellows, 1998), which was revised several times throughout the years resulting to a teacher's edition in 2010. The book continues to evolve and its last revision is available online, under a Creative Commons License (Bell, Witten & Fellows, 2015). Activities and the book itself have been translated to many languages. All material, including CS Unplugged Book edition, can be found free of charge in the classic CS Unplugged website https://classic.csunplugged.org/ (Classic CS Unplugged, n.d.). For most activities presented, when applicable, a connection is made to Great Principles of CS, ACM K12 Curriculum and New Zealand Curriculum (https://classic.csunplugged.org/curriculum-links/) in paragraph "*Curriculum Links*".

As Bell et al. (2009) mention, CS Unplugged project has already had a large international uptake. It gained visibility in U.S.A. after it has been added to the ACM recommendations for the K-12 curriculum (Tucker et al., 2003), in Asia through research done in the department of CS Education at Korea University (Yoo et al., 2006), and later through an enhanced web presence due to sponsorship from Google Inc. There are videos of applying these activities to children, and also a show named "CS Unplugged: The Show", which can be found on UC CS Education's YouTube channel; all relevant information is available on classic CS Unplugged site mentioned above.

Although CT is not mentioned in these first resources, as activities existed before the international explosion of interest in CT in 2006, coping with activities that derive from CS ideas and approaches can help students build CT skills. Bearing this in mind, the CS Unplugged website https://csunplugged.org/en/ has been rebuilt in 2018 and its material has been enriched and presented in new forms, using unit plans, lesson plans, teaching videos, curriculum integration activities, and programming exercises. The new design of activities takes into consideration CT concepts, and in this way every activity presented is connected with basic concepts of CT, with a small explanation on reasoning each connection of activity with specific CT concepts.

Topics presented derive from CS and cover five sections: binary numbers, error detection and correction, kidbots, searching algorithms and sorting networks. Each topic has an age specification; all are suggested for ages 5-7 and 8-10, while sorting networks are also suggested for ages 11-14. Each topic also has its unit plan, several lessons, proposals for curriculum integrations, while in some topics there are programming challenges in an attempt to expand activities further. Lessons are presented with analytical description, remarks for teachers (named Teacher Observations), and Printables which can be downloaded and in some cases produced using scripts in order to fit individual needs. Activity duration is 30, 45 or 60 minutes.

Extensions and variations are also discussed. For example, when working with "*Sorting networks*" teachers are encouraged to test variations after implementing the original activity, even using music notations or music pitches. More information and even more extensions of these variations are presented and discussed further in Bell & Bell (2018).

Bell and Lodi (2019) make clear that CS Unplugged is not a curriculum and is not intended to replace the opportunity for students to write programs on digital devices, but is an adjunct pedagogy to help students learn about CS ideas, without having to learn to program first. Furthermore, unplugged approach is intended to be integrated with learning to program, as it can be explicitly linked to programming through a "plugging it in" follow up to the activities (Bell & Vahrenhold, 2018, cited in Bell & Lodi, 2019). Bell and Lodi (2019) also state that when CS Unplugged originated, classroom computers were rare, but nowadays things have changed, digital devices are available. However, in many countries there is still lack of computing devices and unplugged approach seems more than welcome.

### *Connection With CT Concepts*

CS Unplugged gives a thorough connection of each proposed activity with CT concepts. All activities end up with paragraph "*Seeing the Computational Thinking connections*" where a short discussion of activity's connection with CT concepts takes place. CS Unplugged uses its own approach to CT concepts, as described in section "*Computational Thinking and CS Unplugged*" of its website (https://csunplugged.org/en/computational-thinking/), and CT concepts used are:

- Algorithmic thinking.
- Abstraction.
- Decomposition.
- Generalizing and patterns.
- Logic.
- Evaluation.

For each activity there is an analysis for each CT concept, so teachers could easily understand how approach and teaching of CT concepts is achieved by using the activities.

## Colorado School of Mines and National Center for Women and Information Technology

Following CS Unplugged's publicity and dissemination, Colorado School of Mines (n.d.) adopted original CS Unplugged activities and created a set of lesson plans for teachers to use in middle school classrooms in U.S.A. There are 13 lesson plans which are presented in three main categories: representing information (5), algorithms and problem solving (5), interacting with computers (3). All lessons presented have a detailed lesson plan and relevant worksheets, while for some of them there are supplementary videos or slights for use. The duration of each activity is 50 minutes.

The National Center for Women and Information Technology (NCWIT) has also used CS Unplugged material and the lesson plans from Colorado School of Mines, ending up with the publication of CS in a Box: Unplug your Curriculum, a book available for free download and use (NCWIT, n.d.).

### Connection With CT Concepts

As these resources derive directly from classic CS Unplugged activities, readers are recommended to read "*Connection with CT concepts*" of previous paragraph. Furthermore, authors of Colorado School of Mines (n.d.) suggest that CS Unplugged activities can be used within the K-12 CS Framework (2016), in which CT is taken into consideration and practices including CT are discussed, including several unplugged activities.

## Computing at School

Computing At School (CAS) is an initiative aiming to promote the teaching of CS at school (Computing At School, n.d.), which began acting mainly in the UK, but nowadays there are also local CAS communities in other countries too. UK has proceeded to changes in national curriculum in 2014 (Department for Education, 2014), and has included computing and computational thinking to the heart of the new curriculum. CAS consists of a community of teachers, academics and industry supporters, and provides many interesting resources, such as curriculum guidance for teachers for primary and secondary education, assessment of CT and many activities for use in the classroom. In addition to online material, there are useful books such as a guide for teachers in primary education (Computing At School, 2013), a guide for teachers in secondary education (Computing At School, 2014) and a guide for teachers for CT (Computing At School, 2015).

Among the resources proposed at CAS there are some unplugged activities that can be used for teaching CT, such as "*Human Crane KS1 activity*", "*Plan and Program a Clock*" and "*Printable Scratch Blocks for Programming Unplugged*" for primary education. The last two activities are preparatory ones for Scratch programming, with the latter one using printed Scratch blocks for unplugged programming, encouraging children to even create their own "*commands*" using blank blocks, in order to program partners or events in class.

### Connection With CT Concepts

In the guide for teachers for CT (Computing At School, 2015) CAS publishes concepts of CT, which include logical reasoning and embrace:

- the ability to think algorithmically.
- the ability to think in terms of decomposition.
- the ability to think in generalizations, identifying and making use of patterns.
- the ability to think in abstractions, choosing good representations.
- the ability to think in terms of evaluation.

In addition, in the same guide CAS publishes techniques (reflecting, coding, designing, analyzing, applying) and approaches associated with CT (tinkering, creating, debugging, persevering, collaborating). An interesting remark is that CAS's concepts are in line with CS Unplugged's CT concepts presented earlier in this chapter.

## CS4FN and Teaching London Computing

Another remarkable effort with lots of unplugged material is Computer Science for Fun (CS4FN), initially created in 2005 by Paul Curzon and Peter McOwan of the School of Electronic Engineering and CS of Queen Mary University of London. CS4FN is an online magazine, offering numerous articles, games, magic tricks, puzzles, riddles, crosswords, stories etc. aiming at bringing children (and adults) closer to CS principles and methods. There are hundreds of pages, and CS4FN plans to produce two issues of the magazine a year, available for free download on the site, sending also out physical copies to schools in the UK (CS for Fun, n.d.).

Although CS4FN provides a vast collection of interesting resources, among which an experienced teacher may find and use many interesting items, its initial purpose is not to provide resources for teaching, as it is a magazine. For this reason CS4FN in its site menu provides a link "*Resources for Teachers*", pointing to Teaching London Computing website, which uses useful teaching resources of CS4FN among other ones (e.g. CS4FN booklets). Teaching London Computing (TLC) has been created by Queen Mary University of London in cooperation with Kings College of London, and is a resource hub from CS4FN and CAS London. TLC provides a website rich in material and resources, aiming at supporting educators of computing (Queen Mary University of London & King's College London, n.d.). TLC offers CS4FN resources, such as booklets and many interesting activities, which are connected to primary and secondary education, and are mainly unplugged, making TLC also a valuable toolbox of unplugged CT teaching activities. Activities are also presented in various ways (grouped for primary or secondary education, categorized by teaching subjects, connecting to other activities), making it a lot easier for a teacher to find appropriate resources for his teaching subject. In Table 1 information on many of the unplugged TLC activities is presented, but readers are strongly prompted to visit TLC website for more information. For younger students and the fun side of computing, TLC has created a spin-off from CS4FN called "A Bit of cs4fn" (A Bit of CS4FN, n.d.).

*Table 1. Teaching London Computing Unplugged Activities*

| Lesson Plan | CT Concepts | Time | Age | Resources |
|---|---|---|---|---|
| The Intelligent Piece of Paper Activity | Algorithms | 15 | 8+ | Lesson plan, presentation |
| The Brain-in-a-bag Activity | Computational modelling | 20-30 | 8+ | Lesson plan, presentation |
| The Sweet Learning Computer | Ns | Ns | Ns | Guide, full size board |
| The Invisible Palming Activity | Algorithms | 15-20 | 7+ | Lesson plan, presentation |
| Computational Thinking: Magical Book Magic | Algorithms, logical thinking, evaluation | Ns | Ns | Magical book booklet |
| The Australian Magician's Dream Activity | Algorithms, abstraction, logical reasoning | 20-30 | 11+ | Lesson plan, presentation, videos of activity |
| The Red Black Mind Meld Activity | Algorithms, evaluation, abstraction, logical thinking | 30 | 12+ (8+) | Lesson plan, presentation |

*continues on following page*

*Table 1. Continued*

| Lesson Plan | CT Concepts | Time | Age | Resources |
|---|---|---|---|---|
| The Four Aces Activity | Algorithm, understanding people | 20 | 7+ | Lesson plan, cs4fn magazine |
| The Teleporting Robot (and Melting Snowman) Activity | Understanding people | 15 | 7+ | Lesson plan, puzzles (2), cs4fn magazine, cs4fn article |
| Pixel Puzzle Pictures and Computational Thinking | Representation, abstraction, decomposition, generalization | Ns | Ns | Puzzle sheets (bees, ladybird, christmas, haloween, etc.), solution sheets |
| Kriss-Kross Puzzles | Logical thinking, patterns | Ns | Ns | Presentation, instructions, puzzle/raw puzzle sheets, solution sheets |
| Maths Kriss-Kross | Logical thinking, patterns | Ns | Ns | Puzzle/raw puzzle sheets, solution sheets |
| Word Searches and Computational Thinking | Algorithmic thinking | Ns | Ns | Word search sheets, solution sheets |
| Algorithmic Doodle Art | Algorithmic thinking | Ns | Ns | Doodle sheets, solution sheets |
| The Tour Guide Activity | Algorithms, representation, abstraction | 15+ | 8+ | Lesson plan, video, booklet |
| The Knight's Tour Activity | Algorithms, generalization, patterns, representation | 50-60 | 8+ | Lesson plan, video, booklet |
| The HexaHexaFlexagon Automata Activity | Abstraction, representation, generalization, patterns, evaluation, logical thinking | Ns | Ns | Booklet, presentation, basic template, 4 templates, video |
| Bakuro, Binary and Computational Thinking | Logical thinking, representation | Ns | Ns | Presentation, puzzle sheets, solution sheets, blank grids |
| Computational Thinking: Number Hive Puzzles | Logical thinking, pattern matching, generalization | Ns | Ns | Booklet, puzzle sheets, solution sheets |
| Computational Thinking: Cut Block Logic Puzzles | Logical thinking, pattern matching, generalization | Ns | Ns | Booklet, puzzle sheets, solution sheets |
| Compression Code Puzzles | Algorithmic thinking | Ns | Ns | Activities, solutions |
| Code cracking puzzles | Algorithmic thinking | Ns | Ns | Numerous code cracking activities, solutions |
| Sherlock Syllogisms | Logical thinking | Ns | Ns | Puzzle sheet, solution sheet |
| The Bubble Sort and Sort Dance Activities | Algorithms | Ns | Ns | Presentation, videos, cards, explanation sheets, book |
| The Divide and Conquer Sorting Activity | Algorithmic thinking | Ns | Ns | Presentation, cards, book |
| The Punch Card Sorting Activity | Algorithmic thinking | 10-15 | 11+ | Lesson plan, presentation, cards, book |
| The 20-questions Activity | Algorithms | 20-30 | 11+ | Lesson plan, presentation, 20 questions, booklet |
| The Box Variable Activity | Programming | 15-20 | 10+ | Lesson plan, presentation, booklet |
| The Assignment Dry Run Activity | Programming | 20-50 | 7+ | Lesson plan, worksheet, video |
| The create-a-face activity | Algorithms (programming) | 50 | 9-10+ | Lesson plan, presentation, |
| The Emotion Machine Activity | Abstraction, algorithmic thinking, decomposition | 40-60 | 7-12 | Lesson plan, table of codes, emotion machine sheet |
| The Imp Computer Activity | Algorithms (programming) | 30-40 | 10+ | Lesson plan, presentation, book |
| Microwave Racing Video | Evaluation | Ns | Ns | Lesson plan, video, cs4fn magazine |
| The Locked-in Activity | Algorithms | 20-30 | 11+ | Lesson plan, presentation, 20 questions, booklet |
| The Punch-card searching activity | Algorithms, translating problems | 30-35 | 11+ | Lesson plan, presentation, videos of activity |
| The Spit-not-so Activity | Algorithms, translating problems | 20-30 | 10+ | Lesson plan, presentation, cs4fn magazine |

Ns: Not specified

Paul Curzon has also written the book "*Computing without computers*" which is a gentle introduction to programming, data structures and algorithms for complete novices (Curzon, 2014). The book focuses on understanding concepts instead of teaching programming notation, intended primarily for people with little background in the subject and is free to download from TLC website.

### *Connection With CT Concepts*

CS4FN provides a webpage about CT, where CT is presented as a collection of diverse problem solving skills that result from studying the nature of computation. Reference is made initially to some obviously important skills like creativity, ability to explain and team work, and furthermore to some very specific problem solving skills such as the ability to think logically, algorithmically and recursively. CS4FN states that CT is also about understanding people (Computer Science for Fun, n.d.).

On the other hand, Teaching London Computing is a resource hub from CS4FN and CAS London, so one can suppose that follows CAS's approach of CT, as presented previously. Curzon et al. (2014) state that the concept of computing is wider than that of CT, but many activities that target computing help to develop CT, and in this way they clearly refer to CT concepts in the activities they publish.

## Barefoot Computing

Barefoot Computing is a project originally created in UK in 2014 as a one year program to help primary school teachers across England prepare for the changing computing curriculum and is now funded and run by BT in partnership with Computing at School (Barefoot Computing, n.d.). Barefoot offers resources in three main categories, for classroom use (lesson plans and resources), for teachers (self-teach material and resources) and school workshops. Among the classroom resources offered, there are 24 unplugged activities, which are presented with a lesson plan, curriculum links and resources needed to implement (Table 2).

### *Connection With CT Concepts*

As seen in Table 2, most lesson plans are presented with a characterization about CT concepts and approaches used in them. Barefoot Computing describes CT concepts and approaches followed in the section "*Computational Thinking Concepts and Approaches*" of its website, and CT concepts used are logic, algorithms, decomposition, patterns, abstraction and evaluation, while approaches associated with CT are tinkering, creating, debugging, persevering, and collaborating. Barefoot Computing's CT concepts and approaches are of course in line with CAS's CT concepts and approaches, and also with CS Unplugged's CT concepts, as presented earlier in this chapter.

## Code.org and Hour of Code Initiative

Code.org is a nonprofit organization aiming at expanding computer access in schools and increasing the participation of women and minorities. It is supported by major companies-donors and provides the leading curriculum for K-12 CS education in many districts of US (Code.org, n.d.). Code.org provides many resources for teaching, organized in curricula:

*Table 2. Barefoot Computing Unplugged Activities*

| Lesson Plan | CT Concepts | Time | Age | Resources (lesson plan available for all activities) |
|---|---|---|---|---|
| 2D Shape Drawing Debugging | Algorithms, logic | 40 | 7-11 | Presentation, example, worksheet |
| Abstraction Unplugged Activity | Abstraction | 30 | 7-11 | Presentation, record worksheet, guess-what-cards presentation |
| Crazy Character Algorithms | Logic, decomposition, algorithms | 30 | 5-7 | Worksheet |
| Creating Patterns Activity | Patterns, algorithms | Ns | Ns | Presentation, printable cards (activity, stretch, support), flipchart file, cip file |
| Dance Move Algorithms | Algorithms | Ns | Ns | Picture cards, extension cards |
| Decomposition Unplugged Activity KS1 | Decomposition, algorithms | 30 | 5-7 | Design sheet, presentation |
| Decomposition Unplugged Activity KS2 | Decomposition | 30 | 7-11 | Design sheet, presentation |
| Getting Ready for School Decomposition Activity | Decomposition, algorithms | Ns | Ns | Instruction sheet, cip files |
| Head, Shoulders, Knees and Toes Algorithms | Algorithms | Ns | Ns | Body part cards, repeat cards, cip file |
| House Patterns Activity | Patterns | Ns | Ns | Presentation, worksheet, cip file |
| Logical Number Sequences | Algorithms, logic, patterns | 45 | 9-10 | Presentation, worksheet |
| Logical Reasoning Unplugged Activity | Logic | 25 | 9-11 | Presentation, Sudoku worksheet |
| Modelling the Internet | Abstraction | 50 | 9-11 | Presentation, sets of badges, dns information sheet, 'source code' sheet |
| Network Hunt Activity | Ns | 55 | 7-11 | Worksheet, matching activity |
| Patterns Unplugged Activity: Elephants, Cats and Cars | Patterns | 25 | 5-7 | Presentation, copies of pattern challenge |
| Patterns Unplugged: Reusing Recipes | Patterns | 25 | 7-11 | Copies of pattern challenge, presentation |
| Ranking Search Activity | Ns | 45 | 9-11 | Presentation |
| River Crossing Activity | Algorithms, decomposition, logic | Ns | Ns | Recording sheet & character images, large picture cards, cip file |
| Sharing Sweets Algorithms | Patterns, algorithms | 30 | 5-7 | - |
| Shopping List Activity | Algorithms, abstraction | Ns | Ns | Presentation, differentiated worksheets, images of ingredients, cip file |
| Sorting Objects Activity | Patterns, algorithms | Ns | Ns | Presentation, sorting cats and birds print out, sorting mat, rules cip file |
| Spelling Rules Algorithms | Patterns, algorithms, logic | 20-30 | 5-7 | - |
| Story Sequencing Activity | Algorithms | Ns | Ns | Group/individual worksheets, solution |
| Variables Unplugged Activity | Ns | 35 | 7-11 | Presentation |

Ns: Not specified

- **CS Fundamentals** (K1-5, ages 4-11). Online and unplugged activities to teach students computational thinking, problem solving, programming concepts and digital citizenship, organized in 6 courses, with a duration of 10-25 hours each.
- **CS Discoveries** (Grades 6-10). Teaching of CS as a means to promote creativity, communication, problem solving and fun. It can be taught as a semester or full-year course, and has a duration of 50-150 hours.
- **CS Principles** (Grades 9-12). Teaching subjects include the internet, big data and privacy, as long as programming and algorithms. It is recommended to be taught as a full-year course, with a duration of 100-180 hours.
- **Pre-reader Express** (K1-2, ages 4-8). Courses derive from CS Fundamentals and through a 14-lesson course, with a duration of 10-14 hours, cover the core concepts from the kindergarten and first grade courses at an accelerated pace.

**CS Fundamentals Express** (Grades 3-12, ages 9-18). It is designed as a supplemental resource in an existing technology or programming class, or as an after-school program, with a duration of 30 hours. This course covers all the core concepts from the elementary school curriculum CS Fundamentals. The same material can be modulated for older students. Code.org also supports and organizes the Hour of Code initiative, which urges teachers to spend an hour giving their students the opportunity to code, in an attempt to demystify coding and CS. Hour of code offers more than 500 tutorials, available in over than 45 languages and has already had an international uptake reaching tens of millions of students in more than 180 countries (Hour of Code, n.d.).

Among all these rich resources, Code.org offers unplugged lessons in curriculum CS Fundamentals (CS Fundamentals Unplugged, n.d.). A supply list is provided and course Amazon lists have been created, so that all necessary supplies for teaching the unplugged lessons can be easily made by a teacher. Lessons are presented with a lesson plan and in cases other helpful resources, such as worksheets, lesson videos, teacher videos, assessment sheets and assessment answers. Please note that in each lesson plan, which is an online web page, more resources are provided (downloadable pdfs for teachers and students, images etc.) and an alignment with CS Standards and Cross-curricular opportunities are presented.

### *Connection With CT Concepts*

Code.org neither makes a direct connection of each activity with CT concepts, nor refers to a specific attempt definition. However, it provides two unplugged activities for CT, in the lesson plans in which the main concepts addressed are: algorithms, decomposition, abstraction and pattern matching. On the other hand, there is another activity addressing abstraction, which is a main CT concept, and many of the CS concepts addressed by the activities presented are parts of various CT definitions, as presented in Background.

## EU Code Week

CodeWeek is a grassroots initiative run by volunteers and supported by the European Commission. Almost 3 million people in over than 70 countries participated in 2018's EU Code Week (CodeWeek, n.d.). CodeWeek offers training materials and among them, three unplugged activities: CodyGame, CodyRoby and CodyDuel (https://codeweek.eu/training/coding-without-computers). In the activities one or more students are the coders (Cody) giving instructions, which are in the form of big printed cards, and one or more students are the robots (Roby) executing the instructions by moving on either a big floor board, or moving a Roby piece on a table board. In these three activities there are variations of the main concept, including simple instructions to reach a target, enrichment with questions and answer or even a race competition between robots, always following students' instructions. All material including worksheets, a video presenting main ideas, printable cards, printable board etc. is provided free of charge (http://codemooc.org/codyroby/en/). A second activity offered is "*Computational thinking and problem solving*", however only part of it is unplugged. The unplugged part consists of implementing a binary search in a printed telephone book or dictionary, acting as an introductory activity, continuing with a plugged part in Python.

### Connection With CT Concepts

CodeWeek does not explicitly follow a specific CT definition, but in the activity "*Computational thinking and problem solving*" there is reference to CT concepts of decomposition, pattern recognition, abstraction and algorithm design. Furthermore, the unplugged activity "*CodyRoby*" roughly described above, involves algorithm design and algorithmic thinking.

## CS Inside

Curzon et al. (2009) refer to three research proposals, derived from the three universities they serve: CS Unplugged (University of Canterbury New Zealand), CS4FN (Queen Mary University of London) and CS Inside (University of Glasgow). The first two have already been mentioned, while the third one (CS Inside) follows unplugged philosophy, adding two more aims: the use of technology that is already all around young people nowadays as a motivational key to engage them with the science included (e.g. cell phone predictive texting is used to introduce machine learning). Second aim recognizes that there will never be sufficient numbers of university and science communication staff to reach all school pupils, so it is essential that teachers are able to pick up and use the materials themselves.

### Connection With CT Concepts

Curzon et al. (2009) do not make a specific reference to CT, but only refer to unplugged type of activities for CS Inside. Website of CS Inside (http://csi.dcs.gla.ac.uk/) seems to be offline for a long time, so no further discussion on CT concepts involved can be made.

## Computational Thinking Bins

Morrison, Dorn and Friend (2019) present the guiding philosophy and design of Computational Thinking Bins, which are novel, self-guided, stand-alone activities aiming at helping students engage with CT concepts, using material that usually will fit in a shoe box and with no need of computers, and thus making them unplugged activities. Particular effort is being made so that these activities can be carried out, with no help of a teacher or instructor. The idea is developed at University of Nebraska Omaha and is based on existing STEM Bins of their College of Education. There already exist numerous CT bins, such as "*Robotic Buddies*", "*Beat the Clock*", "*Computer Art*", "*Card Sort*", "*Data Storage Timeline*", "*Enigma Machine*", "*Balance it Out*", "*Caesar Cipher*", "*Scytale Cipher*", "*Computers Thru History*" and "*Vigenere Cipher*", all of which follow same basic design guidelines and are assessed by a rubric for CT Bin Assignment (Morrison et al., 2019). Ideas derive from CS and cover various topics of interest, such as algorithms in general, searching, sorting, cryptography, data storage.

All information related to CT Bins is publicly available through a Creative Commons license, including all information of creating a set, a purchase list, labels and instructional material. There is a lending library at University of Omaha where CT bins can be borrowed, and all information of reproducing CT Bins can be found online (University of Nebraska Omaha, n.d.).

### *Connection With CT Concepts*

There is neither an exact connection of each CT Bin provided with specific CT concepts, nor Morrison et al. (2019) refer to a specific definition of CT. However, all ideas derive from CS, including algorithms and algorithmic thinking, search strategies (e.g. binary search), cryptography, data storage and representation, sorting algorithms, etc., and thus are connected with CT skills. Further research is needed on linking each CT Bin with specific CT concepts, proving its effect on improving student CT skills.

## Computational and Algorithmic Thinking (CAT) and Bebras

Moreover, there are CT contests, which at a first glance deal with problem solving topics without the use of a computer. Two characteristic examples are Computational and Algorithmic Thinking (CAT) (Australian Mathematics Trust, n.d.) and Bebras (Vilniaus Universitetas, n.d.). The latter one is conducted worldwide and is not connected with computer programming, but aims at CT. Both contests do not require specific programming expertise, and moreover in some cases a paper version of the contest is provided, making thus the tasks completely unplugged.

Bebras promotes CT and its implementation in children's education (Dagiene & Stupuriene, 2016). Its tasks however are not freely available for teachers to use, luckily there are some examples in the official Bebras website and also some of the tasks that already have been used in previous contests are partially available in national Bebras webpages.

### *Connection With CT Concepts*

Bebras Challenge is connected with CT, promotes problem solving skills and informatics concepts, such as decomposition, algorithm design, pattern recognition, pattern generalization and abstraction (Vilniaus Universitetas, n.d.). In this manner, by definition all Bebras tasks are relevant with one more CT concepts. Furthermore, Bebras tasks are used by researchers in order to teach and evaluate CT (Dagiene, Stupuriene & Vinikiene, 2016) and more and more for CT assessment (Araujo et al., 2019; Chiazzese et al., 2019; Wiebe et al., 2019).

## Center for Discrete Mathematics and Theoretical CS (DIMACS)

Another serious effort has been made in U.S.A. by the Center for Discrete Mathematics and Theoretical CS (DIMACS), through the project "The Value of Computational Thinking across Grade Levels 9-12 – VCTAL" (DIMACS, n.d.). VCTAL is developing a set of instructional modules, mini-modules, and (ultimately) a book for use in high school classrooms, aiming at cultivating computational thinking in students across different grade levels and subject areas. There are activities referring to real world topics and problems, in which computer is gradually used in activities for processing, representation and visualization of data. Some of the activities, such as network activity, cryptography activities, tomography activity, data privacy, etc. come with either an unplugged part or a discussion which require no use of computer. The main idea of VCTAL however is not using only unplugged activities, so in many activities a computer program is used, even from scratch.

### *Connection With CT Concepts*

There is a clear reference in VCTAL webpage to the Operational Definition of ISTE and CSTA (2011), so activities are expected to be in line with this definition. However there is no direct reference for each activity linking its implementation with specific CT characteristics and dispositions or attitudes.

## Computing at Children's University in Poland

Syslo and Kwiatkowska (2014) describe their involvement at Children's University in Poland, a non-profit initiative where academic teachers organize educational activities for children. The authors use unplugged activities to introduce young students (ages 5-12) to CS concepts and propose the use of various activities coping with calculating devices, towers of Hanoi, Fibonacci numbers, binary search, graph concepts such as Konigsberg bridges, map coloring or knight tour, and shortest paths on graphs. The authors conclude that children are capable to work with abstraction and apply algorithmic (computational) thinking. They also report that to their belief and experience, children are at the appropriate age at which CS teaching could begin.

### *Connection With CT Concepts*

The authors neither provide a specific connection of proposed activities with CT concepts, nor do they use a specific definition of CT. In their conclusions they refer to abstraction and algorithmic thinking (Syslo & Kwiatkowska, 2014).

## Teaching Computing Unplugged in Primary Schools

Caldwell and Smith (2017) wrote the book "*Teaching Computing Unplugged in Primary Schools*", which consists of nine chapters of different, fun and engaging activities, showing how concepts of computing and CT can be explored and developed away from technology. The activities show a large number of ways that computational thinking can be used across the curriculum in everyday activities, such as through drama, outdoor activities, art, music, puzzles, games and practical hand-on activities. All activities suggest ways of collaborative implementation, demonstrating that CT techniques can be applied across curriculum subjects so that they are embedded as a creative problem-solving tool.

There are nine chapters, each of which offers three unplugged activities formed in lesson plans, as shown in Table 3 (age is given as key stages of UK curriculum). In the lesson plans, extensions are also recommended, as well as some plugged activities for anyone wanting to go further, and useful hints and links are provided.

*Table 3. Teaching computing unplugged in primary school activities*

| Chapter | Activity | CT Concepts | Time | Age |
|---|---|---|---|---|
| Robots | Robot hamster playground | Algorithms, logical reasoning, evaluation, decomposition | 60 | Upper KS1, KS2 |
| | Shoot the robot | Algorithms, logical reasoning, decomposition | 50 | Primary school |
| | Robot foodies | Algorithms, evaluation, logical reasoning, decomposition | 55 | KS1, early KS2 |
| Musicians | Just dance | Algorithms, pattern recognition, decomposition | 55 | KS1, KS2 |
| | Human beatbox | Algorithms | 45 | KS2 |
| | Music maker | Algorithms, abstraction, pattern recognition | 60 | KS2 |
| Artists | Sol's solutions | Algorithms, decomposition | 60 | KS1, KS2 |
| | Rocketing numbers and images | Algorithms, logic, abstraction | 80 | KS1, KS2 |
| | Thomas's tangles | Algorithms, evaluation | 85 | KS2 |
| Explorers | Sorting networks | Algorithmic thinking, logical reasoning, abstraction | 50 | KS1, KS2 |
| | Riding the internet | Generalization, abstraction | 30+ | Upper KS2 |
| | Getting 'plugged in' | Abstraction | 35+ | Upper KS1, KS2 |
| Code breakers: Dpef Csfblfst | Using ciphers | Algorithmic thinking, abstraction, evaluation, decomposition, generalization | 60 | KS2 |
| | Cracking ciphers | | 120+ | Upper KS2 |
| | Encoding for transmission | | 60 | Upper KS1, KS2 |
| Magicians | Invisible palming | Algorithmic thinking, decomposition, abstraction, evaluation, logical thinking | 45 | KS2 |
| | Turning the page on book magic | Evaluation, logical thinking | 50 | KS2 |
| | The teleporting robot | Algorithmic thinking, logical thinking, evaluation | 45 | KS2 |
| Gamers | Space race | Algorithms, logic, evaluation | 60 | KS1, KS2 |
| | Can you find the criminal? | Logical reasoning, evaluation | 60 | Upper KS2 |
| | Coding with cards | Logical reasoning, evaluation | 50 | KS1, KS2 |
| Cooks | Wrong recipes | Algorithmic thinking, abstraction | Ns | KS1, KS2 |
| | Recipe detectives | Generalization and patterns, evaluation | Ns | KS1, KS2 |
| | Copy and paste pasta | Generalization and patterns | Ns | KS1, KS2 |
| Scientists | A drizzly bear | Algorithms, decomposition, logical reasoning | 60 | KS1 |
| | Fantastic flying machines | Algorithms, decomposition, logical reasoning | 90 | Upper KS2 |
| | Spaghetti towers | Patterns, evaluation, decomposition | 50 | KS2 |

Ns: Not specified

### Connection With CT Concepts

Caldwell and Smith (2017) state that in their book they use the definition of CT defined by Barefoot and CAS, as presented in the guide published by CAS (Computing At School, 2015).

## The Power of Computational Thinking

Furthermore, Curzon and McOwan (2017), both from Queen Mary University of London and CS4FN project co-creators, wrote a book by the title "*The Power of Computational Thinking*", which consists of 13 chapters, with detailed activities explaining CT in an accessible way using magic tricks, games and puzzles, as well as presenting through real and challenging problems that computer scientists work on. Their work is based on their previous work published in CS4FN and TLC, already reviewed above, in combination with new material. The layout of the book is explanatory, recommended for anyone wanting to deepen further in these activities, which are presented in a descriptive way. Thus, no lesson plans or ready to print worksheets are provided in this book, however it still is a valuable tool for teaching CT.

### *Connection With CT Concepts*

As stated before, both Professors are involved in creating and maintaining both CS4FN and TLC websites, so a logical conclusion is that the definition of CT followed is the one published by CAS (2015). However, in the last chapter the authors refer to and analyze CT skills, reporting even more CT concepts: algorithmic thinking, computational modelling, scientific thinking, heuristics, logical thinking, pattern matching, representation, abstraction, generalization, decomposition, understanding people, evaluation and creativity.

## Children's Literature on Unplugged CT Teaching Tools

Twigg, Blair and Winter (2019) propose children's literature in order to introduce computing principles and concepts in primary schools, using unplugged methodology. Researchers used classic picture books of the appropriate age and identified sequencing, repetition and selection in them, combining this way practices used in children literature with computational ones.

Other authors use children's literature as a medium to teach CT. For example Liukas (2015) in "Hello Ruby – Adventures in Coding" writes nine small stories in Ruby's world, illustrated, which are in fact nine small lessons in CT. The book is designed to be read (worked) with a parent, and there is also a part with activities following the stories, where children are expected to practice. CT concepts used are algorithmic thinking, decomposition, abstraction, logical thinking and pattern recognition. Same author has continued the "Hello Ruby" series, with introduction to the inside of a computer and to the internet world (Liukas, 2017; 2018).

Another fascinating children's book, consists of a central heroine princess who wanders around her kingdom trying to save it, and in this journey principles of CT are applied and explained through the kingdom's situation and everyday problems (Kubica, 2012). Kubica, a Google engineer on machine learning and algorithms, has also written another "*computational fairy tale*" introducing aspects of software development using also a world of castles and magic (Kubica, 2013), and finally another book by the title "The CS detective. An algorithmic tale of crime, conspiracy, and computation", in which CT and search algorithms are presented, in a tale involving crime, detectives etc. (Kubica, 2016). A respectable amount of his work is available online, free of charge (Computational Fairy Tales, n.d.).

## Scratch Unplugged

Finally, unplugged methodology has inspired two developers to create a board game called Scratch Unplugged (Van de Bergh & Callebaut, n.d.). It is based on a simple idea of two opponent teams, using printed Scratch instruction blocks which their robot will follow on the board. Board consists of cells, bombs and walls and the idea is getting to the opponent team's starting point first, without walking on a bomb or reaching a wall. Instruction manual and printables (board, building blocks, poster) are provided free of charge. A similar idea is also used in CAS, as mentioned earlier in this chapter.

### *Connection With CT Concepts*

There is explicit reference to CT, but this activity simply aims at unplugged programming, in an attempt to include all students and help them discover the basics of programming in a fun engaging way, and thus algorithmic thinking is a concept that Scratch Unplugged helps develop.

## Blended Unplugged and Plugged Research Efforts

Unplugged activities, especially CS Unplugged toolbox, have been used for many years. Bell and Lodi (2019) research CS Unplugged activities and their connection to CT, reporting that "*unplugged activities are effective when used in a context where they will be ultimately linked to implementation on a digital device, either through programming, or by helping students to see where these ideas impinge on their daily life*". Greenberg & Reed (2018) use sophisticated error detection magic tricks, some of which belong to CS Unplugged toolbox, but proceed with proposing an innovative, powerful interactive tool for visualizing the underlying concepts. The authors also comment on the fact that some researchers also proceed with creating program exercises motivated by CS Unplugged activities.

The research design using an experimental and a control group testing effectiveness of unplugged activities is used in more studies. Hermans and Aivaloglou (2017) experiment using plugged-in and unplugged first programming lessons, mentioning that their belief is that children should definitely use a computer, but try to identify which approach may suit best as an introducing activity to programming. Their sample is consisted of primary school children at the ages of 8-12. Although their research reports no difference in understanding of programming concepts between the plugged first and unplugged first group, results suggest that after 8 weeks children in the unplugged first group measure significantly better in self-efficacy beliefs and use more different Scratch blocks than children in the plugged first group. Conde et al. (2017) also conduct an experiment using two groups, one carrying out unplugged activities in order to develop CT while the other does not, and report better results for the students who have completed unplugged activities, also reporting differences depending on age, specifically for younger students.

Recent research seems to follow this blended unplugged and plugged procedure. Tsarava et al. (2017) propose a blended method containing both unplugged and plugged-in activities, aiming mainly at programming concepts such as loops, conditionals, events, etc., which are shared concepts between CT and programming / CS education. In addition, Tsarava, Moeller & Ninaus (2018) develop life-size board games as an unplugged introduction to CT, but in their future research direction state that they plan to continue with the same concepts used in these unplugged games in the context of other educational environments.

Dorling and White (2015) discuss ways to tackle the difficulties presented to students by text-based language and provide a pathway from Scratch to Logo and Python, in which they use an unplugged version of Logo, UPL. Koning, Faber and Wierdsma (2017) describe their research on educational strategies for CT in primary education, where among other plugged-in approaches, unplugged, hands-on activities are used for abstract concepts like variables and decomposition. Moreover, Grover, Lundh and Jackiw (2019) present novel non-programming activities for engagement with foundational concepts in introductory programming, where they also use a blended approach, including two unplugged activities among other web-based, interactive ones.

Finally, a blended model of unplugged and plugged-in activities is used in other researches, where usually an introductory unplugged activity is used for students to familiarize with basic concepts and then follow additional plugged-in activities with either use of computer or robotics (Lonati et al., 2017; Miller et al., 2018). Lonati et al. (2015) propose a workshop called Algomotricity Maze Workshop (AlMa) in which they firstly use unplugged activities as introductory ones, students trying to guide a blindfolded person (human robot) through a path. Students gradually create their instructions (unplugged programs), test them, even swap programs, and at the last phase students are given computers and a programming environment in order to write real programs guiding a sprite through a maze.

Saxena et al. (2019) design a blended approach containing both plugged and unplugged activities in order to cultivate CT in early childhood education (ages 3-6). They use Bee-Bot for plugged activities and for the unplugged activities LEGO bricks for patterns and sequencing stories. Teachers use unplugged activities including vocabulary building songs, direction game through cards and Tic-Tac-Toe, in order to help children acquire necessary CT skills and language to continue with algorithm design in Bee-Bot activity. The unplugged activities are used here as warm-up ones. Basic CT concepts dealt with are pattern recognition, sequencing and algorithm design.

CT is not always approached strictly by programming or CS exercises. There are efforts that could be categorized elsewhere, for example Bell and Bell (2018) present their research on integrating CT with a music education context, adopting a sorting network exercise used in CS Unplugged using musical notes or bell pitches instead of numbers for comparing values at the nodes. The authors proceed with analyzing more music examples in which CT can be approached, using both approaches, thus proposing a blended musical method.

Another approach which uses blended methodology differentiating from typical CS deals with looming, first in an unplugged way of finding patterns and reusing them, and then continues with programming such looming patterns in Scratch (Lee & Vincent, 2019).

As a conclusion, Caeli and Yadav (2019) examine the historical route of CT and refer specifically to plugged and unplugged activities, emphasizing on the importance of combining both approaches, giving the opportunity to students to fully understand and take advantage of the power of computing, as unplugged activities can be extremely efficient in understanding concepts, but students should then proceed and be provided with a unified whole, and cope with programming and automation, something that nowadays machine do for humans.

## Bringing it all Together

This section presented a collection of unplugged CT activities and toolboxes. Some of these have already been used by educators in their initial form or combined with plugged activities. Table 4 provides a synopsis of the unplugged CT approaches, in terms of age range, CT concepts and resources that each approach presents.

*Table 4. A synopsis of unplugged CT approaches*

| Approach | Age range | CT Concepts | Resources provided |
|---|---|---|---|
| CS Unplugged | 5-14 | Algorithmic thinking, abstraction, decomposition, generalizing and patterns, logic, evaluation | Online lesson plans, printables, teaching videos |
| Colorado School of Mines | 9-14 | Ns. Activities from CS Unplugged | Lesson plans, worksheets, videos, slights |
| N.C.W.I.T. | 9-14 | Ns. Activities from CS Unplugged | Book with lesson plans and worksheets |
| Computing At School | Primary and secondary | Logical reasoning, algorithmic thinking, decomposition, generalizations and patterns, abstractions and representations, evaluation | Books, teacher guides, several activities, printables |
| CS4FN - Teaching London Computing | 7+,<br>Ns for some activities | Abstraction, representation, algorithmic thinking, decomposition, evaluation, generalization, patterns, logical reasoning, computational modelling, translating problems, understanding people | Lesson plans, presentations, booklets, videos, magazines, worksheets, solution sheets |
| Barefoot Computing | 5-11,<br>Ns for some activities | Abstraction, algorithms, decomposition, logic, patterns | Lesson plans, presentations, worksheets, solutions, printables (cards etc.), cip files |
| Code.org - Hour of Code | Ns | Abstraction, algorithms, programming concepts (conditionals, debugging, events, functions, sequencing, variables, loops), digital citizenship, persistence | Lesson plans, lesson videos, teacher videos, worksheets, assessment sheets, solutions |
| EU Code Week | Ns | Decomposition, pattern recognition, abstraction, algorithmic thinking and algorithm design | Worksheets, video, printables (cards, board) |
| Computational Thinking Bins | 11-18 | Algorithms (general, searching, sorting), cryptography, data storage | Purchase list, labels, instructional material, lesson plans, lending library |
| Bebras | 5-19 | Decomposition, algorithm design, pattern recognition, pattern generalization and abstraction | Tasks, examples |
| DIMAC's | High school | Operational Definition of ISTE and CSTA (2011) | Set of instructional modules, mini-modules, book |
| Children's University in Poland | 5-12 | Abstraction, algorithmic thinking | Description of activities |
| Teaching Computing Unplugged in Primary Schools | Primary schools | Abstraction, algorithmic thinking, decomposition, evaluation, generalization and patterns, logic | Lesson plans, extensions, hints |
| The Power of Computational Thinking | Ns | Algorithmic thinking, computational modelling, scientific thinking, heuristics, logical thinking, pattern matching, representation, abstraction, generalization, decomposition, understanding people, evaluation and creativity | Book |
| Children's literature: Twigg, Blair and Winter (2019) | Up to 7 | Sequencing, repetition and selection | Description in paper |
| Children's literature: Hello Ruby | Ns | Algorithmic thinking, decomposition, abstraction, logical thinking and pattern recognition | Fairy tale and workbook |
| Children's literature: Computational Fairy Tales | Ns | Algorithmic thinking, logic | Fairy tales |
| Scratch Unplugged | Ns | Algorithmic thinking | Printables (board, blocks, poster), instructions |

Ns: Not specified

## Advantages of Using Unplugged Teaching Material

Unplugged is a specific CT teaching approach and, as analyzed above, a rich toolbox collection already exists and can be used in classroom and outdoor activities. There are specific advantages that unplugged approach brings to educational procedures.

Sentance and Czimadia (2017) report their findings on teachers' views about CT, on the verge of changing the computing curriculum in UK in 2014. Their research results refer to strategies that work well for the teachers in teaching Computing, and one of the five key themes reported by teachers' perspective are unplugged type activities. Lee and Junoh (2019) also refer to the significance of implementing unplugged coding with concrete hands-on practices, which enable children to manipulate codes, even in early childhood.

Bell et al. (2008; 2009) report numerous advantages for the use of unplugged methodology:

- Stepping away from the computer, children are able to think about issues that Computer Scientists face beyond simply programming.
- It is easily implemented, as it does not require a significant commitment from the start in time and/or resources.
- There is no need of previous technical experience (e.g. knowledge of specific programming language).
- The activities used are kinaesthetic, fun and engaging for children.
- There is no need of a specific classroom or computer lab and many activities are suitable for use outdoors, which can be useful as a break from being in a classroom, and combines physical activity with problem solving.

It's true that, a scientific evaluation of implementation of unplugged activities is needed, so that advantages and efficiency of this teaching approach can be proved. That's what is going to be discussed in the next section.

## Evaluating Unplugged CT Methodologies

Unplugged approaches already count more than 20 years of presence. One might except that many studies evaluating unplugged methodologies would exist during these years, but, as Brackman et al. (2017) report, there is lack of investigations that prove the effectiveness of the unplugged activities in the development of computational thinking skills. Kalelioglu et al. (2016) also emphasize that, in general the papers of CT implementation lack research designs and researchers can neither guarantee that their implementation proposals deliver CT skills, nor can they assess if students are equipped with predefined skills after their interventions. Table 5 presents an overview of several evaluation attempts concerning unplugged activities, which are further discussed in this section.

One of the earliest evaluation attempts was made by Nishida et al. (2008), who reported positive findings on using CS unplugged resources for teaching informatics. Lambert and Guiffre (2009) also used CS Unplugged activities in three thirty minutes' sessions with fourth grade students, and report improved confidence and interest in CS and Math. On the contrary, Taub, Ben-Ari & Armoni (2009) also use CS Unplugged and report partial success in changing students' views on CS and suggest additions and modifications.

In addition, Feaster et al. (2011) reported that there was little success in implementing unplugged activities to senior high school students, in their attempt to improve students' attitudes towards CS and perceived content understanding. Ten activities, almost all from CS Unplugged, were used to senior high school students (grades 9-12) and the research was repeated for two consecutive semesters with different groups, but no important outcomes were produced. Authors state that in their belief, one of the reasons of this failure is that high school students do not find kinesthetic activities of this kind as exciting as middle or elementary school students. Thies and Vahrenhold (2013) comment on results of Feaster et al. (2011), explaining that CS Unplugged materials can be adequate and effective when integrated in lower secondary education, and in this way agree with Feaster et al. (2011) assumption on the appropriateness of the age of the students. Moreover, Thies and Vahrenhold (2013) report that unplugged activities can be at least as effective as when following more conventional approaches.

Another interesting approach brings CS Unplugged closer to disabled students, and with the help of some digital materials for handicapped people, positive results are also reported in a vocational training course for disabled students (Manabe et al., 2011).

Lamagna (2015) proposes the use of various unplugged tools, mainly puzzles, in order to develop important algorithmic thinking skills. In his research four exercises are presented and analyzed, all of which aim at developing algorithmic thinking. The author reports positive results on the unplugged approach suggested in this work.

In recent years additional research has been conducted in the field. Thies and Vahrenhold (2016) revalidate previous findings across with multiple institutions and a broader student population and report that unplugged activities are equally efficient compared to teaching using textbooks or interactive methods.

Rodriguez et al. (2016) evaluate and choose six CS Unplugged activities, which they use in 7$^{th}$ grade classrooms and conclude to revised Unplugged activities, which they publish in Colorado School of Mines (n.d.), reaching to some very interesting conclusions and advice on designing unplugged activities.

Moreover, Brackman et al. (2017), who report lack of investigations on effectiveness of unplugged activities, particularly for primary schools, conduct a research using unplugged resources, and their conclusions on teaching effectiveness of unplugged approach are mainly positive.

Faber et al. (2017) have designed and implemented an introductory course in computational thinking for students at their final year in primary school, and in their results report that the unplugged aspect of the lesson materials seems to elicit positive reactions from both teachers and students, and state that unplugged programming lessons are a valuable alternative to regular, online programming lessons.

Jagust et al. (2018) use various unplugged activities to teach CS related topics with focus on CT and algorithmic thinking, and although work is still in progress they report positive results and very good reception of the activities by students, parents and teachers.

Jiang and Wong (2017) also conducted a research using programming exercises and unplugged activities and reported moderate to high motivation to learn CT in general, reporting motivation of students by both programming activity and CS Unplugged, but a higher sense of mastery was built after programming activities compared to CS Unplugged however. In addition, Leon et al. (2019) report on work in progress and an educational project called "Llup'ix" in Peru, where they plan to use unplugged activities in primary schools for teaching technology concepts, considering unplugged methodology an effective strategy for public education in their country.

Tsarava et al. (2018) suggest use of board games, introducing three life-size board games called "*Crabs & Turtles: A Series of Computational Adventures*", which aim at introducing basic coding concepts and CT processes to 8 to 9 year-old primary school children. Basic CT processes dealt with are

*Table 5. Evaluation researches of unplugged approach*

| Research | Age | Resources | Findings | Sample, Duration |
|---|---|---|---|---|
| Nishida et al. (2008) | Grade 3 | CS Unplugged | Positive on teaching informatics, effective for students, increased interest, motivated to learn, deeper thinking in Unplugged | 64 students, 10 hours<br>48 students, 9 hours |
| Lambert & Guiffre (2009) | Grade 4 | CS Unplugged | Improved confidence and interest in CS and Math | Three classes (20-25 students each), 3 30-minute sessions plus pre-post test |
| Taub et al. (2009) | Grade 7-8 | CS Unplugged | Partial process of changing students' views, propose additions and modifications | 13 students, 18 2-hour activities, 81 students questionnaire |
| Manabe et al. (2011) | High school (disabled students) | CS Unplugged with digital assistance | Very positive learning attitudes in vocational training for disabled students | A course at a vocational training school |
| Feaster et al. (2011) | Grades 9-12 | CS Unplugged | No statistically significant impact on student attitudes or perceived content understanding | Ten 40-minute sessions, two semesters, two groups (control 15 - experimental 14 students) |
| Thies & Vahrenhold (2013) | Lower secondary (11-12) | CS Unplugged | Teaching using unplugged activities at least as effective as when following more conventional approaches | 25 students, two groups (control – experimental), 3 45-minute slots |
| Lamagna (2015) | Tertiary students | Puzzles | Positive results on unplugged algorithmic thinking approach | Freshman seminar, Junior level course. No of students not specified |
| Thies & Vahrenhold (2016) | Grade 5-6 | CS Unplugged | Revalidation of prior research. Unplugged activities equally efficient to teaching using textbooks or interactive methods | 108 students, two groups (control – experimental), 4 45-minute slots |
| Rodriguez et al. (2016) | Grade 7 | CS Unplugged | Approach that supplements CS Unplugged activities with worksheets | 130 students, 3 deployments of 6 50-minute activities |
| Brackman et al. (2017) | Grade 5-6 (10-12) | Hello Ruby, Code Master, Authors' creation | Unplugged activities help students enhance their CT skills significantly more | 73 students, two groups (control – experimental), 10 hours |
| Faber et al. (2017) | Final grade of primary school | Various (CS Unplugged, Code.org, etc.) | Positive reactions from both students and teachers | 411 students (26 schools), 6 90-minute lessons |
| Jagust et al. (2018) | Elementary to tertiary education | Graph paper - Board games, The network | Good reception of unplugged activities from students, parents, teachers | Students from elementary to tertiary education |
| Jiang & Wong (2017) | Grade 4 (9-11) | CS Unplugged | Equally motivated by CS unplugged and programming, higher sense of mastery after programming compared to CS Unplugged | 220 students, six-week (lesson) course |
| Tsarava et al. (2018) | Primary school (use of experts) | Life-size Board games | Positive results toward the direction of teaching basic coding concepts and CT processes | 17 adult university students, 19 other adults, 3 2-hour gaming sessions |
| Torres-Torres et al. (2019) | Primary education, Families | Avatar, Carpet | Positive observations on CT skills, including girl inclusion and improvement of gender equity | Two groups (34 students, Families), two activities |

decomposition, algorithms, patterns, evaluation and abstraction, according to the authors. Tsarava et al. (2018) describe analytically the three games and proceed with a pilot evaluation with postgraduate students and a specialized group of gamification experts and teachers, and report positive results toward the direction of teaching basic coding concepts and CT processes.

Finally, Torres-Torres, Roman-Gonzalez and Perez-Gonzalez (2019) use unplugged activities to foster CT skills in primary education from a gender perspective, and report positive observations on CT skills, including girl inclusion and improvement of gender equity.

## SOLUTIONS AND RECOMMENDATIONS

A positive finding of this chapter is that there is already a notable amount of CT activities around the world, most of which are distributed freely and in a structured way through the internet, making the CT unplugged toolbox richer and more powerful. Activities already cover many different topics, some age groups and some of them have already been translated to different languages, facts that help teachers make use of them easier, in a more simple and efficient way.

However, the lack of a widely accepted CT definition continues to be a barrier to CT teaching efforts. How can CT be taught and approached, if no consent is reached on what exactly CT is and how wide its components and concepts are? As mentioned earlier in this chapter, some teaching resources' proposals do not follow a specific CT definition, but use CT as a wider concept. Fortunately, in many of the main unplugged toolbox approaches analyzed above, like CS Unplugged, CAS, Barefoot Computing, CS4FN, TLC there seems to be some consent on the CT definition followed, in terms of CT concepts, which makes the use and combination of them easier for teaching or evaluating purposes. In this way, common CT concepts appear to be the following six: algorithmic thinking, abstraction, decomposition, generalization and patterns, evaluation and logical thinking. Some of the rest unplugged toolboxes presented also use part of these six concepts.

Many of the unplugged activities analyzed previously originally aimed at teaching CS concepts (such as CS Unplugged), or at the concept of computing (CAS, Barefoot Computing, etc.), which is wider than that of CT (Curzon et al., 2014), and not at CT itself; however using these activities, as well as teaching programming and CS concepts, helps CT development. CS Unplugged has already linked many unplugged activities with CT concepts. What needs to be done is a wider research on specific unplugged approaches and their connection to specific CT concepts, so that unplugged toolboxes will clearly aim at cultivating these concepts, and furthermore, a connection with other variables such as student age range, sex, cognitive level should be taken into consideration.

Unplugged approach is not extensively examined yet in educational research practice. However there are some studies, which mostly report positive results as presented previously in this chapter, reinforcing in this way the usefulness of unplugged methodology as a CT educational approach. Notwithstanding, the majority of the proposed activities have not been assessed yet for their effectiveness on improving CT skills, an action that requires valid assessment CT tools, or assessment research methods that deal with validity issues, organized in a concrete scientific framework. Moreover, many of the studies presented, usually follow convenience sampling, using a small number of students, which unfortunately leads to results that can't be generalized or reproduced. In one case however, Thies and Vahrenhold (2016) have managed to revalidate their previous research's findings (Thies & Vahrenhold, 2013), using multiple teachers and institutions. It is however remarkable, that more than half of evaluation studies presented in the relevant section use CS Unplugged as teaching resources.

In addition, a blended unplugged and plugged approach seems to be used more and more during the last years, characterizing unplugged as a powerful introductory means to modern CT education. This blended approach makes good use and exploits all advantages of unplugged approach, without moving away from the initial interpretation of CT, which nowadays involves efficient use of computers and digital devices.

Nevertheless, most of the proposed activities require the presence of a teacher, which means that teachers themselves need to be prepared for the changes that inclusion of CT education into classroom practice has brought. So, apart from designing and testing unplugged activities, some effort must be forwarded to teacher training. Researchers proposing CS Inside unplugged activities have already mentioned this gap, stating that there will never be sufficient numbers of university and science communication staff to reach all school pupils, so it is essential that teachers are able to pick up and use the materials themselves (Curzon et al., 2009).

Moreover, teacher perceptions can also be a barrier to implementing unplugged, kinesthetic activities. Teachers who choose not to proceed with implementing unplugged, kinesthetic activities in classroom, although trained at workshops, report feeling uncomfortable with this kind of teaching kinesthetically, others report that it would take more time to prepare, and others feel that the age of their students is inappropriate for this kind of activities (Thies and Vahrenhold, 2016).

Finally, there are some activities especially designed as stand-alone ones, such CT Bins, and some of children's literature might as well act as a stand-alone teaching resource that does not require the presence of a trained teacher, but in most cases and in typical primary and secondary education, a well-trained and fully equipped teacher will be required.

The discussion above leads to three categories for unplugged activities and methodologies: a) activities are designed and intended for use in typical educational systems. Activities of this type could be adapted in curricula of K-12 education, come with lesson plans and usually require the presence of a qualified teacher, b) design and aim of activities is for outreach programs, do not always address specific CT (or CS) concepts and are not always structured in lesson plans and for use in classroom, and c) activities are designed as standalone and could be self-implemented (e.g. a fairy tale, CT bins, crosswords, puzzles, etc.) or could be worked with the presence of an adult (e.g. a parent), who is not necessarily a qualified teacher. Mannila et al. (2014) use a similar categorization referring to "official and formal" ways to teach CT and "informal initiatives", such as clubs, contests and outreach programs.

## FUTURE RESEARCH DIRECTIONS

Open research questions still include the definition of what exactly CT is, as a wider consent has not yet been accomplished. This definition consent will surely help on designing effective educational practices as well as proper assessment of students. Many of the activities presented in the toolbox are not clearly linked to specific CT concepts or dispositions, which is something that also needs to be dealt with. Furthermore, activities that argue for implementing specific CT concepts need to be scientifically researched, so that their effectiveness is properly proved.

Except from connecting specific unplugged activities with specific CT concepts, future research directions should also include the connection or correlation of unplugged activities with specific variables, such as students' age and sex. Moreover, there are countries, e.g. the UK (Department for Education, 2014), which have already included computing and computational thinking into their new curriculum. Proposed activities, in order to be used in formal primary or secondary education, should take into consideration the integration of CT into K-12 classroom level curriculum, which differentiates among countries (Fessakis et al., 2018; Heintz, Mannila & Färnqvist, 2016).

In addition, several researchers combine unplugged and plugged activities, resulting to a blended approach, where unplugged activities are used mainly as introductory to the plugged ones (Saxena et al., 2019), while other researchers already propose plugged activities and only examine if unplugged or plugged activities serve better as introductory ones (Hermans & Aivaloglou, 2017). The blended approach of unplugged and plugged activities and the role of unplugged activities as introductory or as the main educational material need to be part of more extensive research.

Furthermore, much of the research done should be repeated using bigger and more representative samples; most researchers have used convenience sampling, including a small number of students which usually are one or two classrooms. Many of the resources are presented for a wide age range (e.g. primary school), but are only implemented and researched with a specific age (students of only one or two grades), so there are no research results for the whole age range.

Another direction that is worth working on is the involvement of teachers, as some resources are designed as stand-alone ones and other require the presence of a teacher. It would be interesting to research on the effectiveness on CT of unplugged activities that require the presence of a teacher and usually are used in typical educational system, versus other that can be used with a parent or even by the children themselves, such as CT literature.

## CONCLUSION

In this chapter the unplugged approach for teaching CT has been addressed. An exhausting compilation of the most well-known unplugged teaching activities has been reviewed, all of which form a powerful toolbox for teaching CT unplugged. The lack of a commonly accepted CT definition is still a problem of the field, but some of the main unplugged toolboxes use and implement activities on the same CT concepts, providing in this way an informal consent on six basic CT concepts. Algorithmic thinking seems to be present at almost all unplugged approaches, even those that do not clearly state which CT concepts they cover, as algorithmic thinking is directly connected with all programming exercises including sequencing, loops, conditionals etc. Abstraction is also a wide used CT concept that also seems to be present in most unplugged toolboxes.

The majority of these resources are available for use by educators free of charge on the internet, which makes them even more useful as a CT teaching approach. Activities that use a creative commons license are also really helpful and powerful, as the educational community can evolve them and redistribute them.

Some positive findings of use of unplugged activities in teaching CT in K-12, in combination with advantages of unplugged approach provide an extra boost to the wider use of these activities in CT education. Another interesting aspect is that many of the unplugged activities are cross-curricular and can be used in combination with other subjects (literature, music, arts, etc.), making them even more interesting and engaging in this way.

Many researchers report a better understanding in introducing basic CT or computing concepts when using unplugged activities in a blended approach, in which plugged activities follow the unplugged introduction ones, making this schema an attractive one for future CT educational activities implementations.

However, most of the unplugged approaches presented aim at younger students, while some other resources do not specify exact age range. Moreover, it seems that older students are not so intrigued by unplugged, kinesthetic activities. Apart from that, evaluation has not yet been extensively carried out; most studies are for CS Unplugged and there are only few for the rest of the material. This fact in combination with small convenience samples that usually studies have, leads to failure to generalize any positive findings.

Another issue is that many of the activities presented are largely designed for outreach programs, and have already reached a large audience worldwide (Bell at al., 2011); however, as the authors accurately point out, an important issue that needs to be addressed is how these materials can be finally adapted for settings where assessment of specific concepts (a certain level of assessment) will be required. As more countries proceed with including CT and computing in their school curricula (Heintz et al., 2016), this remark becomes more and more important every day, in order to use unplugged activities in formal curricula.

Analysis of the unplugged toolboxes in this chapter shows that most of them do not require expensive materials and moreover, some of them require none; this comes to agreement with the general design instructions of Bell et al. (2008), suggesting that material should be low-cost. Most activities are engaging and many have a sense of story to capture interest and motivate children, which also Bell et al. (2008) mention. However not all activities are kinaesthetic, but an interesting pattern that has showed up is that all of them involve thinking processes, thinking on problem solving and inquiring solutions using CS tools and approaches, even though without using computers, confirming in this practical way the CT definition of Cuny et al. (2010, cited in Wing, 2011).

In addition, analysis of studies referring to unplugged approaches' evaluation shows that unplugged methodologies are mostly applied to primary education students. Table 5 shows that 11 out of 15 studies are implemented in primary education. Feaster et al. (2011) use a high school students' sample but do not report positive findings; on the contrary they suggest that unplugged methodologies might be more appropriate for younger students. Another remarkable finding is that 10 out of 15 studies have used CS Unplugged resources, which sounds logical, considering the large international uptake of CS Unplugged (Bell et al., 2009). Most studies (10 out of 15) report positive findings in terms of dispositions, such as good reception of activities, increased motivation and increased confidence of students. Studies do not usually report on specific CT concepts and developing process of each one; a general acquisition of CT skills is sometimes mentioned. Future research attempts should address the process of CT development.

Finally, this chapter also proposes a new categorization of unplugged activities, which consists of typical activities that can be used in the formal educational system, usually requiring a teacher, outdoor activities that do not necessarily follow a typical educational system pattern and stand-alone ones that can be used out of any framework, without the presence of a teacher or just with the company of a parent.

## REFERENCES


A Bit of CS4FN. (n.d.). *A Bit of CS4FN*. Retrieved from https://abitofcs4fn.org/

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835. doi:10.1093/comjnl/bxs074

Angeli, C., & Jaipal-Jamani, K. (2018). Preparing Pre-service Teachers to Promote Computational Thinking in School Classrooms. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines* (pp. 127–150). Springer. doi:10.1007/978-3-319-93566-9_7

Araujo, A. L. S. O., Andrade, W. L., Guerrero, D. D. S., & Melo, M. R. A. (2019). How Many Abilities Can We Measure in Computational Thinking?:A Study on Bebras Challenge. In *Proceedings of 50th ACM Technical Symposium on CS Education* (pp. 545-551). New York: ACM.

Australian Mathematics Trust. (n.d.). *Computational and Algorithmic Thinking (CAT)*. Retrieved from https://goo.gl/eVCFis

Barefoot Computing. (n.d.). *Barefoot. Unlock the future of your classroom*. Retrieved from https://www.barefootcomputing.org/

Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the CS Education Community. *ACM Inroads*, *2*(1), 48–54. doi:10.1145/1929887.1929905

Bell, J., & Bell, T. (2018). Integrating computational thinking with a music education context. *Informatics in Education*, *17*(2), 151–166. doi:10.15388/infedu.2018.09

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008). CS without computers: new outreach methods from old tricks. In *Proceedings of 21st Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ)*. Auckland, New Zealand: Academic Press.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). CS Unplugged: School students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

Bell, T., Curzon, P., Cutts, Q., Dagiene, V., & Haberman, B. (2011). Overcoming obstacles to CS education by using non-programming outreach programmes. In I. Kalas & R. T. Mittermeir (Eds.), *ISSEP 2011, LNCS 7013* (pp. 71–81). Berlin: Springer. doi:10.1007/978-3-642-24722-4_7

Bell, T., & Lodi, M. (2019). Constructing Computational Thinking Without Using Computers. *Constructivist Foundations*, *14*(3), 342–351.

Bell, T., & Vahrenhold, J. (2018). CS Unplugged – How is it used, and does it work? In H. J. Bockenhauer, D. Komm, & W. Unger (Eds.), *Adventures between lower bounds and higher altitudes: Essays dedicated to Juraj Hromkovic on the occasion of his 60th birthday*. New York: Springer.

Bell, T., Witten, I. H., & Fellows, M. (1998). *CS Unplugged . . .off-line activities and games for all ages*. Retrieved from https://classic.csunplugged.org/wp-content/uploads/2015/01/unplugged-book-v1.pdf

Bell, T., Witten, I. H., & Fellows, M. (2015). *CS Unplugged An enrichment and extension programme for primary-aged students*. Retrieved from https://classic.csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. Retrieved from https://goo.gl/dmWYfU

Brackmann, C., Roman-Gonzalez, M., Robles, G., Moreno-Leon, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 65-72). 10.1145/3137065.3137069

Caeli, E. N., & Yadav, A. (2019). *Unplugged approaches to computational thinking: A historical perspective.* Springer. doi:10.100711528-019-00410-5

Caldwell, H., & Smith, N. (2017). Teaching Computing Unplugged in Primary Schools. London: Learning Matters (SAGE Publications Ltd).

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019). Educational Robotics in Primary School: Measuring the Development of Computational Thinking Skills with the Bebras Tasks. *Informatics (MDPI)*, *6*(43), 1–12. doi:10.3390/informatics6040043

Classic C. S. Unplugged. (n.d.). *Classic CS Unplugged*. Retrieved from https://classic.csunplugged.org/

Code.org. (n.d.). *Code.org About us*. Retrieved from https://code.org/about

CodeWeek. (n.d.). *Code Week*. Retrieved from https://codeweek.eu/about

Colorado School of Mines. (n.d.). *CS Unplugged at Mines for Middle Schools*. Retrieved from http://csunplugged.mines.edu/

Computational Fairy Tales. (n.d.). *Computational Fairy Tales*. Retrieved from http://computationaltales.blogspot.com/

Computational Thinking with Scratch. (2012). *Computational thinking*. Retrieved from https://goo.gl/mxzTnp

Computer Science for Fun. (n.d.). *CS4FN: the fun side of CS*. Retrieved from http://www.cs4fn.org/

Computing at School. (2013). *Computing in the national curriculum. A guide for primary teachers*. Bedford: Newnorth Print.

Computing at School. (2014). *Computing in the national curriculum. A guide for secondary teachers*. Bedford: Newnorth Print.

Computing at School. (2015). *Computational Thinking. A guide for Teachers*. Retrieved from https://community.computingatschool.org.uk/files/8550/original.pdf

Computing at School. (n.d.). *Computing at school*. Retrieved from https://www.computingatschool.org.uk/

Conde, M. Á., Fernandez-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellan-Olivera, V., & García-Peñalvo, F. J. (2017). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. In *Proceedings of International Conference Technological Ecosystems for Enhancing Multiculturality (TEEM 2017)*. 10.1145/3144826.3145355

Curzon, P. (2014). *Computing without computers. A gentle introduction to computer programming, data structures and algorithms*. Retrieved from https://teachinglondoncomputing.org/resources/inspiring-computing-stories/computingwithoutcomputers/

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: a framework*. Retrieved from https://goo.gl/28ScgY

Curzon, P., McOwan, P., Cutts, Q., & Bell, T. (2009). Enthusing & Inspiring with Reusable Kinaesthetic Activities. *ACM SIGCSE Bulletin*, *41*(3), 94–98. doi:10.1145/1595496.1562911

Curzon, P., & McOwan, P. W. (2017). *The Power of Computational Thinking*. London: World Scientific Publishing Europe Ltd. doi:10.1142/q0054

Dagiene, V., & Stupuriene, G. (2016). Bebras – a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in Education*, *15*(1), 25–44. doi:10.15388/infedu.2016.02

Dagiene, V., Stupuriene, G., & Vinikiene, L. (2016). Promoting inclusive informatics education through the Bebras challenge to all K-12 students. Proceedings of CompSys Tech '16. doi:10.1145/2983468.2983517

Department for Education. (2014). *The national curriculum in England. Framework document*. Retrieved from www.gov.uk/dfe/nationalcurriculum

DIMACS. (n.d.). *The Value of Computational Thinking across Grade Levels 9-12 (VCTAL)*. Retrieved from http://dimacs.rutgers.edu/archive/VCTAL/

European Commission. (2016). *The Computational Thinking Study*. Retrieved from https://goo.gl/y5TL2K

Faber, H., Wierdsma, M., Doornbos, R., van der Ven, J., & de Vette, K. (2017). Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Educational Network*, *12*, 13–24.

Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. (2011). Teaching CS Unplugged in the High School (with Limited Success). In *Proceedings of 16th Annual Conference on Innovation and Technology in CS Education (ITiCSE '11)*. 10.1145/1999747.1999817

Fessakis, G., Komis, V., Mavroudi, E., & Prantsoudi, S. (2018). Exploring the scope and the conceptualization of Computational Thinking at the K-12 classroom level curriculum. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines: Foundations and Research Highlights*. Springer. doi:10.1007/978-3-319-93566-9_10

Fundamentals Unplugged, C. S. (n.d.). *Code.org CS Fundamentals Unplugged*. Retrieved from https://code.org/curriculum/unplugged

Greenberg, R. I., & Reed, D. F. (2018). Using magic in computing education and outreach. *Proceedings of 2018 IEEE Frontiers in Education Conference*. 10.1109/FIE.2018.8658626

Grover, S., Lundh, P., & Jackiw, N. (2019). Non-Programming Activities for Engagement with Foundational Concepts in Introductory Programming. In *Proceedings of 50th ACM Technical Symposium on CS Education* (pp. 1136–1142). New York: ACM. 10.1145/3287324.3287468

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Proceedings of IEEE Frontiers in Education Conference (FIE) 2016*. Erie: IEEE. 10.1109/FIE.2016.7757410

Hermans, F., & Aivaloglou, E. (2017). To Scratch or not to Scratch? In *Proceedings of 12th Workshop on Primary and Secondary Computing Education* (pp. 49-56). Nijmegen, Netherlands: Academic Press.

Hour of Code. (n.d.). *Hour of Code activities from Code.org*. Retrieved from https://code.org/hourofcode/overview

ISTE & CSTA. (2011). *Operational definition of Computational Thinking in K-12 education*. Retrieved from https://goo.gl/AlSSp

Jagušt, T., Sovic Krzic, A., Gledec, G., Grgic, M., & Bojic, I. (2018). Exploring Different Unplugged Game-like Activities for Teaching Computational Thinking. In *Proceedings of 2018 Frontiers in Education Conference (FIE)*. IEEE. 10.1109/FIE.2018.8659077

Jiang, S., & Wong, G. K. W. (2017). Assessing primary school students' intrinsic motivation of computational thinking. In *Proceedings of 2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE).* IEEE.

K–12 CS Framework. (2016). *K–12 CS Framework*. Retrieved from http://www.k12cs.org

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic J. Modern Computing*, *4*(3), 583–596.

Koning, J. I., Faber, H. H., & Wierdsma, M. D. M. (2017). Introducing computational thinking to 5 and 6 year old students in Dutch primary schools; an educational design research study. In *Proceedings of 17th Koli Calling International Conference on Computing Education Research* (pp. 189–190). New York: ACM.

Kubica, J. (2012). *Computational fairy tales*. Monee.

Kubica, J. (2013). Best practices of spell design. A computational fairy tale. San Bernardino, CA: Academic Press.

Kubica, J. (2016). *The CS detective. An algorithmic tale of crime, conspiracy, and computation*. San Francisco: No Starch Press.

Lamagna, E. A. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, *30*(6), 45–52.

Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in Colleges*, *24*(3), 118–124.

Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal*, *47*(6), 709–716. doi:10.100710643-019-00967-z

Lee, V. R., & Vincent, H. (2019). An expansively-framed unplugged weaving sequence intended to bear computational fruit of the loom. In P. Blikstein, & N. Holbert (Eds.), *Proceedings of FabLearn 2019*. New York: ACM. 10.1145/3311890.3311907

Leon, J. A., Abad Jiménez, W. E., Barraza Ramírez, J. L., Frontado Aldana, S. A., Puma Alvites, A. T. A., & Solórzano Pinzás, R. L. (2019). Work in progress: Llup'ix. In *Proceedings of 2019 IEEE World Conference on Engineering Education (EDUNINE)*. IEEE. 10.1109/EDUNINE.2019.8875816

Liukas, L. (2015). *Hello Ruby: Adventures in coding*. New York: Feiwel & Friends.

Liukas, L. (2017). *Hello Ruby: Journey inside the computer*. New York: Feiwel & Friends.

Liukas, L. (2018). *Hello Ruby: Expedition to the internet*. New York: Feiwel & Friends.

Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2015). Is coding the way to go? In A. Brodnik & J. Vahrenhold (Eds.), ISSEP 2015, LNCS 9378 (pp. 165–174). Academic Press.

Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2017). Nothing to fear but fear itself: Introducing recursion in lower secondary schools. In *Proceedings of 2017 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*. IEEE. 10.1109/LaTiCE.2017.23

Manabe, H., Kanemune, S., Namiki, M., & Nakano, Y. (2011). CS Unplugged assisted by digital materials for handicapped people at schools. In I. Kalas & R. T. Mittermeir (Eds.), *ISSEP 2011, LNCS 7013* (pp. 82–93). Berlin: Springer. doi:10.1007/978-3-642-24722-4_8

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education. In *Proceedings of Innovation and Technology in Computer Science Education, ITiCSE '14*. New York: ACM. doi:10.1145/2713609.2713610

Miller, B., Major, J. C., Kirn, A., Feil-Seifer, D., Anderson, M., & Jurkiewicz, M. (2018). Unplugged Robotics to Increase K-12 Students' Engineering Interest and Attitudes. In *Proceedings of 2018 IEEE Frontiers in Education Conference (FIE)*. IEEE. 10.1109/FIE.2018.8658959

NCWIT. (2018). *CS-in-a-Box: Unplug Your Curriculum (2018 Update)*. Retrieved from https://www.ncwit.org/resources/computer-science-box-unplug-your-curriculum-2018-update

Nishida, T., Idosaka, Y., Hofuku, Y., Kanemune, S., & Kuno, Y. (2008). New methodology of information education with "CS Unplugged". In R. T. Mittermeier & M. M. Syslo (Eds.), *ISSEP 2008, LNCS 5090* (pp. 241–252).

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS Unplugged Design Pattern. In *Proceedings of 40th SIGCSE Technical Symposium on CS Education, SIGCSE 2009* (pp. 231-235). New York: ACM.

Papert, S. (1993). *Mindstorms: Children, Computers, and Powerful Ideas* (2nd ed.). New York: Basic Books.

Queen Mary University of London & King's College London. (n.d.). *Teaching London Computing*. Retrieved from https://teachinglondoncomputing.org/

Rodriguez, B., Rader, C., & Camp, T. (2016). Using Student Performance to Assess CS Unplugged Activities in a Classroom Environment. In *Proceedings of 2016 ACM Conference on Innovation and Technology in CS Education ITiCSE '16* (pp. 95-100). New York: ACM. 10.1145/2899415.2899465

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2019). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *The Asia-Pacific Education Researcher*, 1–12. doi:10.100740299-019-00478-w

Selby, C. C., & Woollard, J. (2013). *Computational Thinking: The Developing Definition (Project report)*. Retrieved from https://eprints.soton.ac.uk/356481/

Sendurur, P. (2019). Investigation of pre-service CS teachers' CS-unplugged design practices. *Education and Information Technologies*, *24*(6), 3823–3840. doi:10.100710639-019-09964-6

Sentence, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. doi:10.100710639-016-9482-0

Syslo, M. M., & Kwiatkowska, A. B. (2014). Playing with computing at a children's university. In *Proceedings of Workshop in Primary and Secondary Computing Education WIPSCE '14*. New York: ACM. 10.1145/2670757.2670790

Taub, R., Ben-Ari, M., & Armoni, M. (2009). The effect of CS Unplugged on middle-school students' views of CS. In *Proceedings of the 2009 ACM Conference on Innovation and Technology in CS Education, ITiCSE '09* (pp. 99-103). New York: ACM. 10.1145/1562877.1562912

The Royal Society. (2012). *Shut down or restart: The way forward for computing in UK schools*. Retrieved from https://goo.gl/7Qa7fY

Thies, R., & Vahrenhold, J. (2013). On plugging "unplugged" into CS classes. In *Proceedings of the 2013 ACM Special Interest Group on Computer Science Education, SIGCSE '13* (pp. 365-370). New York: ACM.

Thies, R., & Vahrenhold, J. (2016). Back to school: CS unplugged in the wild. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in CS Education, ITiCSE '16* (pp. 118-123). New York: ACM.

Torres-Torres, Y. D., Roman-Gonzalez, M., & Perez-Gonzalez, J. C. (2019). Implementation of Unplugged Teaching Activities to Foster Computational Thinking Skills in Primary School from a Gender Perspective. In *Proceedings of 7th International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 209-215). New York: ACM. 10.1145/3362789.3362813

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, *5*(2), 25–44. doi:10.17083/ijsg.v5i2.248

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. In *Proceedings of European Conference on Games Based Learning* (pp.687-695). Academic Press.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A Model Curriculum for K-12 CS: Final Report of the ACM K-12 Task Force Curriculum Committee*. New York: ACM.

Twigg, S., Blair, L., & Winter, E. (2019). Using children's literature to introduce computing principles and concepts in primary schools: work in progress. In *Proceedings of 14th Workshop in Primary and Secondary Computing Education, WIPSCE '19*. New York: ACM. 10.1145/3361721.3362116

University of Canterbury. (n.d.). *CS Education Research Group. Computational Thinking and CS Unplugged*. Retrieved from https://csunplugged.org/en/computational-thinking/

University of Nebraska Omaha. (n.d.). *Computational Thinking Bins*. Retrieved from https://www.unomaha.edu/college-of-information-science-and-technology/computer-science-education/community-engagement/ct-bins.php

Unplugged, C. S. (n.d.). *CS Unplugged*. Retrieved from https://csunplugged.org/en/

Van de Bergh, B., & Callebaut, G. (n.d.). *Scratch unplugged. Computerless programming with Scratch*. Retrieved from http://www.scratchunplugged.com/

Vilniaus Universitetas. (n.d.). *Bebras - International Challenge on Informatics and Computational Thinking*. Retrieved from https://goo.gl/7cWDp2

Waite, J., Maton, K., Curzon, P., & Tuttiett, L. (2019). Unplugged computing and semantic waves: Analysing crazy characters. In *Proceedings of 1st UK & Ireland Computing Education Research Conference (UKICER)*. ACM. 10.1145/3351287.3351291

Wiebe, E., London, J., Aksit, O., Mott, B. W., Boyer, K. E., & Lester, J. C. (2019). Development of a Lean Computational Thinking Abilities Assessment for Middle Grades Students. In *Proceedings of 50th ACM Technical Symposium on CS Education (SIGCSE '19)* (pp. 456-461). New York: ACM. 10.1145/3287324.3287390

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wing, J. M. (2011). Research Notebook: Computational Thinking - What and Why? *The Link*. Retrieved from https://goo.gl/ifrqcO

Yoo, S.-W., Yeum, Y.-C., Kim, Y., Cha, S.-E., Kim, J. H., Jang, H.-S., ... Lee, W.-G. (2006). Development of an Integrated Informatics Curriculum for K-12 in Korea. In R. T. Mittermeir (Ed.), *Proceedings of International Conference on Informatics in Secondary Schools (ISSEP) 2006. Lecture Notes in CS 4226*. Berlin: Springer. 10.1007/11915355_19

## ADDITIONAL READING


Christian, B., & Griffiths, T. (2016). *Algorithms to live by. The CS of human decisions*. New York: Henry Holt and Co.

Curzon, P. (2014). Unplugged computational thinking for fun. In *Proceedings of Key Competencies in Informatics and ICT* (pp. 7–15). KEYCIT.

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: a framework*. Southampton Education School.

Curzon, P., & McOwan, P. W. (2008). Engaging with CS through magic shows. In *Proceedings of ItiCSE '08*. New York: ACM. 10.1145/1384271.1384320

Denning, P. J., & Tedre, M. (2019). *Computational thinking*. Cambridge, Massachusetts: THE MIT PRESS. doi:10.7551/mitpress/11740.001.0001

Fincher, S. A., & Robins, A. V. (Eds.). (2019). *The Cambridge handbook of computing education research*. Cambridge: Cambridge University Press. doi:10.1017/9781108654555

Guzdial, M. (2016). *Learner-centered design of computing education. Research on computing for everyone*. Williston: Morgan & Claypool.

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing Computational Thinking in CS Unplugged Activities. In M. E. Caspersen, S. H. Edwards, T. Barnes, & D. D. Garcia (Eds.), *Proceedings of the 2017 ACM SIGCSE Technical Symposium on CS Education* (pp. 501-506). New York: ACM. 10.1145/3017680.3017779


## KEY TERMS AND DEFINITIONS

**Abstraction:** Hiding/reducing unnecessary detail, so that focus is on the crucial components of a problem or situation.

**Algorithmic Thinking:** Thinking process used when creating algorithms, standing for using clear, well-defined and efficient steps in a problem-solving process.

**Computer Science (CS) Unplugged:** Resources and project for teaching CS without the use of computers or digital equipment, originally generated by a project of CS Education Research Group at the University of Canterbury, New Zealand.

**Decomposition:** Breaking a problem or process into smaller, easier to manage or solve, pieces.

**Evaluation:** Process of ensuring that a proposed solution or algorithm is efficient and its final outcome is exactly what expected and described.

**Generalization:** Used often with pattern matching. Process involving pattern recognition or matching, so that a solution of a specific problem can be modified and applied to several similar problems.

**Kinaesthetic (Activity):** Using parts of human body, meaning that an activity is designed for students to move, using their legs, arms, etc. during implementing it, and not just sit on a chair or in front of a computer.

**Logical Thinking:** Thinking process where facts and rules are applied so that any outcome or conclusion is not arbitrary, but emerges from data and facts and rules, which can be verified.

**Plugged:** Educational process or activity making use of computers or digital equipment.

**Unplugged:** Educational process or activity, usually kinaesthetic, which makes no use of computers or digital equipment.

## Section 3
# STEM and Educational Robotics

# Chapter 10
# Measuring the Impact on Student's Computational Thinking Skills Through STEM and Educational Robotics Project Implementation

**Avraam Chatzopoulos**
https://orcid.org/0000-0002-2569-4147
*University of West Attica, Greece*

**Michail Kalogiannakis**
https://orcid.org/0000-0002-9124-2245
*University of Crete, Greece*

**Michail Papoutsidakis**
*University of West Attica, Greece*

**Sarantos Psycharis**
*School of Pedagogical and Technological Education, Greece*

**Dimitrios Papachristos**
*University of West Attica, Greece*

## ABSTRACT

*In this chapter, the authors present their research on how P12 students apply computational thinking (CT) skills when they are assigned simple science, technology, engineering, mathematics (STEM) problems, which they are called upon to solve with the help of educational robotics (ER) activities. The reason for this research was the high participation and increased interest shown in an ER event, where distributed questionnaires recorded students' views on ER, STEM, and CT. Their answers were the spark to conduct a pilot study on primary school students in the form of an experiential seminar to investigate the possibility of developing their CT skills by applying ER activities when they are asked to solve authentic STEM problems. The results showed that students may develop CT skills when involved in ER activities and that educational robots enhance students' engagement with programming and create a more favorable environment for developing students' CT skills.*

DOI: 10.4018/978-1-7998-4576-8.ch010

## INTRODUCTION

Today many academics, businessmen, politicians, media outlets refer to the 4th Industrial Revolution, (Marr, 2016, 2017, 2018) where innovation contributes to the exchange of data on production technologies or the interconnection of the physical, digital and biological worlds (Marr, 2016). Western societies are already under its influence, and the World Bank president has said that 150 million workers will lose their jobs by 2022, while 300 million newcomers will not find work (Higgins, 2013). By one popular estimate, 65% of children entering primary school today will ultimately end up working in completely new job types that don't yet exist (Schwab & Samans, 2016). In such a rapidly evolving employment landscape, the ability to prepare for the future skills requirements is increasingly critical, and education models must adapt to equip children with the skills to create a more inclusive, cohesive and productive world (World Economic, 2020). On the other hand, there is a growing disconnect between education systems and labor markers. Many of today's students will work in new job types that do not yet exist, with an increased demand for both digital and social-emotional skills (World Economic, 2019). *Education 4.0* initiative emerged from the need to prepare students and teachers for the requirements of 4th Industrial Revolution, and it aims to create a common agenda to transform education systems to ensure future-readiness among the next generation of talent (World Economic, 2019).

In this context students need new skills for their future. For instance, Dr. Tony Wagner (co-director of Harvard's Change Leadership Group) suggests the following seven survival skills (Singmaster, 2008):

1. critical thinking and problem-solving,
2. collaboration and leadership,
3. effective oral and written communication,
4. accessing and analyzing information,
5. curiosity and imagination,
6. initiative and entrepreneurialism,
7. agility and adaptability.

Respectively, the non-for-profit organization "Battelle for Kids" suggest the P21's Frameworks for 21st Century Learning, (Figure 1) a framework developed with input from teachers, education experts, and business leaders to define and illustrate the skills and knowledge students need to succeed in work and life (Battelle for Kids, 2019).

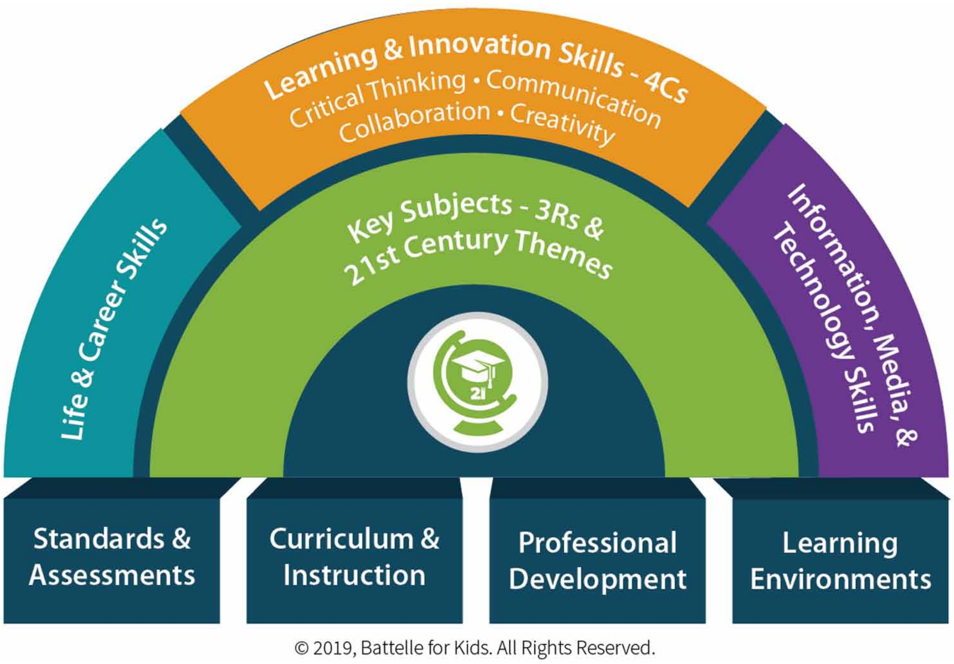
In this framework, *learning and innovation*–among others- skills- are mentioned as needs for the increasingly complex life and work environments in today's world.

These skills include:

1. creativity and innovation,
2. critical thinking and problem solving,
3. communication, and
4. collaboration.

*Science Technology Engineering and Mathematics (STEM)*, *Educational Robotics (ER)*, and *Computational Thinking (CT)* will appear to support the effort to fulfill this need, to prepare students for the 21st century and 4th industrial revolution requirements.

*Figure 1. Framework for 21st Century Learning (Battelle for Kids, 2019)*




## STEM AND ER IN EDUCATION

Old teaching methods mechanisms are no longer beneficial to the students. In traditional instructional methodology, where the lecture classes are perceived to be tedious by students, the gamification technology has a great advantage to solve the problem as it can improve learning motivation of students. Various studies have shown that gamification under appropriate conditions may create an environment conducive to learning and lead to large increases in students' interest in programming and STEM activities (Papadakis, 2018; Papadakis & Kalogiannakis, 2018; Vidakis et al., 2019).

STEM term was firstly introduced by the NSF (National Science Foundation) in the 1990's as SMET (Wikipedia, 2018), and it was used to refers to teaching and learning in the fields of science, technology, engineering, and mathematics, or it is used as a generic label for any action, policy, program or practice that involves one or more of the its disciplines (Gonzalez & J.Kuenzi, 2012).

In the literature, there is a big variety of the STEM education term's definition. Ioannou M & Bratitsis T defines it as an integrative approach to curriculum and instruction, content and skills, approaching all its areas as one, without any boundaries between them (Ioannou & Bratitsis, 2016). Vasquez, Sneider, and Comer point out that it is not a curriculum, but a way of organizing and delivering instruction (Vasquez, Comer, & Sneider, 2013), and by integrating STEM derivatives there are many benefits that could improve the science and mathematics education (Heilig, 2015). For Ejiwale is a "meta-discipline" which means the creation of discipline is based on the integration of other disciplinary knowledge into a new 'whole' rather than in bits and pieces (Ejiwale, 2013). Tsupros, Kohler, & Hallinen agree that it is an interdisciplinary approach where learning happens by integrating the four disciplines into one cohesive teaching and learning paradigm (Tsupros, Kohler, & Hallinen, 2009). And, according to Chatzopoulos, Papoutsidakis, Kalogiannakis, & Psycharis, the term STEM is linked to the teaching and learning approach that integrates the content and skills of the above terms that make up it (Chatzopoulos, Papoutsidakis, Kalogiannakis, & Psycharis, 2019; Texley & Ruud, 2018) where students can engage in and benefit (Traurig & Feller, 2008).

There are two different approaches to integrate STEM into education (Psycharis & Kotzampasaki, 2019; Roehrig, Moore, & Wang, 2012; Saito, Gunji, & Kumano, 2015):

1. The *content integration* that focuses on merging content fields into a single teaching activity to highlight "big ideas" from multiple content areas.
2. The *contextual integration* that focuses on the content of a single scientific field, while frameworks from other disciplines are used to make the subject more relevant.

One such STEM integration is *Educational Robotics (ER)*, a broad term that refers to a collection of activities, educational programs, technology platforms, educational resources, and pedagogical theories of learning within and outside schools (Chatzopoulos et al., 2019; Daniela & Lytras, 2018).

Robots and specifically ER have been gaining popularity in recent years (Zygouris et al., 2017) and in Greece, the main mobility currently seen in STEM education in schools concerns ER applications (Chatzopoulos et al., 2019). There has been a great interest among researchers (Mavrovounioti, Chatzopoulos, Papoutsidakis, & Piromalis, 2018), and educators (Melkonian, Chatzopoulos, Papoutsidakis, & Piromalis, 2018), since ER (Xatzopoulos, Papoutsidakis, & Chamilothoris, 2013) is a powerful learning and supportive tool for the development of cognitive-social skills.

ER first emerged in 1960 when Seymour Papert, a mathematician and the director of the MIT Logo Group (Marina Umaschi Bers, 2008, p. 14), began developing new technologies for children and was continued by Mitchel Resnick's Lifelong Kindergarten research group at MIT Media Lab, who has been involved in the connection between games, computer and learning since 1980 (Foundation, 2019; Ronsivalle, Boldi, Gusella, Inama, & Carta, 2018).

In 1984, LEGO owner Kjeld Kirk Kristiansen was fascinated by a Seymour Papert's speak in a television show, who was demonstrating his programming language, LOGO, for children. As a consequence, Kjeld Kirk asked for a meeting to be set up between the two parties, eventually resulting in the collaboration between Lego and MIT. In 1987, the Lego Group launch a new product, Lego Technic Control 0, a product that can be programmed with a special version of LOGO developed by Seymour Papert (Lego, 2019). Later, in the mid-1980s Mitchel Resnick, Steve Ocko, and Fred Martin (Resnick, Ocko, & Papert, 1988), in Papert's lab at MIT, started to develop the first LEGO–Logo program, but the most notable ER product conceived by the collaboration was Lego MINDSTORMS™ launched in 1998. Since, Lego products gave students a hands-on STEM solution, ER attracted the interest of researchers and teachers as a powerful teaching tool to support learning and develop students' cognitive-social skills (Alimisis, 2009, 2013).

ER is introduced in many learning environments as an innovative teaching and learning tool (Alimisis, 2014; Ospennikova, Ershov, & Iljin, 2015) that supports students (Blanchard, Freiman, & Lirrete-Pitre, 2010; Caballero-Gonzalez, Muñoz-Repiso, & García-Holgado, 2019; Çalik, Ebenezer, Özsevgeç, Küçük, & Artun, 2015; Çalik, Özsevgeç, Ebenezer, Artun, & Küçük, 2014):

1. developing high-level skills,
2. creating multiple representations of understanding the object,
3. constructive communication and collaboration between them (Caballero-Gonzalez et al., 2019),

4. developing and improving their learning by solving complex authentic problems,
5. implementing abstract design ideas to reflect and immediately notice the results of this effort,
6. facilitating student learning through research and experimentation by contributing to the development of knowledge in the STEM areas.

In addition, STEM and ER activities promote problem-based learning and critical thinking (Psycharis, 2018; Psycharis, Kalovrektis, Sakellaridi, & Korres, 2017; Sullivan & Heffernan, 2016; Swaid, 2015), as they focus on research and analysis of a complex real-world problem and it is also important the *play aspect* involved in making them particularly attractive to students especially in primary education (Atmatzidou, Markelis, & Demetriadis, 2008; Chevalier, Riedo, & Mondada, 2016). ER suggests learning through design and includes activities such as constructing and operating robot platforms and are also used as tools for advancing *computational thinking (CT)*, coding, and engineering. They offer an appropriate platform for developing skills in a fun and meaningful way and also engage students to deal with a range of disciplines such as science, technology, engineering, and mathematics (STEM), literacy, social studies, dance, music, and art (Ioannou & Makridou, 2018).

However, there is an internationally recognized under-representation of women in sectors related to IT, both in academia and STEM-related industries. Various researchers have attempted to investigate the reasons why women tend to be un-interested in the computer science already from the period of their secondary education. Some of them interpret this lack of disposition as an after-effect from the fact that boys might have a more frequent access to computers, search for technology information more often and/or receive a more positive feedback about computer learning, without ignoring the role of the hidden analytical program and of the school textbooks. Other researchers state that girls continue to reveal lower self-esteem regarding their digital skills than their classmates and in contrast to boys they confront programming and other activities affiliated with Information science as difficult, non-interesting and/or boring (Papadakis, 2018a; Papadakis, Polychronaki, & Tousia, 2018).

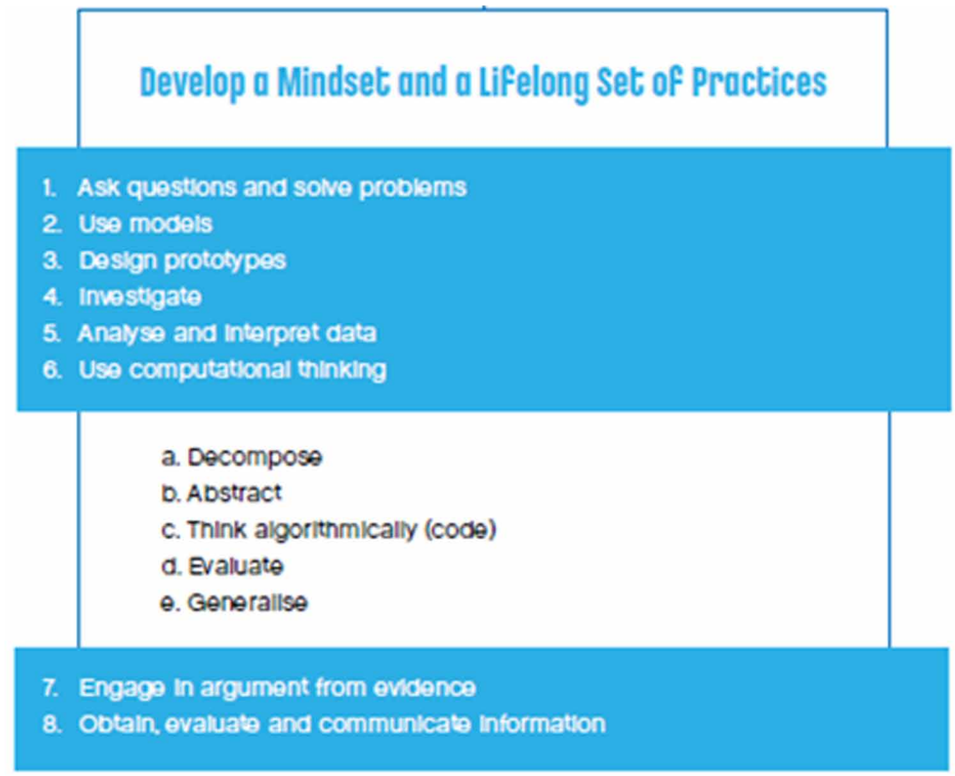
## COMPUTATIONAL THINKING

In the Lego's "Teacher's Guide" educational manual offered to support its STEM and ER activities, a block diagram for developing a mindset and a lifelong set of practices is presented (Figure 2) as a tool to achieve good STEM practices. In this figure CT is briefly presented as a series of the following actions:

1. Decompose,
2. Abstract,
3. Think Algorithmically (code),
4. Evaluate and
5. Generalize (Lego, 2016a, 2016b).

CT term first used by Seymour Papert to describe not just the procedures and concepts used for the solution of problems and the design of computer systems, but also the application of the above in the perception and comprehension of natural phenomena (Papadakis, Kalogiannakis, & Zaranis, 2016; Papert, 1996). Later, Wing, defined it in-depth as a rich set of analytical methods that effectively involve the human and the machine element in the solution of various problems (Wing, 2006). These methods

*Figure 2. Developing a mindset and a lifelong set of practices*




include specific tasks such as *programming*, *testing* and *debugging*, and the use of *abstract* concepts such as *data representation* (Papadakis et al., 2016).

While there is a debate over providing CT's definition (Ioannidou, Bennett, Repenning, Koh, & Basawapatna, 2011), it can be described as a set of skills that everyone can use in their daily lives to solve problems. It is another STEM practice where allows students to ask questions, design solutions, and communicate results. One of its main skills is *algorithmic thinking*, where "*code*" or "*coding*" are used to describe the action of an "*algorithm*". The code is a vehicle for the children's CT's development in a STEM context (Lego, 2016a, 2016b).

CT is regarded as a cognitive process of problem-solving using the above skills, and in primary education consists of a new philosophy for students approaching everyday life problems (Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011).

On a simple basis, CT involves six different *concepts* (*Logic*, *Algorithms*, *Decomposition*, *Patterns*, *Abstraction*, and *Evaluation*) and five *approaches* (*Tinkering*, *Creating*, *Debugging*, *Persevering* and *Collaborating*) to work (Atmatzidou & Demetriadis, 2014; Shute, Sun, & Asbell-Clarke, 2017; Zahilah Mohamed Zaki, Wong, & Ridzwan Yaakub, 2019). Other researchers suggest more, e.g. Brennan & Resnick's framework involves seven concepts that are highly useful in programming (and non-programming) contexts: *sequences*, *loops*, *parallelism*, *events*, *conditionals*, *operators*, and *data* (Brennan & Resnick, 2012; Pugnali, Sullivan, & Umashi Bers, 2017). In the literature there are also references to CT's *components, dimensions.*

The **CT skills** can be defined as the following:

- **Decomposition (or Segmentation):** The ability to simplify a problem into smaller parts in order to ease the process of finding a solution. In this way the problem becomes easier to be explained to another person, or to be separated into tasks. Decomposition frequently leads to Generalization.
- **Generalization (or Pattern Recognition):** The ability to recognize the parts of a task that are known, or have been seen somewhere else, that frequently leads to easier ways of designing algorithms.
- **Algorithmic Thinking:** The ability to create an ordered series of steps with the purpose of solving a problem.
- **Evaluating (or Debugging):** The ability to verify whether or not a prototype works as intended, and if not, to identify what needs to be improved. It is the process to find and correct mistakes within a program.
- **Abstraction:** The ability to explain a problem or a solution by removing unimportant details, meaning to being able to conceptualize an idea.

### Using an Engineering Design Process to Develop CT skills

A process to develop students' CT skills is to use the engineering design process, a series of phases that guide them toward a solution. These phases use and develop some of their CT skills. A simplify CT developing skills process is presented below (Lego, 2016a):

1. Define the problem. Usually the problem or project has many details. It is easier to solve it if can be broken down into smaller easier to understand parts (*Decomposition* or *Segmentation* CT skill).
2. Plan the solution. Imagine different solutions to the program and try to solve it by making a detailed plan. The plan will define clearly the steps need to be done to reach the solution. By identifying the parts of the task that might have seen before, *Generalization* CT skill is developed.
3. Try. Build and program the solution using programing languages and ER tools. By writing the code of the solution *Algorithmic Thinking* CT skill is developed.
4. Modify. Evaluate the solution if it meets the success criteria. If it is not, modify it and go through the above steps one more time. By changing, fixing, debugging, or improving some part of the solution *Evaluating (or Debugging)* CT skill is developed.
5. Communicate. By explaining the final solution that meets the success criteria. with the right level of detail, abstracting the unnecessary details, *Abstraction* and *Communication* CT skill are developed.

## EDUCATIONAL ROBOTCIS EVENT

In November 2019 the Municipality of Agia Varvara in collaboration with the University of West Attica handled an ER physical event, to help students and their parents learn about STEM Education and ER, and on the other hand to measure their interest in future ER seminars within the municipality (Municipality, 2019a; Vamvakopoulos, 2019).

The interest for the event turned out to be unexpectedly large and the event's venue "Ioannis Ritsos" theater - cinema (400 people capacity) was filled, as the -relatively small- Municipality of Agia

Varvara has 26.550 residents (2011 census), while the total number of students from the 8 preschools, 6 elementary schools, 3 junior high schools and 2 high schools do not exceed 1800 persons (Municipality, 2019b; Wikipedia, 2019).

Pre and post questionnaires were distributed to record students and parents' views and their interest in STEM education, ER and CT. In the pre-event questionnaire (Figure 3) students and parents were asked to provide answers to a series of questions related to their demographics, general background (about STEM education, ER, CT, etc.), and their intention to participate in STEM Education seminars, to design and develop robots, and in general to engage in ER workshops. A small sample of the pre-questionnaire is presented below in Appendix Table 1.

In the post-event questionnaire (Figure 4) students and parents were asked to provide answers to a series of questions related to their demographics, and their understanding about STEM education and CT. A small sample of the post-questionnaire is presented below in Appendix Table 2.

Questionnaires data were recorded, organized, cleaned and analyzed with the use of SPSS software. The result of these data's comprehension and interpretation was the decision to conduct a pilot research on a sample primary school students in the form of an experiential seminar, to investigate the possibility of developing their CT skills by applying ER activities when they are asked to solve authentic STEM problems.

## RESEARCH

Data evaluation of the ER event, lead to the decision of conducting a pilot research, on sample students in the form of an experiential seminar and workshop, to investigate the possibility of developing their CT skills by applying ER activities when they are asked to solve authentic STEM problems (Figure 5 below shows a block diagram of the research's procedure followed).

Several studies are exploring the learning benefits from ER activities that are related to CT's skills, such as algorithmic thinking, collaboration, problem solving, etc. (Atmatzidou & Demetriadis, 2016; Caballero-Gonzalez et al., 2019; Catlin & Woollard, 2014; Fronza, El Ioini, & Corral, 2017; Ioannou & Makridou, 2018; Muñoz-Repiso & Caballero-González, 2019). However, these studies do not converge, as some report improving skills (Castledine & Chalmers, 2011), while others conclude that the identified benefits –if any- are not significant (Turner & Hill, 2007). Another important observation is that studies often do not refer to the degree of guidance to support the development of skills.

### Research Objectives & Questions

On the basis of the above observations, the purpose of the present research is to investigate the possibility of developing CT skills in primary school students by applying ER activities when they are asked to solve authentic STEM problems. Specifically, the research focuses on the CT's skills of *abstraction*, *generalization*, *algorithm*, *modularity*, *segmentation*, *debugging*, and *collaboration*. For this reason, the **research questions** that have been arisen and are to be answered are:

1. Do educational robotics activities enhance students' CT skills?
2. Do educational robots in authentic STEM applications create a more favorable environment for developing students' CT?

Figure 3. Pre-questionnaire's block diagram

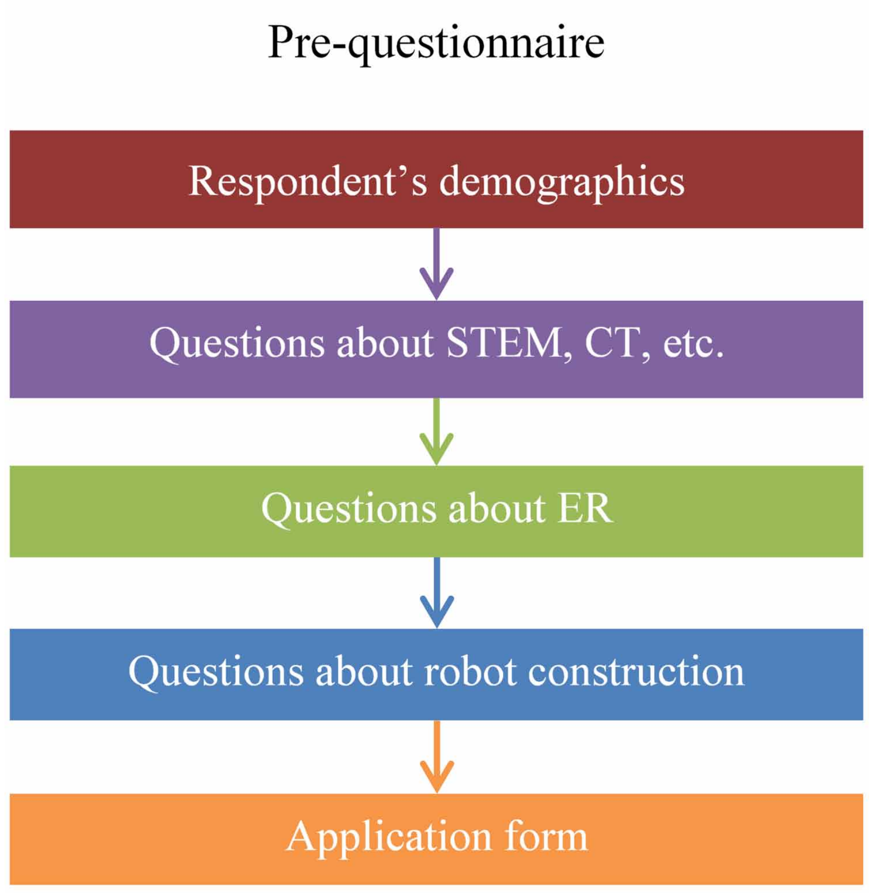



Figure 4. Post-questionnaire's block diagram

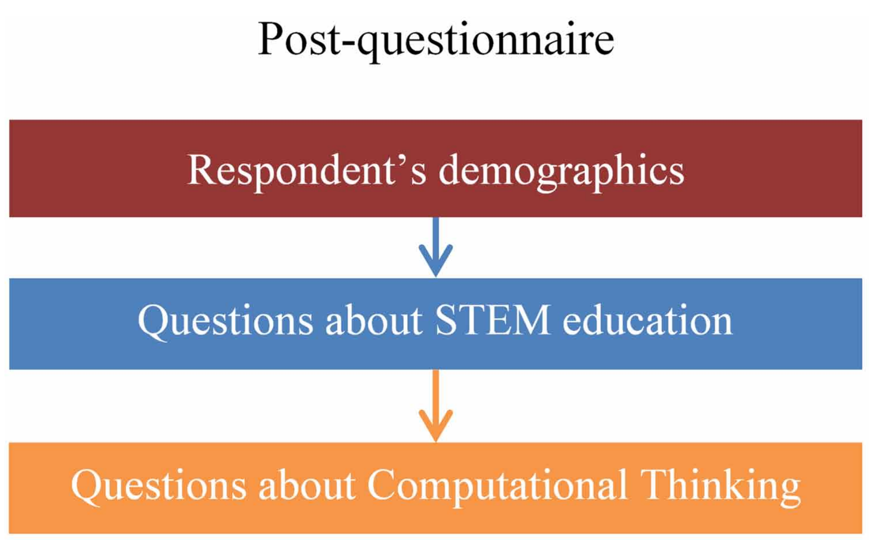
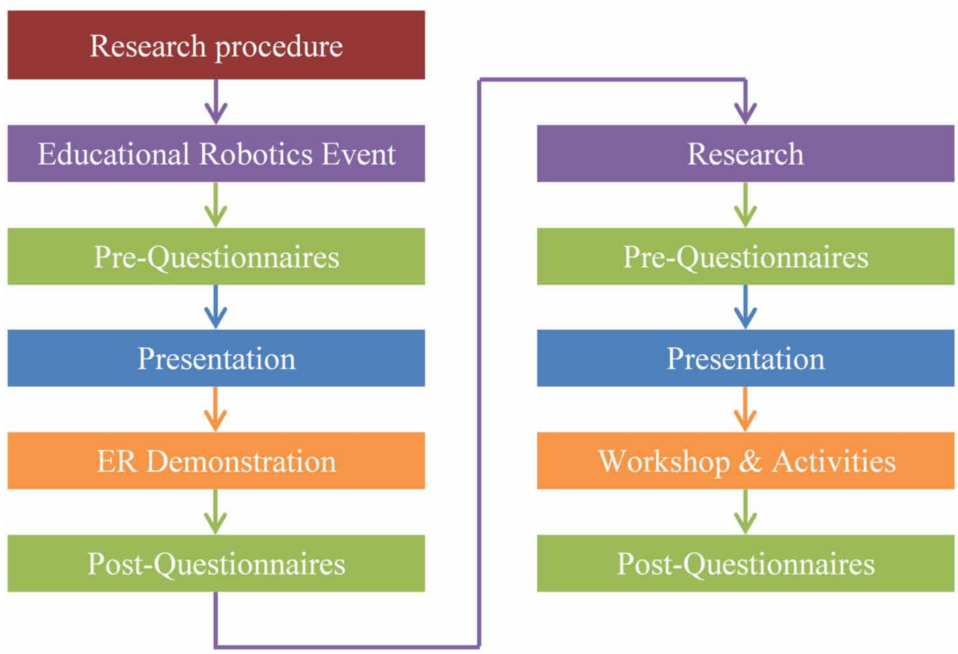


Figure 5. Block diagram of the research procedure




3. Do educational robots enhance students' engagement with programming?

## Methodology

In the following subsections, the research methodology is described including, the participants' recruiting, the research and teaching model, the support and guidance techniques, the workshop structure, and the whole procedure of the project. Prior the research, all necessary permissions were taken from the University of West Attica (School of Engineering - Department of Industrial Design and Production Engineering). In addition, all ethical issues (Cohen, Manion, & Morrison, 2018; Denzin & Lincoln, 2005; Kindon, Pain, & Kesby, 2007; Locke, Alcorn, & O'Neill, 2013) were taken seriously and participants/guardians were given a consent form (Cohen et al., 2018, p. 145), which had to be signed by their parents, which clearly stated the terms and procedure of the research. All the participants and their parents/guardians were well informed of the objectives of the study (Muñoz-Repiso & Caballero-González, 2019).

## CT's Skills and Problem-Solving Framework

Based on the research review (Marina Umaschi Bers, Flannery, Kazakoff, & Sullivan, 2014; Eguchi, 2014b; Grover, 2011; Kazakoff, Sullivan, & Bers, 2013; Lee et al., 2011; Touretzky, Marghitu, Ludi, Bernstein, & Ni, 2013) the proposed framework was a modified combination of the *model of the CT skills* (Atmatzidou & Demetriadis, 2014), and the *teacher guidance protocol* (Atmatzidou, Demetriadis, & Nika, 2018).

## CT Skills Model

The CT skills model proposed by (Atmatzidou & Demetriadis, 2014) creates the appropriate teaching environment for the CT skills development, which integrates the theoretical approach of the CT into ER activities. The model focuses on these five CT skills: *abstraction generalization*, *algorithm*, *modularization* and *segmentation*. The model is presented below in Appendix Table 3 (Atmatzidou & Demetriadis, 2016). In addition, guidance and supportive teaching techniques were used to support ER activities more effectively (see section *Supportive teaching techniques*).

## Teacher Guidance Protocol

Teacher guidance protocol (Atmatzidou et al., 2018) refers to the teachers' instructional interventions for supporting the development of the students' *metacognitive (MC)* and *problem solving (PS) skills*. It is based on the *Schoenfeld model* (Schoenfeld, 1992) where:

1. teacher has the role of facilitator and consultant, providing support in the form of hints, prompts, feedback etc.,
2. students are prompted to externalize their thinking through *think-aloud protocols* (Lochhead & Whimbey, 1987; Pate & Miller, 2011), and
3. students are guided to externalize their reflections on how they applied their strategies.

The modified protocol is presented below in Appendix Table 4.

## Supportive Teaching Techniques

### The Scenario

The *scenario* describes how students need to collaborate, such as roles and tasks distribution, the rules, the work phases, the deliverables, etc. A *collaborative scenario* or *script* is an explicit teaching contact between teacher and students on how to cooperate (O'Donnell & Dansereau, 1992). In this research three collaboration scenarios (*jigsaw technique*, *role play* and *think aloud*) were used and mixed together during the ER activities.

### Jigsaw Technique

The first collaboration scenario used was the *Jigsaw technique* (Barkley, Major, & Cross, 2014). According to this scenario (Aronson, 1978; Barkley et al., 2014; Callan, 2013):

1. Students are divided into groups of experts who specialize in a particular subject of the overall knowledge they need to acquire by the end of the lesson.
2. After understanding the teaching object, these groups are again divided and create new jigsaw groups, each consisting of a specialist in each subject.
3. In these new groups, each student has the responsibility to teach his or her piece to others.This technique is useful because it motivates students to take responsibility for learning something well,

so they can teach it to other members of the group. The Jigsaw technique is also an effective, strategy for explaining the learning scope and degree because students learn many topics at the same time during the activity. This research used a modified version of the Jigsaw technique which is presented below in Appendix Table 5.

### Role Play

The second collaboration scenario used was the *role play* (Kodotchigova, 2002), where students take on some roles within the group. In this research, the two roles (*Maker* and *Programmer*) are rotated cyclically after the end of each ER activity, so that all members of the team to knows and become familiar with all the roles. Collaboration is achieved through students' observance of the roles. The roles are clearly defined and each member know his/her responsibilities, however, it is often the case that the responsibilities of the different roles are confused in order to work better together.

### Think-Aloud Pair Solving Problem (TAPPS)

The third collaboration scenario used was the *Think-Aloud Pair Solving Problem (TAPPS)* (Alimisis, Moro, & Menegatti, 2016; Lochhead & Whimbey, 1987; Pate & Miller, 2011). At the end of the ER training activities, students were asked to solve an activity by following the TAPPS scenario which involves pairs of students taking the role of the *"listener"* and the *"solver"*. The solver solving the problem while the listener asks questions to prompt solvent to verbalize his thoughts and clarify his thinking (Pate & Miller, 2011). In this way students express their thoughts aloud while engaging in problem-solving activities to externalize the thinking process. However, listeners are not allowed to solve the problem or ask questions or make statements that guide the problem solver toward a solution (Lochhead & Whimbey, 1987). In this research, the solver role was played by each student and the listener role was the trainer. Students who solve the problem thinks aloud about the steps they have to take in order to reach the solution of the activity. The listener (trainer) listens to the steps suggested by the solver and tries to understand the rationale behind the steps suggested. He also suggests improvements or helps the solver through some key questions. This collaborative technique emphasizes the problem-solving process rather than the outcome, helping students to identify errors in logic. It also improves students' analytical skills and contributes to a deeper understanding of concepts and their use in new situations.

## Research Procedure

The study was conducted by a one-day duration seminar with a workshop (total duration of 6 hours). In all activities, the Lego WeDo 2 educational robotic platform was used (Education, 2019). Research's schedule is presented below in Appendix Table 6.

Both, research's objective and research's questions were measured using qualitative and quantitative analysis (see section: *Evaluation Tools*). The research was structured based on two stages: the *experiential seminar* and the *workshop*. At the beginning and at the end of the research, *questionnaires* were filled and *a group discussion* was held to gather quantitative and qualitative data.

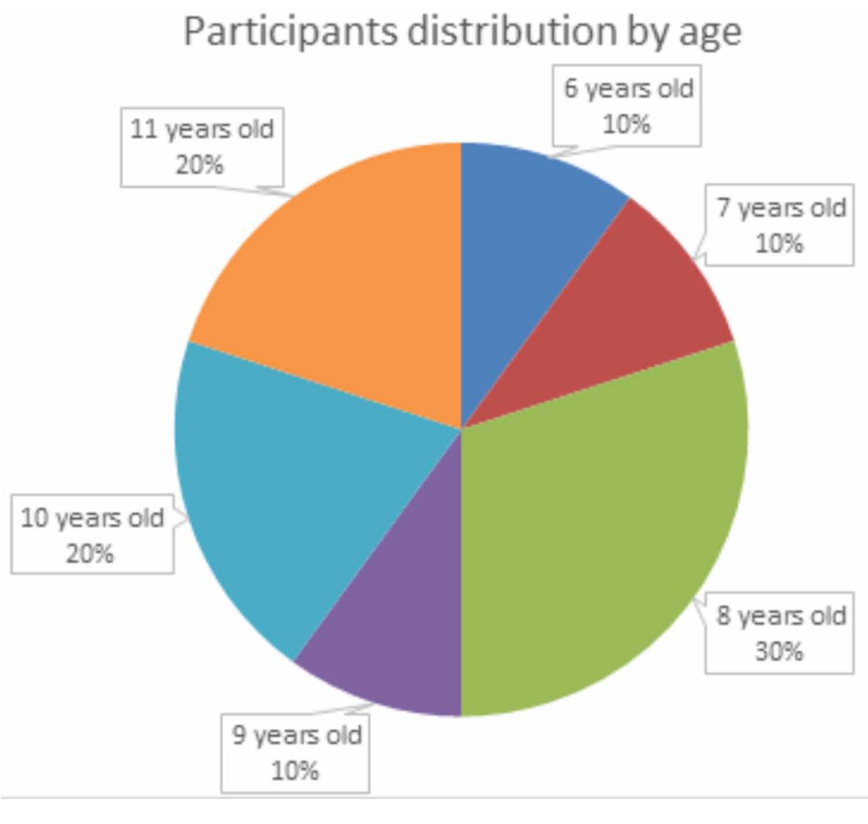
The purpose of the *introductory experiential seminar* was to introduce students to STEM education, ER and robots in general, team-based learning, project-based learning, role-play, CT, and how they all interact and participate in knowledge and skills building. Particular emphasis was given to the CT concepts, while a model for completing STEM problems that also involves CT stages, was presented. To make the model more comprehensible, a case study of a robot based on Lego parts was presented.

The workshop consisted of three phases: *introductory phase*, *training phase* and *practice (or challenge) phase* (see section: *Workshop's organization and structure*). Throughout the research, the role of the trainers was crucial. They guided, coordinated, motivated, supported the learning process for students to acquire skills and enrich their knowledge. Learning activities were individual and collaborative and the final conclusion was a very practical pedagogical approach.

## Participants

The research was held at the University of West Attica in November 2019. Participants were primary education students: 7 boys and 3 girls (70% boys – 30% girls) aged 6 – 11 years old (see Figure 6). Gender balance was not achieved in the sample, because participants were randomly selected from the application forms, filled at the previously educational robotics event that was took place in the municipality of Agia Varvara. The mean age of the participants was 8,8 years old (M=8,8). The seminar was organized by the lead researcher (author of this work), while trained primary education teachers and postgraduate students assisted in the practical aspects of the workshops/activities. Both researchers and the trained students/teachers (the *trainers*) were present at the introductory sessions and follow the same instructions, thus ensuring the equivalence of the training provided. Students were assigned to one of five groups according to their age, while two of them had an earlier basic acquaintance with educational robotics.

*Figure 6. Participants distribution by age*



## Workshop's Organization and Structure

### Roles and Groups

Students were divided into groups of two members based on their age, in order to work together to implement the ER activities. Each member of the group in each activity was assigned a role at a time and in each activity so as to be able to gain the experience of every role offers in problem solving process. Specifically, the roles assigned to the students were:

1. The *Maker* who is responsible for assembling the robot.
2. The *Programmer* who is responsible for observing and understanding the actions to be performed and authoring the program.

However, because the teams have consisted of only 2 members, students often had to perform two roles. The above organization of the groups was not accidental but was based on the above model adopted by this research.

### Workshop's Structure

Research's workshop was divided into three phases: i) *introductory phase*, ii) *training phase*, and iii) *practice phase*. A brief description of workshop's learning activities is presented below in Appendix Table 7.

### Introductory Phase

At the end of the *introductory experiential seminar (*see section *Research Procedure),* the *workshop's introductory phase* was used to introduce students to robotic platform Lego WeDo 2, exploring its characteristics and achieving a general understanding of its resource's functionalities. Specifically, all the key elements of Lego WeDo 2 were introduced such as *Lego bricks, building elements and mechanisms*, *WeDo 2.0 Smarthub*, *Medium Motor*, *Motion Sensor*, *Tilt Sensor*, *Batteries*, *Bluetooth dongle* and the *Lego WeDo 2.0 v1.9.30 programming environment*. Students were navigated and explored the Lego WeDo programming environment (Education, 2017) software and it's blocks (Figure 7). Then a simple program "Turn Lego Hub's Led On" (Appendix Table 7 - Activity 1.1) was implemented with the collaboration of the students to see how it works and how easy it is to create a complete program with only a few commands (blocks) and to arouse their interest.

In fact, when designing the program, children were asked to step by step how they would implement the problem even without programming knowledge. These steps were transferred to the programming environment and turned into blocks. In addition, a step-by-step program control was introduced to correct any mistakes made, introducing CT's concept of *Debugging*.

### Training Phase

In the *training phase*, the trainer provides introductory training ER activities to support participants with CT methodology. The *Lego's STEM project development model* and a simplified *process for developing*

*Figure 7. A subset of LEGO® Programming Blocks*


*CT skills* (Figure 8) was introduced and explained, to develop participants with a mindset and a lifelong set of practices. Among these practices students learned about CT; was described as a group of skills (ask questions, design solutions, create an algorithm, communicate results) to solve every day problems (Lego, 2016b).

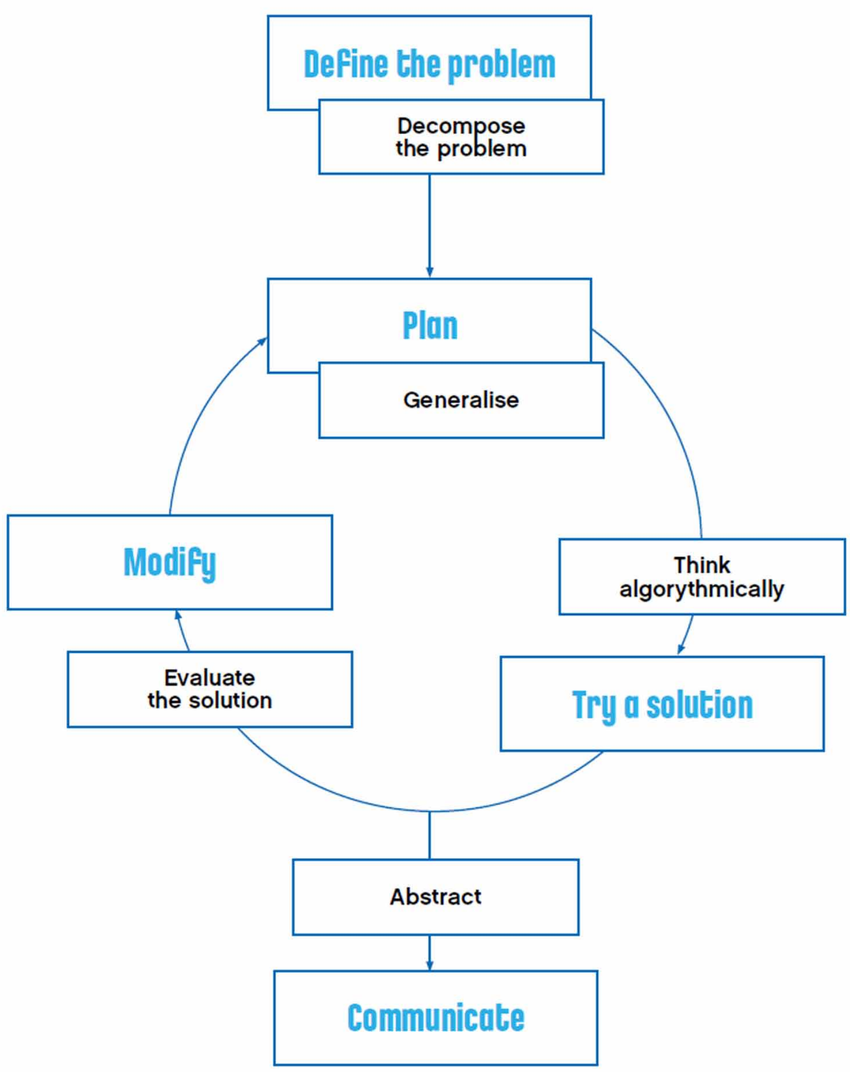
The workshop's training activities were based on Barrow's basic model for *problem-based learning* (Barrows, 1996). Learning is *student-centered* with small work groups led by researchers-trainers. The ER activities contain authentic STEM problems through the analysis and solution of which knowledge was acquired through self-directed learning. The role of the *trainers* is mediating, guiding, supporting the work and cooperation between students. In cases of mismanagement, they supported students and guided them in identifying the cause and correcting the error. In addition to discussing with the group, questions such as:

- *Why is this happening?*
- *Why isn't the other happening?*
- *Explain how ..?*
- *What if ...?*
- *How would you describe in steps what solution?*
- *What else does the problem look like?*
- *What other features could we add?*

to help students understand and deepen concepts related to problem-solving and CT.

Students are divided into groups of two; the choice for small groups was a necessity due to the small number of participants. One member becomes the *leader* of the current ER activity and takes the responsibility to teach the other team member. For keeping equality, in every activity, the *leader* is changed. There are only two key roles assigned to team members: the *maker* and the *programmer*. *Maker*

Figure 8. Lego's process for developing CT skills (Lego, 2016b, p. 9)




is responsible for the robot's hardware development and assembly, while the *programmer* is responsible for observing and understanding the actions to be performed and authoring the program. Each role is clearly defined and each member knows his/her responsibilities. In this way, students are motivated to take responsibility for learning something well, so they can teach it to the other member of the group. Students research their part of their problem and collaborate to exchange information. Collaboration is achieved through students' observance of the roles. Members' roles are rotated cyclically after the end of each ER activity so that all members of the team to know and become familiar with all the roles. Trainers oversee the whole process and intervene if necessary. They

1. emphasize to students the importance of the teamwork and collaboration,
2. enhance the kindness of each other,

3. approach the problems created between team members and uses them to provide feedback to the class, thereby helping the collaboration between students, and
4. make sure they understand that ultimately everyone will be a winner, actively participating in the above educational process.

At the end of the ER activities, teamwork and student evaluation are given. Post-questionnaires and group-discussion are taking place. The workshop's ER activities were organized based on the planned objectives:

"1st training activities" worked on the *algorithm* CT's dimension. Children had to create *linear sequences* of instructions to "turn Lego Hub's Led On/Off in various ways: turn On/Off, change color, blink, etc.".

"1st CT activities" worked on *generalization* CT's dimension. Children had to expand the existing solution "turn Led On/Off" in a given problem "Operate Led as a traffic light" in order to cover more possibilities.

"2nd training activities" worked on the *algorithm and generalization* CT's dimensions. Children had to create more advanced *linear* and *parallel sequences* of instructions to "turn Lego Hub's Led On/Off in various ways and play sounds alone or simultaneously".

"2nd CT training activities" worked on the *segmentation* and *modularization of* CT's dimensions. Children had to break apart activities "turn Lego Hub's Led to blinks while plays two sounds simultaneously" into smaller/single ones "turn Lego Led blinks" & "play a sound" that are easier to be solved. Children had also to develop autonomous sections of code to be used for the same or different problems.

"3rd training activities" worked on the *algorithm, generalization, segmentation,* and *modularization* of CT's dimensions. Children had to create even more advanced *linear* and *parallel sequences,* segment difficult problems and modularize the development of autonomous processes that perform a specific function.

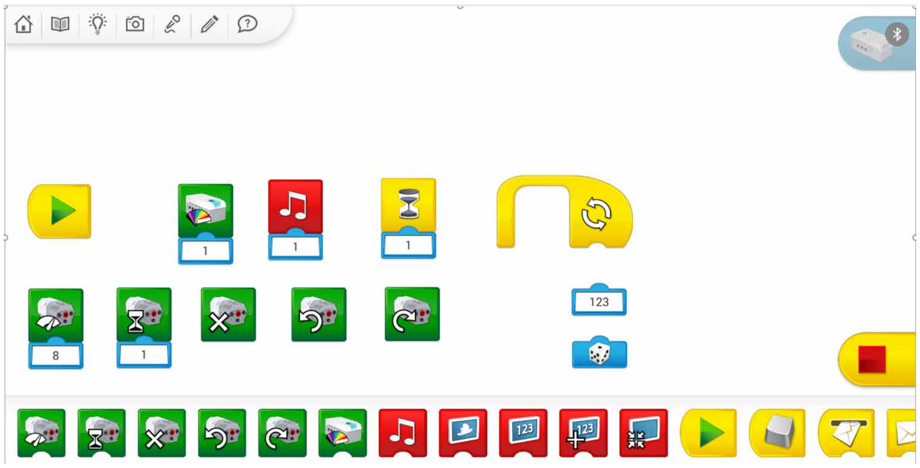
"3rd CT training activities" also worked on the *algorithm, generalization, segmentation,* and *modularization* of CT's dimensions. Students has given a more difficult problem (activity) to develop.

Finally, "4td training and CT activities" (*workshop's practice phase*) worked on all CT's dimensions including *abstraction.* Students among others had also to separate the important from the redundant information, analyze and specify common behaviors or programming structures between different scripts and identify abstractions. Simultaneously and in all the above activities, children were introduced to the *debugging dimension* and its necessity. They were provided with simple sequences containing errors that they had to detect and correct to successfully complete the challenge.

## Practice Phase

In the *practice* or *challenge phase*, the trainer provides participants a free-form demanding project but within their means, to develop. Participants –as always- work in groups, collaborate and perform the activities of CT's solving. Upon projects' completion, an open discussion is placed on what happened and why, by analyzing the strengths and weaknesses of groups' solutions and strategies they followed.

*Figure 9. Lego WeDo 2.0 v1.9.30 programming environment*



## Project's Scenario

The primary goal of the seminar was the participants to focus on solving an authentic real STEM problem through the programming of a robot, in particular, a robot model build on the Lego WeDo 2 educational robotic platform (Education, 2019).

The students (Figure 10) are given clear instructions not to deal with vehicle's aesthetics so that it would bring to a fire truck, but focus more on its operation in order to follow the below authentic problem scenario:

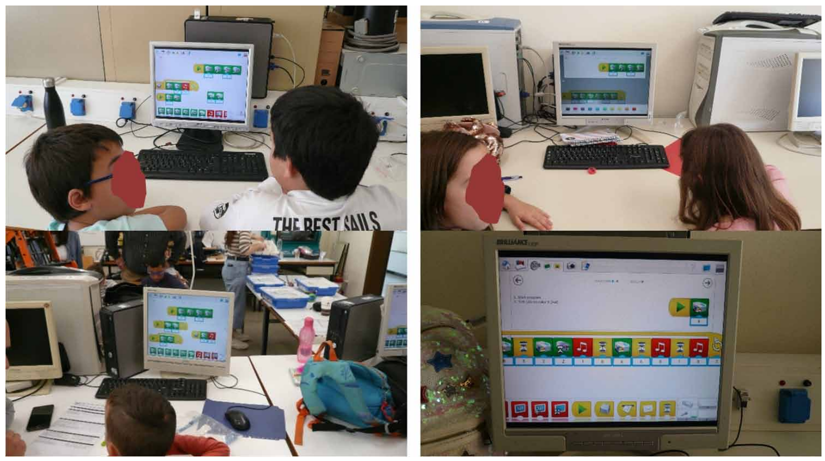
*When the robot (fire truck) starts up, it moves forward at low speed. When the robot's operator presses the F key on PC's keyboard (F for Fire), then the robot moves at its maximum speed, flashing its beacon (switching its led from red to white), and sounding its siren to simulate the siren of a real fire truck. When the robot's operator presses the B key (B for Begin), the robot moves forward at low speed, its beacon light is off and, its siren is not sounding. Finally, when S key is pressed (S for Stop), the robots stops moving, beacon and siren switch off (no led flashing, no sound).*

The above scenario was chosen because of the simple and flexible implementation of the fire-truck (robot) since it only needed a single motor and the Lego's brain (Lego Hub) to be built. Lego WeDo 2.0 v1.9.30 was used as the programming environment for the robot project. It uses -an easy to develop-block-based language suitable for primary education students.

## Evaluation Tools

In this research, CT development was investigated by the combination of quantitative and qualitative data collection and analysis tools. For the qualitative evaluation were used:

*Figure 10. Participants groups*



1. Trainers' *observations* during students' CT activities. The learning level of students' CT was observed through the performance achieved by the participants in each of the workshop's activities. *Brennan & Resnick CT framework* was exploited so *sequences*, *patterns*, and *debugging* CT's dimensions were explored (Brennan & Resnick, 2012; Caballero-Gonzalez et al., 2019). In addition, from *Bers' TangibleK curriculum,* two more variables: *correspondence* and *control flow,* were observed and assessed (Marina U. Bers, 2010; Marina Umaschi Bers et al., 2014).
2. The trainers' *rubrics record grid* (Lego, 2016b, p. 32) that records any type of observation they believe is important for each student. This observation rubric used to i) evaluate students' performance at each step of the process, and ii) provides constructive feedback to help them with their progress. This rubric is based on the following progressive stages (Appendix Table 8):
    a. *Emerging stage*, where students are at the beginning stage of content knowledge development. It is the ability to understand and apply content, and/or demonstration of coherent thoughts about a given topic.
    b. *Developing stage*, where students are able to present basic knowledge only, but cannot yet apply content knowledge or demonstrate comprehension of the concepts being presented.
    c. *Proficient stage*, where students have concrete levels of comprehension of the content and concepts and can demonstrate the topics, content, or concepts being taught.
    d. *Accomplished stage*, where students can take concepts and ideas to the next level, apply concepts to other situations, synthesize, apply, and extend knowledge to discussions that include extensions of ideas.
3. *Students produced material* such as artifacts, programs, notes.
4. *Group discussion* before and after the workshop's ER activities. The final discussion was conducted in order to better capture students' views of the whole activity, as most of the questions in the post-questionnaire were closed-ended. In addition, some of the some of the questions contained in the post-questionnaire were repeated in the interview in order to verify the views that the students recorded in the questionnaires.

For the quantitative evaluation were used:

1. *Pre-questionnaires*. Before starting the seminar, a *pre-questionnaire* was conducted to obtain students' demographics and previous knowledge background about programming, robotics, their core interests, and how they relate to digital technology and ER (see sample below in Appendix Table 9).
2. A *self-assessment CT rubric*. This *CT rubric* corresponds to an adaptation of the *self-assessment statement* used by Lego Education CT Teacher's Guide (Lego, 2016a, p. 31), that reflects on the work students have done, and it helps to encourage reflection and set goals for the next project (Appendix Table 10). Students recorded their behaviors at the end of ER activity, using a Likert scale from 1 to 5 (1 = never, 2 = almost never, 3 = sometimes, 4 = frequently, 5 = always).
3. *Post-questionnaires*. The CT post-questionnaire was individualized and given to students after the end of the workshop's practice phase, in order to investigate:
   a. students' participation and understanding in ER activities when assigned an authentic STEM problem.
   b. basic indications of students' happiness and involvement when using/working with robots and programming,
   c. whether students can use the CT concepts in problem-solving, and
   d. the participation/teamwork.

The CT post-questionnaire (Appendix Table 11) was formulated by the researchers and it was adapted from the CSTA K–12 CS standard (Lego, 2016a) and Brennan & Resnick's work (Brennan et al., 2019). It contained four open-ended and sixteen closed-ended questions using a Likert scale from 1 to 5. Each questionnaire's question hides a CT skill in order to check whether students correctly used some concepts to solve the various problems and to what extent.

## RESULTS

In this study, the collected data from the questionnaires were examined by descriptive statistics to explore the group's means. The pre-questionnaires revealed the demographic profile of the participants and their relationship to digital technologies, programming, and robotics.

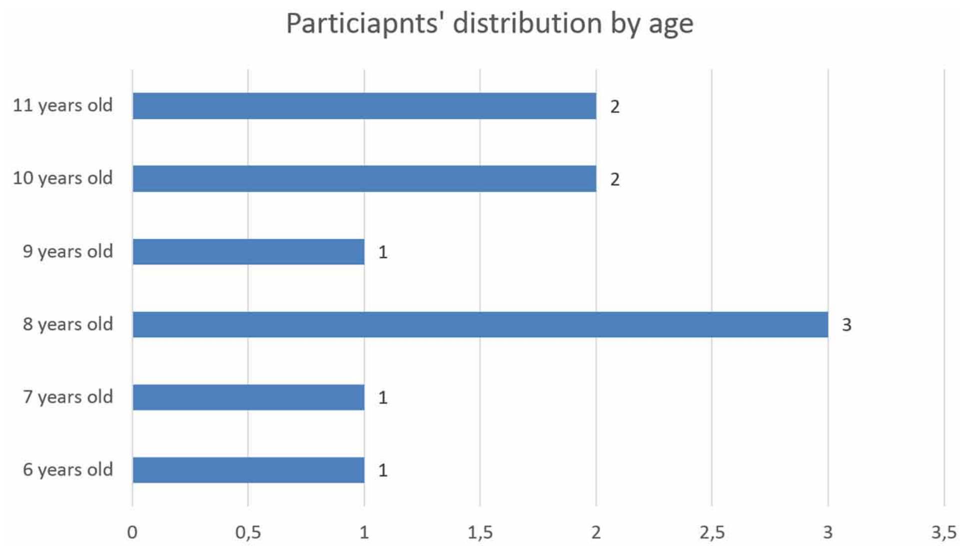
### Demographic Profile

Participants were primary education students from aged 6 to 11 years old (Figure 11), of which 7 boys and 3 girls. Gender balance was not achieved in the sample, because they were randomly selected from application forms. The mean age of the participants was 8,8 years old (M=8,8).

### Participants' Relationship to Digital Technologies

The following results were found in the imprint of the pre-questionnaire results (Appendix Table 12 & Table 13). On the pre-questionnaires, the whole majority (100%) of the participants have a computer, a tablet and Internet access from its home. However, only half the majority (50%) feel well familiar with

*Figure 11. Participants' distribution by age*




working with these devices. Participants mainly use them to play games (70%) and browse the Internet (60%). A small majority of them (30%) use them for school (or other) homework and do other things such as to watch movies or to practice in basic digital skills (see Figure 12).

## Participants' View of Robotics and Programming

The whole majority (100%) of the participants knew what a robot is, only half of them (50%) knew what is programming is, and the majority (80%) had dealt in the past with robots or robotics. In the question "*Why do you like robots and robotics*" the big majority of the participants answered that: "*they want to participate in national robotics competition*" (70%) or "*other robotics competition*" (60%) and "*they like to play with robots*" (60%). Half of the majority (50%) answered that "*they find it an interesting activity*", "*they think it's something creative*" and "*they want to learn to program*". However, only a half majority (50%) seems to "*want to know more about robotics*". The above observations may lead us to conclude that the participants find robot / robotics interesting because they relate to play (see below question 17.1 results) and competition (see below questions 17.8 and 17.9 results).

In terms of dealing with the *programming*, although the participants didn't have sufficient knowledge of what it is (see above Q7 results), or related experience (Q9 results), however, there is a moderate interest in engaging with it (Q17.6 results). Concerning *collaboration*, participants' opinions are divergent, some (60%) likes teamwork while others (30%) do not. This is particularly interesting because below after the workshop's completion, this trend seems to have been reversed.

## Post-Questionnaires

At the end of the workshop, participants completed the post-questionnaire accessing their overall experience and CT skills. All participants completed the post-questionnaires expect one, who did not feel well at the end of the workshop and his parent found it more appropriate to leave. That's why in all pre-questionnaires' results there is one "Did not answer". The following interesting results were found:

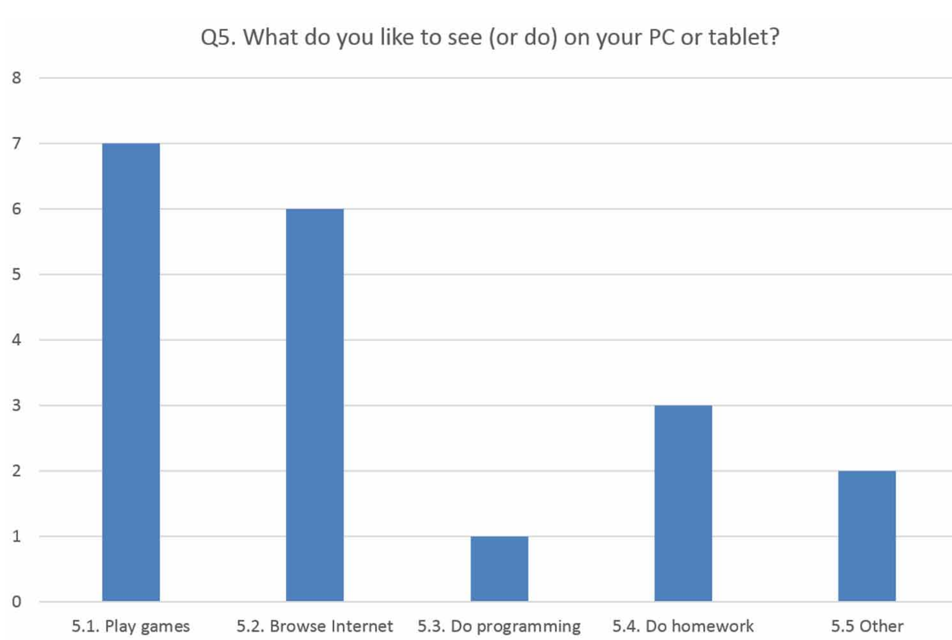
## Participants Engage to Robotics

The vast majority (90%) of the participants liked robotics after participant in the ER activities (Appendix Table 14). They enjoyed working with robots (see below Q1 results: 90%) and liked to engage with robots (Q3 results: 90%). This finding should probably not come as a surprise since the participants in their pre-questionnaires answers indicated that they knew about robots and robotics and dealt in the past with them. In addition, they volunteered to participate in the educational robotics event. However, the conclusion is that participants maintained their high expectations regarding robotics, even after participating in the ER activities.

## Participants' Relationship to Programming

The vast majority of the participants enjoyed programming (Q13 results: 90%) and feel quite confident (Q2 results: 80%) to program them after participating in ER activities (Appendix Table 15). They also think (Q14 results: 80%) that ER activities helped them to understand better programming. In the Q15 question "*How would you characterize yourself as a programmer?*" (Figure 13), the vast majority of them (Q15 results: 70%) thinks that he/she is a very good programmer. This personal perception may be related to participants' satisfaction (Q2 & Q13 results) about programming robots. However, it is interesting to know, that after participant to ER activities, the majority (Q16 results: 60%) of them would like to get more involved in programming, while a significant 30% was not very sure about that.

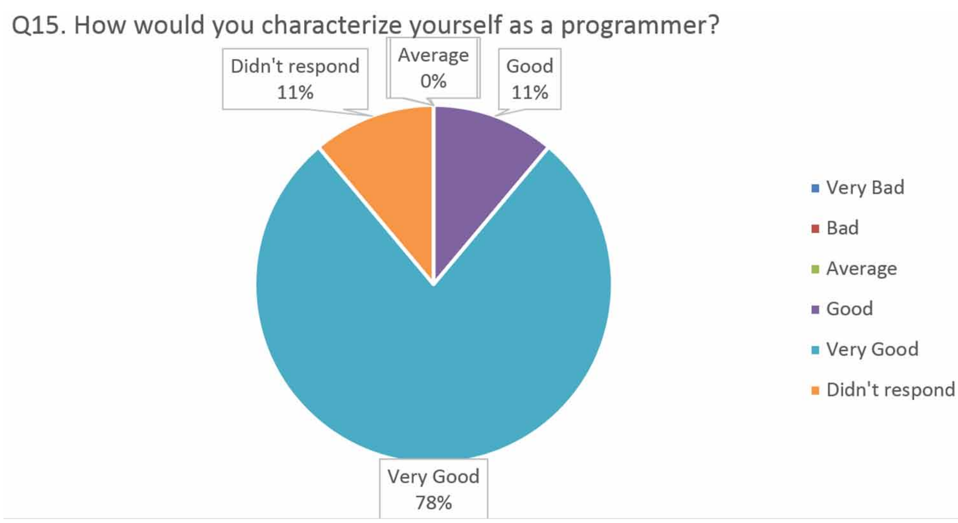
*Figure 12. Results of Question No5: "What do you like to see (or do) on your PC or tablet"*



## Participants' View of Collaboration

On the pre-survey, participants' opinions concerning collaboration were divergent, the majority (60%) of them stated that they like teamwork, while a significant 30% didn't like it (Appendix Table 16 & 17). However, after the workshop's ER activities this trend has been reversed since the vast majority (Q18 results: 70%) of the participants liked working with others. In the Q19 question "*Did you like that everyone in the team had their own role?*", a majority of 60% liked it while a significant majority of 30% didn't like it at all. This can be better explained below from researchers' observations regarding participants' teams. Finally, regarding the team's role (Figure 14), participants' majority (Q17 results: 60%) stated that they like both roles. The correlation of the above results may lead us to conclude that while students like to participate in groups, however, they prefer to alternate the team's roles during ER activities.

*Figure 13. Results of Question No15: "How would you characterize yourself as a programmer?"*



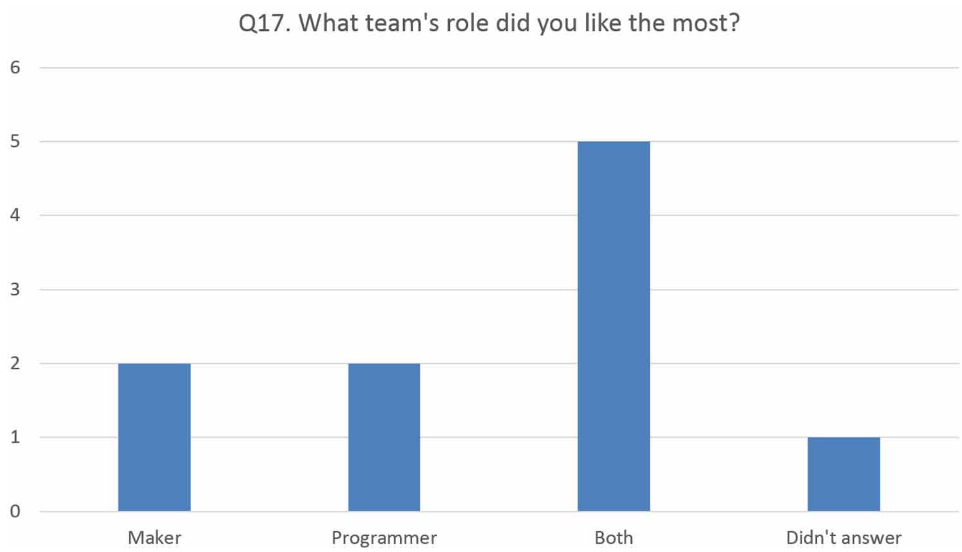

## Developing Participants CT Skills

In order to draw conclusions on the possibility of developing participants' CT skills, this research utilized data from participants' post-questionnaires and self-assessment CT rubric, along with the researchers'/trainers' notes and rubrics record grids. Appendix Table 18 summarizes the correlation of the questions with the respective CT skills.

Maybe, some of the questions above do not seem to clearly correlate with the respective skills such as R1, R2, R3, and R4, for this and additional explanations are given:

*R1. I defined the question or problem.* Many times a problem can have a lot of details, so in order to make it easier to solve, it has to be broken down into smaller parts (segments). By defining the problem in a simple way and by identifying some success criteria, a CT skill called *Segmentation* (or *Decomposition*) is developed.

*Figure 14. Question No17. What team's role did you like the most?*




*R2. I built a model*. To build a model means first to imagine different solutions to the problem and then make a detailed plan for executing one of them. Planning is the prerequisite to clearly define all the steps to complete the solution. By identifying the parts of the task that might have been seen before, a CT skill called *Generalization* is developed. Usually a model is complex enough and may be composed by a lot of details, that have to be broken down (*Decomposition* or *Segmentation*) into smaller parts. The ability to distinguish these components (in a software or hardware context) with clearly defined areas of responsibility is another CT skill called *Modularization*.

*R3. I programmed a solution (for my model)*. With model's completion, it is the turn of programming it, an important task that involves: - *Software Segmentation*, to simplify software problem into smaller parts to ease find a solution,- *Software Modularization*, to design a software consisting of distinct processes (e.g. subroutines or Blocks) with clearly defined areas that will be used in the main program (one or more times), or in different programs and perform a specific function, and- *Algorithmic Thinking* (or *Algorithm*), the ability to create an ordered series of steps with the purpose of solving a problem.

*R4. I tested my solution and made improvements*. To test a solution means to evaluate according to whether or not the program and model are meeting the success criteria. So, it is necessary to determine what is needed to change, fix, debug or improve, evaluation skills that constitute the *"Debugging"* CT skill.

Furthermore, the fact that all activities were done collaboratively explains the association of most questions with the *Collaboration* CT skill. Appendix Table 19 presents post-questionnaires questions' analysis. Appendix Table 20 presents participants' computational thinking self-assessment rubric's analysis. The comparison of these rubrics was used to draw conclusions related to CT skills development.

To investigate the possibility of developing CT skills in participants who participated in ER activities when they are asked to solve authentic STEM problems, mean score values (before and after) were calculated from self-assessment CT rubrics. Appendix Table 21 presents these results where a slight increase is imprinted. However, several questions were not answered by the participants (such as final's question R3 where the abstinence was a remarkable 30%) so it is not advisable to obtain reliable results. The results of each CT skill are illustrated separately in Appendix Table 22 and were calculated by the weights of each associated question as it is presented in Appendix Table 18.

The above data are summarized in the Appendix Table 23 where it appears that the **majority of participants' develop CT skills** when they are engaged with ER activities to solve authentic STEM problems. The highest scores are recorded in the skills of *Modularization* (75%), following with *Collaboration* (65%), *Abstraction* (64%), *Segmentation* (62%), *Debugging* (60%), *Generalization* (50%), and finally *Algorithm* (42%). These results show only basic trends since the participants' sample was very small and the percentage that did not answer the questions was significant (a mean of 17%). However, these results are verified by triangulating researchers' observations (see section *Qualitative data analysis*) and *Trainers' rubrics record grid* and appear to have a reasonable basis.

The first overall observation of the trainers was that the majority of the participants were able to understand the question at hand and define the problem of the CT activity. In particular, it was not difficult for them to break it down into smaller parts to solve. This is verified by the 62% score in *Segmentation*. They also had great flexibility in explaining the problem or solution easily, removing unimportant details, which is reflected in the 64% *Abstraction* score.

With respect to problem's program, while the participants showed great flexibility in *Modularization*, and easily reused previously written Blocks (is verified by the 75% score), many times they found difficulties in making their code works as it was supposed to do. This is reflected by the low 42% score in the *Algorithm*. Despite their difficulties in finding the correct algorithms, the participants showed a complete understanding of whether or not their project works as it was intended and they had the ability to identify what had needed to be improved. This is explained by the 60% score in *Debugging*.

Participants were not very sure of using the *Generalization* CT skill (50% score) as they usually couldn't recognize patterns, that is the parts of a task that are known or that had been seen somewhere else. In this way, they could be driven to easier ways of designing their algorithms, which is verified by the equally low score in *Algorithm* (42% score).

Finally, on *Collaboration* CT skill (65% score), the majority of the participants had established a good collaboration from the beginning of the workshop or improved it subsequently (see section *Participants' view of collaboration*).

## Qualitative Data Analysis

Following is the qualitative data of the present study, which was drawn from the structured observation and the researchers' notes. Through the oral communication of the participants, their attitudes and gestures, data were collected that provide a comprehensive understanding of their cooperation, their enjoyment of the ER activities, their preference for their role in the team, the advice and encouragement they exchanged, their confidence and the end of disagreements and corrections that took place during the process. The results shown below are for each individual group.

### 1st Group

The 1st group consisted of two boys 11 years old. Both members had previously been involved in robotics and programming. From the beginning, both group members worked together smoothly. They took on their roles without controversy, but along the way, they agreed to keep them. The researcher did not interfere with their decision as their collaboration was exemplary; they discussed, exchanged information and thoughts and worked in support of one another.

Participant A took on the role of the programmer while participant B took on the role of the maker. The two were experimenting together, in addition, participant A was constantly testing new Block combinations, while participant B was exploring alternatives solutions. Although participant B had the role of the maker he was actively involved in solving the algorithms.

Both members were confident in their choices. Without showing their enthusiasm, they worked tirelessly and together to solve all the CT activities. There were no conflicts or disputes. It was the team that came first, very close to the project's scenario completion, and while not completing it, they jointly demonstrated their satisfaction with their performance. These participants liked the project's challenge, due to the fact that it was "food for thought" for them.

## 2nd Group

The 2nd group consisted of two boys (10 and 8 years old respectively). Both members had previously been involved in robotics and programming. From the beginning, participants had a good collaboration. Participant B was a little more experienced in programming so in the beginning, he took on the role of the programmer. Participant A took on the role of the maker. However, they agreed to keep these roles along the way. The researcher respected their decision as they showed that they can collaborate effectively and segment CT activities.

Participant A was particularly expressing admiration gestures and shouts of joy when successfully completing the task. Participant B was more cooperative, and he also expressed his joy when they completed the activities. His interest grew when he saw the robot moving, he immediately wanted to make improvements by trying out new combinations of code. Towards the end of the workshop participant B left because he felt sick, participant A remained alone but his interest maintained.

Both participants asked questions to the researcher in order to understand and overcome their obstacles. Despite their queries, they seemed to be self-confident and with their good cooperation, they completed successfully the ER activities.

## 3rd Group

The 3rd group consisted of a boy and a girl (6 and 7 years old respectively). Girl had dealt with robotics in the past, while the boy did not. From the beginning of the ER activities, the girl had been imposing the boy. Although the modified Jigsaw technique that was followed by this research was not to appoint a leader, the girl assumed a leading role. She had full control of the computer mouse, despite all the boy's attempts to direct it at times. She also wanted to be involved in both roles (programmer and maker) at the same time. Possible, the age difference –which is significant in this target group- and her past experience in robotics, have led the girl to take the initiative role and build confidence.

In the first activity, she was assigned the role of a programmer and the boy the role of maker. He seemed to be out of control, having difficulty communicating and concentrating. However, his interest remained undiminished and then amplified when the roles changed and he took on the role of programmer.

At the beginning of the training the boy was received instructions and observations from the girl, but then the relationship between them was balanced and communication between them was restored. Later they were working in teams, expressing great joy when completing activities and communicating more effectively. During the activities, both members expressed particular interest in resolving the activities and they frequently asked questions to their researcher-trainer in order to understand and resolve them.

### 4th Group

The 4th group consisted of two boys 8 years old. Both members had previously been involved in robotics. From the beginning participant A wanted to play a leading role and did not want to collaborate with participant B. He was so persistent and forceful that the researcher had to intervene several times to restore balance in the team. At the beginning of the ER activities participant A took over the role of the programmer but he wanted to be actively involved in both roles, causing participant B's reaction. Participant B was more sympathetic and willing to cooperate. However, participant A's interventions sometimes led him to more dynamic reactions.

Initially, team members' communication and collaboration was problematic, but in the course of time, they cooperated better. In their positives were that they quickly and successfully completed all their ER activities and experimented with several solutions.

At the CT skills level, both members had proven to knowledge and skills throughout the activities, especially in algorithm, segmentation, process and debugging. At the end of the workshop, both members indicated that they were satisfied and expressed their interest in applying themselves in future workshops. This was particularly positive as it appeared that through their confrontations they were able to strike a balance and overcome the difficulties of mutual collaboration.

### 5th Group

The 5th group consisted of two girls 10 years old. Both members had previously been involved in robotics. From the workshop's beginning participants had an excellent collaboration. Participant A took on the programmer role and participant B took on the maker role, though it seemed that she had more programming experience than the other. This was not an obstacle as participant A was consulted by participant B, and B was willing to help A to understand ER activities algorithms. They took on their roles without controversy, but along the way, they agreed to keep them. The researcher did not try to change their decision as their collaboration was exemplary; they discussed, helped each other, exchanged information and thoughts and worked in support of one another. They both interacted with the researcher, asking questions to understand the algorithms, suggesting and testing solutions to complete the CT activities.

Participant A seemed to fully trust participant B, she asked her questions and listen carefully for her answers. She tried several programs' variations with the help of her partner. Participant B took on the role of supporting A, she was helping her and learning her programming. She successfully built the robot model by following closely the given instructions. Participant A was also involved in the construction providing assistance and support. After the robot's implementation, both focused on testing several programs. It was an exemplary group, with no rivalries and disputes with sincere support and participation in achieving the goal.

## Answers to Research's Questions

To map the results related to research questions, Appendix Table 24 summarizes the correlation of the questions that were utilized from participants' post-questionnaires and self-assessment CT rubric, and Appendix Table 25 summarize research's questions analysis.

In respect to this research, the following results were found in the imprint research's questions analysis:

1. by a vast majority (75%) it is confirmed that educational robots in authentic STEM applications create a more favorable environment for developing students' CT,
2. educational robots enhance students' engagement with programming since a significant vast majority (68%) is advocating this, and
3. educational robotics activities enhance students' CT skills, which is proved by the above section "*Developing participants CT skills*" and the 60% score in this question's analysis.

## DISCUSSION

The present study aimed to examine whether the implementation of ER activities and authentic STEM problems may develop students computational and digital skills. For the purpose of this study the activities that were utilized allowed P12 students to familiarize themselves with ER activities and use CT to fulfill the proposed activities. The results showed that robots, ER, and in general, STEM, are especially attractive to students due to their playful environment and they engage them in game-based problem-solving activities. Students developed CT and digital skills, that are abilities to abstract, generalize, collaborate, decompose, program (code) and debug, that made them solving the ER and CT activities easier. They participated with joy and pleasure, recorded by the researchers' notes.

Particularly, results showed that participants highest scores were recorded in the skills of *Modularization* (75%), and *Collaboration* (65%), moderate scores founded in *Abstraction* (64%), *Segmentation* (62%), and *Debugging* (60%), while they had more difficulty with the skills of *Generalization* (50%), and *Algorithm* (42%).

This study's findings are confirm numerous previous research results (Alimisis, 2013; Atmatzidou & Demetriadis, 2016; Marina Umaschi Bers et al., 2014; Eguchi, 2014a, 2014b, 2015; Kanaki & Kalogiannakis, 2018; Papadakis et al., 2016; Zhang & Nouri, 2019). It provides evidence that P12 students can learn to use and adapt CT skills. At the end of the intervention the majority of them were able to:

- "*define a problem*" in a simple way and by identifying key criteria, so decompose (segment) problem in smaller -easier to solve- parts,
- *"build a model"*, means imagining different solutions to the problem, making detailed plans, identifying components and tasks -with clearly defined areas of responsibility- of the problem's parts after decomposition, and reusing software and hardware components,
- *"program a solution for the model"*, that is *thinking algorithmic*, creating an ordered series of steps to solve the problem and using software's *segmentation* and *modularization*,
- *"test the solution and made improvements"*, means to evaluate and test the problem's solution and if it doesn't meet success criteria to debug it, then discover errors, change, fix and improve them.

- *"collaborate"*, with role assignments, discussion, exchanging information and thoughts, helping each other, and working in support of one another.

However, there are several limitations to this study that need be acknowledged:

- the small sample size,
- the sample selection method, although students were randomly selected, it was based on the application forms submitted to the ER event,
- the larger age difference of the students due to the above sample selection method,
- the students' ER and robots background variety, due to the sample selection method,
- the narrow geographical focus, all students came from the neighborhoods of West Attica, and
- the short duration of the implementation training of robotic activities.

Additionally, this research does not answer questions related to CT skills and i) solo vs group, ii) gender-based, and iii) age-based, CT activities implementation. Many of these limitations are expected to be fulfilled in the near future when a wider sample and improved evaluation tools will be used in the implementation of the same research.

## REFERENCES

Alimisis, D. (2009). *Teacher education on robotics-enhanced constructivist pedagogical methods*. Athens: School of Pedagogical and Technological Education. Retrieved from http://www.terecop.eu/en/Products1.html

Alimisis, D. (2013). Educational robotics : Open questions and new challenges. *Themes in Science & Technology Education*, *6*(1), 63–71.

Alimisis, D. (2014). Educational Robotics in Teacher Education: an Innovative Tool for Promoting Quality Education. In L. Daniel, I. Lūka, L. Rutka, & I. Žogla (Eds.), *Teacher of the 21st Century: Quality Education for Quality Teaching* (pp. 14–27). Newcastle, UK: Cambridge Scholars Publishing. Retrieved from https://books.google.gr/books?hl=el&lr=&id=x3ExBwAAQBAJ&oi=fnd&pg=PA14&dq=%5B7%5D%09Alimisis,+D.+(2014).+Educational+Robotics+in+Teacher+Education:+an+Innovative+Tool+for+Promoting+Quality+Educatio.,+In+L.+Daniela,+I.+Lūka,+L.+Rutka+and+I.+Žogla+(Ed.)+"Te

Alimisis, D., Moro, M., & Menegatti, E. (2016). *Educational Robotics in the Makers Era. Advances in Intelligent Systems and Computing* (Vol. 560). Springer; doi:10.1007/978-3-319-55553-9

Aronson, E. (1978). *The jigsaw classroom*. Retrieved January 2, 2020, from http://psycnet.apa.org/psycinfo/1980-51351-000

Atmatzidou, S., & Demetriadis, S. (2014). How to Support Students ' Computational Thinking Skills in Educational Robotics Activities. *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, 43–50.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661–670. doi:10.1016/j.robot.2015.10.008

Atmatzidou, S., Demetriadis, S., & Nika, P. (2018). How Does the Degree of Guidance Support Students' Metacognitive and Problem Solving Skills in Educational Robotics? *Journal of Science Education and Technology*, *27*(1), 70–85. doi:10.100710956-017-9709-x

Atmatzidou, S., Markelis, I., & Demetriadis, S. (2008). The use of LEGO Mindstorms in elementary and secondary education : game as a way of triggering learning. *Workshop Proceedings of International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 22–30.

Barkley, E. F., Major, C. H., & Cross, K. P. (2014). *Collaborative Learning Techniques: A Handbook for College Faculty*. John Wiley & Sons, Inc. Retrieved from https://www.wiley.com/en-us/Collaborative+Learning+Techniques%3A+A+Handbook+for+College+Faculty%2C+2nd+Edition-p-9781118761557

Barrows, H. S. (1996). Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*, *1996*(68), 3–12. doi:10.1002/tl.37219966804

Battelle for Kids. (2019). Framework for 21st Century Learning. *Partnership For 21st Century Learning*. Retrieved from https://www.battelleforkids.org/networks/p21/frameworks-resources

Bers, M. U. (2008). *Blocks to Robots Learning with Technology in the Early Childhood Classroom*. New York, NY: Teachers College Press.

Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, *12*(2), 1–20.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. doi:10.1016/j.compedu.2013.10.020

Blanchard, S., Freiman, V., & Lirrete-Pitre, N. (2010). Strategies used by elementary schoolchildren solving robotics-based complex tasks: Innovative potential of technology. *Procedia: Social and Behavioral Sciences*, *2*(2), 2851–2857. doi:10.1016/j.sbspro.2010.03.427

Brennan, K., Chung, M., Martin, W., Cervantes, F., Tally, B., & Resnick, M. (2019). *Supporting computational practices*. Harvard Graduate School of Education.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking* (pp. 1–25). Vancouver, BC, Canada: Annual American Educational Research Association Meeting. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Caballero-Gonzalez, Y.-A., Muñoz-Repiso, A. G.-V., & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics. In *TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 19–23). 10.1145/3362789.3362874

Çalik, M., Ebenezer, J., Özsevgeç, T., Küçük, Z., & Artun, H. (2015). Improving Science Student Teachers' Self-perceptions of Fluency with Innovative Technologies and Scientific Inquiry Abilities. *Journal of Science Education and Technology*, *24*(4), 448–460. doi:10.100710956-014-9529-1

Çalik, M., Özsevgeç, T., Ebenezer, J., Artun, H., & Küçük, Z. (2014). Effects of "Environmental Chemistry" Elective Course Via Technology-Embedded Scientific Inquiry Model on Some Variables. *Journal of Science Education and Technology*, *23*(3), 412–430. doi:10.100710956-013-9473-5

Callan, N. (2013). *ESL Jigsaws*. Retrieved from https://www.esljigsaws.com/assets/lessons/samplelesson.pdf

Castledine, A., & Chalmers, C. (2011). LEGO Robotics: An authentic problem solving tool? *Design and Technology Education: An International Journal, 16*(3), 19–27. Retrieved from https://ojs.lboro.ac.uk/DATE/article/view/1661/1562

Catlin, D., & Woollard, J. (2014). Educational Robots and Computational Thinking. Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, 144–151. 10.1017/CBO9781107415324.004

Chatzopoulos, A., Papoutsidakis, M., Kalogiannakis, M., & Psycharis, S. (2019). Action Research Implementation in Developing an Open Source and Low Cost Robotic Platform for STEM Education. *International Journal of Computers and Applications*, *178*(24), 33–46. doi:10.5120/ijca2019919039

Chevalier, M., Riedo, F., & Mondada, F. (2016). Pedagogical Uses of Thymio II: How Do Teachers Perceive Educational Robots in Formal Education? *IEEE Robotics & Automation Magazine*, *23*(2), 16–23. doi:10.1109/MRA.2016.2535080

Cohen, L., Manion, L., & Morrison, K. (2018). *Research Methods in Education* (8th ed.). London, UK: Routledge.

Daniela, L., & Lytras, M. D. (2018). *Educational Robotics for Inclusive Education*. Technology, Knowledge and Learning; doi:10.100710758-018-9397-5

Denzin, N. K., & Lincoln, Y. S. (2005). The Sage Handbook of Qualitative Research (3rd ed.). London, UK: SAGE Publications Ltd.

Education, L. (2017). Education WeDo Toolbox, (v1), 31.

Education, L. (2019). *LEGO Education WeDo 2.0*. Retrieved January 4, 2020, from https://education.lego.com/en-us/elementary/intro/wedo2

Eguchi, A. (2014a). Educational Robotics for promoting 21st century skills. *Journal of Automation*. *Mobile Robotics & Intelligent Systems*, *8*(1), 1–23. doi:10.14313/JAMRIS

Eguchi, A. (2014b). Learning experience through robocupjunior: Promoting engineering and computational thinking skills thourgh robotics competition. *ASEE Annual Conference and Exposition, Conference Proceedings*.

Eguchi, A. (2015). RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. *Robotics and Autonomous Systems*, *75*, 692–699. doi:10.1016/j.robot.2015.05.013

Ejiwale, J. A. (2013). Barriers to Successful Implementation of STEM Education. *Journal of Education and Learning*, *7*(2), 63–74. doi:10.1007/978-3-319-24436-5_20

Foundation, F. A. B. (2019). *Scaling a Community of Practice for Education in STEM through Digital Fabrication Reflection and Playbook*. Academic Press.

Fronza, I., El Ioini, N., & Corral, L. (2017). Leveraging Robot Programming to Foster Computational Thinking. doi:10.5220/0006310101090116

Gonzalez, H. B., & Kuenzi, J. J. (2012). Science, technology, engineering, and mathematics (STEM): A Primer. *Congressional Research Service*, 1–15. Retrieved from https://www.ccc.edu/departments/Documents/STEM_labor.pdf

Grover, S. (2011). Robotics and engineering for middle and high school students to develop computational thinking. *Annual Meeting of the American Educational Research Association,* (650), 1–15. Retrieved from http://web.stanford.edu/~shuchig/docs/AERA2011-Shuchi Grover-Robotics and Engineering for Middle and High School Students to Develop Computational Thinking.pdf

Heilig, S. (2015). K-12 STEM Education. *The Institute for the Promotion of Teaching Science and Technology, 1*(2), 112.

Higgins, J. M. (2013). The Future of Jobs. *World Future Review*, *5*(, 11–23. doi:10.1177/1946756712473437

Ioannidou, A., Bennett, V., Repenning, A., Koh, H., & Basawapatna, A. (2011). Computational Thinking Patterns. *Annual Meeting of the American Educational Research Association, 2*. Retrieved from http://www.agentsheets.com

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544. doi:10.100710639-018-9729-z

Ioannou, M., & Bratitsis, T. (2016). Utilizing Sphero for a speed related STEM activity in Kindergarten. *Hellenic Conference on Innovating STEM Education*.

Kanaki, K., & Kalogiannakis, M. (2018). Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses. *Education and Information Technologies*, *23*(6), 2673–2698. doi:10.100710639-018-9736-0

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255. doi:10.100710643-012-0554-5

Kindon, S., Pain, R., & Kesby, M. (2007). *Participatory Action Research Approaches and Methods*. London: Routledge.

Kodotchigova, M. A. (2002). Role Play in Teaching Culture: Six Quick Steps for Classroom Implementation. *The Internet TESL Journal, 8*(7). Retrieved from http://iteslj.org/Techniques/Kodotchigova-RolePlay.html

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. doi:10.1145/1929887.1929902

Lego, E. (2016b). *Teacher ' s Guide LEGO Education WeDo 2. 0 Computational Thinking*. The LEGO Group.

Lego. (2016a). *LEGO Education WeDo 2. 0 - Computational Thinking - Teacher's Guide*. LEGO Group. Retrieved from https://le-www-live-s.legocdn.com/sc/media/files/user-guides/wedo-2/computational-thinking-teacher-guides/computationalthinkingteacherguide-en-gb-v1-69cd677d61d-21b2a44999328c40e7122.pdf

Lego. (2019). *LEGO Education – LEGO History*. Retrieved December 30, 2019, from https://www.lego.com/en-us/lego-history/lego-education-604484c3b3de4314a678ef280d04d216

Lochhead, J., & Whimbey, A. (1987). Teaching analytical reasoning through thinking aloud pair problem solving. *New Directions for Teaching and Learning,* (30), 73-92. Retrieved from https://eric.ed.gov/?id=EJ355732

Locke, T., Alcorn, N., & O'Neill, J. (2013). Ethical issues in collaborative action research. *Educational Action Research*, *21*(1), 107–123. doi:10.1080/09650792.2013.763448

Marr, B. (2016). Why Everyone Must Get Ready For The 4th Industrial Revolution. *Forbes*. Retrieved from https://www.forbes.com/sites/bernardmarr/2016/04/05/why-everyone-must-get-ready-for-4th-industrial-revolution/print/

Marr, B. (2017). *Kroger: How This U.S. Retail Giant Is Using AI And Robots To Prepare For The 4th Industrial Revolution.* Bernardmarr.Com.

Marr, B. (2018). What is Industry 4.0? Here's A Super Easy Explanation For Anyone. *Forbes*, 4–7. Retrieved from https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/#5b3ba6539788

Mavrovounioti, V., Chatzopoulos, A., Papoutsidakis, M., & Piromalis, D. (2018). Implementation of an 2-wheel Educational Platform for STEM Applications. *Journal of Multidisciplinary Engineering Science and Technology*, *5*(10), 8944–8948.

Melkonian, S., Chatzopoulos, A., Papoutsidakis, M., & Piromalis, D. (2018). Remote Control via Android for a Small Vehicle ' s 2-Wheels Balancing. *Journal of Multidisciplinary Engineering Science and Technology*, *5*(10), 8964–8967.

Municipality, A. V. (2019a). *Educational Robotics Event for Elementary-High School-High School Students*. Retrieved January 2, 2020, from https://agiavarvara.gr/event/ekdilosi-ekpaideytikis-rompotikis-gia-mathites-dimotikoy-gymnasioy-lykeioy/

Municipality, A. V. (2019b). *Giannis Ritsos Municipal Theater*. Retrieved January 2, 2020, from https://agiavarvara.gr/building/dimotiko-kinimatotheatro-giannis-ritsos/

Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar*, *27*(59), 63–72. doi:10.3916/C59-2019-06

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performances. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interactions in cooperative groups: the theoretical anatomy of group learning* (pp. 120–141). Cambridge: Cambridge University Press.

Ospennikova, E., Ershov, M., & Iljin, I. (2015). Educational Robotics as an Inovative Educational Technology. *Procedia: Social and Behavioral Sciences*, *214*(June), 18–26. doi:10.1016/j.sbspro.2015.11.588

Papadakis, S. (2018a). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2018b). The use of computer games in classroom environment. *International Journal of Teaching and Case Studies*, *9*(1), 1. doi:10.1504/IJTCS.2018.090191

Papadakis, S., & Kalogiannakis, M. (2018). Using gamification for supporting an introductory programming course. The case of classcraft in a secondary education classroom. *Lecture Notes of the Institute for Computer Sciences*. *Social-Informatics and Telecommunications Engineering, LNICST*, *229*, 366–375. doi:10.1007/978-3-319-76908-0_35

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187. doi:10.1504/IJMLO.2016.077867

Papadakis, S., Polychronaki, K., & Tousia, C. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142. doi:10.1504/IJTCS.2018.090963

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, *1*(1), 95–123. doi:10.1007/BF00191473

Pate, M., & Miller, G. (2011). Effects of Think–Aloud Pair Problem Solving on Secondary–Level Students' Performance in Career and Technical Education Courses. *Journal of Agricultural Education*, *52*(1), 120–131. doi:10.5032/jae.2011.01120

Psycharis, S. (2018). STEAM in educations: A literature review on the role of Computational Thinking, Engineering Epistemology and Computational Science. *Computational STEAM Pedagogy*, *4*(2), 51–72. doi:10.5281/zenodo.1214565

Psycharis, S., Kalovrektis, K., Sakellaridi, E., & Korres, K. (2017). Unfolding the Curriculum: Physical Computing, Computational Thinking and Computational Experiment in STEM's Transdisciplinary Approach. *European Journal of Engineering Research and Science*, 19–24.

Psycharis, S., & Kotzampasaki, E. (2019). The Impact of a STEM Inquiry Game Learning Scenario on Computational Thinking and Computer Self-confidence. *Eurasia Journal of Mathematics, Science and Technology Education*, *15*(4). doi:10.29333/ejmste/103071

Pugnali, A., Sullivan, A., & Umashi Bers, M. (2017). The Impact of User Interface on Young Children's Computational Thinking. *Journal of Information Technology Education: Innovations in Practice*, *16*, 171–193. doi:10.28945/3768

Resnick, M., Ocko, S., & Papert, S. (1988). LEGO, LOGO, AND DESIGN. *Children and Interactive Electronic Environments, 5*(4), 14–18. Retrieved from https://www.jstor.org/stable/41514692

Roehrig, G. H., Moore, T. J., & Wang, H. (2012). *Is Adding the E Enough?: Investigating the Impact of K-12 Engineering Standards on the Implementation of STEM Integration*. School of Engineering Education Faculty Publications.

Ronsivalle, G. B., Boldi, A., Gusella, V., Inama, C., & Carta, S. (2018). *How to Implement Educational Robotics' Programs in Italian Schools: A Brief Guideline According to an Instructional Design Point of View*. Technology, Knowledge and Learning; doi:10.100710758-018-9389-5

Saito, T., Gunji, Y., & Kumano, Y. (2015). The Problem about Technology in STEM Education: Some Findings from Action Research on the Professional Development & Integrated STEM Lessons in Informal Fields. *K-12 STEM Education, 1*(2), 85–100.

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. *Handbook for Research on Mathematics Teaching and Learning*, 334–370.

Schwab, K., & Samans, R. (2016). Global Challenge Insight Report: The Future of Jobs. World Economic Forum, 1–167. 10.1177/1946756712473437

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. doi:10.1016/j.edurev.2017.09.003

Singmaster, H. (2008). *Seven Skills Students Need for Their Future*. Retrieved from https://asiasociety.org/education/resources-schools/professional-learning/seven-skills-students-need-their-future

Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, *48*(2), 105–128. doi:10.1080/15391523.2016.1146563

Swaid, S. I. (2015). Bringing Computational Thinking to STEM Education. *Procedia Manufacturing*, *3*, 3657–3662. doi:10.1016/j.promfg.2015.07.761

Texley, J., & Ruud, R. M. (2018). *Teaching STEM literacy: a constructivist approach for ages 3 to 8* (1st ed.). St. Paul, MN: Redleaf Press.

Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013). Accelerating k-12 computational thinking using scaffolding, staging, and abstraction. SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education, 609–614. 10.1145/2445196.2445374

Traurig, A., & Feller, R. (2008). Preparing Students for STEM Careers. *STEM Career Magazine*, *53*(9). doi:10.1017/CBO9781107415324.004

Tsupros, N., Kohler, R., & Hallinen, J. (2009). *STEM Education in Southwestern Pennsylvania the missing components*. Retrieved from https://www.cmu.edu/gelfand/documents/stem-survey-report-cmu-iu1.pdf

Turner, S., & Hill, G. (2007). Robots in problem - solving and programming. *8th Annual Conference of the Subject Centre for Information and Computer*, 82–85. 10.13140/2.1.2425.6800

Vamvakopoulos, P. (2019). *Event for Educational Robotics, with great interest*. Retrieved January 2, 2020, from http://proodeutiki.gr/?p=85159

Vasquez, J. A., Comer, M., & Sneider, C. (2013). *STEM Lesson Essentials, Grades 3-8: Integrating Science, Technology, Engineering, and Mathematics*. Portsmouth, NH: Heinemann. Retrieved from https://www.amazon.com/STEM-Lesson-Essentials-Grades-3-8/dp/0325043582

Vidakis, N., Barianos, A. K., Trampas, A. M., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating education in-game data: The case of an ancient theatre serious game. CSEDU 2019 - Proceedings of the 11th International Conference on Computer Supported Education, 1, 36–43. 10.5220/0007810800360043

Wikipedia. (2018). *Science, technology, engineering, and mathematics.* Retrieved March 6, 2018, from https://en.wikipedia.org/wiki/Science,_technology,_engineering,_and_mathematics

Wikipedia. (2019). *Agia Varvara Attikis.* Retrieved January 2, 2020, from https://el.wikipedia.org/wiki/Αγία_Βαρβάρα_Αττικής

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

World Economic Forum. (2019). *Education 4.0*. Retrieved January 15, 2020, from https://www.weforum.org/projects/learning-4-0

World Economic Forum. (2020). *Schools of the Future Defining New Models of Education for the Fourth Industrial Revolution World Economic Forum*. Retrieved January 18, 2020, from https://www.weforum.org/reports/schools-of-the-future-defining-new-models-of-education-for-the-fourth-industrial-revolution

Xatzopoulos, A., Papoutsidakis, M., & Chamilothoris, G. (2013). Mobile Robotic Platforms as Educational Tools in Mechatronics Engineering. In *International Scientific Conference eRA – 8* (pp. 41–51). Pireaus. Retrieved from era.teipir.gr

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, (2), 465–470. 10.1145/1953163.1953297

Zahilah Mohamed Zaki, F., Wong, S. L., & Ridzwan Yaakub, M. (2019). A Review of Common Features inComputational Thinking Frameworks in K-12 Education. *IOP Conference Series. Materials Science and Engineering*, *551*, 012063. doi:10.1088/1757-899X/551/1/012063

Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*(June), 103607. doi:10.1016/j.compedu.2019.103607

Zygouris, N. C., Striftou, A., Dadaliaris, A. N., Stamoulis, G. I., Xenakis, A. C., & Vavougios, D. (2017). The use of LEGO mindstorms in elementary schools. *IEEE Global Engineering Education Conference, EDUCON*, 514–516. 10.1109/EDUCON.2017.7942895

## ADDITIONAL READING


Dorouka, P., Papadakis, St., & Kalogiannakis, M. (2020). Tablets & apps for promoting Robotics, Mathematics, STEM Education and Literacy in Early Childhood Education. *International Journal of Mobile Learning and Organisation*, *14*(2), 255–274. doi:10.1504/IJMLO.2020.106179

Kalogiannakis, M., & Papadakis, S. (2017). Pre-service kindergarten teachers acceptance of "ScratchJr" as a tool for learning and teaching computational thinking and Science education. In *proceedings of the 12th Conference of the European Science Education Research Association (ESERA), 'Research, practice and collaboration in science education' (pp. 21-25)*. Dublin City University and the University of Limerick, Dublin, Ireland.

Kalogiannakis, M., & Papadakis, St. (2018). A proposal for Teaching ScratchJr Programming Environment in Preservice Kindergarten Teachers. In *Finlayson, O., McLoughlin, E., Erduran, S., & Childs, P. (Eds.), Electronic Proceedings of the ESERA 2017 Conference. Research, Practice and Collaboration in Science Education, Part 15/Strand 15 (co-ed. Bodil Sundberg & Maria Kallery, Early Years Science Education)*, (pp. 2095-2105). Dublin, Ireland: Dublin City University. ISBN 978-1-873769-84-3

Kalogiannakis, M., & Papadakis, S. (2019). Pre-service kindergarten teacher's acceptance of "ScratchJr" as a tool for learning and teaching computational thinking and Science education. [JES]. *The Journal of Emergent Science*, *15*, 31–34.

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2017). Using Gamification for Supporting an Introductory Programming Course. The Case of ClassCraft in a Secondary Education Classroom. In *Interactivity, Game Creation, Design, Learning, and Innovation* (pp. 366–375). Cham: Springer.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *Int. J. Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed.), *Early Childhood Education* (pp. 1–19). London, UK: IntechOpen Limited; https://www.intechopen.com/online-first/evaluating-a-course-for-teaching-advanced-programming-concepts-with-scratch-to-preservice-kindergart/ doi:10.5772/intechopen.81714

Papadakis, S., & Kalogiannakis, M. (2020). A Research Synthesis of the Real Value of Self-Proclaimed Mobile Educational Applications for Young Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

## KEY TERMS AND DEFINITIONS

**Computational Thinking:** A term to describe a set of skills or analytical methods that involve human and machine elements to solve problems. These skills usually include problem decomposition, generalization, algorithmic thinking, evaluation, and abstraction.

**Evaluation Tools:** In educational evaluation, this term is used to describe instruments, tools, and methods to collect evidences of the student’s achievement that are generally classified into quantitative and qualitative techniques.

**Educational Robotics:** A broad term that refers to a collection of educational activities, programs, resources, technology platforms, and pedagogical theories of learning within and outside schools.

**Educational Robotics Platform:** A robot technology platform that consists of hardware, software and educational material for educational use.

**Fourth Industrial Revolution:** A term used to describe the 21st century’s economic, social, political, and cultural changes. It is often used interchangeably with Industry 4.0, however, the last one is a subset of it that concerns the industry.

**STEM:** An acronym (science, technology, engineering, and mathematics) that is used to refer to teaching and learning in the fields of science, technology, engineering, and mathematics.

**WeDo 2.0:** The term WeDo 2.0 (or Lego WeDo 2.0) refers to Lego’s educational robotic hardware and software platform specifically designed for Kindergarten to Grade 2 students that follows a process for developing computational thinking skills.

## APPENDIX

*Table 1. Sample of E.R. event's pre-questionnaire*

| General questions | |
|---|---|
| Do you know what STEM Education is? | ☐ No ☐ Maybe ☐ Yes |
| Do you know what Educational Robotics is? | ☐ No ☐ Maybe ☐ Yes |
| Do you know what Computational Thinking is? | ☐ No ☐ Maybe ☐ Yes |
| Do you know what Free/Open Software is? | ☐ No ☐ Maybe ☐ Yes |
| Has your child participated in an educational Robotics / STEM seminar? | ☐ No ☐ Yes |
| **Questions about robots** | |
| Choose which of the educational robotics platforms / robots your child knows? | ☐ Lego ☐ Thymio ☐ Edison ☐ BBC Microbit ☐ Makeblock ☐ BeeBot ☐ Arduino ☐ Raspberry ☐ Parallax ☐ Don't know ☐ Other (please specify) |
| Does your child wants to build his own robot?; | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| If your child could build his own robot how much time would he has to build? | ☐ doesn't want to have time / he wants it ready ☐ 1 – 2 hours ☐ 3 - 4 hours ☐ 4 - 8 hours ☐ he will have unlimited hours, as long as it is completed |
| Would you be involved in robot developing / programming with your child? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |

*Table 2. Sample of E.R. event's post-questionnaire*

| What does STEM education mean to you? | |
|---|---|
| Emphasis on applying knowledge to real world problems | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| It supports critical thinking | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| Helps solve problems | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| Boosts creativity | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| It is important for the economic development of the country | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| **What does Computational Thinking mean to you?** | |
| Helps solve problems | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| It is useful for utilizing Information Technology and Science | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| Linked to STEM education | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |
| It is distinguished for its interdisciplinary application | o Don't know / Don't answer o Strongly disagree o Disagree o Neither disagree, nor agree o Agree o Strongly agree |

*Table 3. A model for CT skills (Atmatzidou & Demetriadis, 2014, 2016)*

| CT skills | Definitions of CT skills | Guidance for development CT skills |
|---|---|---|
| Abstraction | Abstraction is the process of creating something simple from something complicated by leaving out the irrelevant details, by finding the relevant patterns, and by separating ideas from tangible details | 1. Separate the important from the redundant information.<br>2. Analyze and specify common behaviors or programming structures between different scripts.<br>3. Identification of abstractions between different programming environments. |
| Generalization | Generalization is transferring a problem-solving process to a wide variety of problems. | 1. Expanding an existing solution in a given problem in order to cover more possibilities / cases.<br>2. Use variables in solution |
| Algorithm | Algorithm is a practice of writing step-by-step, specific and unambiguous, instructions for carrying out a process. | 1. Explicit wording of the steps of the algorithm.<br>2. Possibility of different algorithms for the same problem.<br>3. Effort to find the most effective algorithm |
| Modularization | Modularization is the development of autonomous processes, which encapsulate a set of often used commands that perform a specific function and might used in the same or different problems. | Develop autonomous sections of code to be used for the same or different problems. |
| Segmentation | *Segmentation* (or *decomposition*) is the process of breaking problems down into smaller parts that may be more easily solved. | Breaking apart problems into smaller / single ones that are easier to be solved. |

*Table 4. A metacognitive and problem-solving guidance protocol based on (Atmatzidou et al., 2018)*

| MC and PS strategies | Questions and prompts that guide students to improve MC and PS skills |
|---|---|
| Understanding the problem | 1. Read the problem carefully as many times as you need to make sure that you understand what it asks for.<br>2. Read anything that you might not understand, and discuss it with the other member of your group.<br>3. Focus to the goals of the problem. |
| Design-implementation of the solution | 1. Does it remind you of another problem?<br>3. Divide the problem into smaller parts.<br>4. Which blocks will you use and what settings would you do? |
| Monitoring-evaluation of the solution | 1. Does the robot do what the problem asks for?<br>2. If not, what did not work properly?<br>3. At which point of the code do you identify the problem?<br>4. What changes and what arrangements do you need to make to fix the problem?<br>5. Check the following code ...does it look like your own code? What are the differences? |
| Evaluation of the procedure | 1. Did the steps that you followed work well for you?<br>2. Which of these steps would you improve next time?<br>3. Is there something that you have learned from this activity and you consider useful for future activities?<br>4. Find if there was something that made it difficult to solve the problem. |

*Table 5. Differences between Jigsaw techniques*

| Step Number | Short description of Jigsaw step | Short description of this research step | Differences |
|---|---|---|---|
| 1 | Students are divided into groups | Students are divided into groups (Same with Jigsaw) | No |
| 2 | Appoint a leader | Equality between members; only two members in the group (Same with Jigsaw) | Yes |
| 3 | Learning objects are divided by the number of students in each group. | Only two members in the group; maker and programmer | Yes |
| 4 | Outsourcing part of a learning object to students | Students research their part of their problem | Yes |
| 5 | Students read the section that is appropriate for them | Students collaborate and exchange information | Yes |
| 6 | Creation of temporary expert groups | - | No |
| 7 | Students return to their original group | - | Yes |
| 8 | Students tell the team that they have learned | - | Yes |
| 9 | Teacher oversees the whole process and intervenes if necessary | Same with Jigsaw | No |
| 10 | Quiz questions are given<br>Teamwork evaluation | Post-Questionnaire (Same with Jigsaw) | No |

*Table 6. Research's schedule*

| Activity | Notes |
|---|---|
| Welcome – seminar opening | |
| Brief Introduction | Getting to know the trainers - Getting to know the participants<br>Information about the seminar's schedule |
| Record participants background | Pre-questionnaire: Filling out a questionnaire with the help of parents/guardians |
| Group discussion | The researcher asks questions and listen to participants' answers e.g. What is a robot (definition), robots' types, shapes, usages, advantages, disadvantages, etc.? Demonstration of educational and industrial robots (robotic arms) |
| Presentation | Brief PowerPoint presentation (Purpose, usage, and concepts of Educational Robotics, STEM, Computational Thinking, 21st-century skills, etc.) |
| Workshop<br>Introductory phase | The researcher demonstrates the robotic platform. Participants are getting familiarized with Lego Education WeDo Core Set Hardware & Software. |
| Break | |
| Workshop<br>Training phase | Participants introductory ER activities to support participants with CT methodology, hardware assembly and programming. |
| Break | |
| Workshop<br>Practice phase | Task-driven phase, the researcher provides participants an authentic STEM project to develop. Participants work in groups and collaborate. |
| Group discussion | The researcher receives valuable feedback from the participants and listens to their comments, concerns, views. |
| Record participants CT skills | Post-questionnaire: Filling out a questionnaire with the help of parents/guardians |
| Greetings – seminar closing | |

*Table 7. Workshop’s phases and learning activities*

| | |
|---|---|
| **Workshop Introductory phase** | |
| Lego Education WeDo 2.0 Core Set Demonstration | Lego Bricks and building elements, WeDo 2.0 Smarthub, Medium Motor, Motion Sensor, Tilt Sensor, Batteries, Bluetooth dongle. |
| Lego WeDo 2.0 Software Demonstration | Familiarize with Lego WeDo 2.0 v1.9.30 programming environment.<br>Navigate and explore software |
| **Workshop Training phase** | |
| 1st training activities | Program Lego Hub’s Led. Get familiar with *Start Block, Light Block, Wait For Block, Repeat Block, Number Input Block* |
| | 1.1 Turn Lego Hub’s Led On<br>Variations:<br>a. Turn Lego Hub’s Led Off<br>1.2 Turn Lego Hub’s Led On to a different color e.g. Yellow<br>Variations:<br>b. As 1.2 but color would be: Green, Cyan, Magenta, White and Black<br>1.3 Turn Lego Hub’s Led to turn On/Off. Led Blinks once with red color at a rate of 1sec On – 1 sec Off<br>Variations:<br>a. As 1.3 but rate would be: 2sec On – 1 sec Off<br>b. As 1.3 but rate would be: 1sec On – 2 sec Off<br>c. As 1.3 but rate would be: 0,5sec On – 1,5 sec Off<br>d. As 1.3 but rate would be: 1,5sec On – 0,5 sec Off<br>e. As 1.3 but rate would be: 0,05sec On – 0,05 sec Off<br>f. As 1.3 but rate would be: 0,01sec On – 0,01 sec Off<br>*Trainer questions: What do you notice? Why does this happen?*<br>1.4 Turn Lego Hub’s Led to turn On/Off. Led Blinks once with red/yellow color at a rate of 1sec Red – 1 sec Yellow<br>Variations:<br>a. As 1.4 but blinking colors would be: Green-Cyan,<br>b. As 1.4 but blinking colors would be: Magenta-White<br>c. As 1.4 but blinking colors would be: Blue-Yellow<br>1.5 Turn Lego Hub’s Led to On/Off. Led Blinks for infinity with red/yellow color at a rate of 1sec Red – 1 sec Yellow<br>Variations:<br>a. As 1.5 but rate would be: 2sec Green – 1 sec White<br>b. As 1.5 but rate would be: 0.5sec Blue – 0.5 sec White<br>1.6 Turn Lego Hub’s Led to On/Off. Led Blinks for 5 times with red/yellow color at a rate of 1sec Red – 1 sec Yellow<br>Variations:<br>a. As 1.6 but times would be: 10<br>a. As 1.6 but times would be: 10 and rate: 0,1sec Red – 0,1 sec Yellow<br>b. As 1.6 but blinking colors would be: Magenta-White and times: 3<br>c. As 1.6 but blinking colors would be: Green-White, times: 3 and rate: 0,5sec Green – 0,5sec White<br>d. As 1.6 but blinking colors would be: Green-White, for 2 times at a rate: 1sec Green – 1sec White and then Red-Yellow, for 2 times at a rate: 1sec Red – 1sec Yellow |
| 1st CT activities | Can you program Lego Hub’s Led to operate as a traffic light?<br>A traffic light switches for ever its color in the following way:<br>Red color for 5 sec -> Green color for 7 sec -> Orange color for 2 sec -><br>Red color for 5 sec -> Green … etc. |
| 2nd training activities | Introduce to Lego WeDo strings: *Linear Sequence* and *Parallel Sequence*. Get familiar with *Start Block, Light Block, Wait For Block, Repeat Block, Number Input Block.* |

*continues on following page*

*Table 7. Continued*

| | |
|---|---|
| | 2.1 Play a sound<br>Variations:<br>a. Choose a different sound from the list<br>b. Record a sound and choose it as the playing sound<br>2.2 Turn Lego Hub's Led On and play a sound<br>Variations:<br>a. Turn Lego Hub's Led to turn On/Off and play a sound.<br>b. Led Blinks once with red color at a rate of 1sec On – 1 sec Off and play a sound.<br>c. Variations:<br>d. As 2.2 but rate would be: 0,5sec On – 0,5 sec Off<br>e. As 2.2 but Led blinks once with red/yellow color at a rate of 1sec Red – 1 sec Yellow<br>f. As 2.2 Led Blinks for infinity with green/blue color at a rate of 0,5sec Green – 0,5 sec Blue<br>g. As 2.2 Led Blinks for 5 times with green/blue color at a rate of 0,5sec Green – 0,5 sec Blue and play a sound, and then Led blinks for another 5 times with red/yellow color at a rate of 0,5sec Red – 0,5 sec Yellow and play a different sound |
| 2nd CT activities | 1. Can you turn Lego Hub's Led to blinks for infinity with green/blue color at a rate of 0,5sec Green – 0,5 sec Blue and play two sounds simultaneously?<br>2. Can you synchronize Lego Hub's Led blinking with playing sounds in the following way?<br>When the Start block is pressed:<br>Lego Hub's Led blinks for 5 times green/blue color at a rate of 0,5sec Green – 0,5 sec Blue and one sound is played simultaneously, and then,<br>Led blinks for another 5 times red/yellow color at a rate of 0,5sec Red – 0,5 sec Yellow and another sound is played simultaneously.<br>3. Can you reproduce the sound of a fire siren, that is, to switch between two sounds for a specific length of time?<br>4. Can you extend 1st CT activity "traffic light" program to play a different sound according to the traffic light color? |
| 3rd training activities | Advanced *linear sequence* and *parallel sequence* programming. Get familiar with *Start On Key Press Block* |
| | 3.1 Play a sound when A key is pressed<br>Variations:<br>a. As 3.1 but also, when B key is pressed then another sound is played<br>3.2 Turn Lego Hub's Led On when A key is pressed. When the key is released the Led turns Off.<br>Variations:<br>a. As 3.2 but when key is released the Led remains On.<br>b. As 3.2 but when R key is pressed Led turns Red (On) when W key is pressed Led turns White (On), and when B key is pressed Led turns Off. |
| 3rd CT activities | 1. Can you extend 1st CT activity "traffic light" program to have an emergency Stop in the following way?<br>2. When S key is pressed then the traffic light color becomes immediately Orange for Orange color for 2 sec and then Red color for 5 sec. Subsequently, the traffic light continues its normal operation.<br>3. Can you turn Lego Hub's Led On/Off when L key is pressed in the following way?<br>When the Start block is pressed Led is Off. When L key is pressed once (1st times) the Led turns On. When L key is again pressed (2nd times) the Led turns Off. When L key is again pressed (3rd times), Led turns On, etc. |
| 4td training activities | Build a Lego WeDo 2 robot model. Introduce to *Lego Bricks, building element* and *mechanisms* such as *Medium Motor, Beams, Gears, Pulleys, Wheels.* Get familiar with *Motor This Way Block, Motor That Way Block, Motor Power Block, Motor On For Block* and *Motor Off Block.* |

*continues on following page*

Table 7. Continued

| | |
|---|---|
| 4td CT activities | 4.1 Build and program Lego Drive model to move forward.<br>Variations:<br>a. As 4.1 but the robot moves backward.<br>b. As 4.1 but the robot moves forward for 1 sec and then stops.<br>c. As 4.1 but the robot moves forward for 1 sec, backward for 1 sec, and then stops.<br>d. As 4.1 the but robot moves forward with a decreased speed and for 1 sec and then stops.<br>e. As 4.1 but the robot moves forward with a decreased speed for 1 sec, backward with increased speed for 1 sec, and then stops.<br>f. As 4.1 but the robot moves forward with a decreased. When F key is pressed robot moves forward with increased speed. When D key is pressed robot moves forward with decreased speed. When S key is pressed robot stops moving.<br>4.2 Robot experiments with mechanisms. Try to change pulleys and write down your observations:<br>a. Use a large pulley. Pulley up: a large pulley drives a small pulley to produce more rotations.<br>b. Use a small pulley. Pulley down: a small pulley drives a large pulley to produce less rotations.<br>c. c) Twist pulley. Pulley twist: it is used to make shafts that are parallel but rotate in opposite directions. |
| **Workshop Practice (challenge) phase** | |
| 4th CT activity | Can you program the robot simulate a fire-truck? |
| Decomposition, Generalization (Pattern Recognition), Algorithmic Thinking, Evaluating (Debugging) | Your robot should follow the below scenario:<br>When the robot (fire truck) starts up, it moves forward at low speed. When F key (F for Fire) is pressed, then the robot moves at its maximum speed, flashing its beacon (switching its led from red to white), and sounding its siren to simulate the siren of a real fire truck. When the B key (B for Begin) is pressed, the robot moves forward at low speed, its beacon light is off and, its siren is not sounding. Finally, when S key (S for Stop) is pressed, the robots stops moving, beacon and siren switch off (no led flashing, no sound). |

Table 8. Trainers' rubrics record grid

| Trainers' rubrics record grid | | | |
|---|---|---|---|
| Name: | Name: | Name: | Name: |
| 1. Emerging | 2. Developing | 3. Proficient | 4. Accomplished |
| Notes | | | |

*Table 9. A sample of E.R. event's pre-questionnaire*

| General purpose questions | |
|---|---|
| 1. Do you have a computer? | ☐ Yes ☐ No |
| 2. Do you have a tablet? | ☐ Yes ☐ No |
| 3. Do you have Internet access? | ☐ Yes ☐ No |
| 4. How well are you familiar with computer/tablet | ☐ Not at all ☐ Little ☐ Moderate ☐ Very ☐ Too much |
| 5. What do you enjoy doing on your PC / tablet? | ☐ Games ☐ Internet ☐ Programming ☐ Homework ☐ Other (describe) |
| 6. What is your favorite lesson (or what do you enjoy doing most)? | ☐ Arts/Theater/Music ☐ Foreign Languages ☐ Informatics ☐ Mathematics<br>☐ Physics/Chemistry/Biology ☐ Gymnastics ☐ History ☐ Other (describe) |
| **Questions about programming** | |
| 7. Do you know what programming is? | ☐ No ☐ Yes |
| 9. Are you involved in programming? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 10. If so, what programming languages do you know or describe exactly what you know about it? | |
| **Questions about robotics** | |
| 11. Do you know what a robot is? | ☐ No ☐ Yes |
| 13. Have you dealt with Robotics or Robots? | ☐ No ☐ Yes |
| 17. Why do you like robots and robotics?<br>(Reply to all options) | I like playing with robots:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I find it an interesting activity:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I think it's something creative:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I like to solve problems with robots:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I want to learn programming:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I like teamwork:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I want to participate in a robotics competition:<br>☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always<br>I don't know / I don't like it:<br>☐ I don't know ☐ I don't like robots / robotics |

*Table 10. Student computational thinking self-assessment rubric*

| Student computational thinking self-assessment rubric | | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1. I defined the question or problem. | | | | | |
| 2. I built a model. | | | | | |
| 3. I programmed a solution (for my model). | | | | | |
| 4. I tested my solution and made improvements. | | | | | |
| 5. One thing I did really well was: | | | | | |
| 6. One thing I want to improve on for next time is: | | | | | |
| 7. Notes: | | | | | |

*Table 11. A sample of E.R. event's post-questionnaire*

| Questions about the robot's development | |
|---|---|
| 1. Did you enjoy work with robots? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 2. Do you think after the ER activities, you learned to program the robot to a satisfactory level? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 3. Would you like to be engaged with robots? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 6. Did you find the instructions given in the activities helpful? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 7. Do you think that in different activities you can find common behavior of the robot (i.e. behavior that happens elsewhere)? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 8. Can you think and suggest a general solution to a problem? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 9. Do you think you can evaluate the solution (to see if it is right or wrong) that you gave to each problem? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 10. Do you believe, the way you thought solutions to the ER activities, will help you think and solve a problem in another lesson? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 11. Is there anything about the given instructions that you would like to change or improve? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| Questions about programming | |
| 13. Do you enjoy programming (e.g. robots)? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 14. Did ER activities help you understand programming better? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 15. How would you characterize yourself as a programmer? | ☐ Very Bad ☐ Bad ☐ Moderate ☐ Good ☐ Very good |
| 16. After your ER experience would you like to get more involved in programming? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| **Questions about the team's collaboration** | |
| 17. What team's role did you like the most? | ☐ None ☐ Maker ☐ Programmer ☐ Both |
| 18. Did you like working with other people? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |
| 19. Did you like that everyone in the team had their own role? | ☐ Never ☐ Rarely ☐ Sometimes ☐ Often ☐ Always |

*Table 12. Pre-questionnaires' Yes/No questions' analysis*

| Question (No. & Description) | No (n,%) | Yes (n,%) | Did not answer (n,%) |
|---|---|---|---|
| 1. Do you have a computer? | 0 (0%) | 10 (100%) | 0 (0%) |
| 2. Do you have a tablet? | 0 (0%) | 10 (100%) | 0 (0%) |
| 3. Do you have Internet access? | 0 (0%) | 10 (100%) | 0 (0%) |
| 5. What do you like to see (or do) on your PC or tablet? | | | |
| 5.1. Play games | 3 (30%) | 7 (70%) | 0 (0%) |
| 5.2. Browse Internet | 4 (40%) | 6 (60%) | 0 (0%) |
| 5.3. Do programming | 9 (90%) | 1 (10%) | 0 (0%) |
| 5.4. Do homework | 7 (70%) | 3 (30%) | 0 (0%) |
| 5.5. Other | 8 (80%) | 2 (20%) | 0 (0%) |
| 7. Do you know what programming is? | 5 (50%) | 5 (50%) | 0 (0%) |
| 11. Do you know what a robot is? | 0 (0%) | 10 (100%) | 0 (0%) |
| 13. Have you dealt with Robotics or Robots? | 2 (20%) | 8 (80%) | 0 (0%) |

*Table 13. Pre-questionnaires Likert questions' analysis*

| Question (No. & Description) | Strongly Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| 4. How well are you familiar with computer/ tablet | 1 (10%) | 1 (10%) | 3 (30%) | 3 (30%) | 2 (20%) | 0 (0%) |
| 9. Are you involved in programming? | 5 (50%) | 3 (30%) | 2 (20%) | 0 (0%) | 0 (0%) | 0 (0%) |
| 17. Why do you like robots and robotics? | | | | | | |
| 17.1 I like playing with robots | 0 (0%) | 1 (10%) | 1 (10%) | 2 (20%) | 6 (60%) | 0 (0%) |
| 17.2. I find it an interesting activity | 0 (0%) | 1 (10%) | 1 (10%) | 3 (30%) | 5 (50%) | 0 (0%) |
| 17.3. I think it's something creative | 0 (0%) | 1 (10%) | 2 (20%) | 2 (20%) | 5 (50%) | 0 (0%) |
| 17.4. I want to know more about robotics | 0 (0%) | 0 (0%) | 2 (20%) | 5 (50%) | 3 (30%) | 0 (0%) |
| 17.5. I like to solve problems with robots | 0 (0%) | 2 (20%) | 2 (20%) | 3 (30%) | 3 (30%) | 0 (0%) |
| 17.6. I want to learn to program | 0 (0%) | 0 (0%) | 1 (10%) | 4 (40%) | 5 (50%) | 0 (0%) |
| 17.7. I like teamwork | 0 (0%) | 3 (30%) | 1 (10%) | 2 (20%) | 4 (40%) | 0 (0%) |
| 17.8. I want to participate in the national robotics competition | 0 (0%) | 0 (0%) | 1 (10%) | 2 (20%) | 7 (70%) | 0 (0%) |
| 17.9. I want to participate in other robotics competition | 0 (0%) | 0 (0%) | 1 (10%) | 2 (20%) | 6 (60%) | 0 (0%) |

*Table 14. Post-questionnaires "related to robots" questions' analysis*

| Question (No. & Description) | Completely Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| 1. Did you enjoy work with robots? | 0 (0%) | 0 (0%) | 0 (0%) | 2 (20%) | 7 (70%) | 1 (10%) |
| 3. Would you like to be engaged with robots? | 0 (0%) | 0 (0%) | 0 (0%) | 2 (20%) | 7 (70%) | 1 (10%) |

*Table 15. Post-questionnaires "related to programming" questions' analysis*

| Question (No. & Description) | Completely Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| 2. Do you think after the ER activities, you learned to program the robot to a satisfactory level? | 0 (0%) | 1 (10%) | 0 (0%) | 3 (30%) | 5 (50%) | 1 (10%) |
| 13. Do you enjoy programming (e.g. robots)? | 0 (0%) | 0 (0%) | 0 (0%) | 2 (20%) | 7 (70%) | 1 (10%) |
| 14. Did ER activities help you understand programming better? | 0 (0%) | 0 (0%) | 1 (10%) | 3 (30%) | 5 (50%) | 1 (10%) |
| 16. After your ER experience would you like to get more involved in programming? | 0 (0%) | 0 (0%) | 3 (30%) | 2 (20%) | 4 (40%) | 1 (10%) |
| **Question (No. & Description)** | **Very Bad (n,%)** | **Bad (n,%)** | **Average (n,%)** | **Good (n,%)** | **Very Good (n,%)** | **Did not respond (n,%)** |
| 15. How would you characterize yourself as a programmer? | 0 (0%) | 1 (10%) | 0 (0%) | 1 (10%) | 7 (70%) | 1 (10%) |

*Table 16. Post-questionnaires "related to collaboration" questions' analysis*

| Question (No. & Description) | Completely Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| 18. Did you like working with other people? | 1 (10%) | 0 (0%) | 1 (10%) | 4 (40%) | 3 (30%) | 1 (10%) |
| 19. Did you like that everyone in the team had their own role? | 3 (30%) | 0 (0%) | 0 (0%) | 3 (30%) | 3 (30%) | 1 (10%) |

*Table 17. Question's No 17: "What team's role did you like the most?" results*

| Question (No. & Description) | Maker (n,%) | Programmer (n,%) | Both (n,%) | Did not respond (n,%) |
|---|---|---|---|---|
| 17. What team's role did you like the most? | 1 (10%) | 2 (20%) | 6 (60%) | 1 (10%) |

*Table 18. Associating questions with CT skills*

| Questions from participants' pre-questionnaires and participants' self-assessment CT rubric<br>Note: Letter Q indicates pre-questionnaires questions Letter R indicates CT rubric questions | Computational Thinking Skill | | | | | | |
|---|---|---|---|---|---|---|---|
| | Abstraction | Generalization | Algorithm | Modularization | Segmentation | Debugging | Collaboration |
| Q6. Did you find the instructions given in the activities helpful? | X | | | | | | |
| Q7. Do you think that in different activities you can find common behavior of the robot (i.e. behavior that happens elsewhere)? | | X | | | | | |
| Q8. Can you think and suggest a general solution to a problem? | | X | X | | X | | |
| Q9. Do you think you can evaluate the solution (to see if it is right or wrong) that you gave to each problem? | | | | | | X | |
| Q10. Do you believe, the way you thought solutions to the ER activities, will help you think and solve a problem in another lesson? | X | X | | | | | |
| Q11. Is there anything about the given instructions that you would like to change or improve? | X | | | | | | |
| Q17. What team's role did you like the most? | | | | | | | X |
| Q18. Did you like working with other people? | | | | | | | X |
| Q19. Did you like that everyone in the team had their own role? | | | | | | | X |
| R1. I defined the question or problem. | | | | | X | | X |
| R2. I built a model. | | X | | X | X | | X |
| R3. I programmed a solution (for my model) | | | X | X | X | | X |
| R4. I tested my solution and made improvements | | | | | | X | X |

*Table 19. Post-questionnaires questions' analysis*

| Question (No. & Description) | Completely Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| Q6. Did you find the instructions given in the activities helpful? | 0 (0%) | 1 (10%) | 0 (0%) | 5 (50%) | 3 (30%) | 1 (10%) |
| Q7. Do you think that in different activities you can find common behavior of the robot (i.e. behavior that happens elsewhere)? | 0 (0%) | 1 (10%) | 3 (30%) | 2 (20%) | 3 (30%) | 1 (10%) |
| Q8. Can you think and suggest a general solution to a problem? | 1 (10%) | 0 (0%) | 5 (50%) | 2 (20%) | 1 (10%) | 1 (10%) |
| Q9. Do you think you can evaluate the solution (to see if it is right or wrong) that you gave to each problem? | 0 (0%) | 1 (10%) | 3 (30%) | 2 (20%) | 3 (30%) | 1 (10%) |
| Q10. Do you believe, the way you thought solutions to the ER activities, will help you think and solve a problem in another lesson? | 0 (0%) | 0 (0%) | 2 (20%) | 4 (40%) | 3 (30%) | 1 (10%) |
| Q11. Is there anything about the given instructions that you would like to change or improve? | 5 (50%) | 0 (0%) | 1 (10%) | 0 (0%) | 3 (30%) | 1 (10%) |
| Q18. Did you like working with other people? | 1 (10%) | 0 (0%) | 1 (10%) | 4 (40%) | 3 (30%) | 1 (10%) |
| Q19. Did you like that everyone in the team had their own role? | 3 (30%) | 0 (0%) | 0 (0%) | 3 (30%) | 2 (20%) | 1 (10%) |

*Table 20. Participants computational thinking self-assessment rubric's analysis*

| Question (No. & Description) | Completely Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| R1. I defined the question or problem. | 3 (15%) | 2 (10%) | 0 (0%) | 4 (20%) | 6 (30%) | 5 (25%) |
| R2. I built a model. | 0 (0%) | 0 (0%) | 0 (0%) | 8 (40%) | 10 (50%) | 2 (10%) |
| R3. I programmed a solution (for my model). | 0 (0%) | 2 (10%) | 0 (0%) | 2 (10%) | 10 (50%) | 6 (30%) |
| R4. I tested my solution and made improvements. | 1 (5%) | 0 (0%) | 1 (5%) | 2 (20%) | 11 (55%) | 5 (25%) |

*Table 21. Mean scores of CT activities*

| Question (No. & Description) | N | Mean |
|---|---|---|
| **Scores of the first CT activity:** | | |
| R1. I defined the question or problem | 7 | 3,14 |
| R2. I built a model | 8 | 4,50 |
| R3. I programmed a solution (for my model) | 8 | 4,00 |
| R4. I tested my solution and made improvements | 8 | 4,75 |
| First CT activity's overall mean score | | **4,29** |
| **Scores of the last CT activity:** | | |
| R1. I defined the question or problem | 8 | 3,75 |
| R2. I built a model | 10 | 4,60 |
| R3. I programmed a solution (for my model) | 6 | 5,00 |
| R4. I tested my solution and made improvements | 6 | 4,67 |
| Last CT activity's overall mean score | | **4,41** |
| **Scores Difference** | | 0,12 |

*Table 22. Participants' CT skills analysis*

| Computational Thinking Skill | Completely Disagree (n,%) | Disagree (n,%) | Neither Agree / Disagree (n,%) | Agree (n,%) | Strongly Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|---|---|
| Abstraction | 5 (18%) | 1 (4%) | 3 (11%) | 9 (32%) | 9 (32%) | 1 (4%) |
| Generalization | 2 (4%) | 7 (13%) | 11 (20%) | 10 (18%) | 18 (32%) | 8 (14%) |
| Algorithm | 1 (3%) | 8 (22%) | 5 (14%) | 4 (11%) | 11 (31%) | 7 (19%) |
| Modularization | 0 (0%) | 2 (5%) | 0 (0%) | 10 (25%) | 20 (50%) | 8 (20%) |
| Segmentation | 4 (6%) | 4 (6%) | 5 (7%) | 16 (23%) | 27 (39%) | 14 (20%) |
| Debugging | 1 (3%) | 1 (3%) | 4 (13%) | 4 (13%) | 14 (47%) | 6 (20%) |
| Collaboration | 8 (8%) | 4 (4%) | 2 (2%) | 23 (23%) | 42 (42%) | 20 (20%) |

*Table 23.Aggregated CT skills analysis results*

| Computational Thinking Skill | Disagree | Neutral | Agree | Did not respond |
|---|---|---|---|---|
| Abstraction | 22% | 11% | 64% | 4% |
| Generalization | 17% | 20% | 50% | 14% |
| Algorithm | 25% | 14% | 42% | 19% |
| Modularization | 5% | 0% | 75% | 20% |
| Segmentation | 12% | 7% | 62% | 20% |
| Debugging | 6% | 13% | 60% | 20% |
| Collaboration | 12% | 2% | 65% | 20% |

Table 24. Correlation between rubric questions and research

| Questions from participants' pre-questionnaires and participants' self-assessment CT rubric<br>Note: Letter Q indicates pre-questionnaires qu estions<br>Letter R indicates CT rubric questions | Research's Questions | | |
|---|---|---|---|
| | *Do educational robotics activities enhance students' CT skills?* | *Do educational robots in authentic STEM applications create a more favorable environment for developing students' CT?* | *Do educational robots enhance students' engagement with programming?* |
| Q2. Do you think after the ER activities, you learned to program the robot to a satisfactory level? | | | X |
| Q3. Would you like to be engaged with robots? | | X | |
| Q6. Did you find the instructions given in the activities helpful? | X | | |
| Q7. Do you think that in different activities you can find common behavior of the robot (i.e. behavior that happens elsewhere)? | X | | |
| Q8. Can you think and suggest a general solution to a problem? | X | | |
| Q9. Do you think you can evaluate the solution (to see if it is right or wrong) that you gave to each problem? | X | | |
| Q10. Do you believe, the way you thought solutions to the ER activities, will help you think and solve a problem in another lesson? | X | | |
| Q11. Is there anything about the given instructions that you would like to change or improve? | X | | |
| Q 13. Do you enjoy programming (e.g. robots)? | | | X |
| Q 14. Did ER activities help you understand programming better? | | | X |
| Q16. After your ER experience would you like to get more involved in programming? | | | X |
| Q17. What team's role did you like the most? | X | | |
| Q18. Did you like working with other people? | X | | |
| Q19. Did you like that everyone in the team had their own role? | X | | |
| R1. I defined the question or problem. | X | | |
| R2. I built a model. | X | | |
| R3. I programmed a solution (for my model) | X | | |
| R4. I tested my solution and made improvements | X | | |

Table 25. Research's questions analysis

| Research's Questions | Disagree (n,%) | Neutral (n,%) | Agree (n,%) | Did not respond (n,%) |
|---|---|---|---|---|
| Do educational robotics activities enhance students' CT skills? | 21 (13%) | 16 (10%) | 96 (60%) | 26 (16%) |
| Do educational robots in authentic STEM applications create a more favorable environment for developing students' CT? | 0 (0%) | 0 (0%) | 15 (75%) | 5 (25%) |
| Do educational robots enhance students' engagement with programming? | 1 (3%) | 4 (10%) | 27 (68%) | 8 (20%) |

# Chapter 11
# Learning Computational Thinking Development in Young Children With Bee-Bot Educational Robotics

**Stamatios Papadakis**
https://orcid.org/0000-0003-3184-1147
*Department of Preschool Education, University of Crete, Greece*

**Michail Kalogiannakis**
https://orcid.org/0000-0002-9124-2245
*Department of Preschool Education, University of Crete, Greece*

## ABSTRACT

*It is widely known that when used intentionally and appropriately, technology and interactive media are effective tools to support learning and development. In recent years, there has been a push to introduce coding and computational thinking in early childhood education, and robotics is an excellent tool to achieve this. This chapter presents some results obtained in the development of a learning experience in computational thinking using Bee-Bot educational robotics. The experience involved 47 preschoolers of a kindergarten in Crete, Greece during the period 2019-2020. The study reports statistically significant learning gains between the initial and final assessment of children's computational thinking skills. It was found that children in the treatment group who engaged in the robotic curricular intervention performed better on CT tests. This finding shows that an enhanced teaching experience using robots was beneficial for improving young children's computational thinking skills. The implications for designing appropriate curricula using robots for kindergarteners are addressed.*



DOI: 10.4018/978-1-7998-4576-8.ch011

## INTRODUCTION

Over the last few years, increasing attention has been focused on the development of children's acquisition of 21st-century skills and digital competences (Kalogiannakis & Papadakis, 2020). Consequently, many education scholars have argued that teaching technology – the "T" in STEM (Science, Technology, Engineering, and Mathematics)– to young children is vital in keeping up with 21st-century employment patterns (Lee, 2019). When used intentionally and appropriately, technology and interactive media are effective tools to support learning and development (NAEYC & Fred Rogers Center & Joint Position Statement, 2012). In early childhood, new interactive and smart screen technologies create opportunities to enhance young children's growing, learning, and playing (Bers 2008). Technologies, such as those that involve robotics or coding apps, come at a time when the demand for computing jobs around the globe is at an all-time high while its supply is at an all-time low (European Commission, 2018; Papadakis, 2020; Servoz, 2018). At the same time, researchers and scholars have highlighted the vast cognitive benefits of introducing Computational Thinking (CT) skills to young children (Govindarajan, 2019; Orfanakis & Papadakis, 2016). For instance, Bers (2018a) highlights the fact that children as young as four years old can learn foundational computational thinking concepts and this kind of learning can support their language, mathematical, cognitive, and socio-emotional development.

The term 'Computational Thinking' can be defined as solving problems algorithmically and developing a sense of technological fluency (Bers, 2018b) which involves, but is not limited to, critical thinking, problem solving and creativity (Papavlasopoulou, Sharma, & Giannakos, 2019). Central to this approach is the notion of by taking part in coding activities, children, including preschoolers, are exposed to Computational Thinking skills (Wing, 2006). According to Herdzina (2019) this is a main reason that coding, is seen as the new modern literacy for today's young children. It is widely known that there is women underrepresentation in science, technology, engineering, and math (STEM) and as a result many young women have fewer opportunities to contribute to and benefit from careers in computer science and engineering (Papadakis, Tousia, & Polychronaki, 2018; Master, Cheryan, Moscatelli, & Meltzoff, 2017). Research suggests that coding experience enhances children's interest in knowledge and skills include engineering-science, technology, engineering, and mathematics (STEM) areas, and reduces gender-biased stereotypes associated with STEM careers (Master et al., 2017; Sullivan, 2016).

Following technological advancements, international trends, and research developments, robotics and computer programming initiatives are growing in popularity amongst early childhood researchers and educators (Sullivan & Bers, 2016). Studies on the inclusion of concepts such as computational thinking and coding in the education system have had an important impact on the educational goal for many countries (Çiftci & Bildiren, 2019; Papadakis & Kalogiannakis, 2019; Sáez-López et al., 2016). In an effort to improve current educational programs, national educational programs and private initiatives such as the 'Hour of code' (https://hourofcode.com) and the 'Code week' (https://codeweek.eu) are focusing on STEM literacy and making coding and computational thinking a priority for education (Bers, González-González, & Armas–Torres, 2019; Papavlasopoulou et al., 2019).

Scholars have long recognized that preschool children can engage in and complete basic programming and robotics tasks (Sullivan & Bers 2016). Today, there is a newfound abundance of technological tools aimed at young children on the commercial landscape (Sullivan, 2016). A number of mobile applications (apps) that are designed to teach young children coding skills in a fun, game-like way has become popular in recent years (Papadakis, Zaranis, & Kalogiannakis, 2019). For instance, apps such as the Daisy the Dinosaur, the Kodable, and Scratch Jr., have been found that when used intentionally and appropriately they can support CT competencies such as abstraction, debugging/troubleshooting, pattern recognition, simulations, etc. (Hamilton, Clarke-Midura, Shumway & Lee, 2019). Even Elsa and Anna from Disney's Frozen have gotten in on the coding fun (Herdzina, 2019).

Except apps, the last decade, especially, kindergarten educators have focused on various forms of ICT (Information & Communication Technologies) such as robotics as methods of teaching academic skills to kindergarteners through hands-on experiences with new technologies (Papadakis, Kalogiannakis & Zaranis, 2018). Several robotic platforms exist that are used in different educational stages, depending on their hardware and programming complexity. For example, although there are platforms mainly used at the university level, there exist many robotics construction kits such as the KIBO and the Bee-Bot, and the Thymio (Vitanza, Rossetti, Mondada, & Trianni, 2019). It is the advent of these newer, more modern robotics kits that have been evolved to become the modern generation of learning manipulatives that help children develop a stronger understanding of mathematical concepts, such as number, size, and shape, in much the same way that traditional materials like pattern blocks, beads, and balls once did (González-González et al., 2019). Additionally, robotics can be an effective way to introduce computational thinking as it involves students being able to systematically process tasks and developing the sequenced step-by-step coding commands needed to program a robot (Chalmers, 2018). Young children can become engineers by playing with motors and sensors as well as storytellers by creating and sharing personally meaningful projects that react in response to their environment (Bers 2008; Sullivan & Bers, 2016). In robotic programming, reflective thinking skills based on problem solving have a significant importance as they materialize the reflections of an individual's learning in a physical environment following the process of abstract programming (Durak, Yilmaz & Yilmaz, 2019). Besides, robotic manipulatives allow children to work on skills that are important for healthy child development, such as fine motor skills and hand-eye coordination, while also engaging in collaboration and teamwork (González-González et al., 2019). Additionally, unlike many digital games developed for children, building with robotics does not typically involve sitting alone in front of a screen (Sullivan & Bers, 2016).

The paper proceeds as follows: in section 2 the theoretical framework of the study as well as the Bee-Bot robotic kit is presented; section 3 focuses on study objectives; the methodological approach and research design are shown in section 4; the results are provided in section 5; and, finally, the discussion, limitations, conclusions, acknowledgments and references used are presented, respectively.

## LITERATURE REVIEW

Digital technologies knowledge, understanding and skills, such as the literacy of coding, will form the foundation of the digital literacies practices that young children will draw upon as they transition to school (Walsh & Campbell, 2018). A reason is that coding-based systems are everywhere in society and children's lives, and children experience and manipulate these systems without recognizing them daily (Lee, 2019). Additionally, as Bers (2018a) notes while more and more people learn to code, computer programming leaves the exclusive domain of computer science and becomes central to other professions and to new ways of thinking, coding takes on the civic dimension of literacy. The ability to code is now accepted as an important skill for students to learn before they enter the technology driven, disruptive, global labor market that defines the early twenty-first century (Çiftci & Bildiren, 2019). According to Wing (2006), the children should be taught not only reading, writing and arithmetical ability but also the contribution of the information to problem solving process and how to apply Computational

Thinking skills and make logical analysis during early childhood education (Durak, Yilmaz & Yilmaz, 2019). Computational Thinking focuses on skills children develop from practicing programming and algorithms and enables the development of qualities such as pattern recognition, and logical reasoning (Angeli, 2019). It is not a surprise that Computational Thinking (CT) has been described by Wing (2006) as a necessary skill for all and deemed fundamental to 21st Century professionals (Gomes, Falcão & Tedesco, 2018) as it involves a number of skills, like problem decomposition, developing algorithms, and abstraction (Angeli, 2019).

The provision of early coding/precoding experiences in a developmentally appropriate manner for young age children has been a major concern of most Western governments such as Australia, Canada, the United Kingdom and Sweden (Elkin, Sullivan, & Bers, 2016; Lee, 2019). Consequently, their inclusion in the basic curriculum is seen to be of the utmost importance for citizens of an increasingly programmable world (Kalogiannakis & Papadakis, 2008; Papadakis & Kalogiannakis, 2010). It is known that both from an economic and a developmental standpoint, educational interventions that begin in early childhood have lower costs and durable effects (Bers, 2018b). Although still not a consensus, over the past few years CT concepts have become mandatory in the curriculum of several countries (Gomes, Falcão & Tedesco, 2018). As a result, recently, there has been an increasing interest about children's acquisition of thinking skills and digital competences (Angeli, 2019). For instance, as Bers, González-González, & Armas–Torres, (2019) note in the United States, new initiatives focused on 21st century skills suggest programming and tech literacy skills as a priority for early childhood. At the same time Bers (2018a) also notes that the goal of children introducing to coding activities and developing their CT skills is not to prepare students of all ages for computer science degrees and careers (due to the shortage of programmers and software developers in the industry), but to provide them with the intellectual tools to serve a role in civic society. According to Code.org, there are less than 50,000 Computer Science graduates in 2017. But there are over 500,000 open computing positions in the United States. This could mean that in 2020, the available seats for this position will exceed qualified applicants by a million which could widen the gap even more. As a result, in 2020, an estimated 1 million computer programming-related jobs in the US are expected to be unfilled (Full Scale, 2019). But coding is more than a technical skill; it is a way to achieve literacy in the 21st century, like reading and writing (Bers, 2018a). Through learning the literacy practice of coding, in much the same way they learn the literacy practices of reading and writing, young children also develop computational thinking skills where they engage in problem-solving (Walsh & Campbell, 2018). Indeed, robotics and coding activities in early childhood education can also foster the development of a range of cognitive and social skills (Sullivan, 2016).

In this aspect, a number of commercially available tools for young children to learn computer programming have emerged aiming to teach children as young as four fundamental engineering and programming concepts (Papadakis & Kalogiannakis, 2020; Sullivan, 2016). Scholars and educators welcome the availability and diversity of computational devices, including coding apps and programming tools such as ScratchJr, Daisy the Dinosaur, Scratch, Dash and Dot, Robot Turtles, Cubetto, Bee-Bot, and KIBO (Hamilton, Clarke-Midura, Shumway & Lee, 2019; Sullivan, 2016). Using easy to- use visual programming tools enables students to develop simple algorithms as they drag and drop actions into coding sequences algorithms help young age children, including preschoolers to learn modularity, control structures, representation, hardware/software, design process, and debugging (Bers, 2018b). The robot must be programmed step by- step in the precise sequence that actions are required to occur and, just like following a recipe, the robot performs the programmed sequence of steps that specifies the actions that need to be followed (Chalmers, 2018). This integration can be done using programmable robotic devices and the application of project-based learning methodologies (Papadakis, Kalogiannakis, Orfanakis, & Zaranis, 2019). These different technological interfaces promote distinctive types of joint engagement, as they invite varying kinds of interactions given unique design features (Govindarajan, 2019).

Especially, educational robotics is a didactic approach that can be integrated into preschool education by using programmable robotic devices and the application of project-based learning methodologies (González-González et al., 2019). With the term ‘educational robotics’ we refer to the use of robotics and computer programming to teach. By incorporating mobile devices and robotics into their teaching repertoires, early years educators can create technologies-rich environments that place children’s digital play at the center, with the goal of assisting them to acquire coding as a literacy proficiency (Walsh & Campbell, 2018). This approach imparts knowledge about robot programming or the design, creation, and assembly of robots to enhance learning (Papavlasopoulou, Sharma, & Giannakos, 2019). Nam, Kim, & Lee (2019) highlight the fact that when children are provided with complementary support, they can learn computer programming as a problem-solving or thinking skill. In general, the use of robots for complementary actions can be concrete through the following robotic activities: collaborative planning, representing tasks, practicing their strategies, and applying their plan in the real world (Nam et al., 2019).

There is no doubt, that coding with robotics is a wonderful tool for learners of all ages as it provides a catalyst to introduce them computational thinking, algorithmic thinking and project management (Walsh & Campbell, 2018). Even pre-kindergarten students can learn computational thinking skills while working with robots (Chalmers, 2018). During the process of designing, constructing, and programming robots’ students are exposed to computational concepts such as sequencing, pattern recognition, and loops (Chalmers, 2018). Robotic is a strong and flexible education tool which enables students to conduct both robotic programming and control activities by using special programming tools (Durak, Yilmaz & Yilmaz, 2019). Research suggests that children as young as 4 years old can successfully build and program simple robotics projects while learning a range of engineering and robotics (Sullivan & Bers, 2016). In robotic activities, students design to handle their complex problems and receive immediate feedback about the outputs of the programs they write by testing their solutions (Atmatzidou, Demetriadis, & Nika, 2018). Learning to code robots in prior to school settings provides young children with twenty-first century skills that include coding as a literacy practice (Walsh & Campbell, 2018). There is evidence that even pre-kindergarten students can learn computational thinking skills while working with robots (Chalmers, 2018). However, the teaching and learning of concepts related to technology and engineering, should not be the exclusive focus of concern for introducing robotics and coding in early childhood education as the intentionally and appropriately introduction of these technologies can support the development of a range of cognitive and social milestones (Sullivan & Bers, 2016).

Researchers such as Vitanza, Rossetti, Mondada, & Trianni (2019) highlight the fact that studies around the globe identify that the use of robotics in preschool education large benefits of collaborative over individual learning and also positively evaluate the difference in the learning experience of young age children when associated with robots with respect to solitary activities. Robotic programming is considered a developmentally appropriate strategy for kindergarten classrooms (Papavlasopoulou, Sharma, & Giannakos, 2019). A robot can cover the role of (a) a tool, (b) a tutor or (c) a peer in the learning process, and, in this context, it shows an exceptional ability to attract the students’ attention (Vitanza et al., 2019). Robotics and computer programming in early childhood education can also foster the development of a range of cognitive and social skills (Sullivan, 2016).

Additionally, as Sullivan (2016) highlights the use of a developmentally appropriate robotics curriculum can help to change negative stereotypes and ideas children may initially have about technology and engineering (Papadakis, 2018a). Girls' interest in computer science from a young age possibly fades because of a gendered or non-appropriate pedagogical approach (Papavlasopoulou, Sharma, & Giannakos, 2019).This is important to consider as stereotypes may play a role in children's engagement and performance in curricular activities (Sullivan, 2016) as research suggests that children who are exposed to STEM curriculum and computer programming at an early age demonstrate fewer gender-based stereotypes regarding STEM careers (Sullivan & Bers, 2016).

## Bee-Bot Robotics kit

At present, there are several resources of educational robotics that allow the introduction of programming at early ages (Muñoz-Repiso & Caballero-González, 2019). The Bee-Bot® robotics kit is used for this research. Bee-bot is a robotic floor robot shaped like a bee. It is also a screen-free and app-free robot and it has directional keys, on its back. Hamilton et al. (2019) categorize the Bee-bot as a 'Button-Operated Robot.' Except that Bee-Bot® robotics kit combines resilience and subtlety at the same time, other factors in its favor are its dimensions, which allow for easy handling. In addition, its colors, sounds, and movements make it a suitable resource for use with young children between the ages of 3 and 7 (Muñoz-Repiso & Caballero-González, 2019). The robot has buttons to program the sequence of movements it must perform: advance, reverse, turn left or right, start to move, pause the movements, and delete the previous commands and an 'OK' button to validate. The buttons located on the robot's back are: 'Move forward 15cm', 'Move backward 15cm', 'Turn left 90°', 'Turn right 90°', 'Execute the sequence of instructions', 'Pause', 'Reset' (format the robot). Bee-Bot can memorise up to 200 steps, thus enabling the creation of more complex programs (Muñoz-Repiso & Caballero-González, 2019). Bee-Bot blinks and beeps at the conclusion of each command to allow children to follow Bee-Bot through the program they have entered and then confirms its completion with lights and sound. As far as the technical specifications of the Bee-Bot robot its dimensions are: 125 x 100 x 75 mm and it is rechargeable with USB cable make it suitable for the preschool classroom settings. Playing with the Bee-Bot® robotics kit, inputting commands by pressing the buttons and simply engaging with the device are examples of children familiarising themselves with digital devices, developing their understandings of robots, and learning how to code or the literacy practices of coding (Walsh & Campbell, 2018).

Of course, according to Walsh & Campbell (2018) learning to code a Bee-Bot® robotics kit has also some disadvantages. As the researchers note without first manipulating 'code you can touch' it can be difficult for young children because it is too abstract. The children cannot 'see' the code once they have pushed the buttons on top of the Bee-Bot. They also cannot watch the Bee-Bot execute the coded sequence they entered alongside the symbols, nor understand how their coding was entered incorrectly if the Bee-Bot does not go where they wanted it to go (Walsh & Campbell, 2018). There is also a more advanced version of the Bee-Bot. It is called Blue-Bot and unlike Bee-Bot, it can do 45° turns, and can integrate repetitions in its algorithm. Its case is transparent, allowing kids to see the different electronic components that the robot needs to function. Like the educational robot Bee-Bot, a Blue-Bot move in 15cm steps and do 90° rotations.

Figure 1. The front sides of a Bee-Bot robot

Figure 2. The back side of a Bee-Bot robot

## RESEARCH QUESTIONS

The main goal of this study was to examine how kindergarteners respond to coding with a robot called Bee-Bot by using a teaching intervention to develop children's Computational Thinking skills (sequencing and problem-solving skills). The teaching intervention was designed for preschoolers to perform complementary planning and interactions with a robot via the adoption of with the researchers' scaffolding support. The research questions used for this study are the following and adapted from the study of Muñoz-Repiso & Caballero-González (2019): 1) Is it possible to develop the computational thinking of children in the early childhood education stage through robotic activities in the classroom? 2) Can children improve their ability to sequence actions by responding to a challenge through programming activities using educational robots? 3) Can children improve their ability to identify and correct existing errors in a programming sequence?

## METHODOLOGICAL APPROACH

### Participants

The study sample consisted of N=47 children from 2 classes at an urban public kindergarten in Crete, Greece. The final sample consisted of all children that participated in both pretests and posttests. Children that did not took part to two tests because they were not present at the time of administration due to illness or that parents did not want to burden them with participation were excluded from the study. At the time of the experiment, the children ranged in age from 5 to 6 years at the start of this study. There were 26 girls and 21 boys. The groups consisted of 24 children (girls = 14, boys = 10) in the treatment group and 23 children (girls = 12, boys = 11) from one class in the comparison group. Both classes utilize the national curriculum for kindergarten.

An informed consent was provided to all research participants. In the case of the preschoolers, the informed consent was signed by their parents. The selection of the school was by invitation of the research group. The objective was to study the treatment group by measuring the effectiveness of the intervention through the learning of coding and the development of computational thinking skills.

## Method

For the study described in this communication, we adopted and adapted the teaching interventions described by the Caballero-Gonzalez, Muñoz-Repiso, & García-Holgado (2019); Muñoz-Repiso & Caballero-González (2019) and Nam et al. (2019) in an effort to have comparable results with other international studies. We also used the Bee-Bot activities as described in the ebook entitled 'Let's Go with Bee-Bot' produced by TTS Group Ltd. The book is freely available on the web in the following address https://www.generationrobots.com/media/50_ways_to_make_the_most_of_your_Bee-Bot.pdf. Additionally we also adapted the activities described in the ebook entitled 'KS1 Bee-Bots 1,2,3 Activity: An introduction to programming with Bee-Bots' which is also free available in the following address: https://www.generationrobots.com/media/Bee-Bots-1-2-3-Activity-Barefoot-Computing.pdf. The main goal of this study was to teach children fundamental computational thinking and coding skills. In general, we used a robot-based approach as a tangible programming interface. The preschoolers who participated in the teaching intervention used the Bee-Bot® educational robotics kit to make decisions, construct a sequence of movements and explore the results of actions. The robotics kit used in the teaching intervention satisfied the age-related needs of young children, as is the case with Bee-Bot. It is known that by the time children leave preschool and enter kindergarten, around the age of 5, they can follow multi-step instructions and retell stories that they know well in the correct order (Bers et al., 2019). Taking into these considerations, the activities contributed to strengthen the teaching learning process of basic concepts of computational thinking, in a work environment that favored the play aspects and social interaction of the children.

The study was developed using a quasi-experimental design (Gribbons & Herman, 1996), with pre-test and post-test measurements in two groups (experimental and control) without random assignment. The students were divided into two groups, the experimental and the control group. The members of the experimental group performed the teaching intervention while the members of the control group did not take part in the robotics activities. In the first stage of the research the initial measurement of the dependent variable (pre-test) took place, then the teaching intervention followed, and finally at the third stage the second measurement of the dependent variable (post-test) took place. The intervention sessions were based on the study of Muñoz-Repiso & Caballero-González (2019). The intervention consisted of the development of 7 working sessions with the children in the experimental group. The first was an introduction to the use of the devices, and in the following 6 sessions the children explored concepts and carried out practices on programming. During the activities, the preschoolers worked in small groups (up to 3 members) collaboratively. Each group operated a Bee-Bot and collaborated to solve the problem. If the children wanted to retry the activity after a failed attempt at problem-solving, they were allowed to do so. Each session took place during a school day. Unlike the duration of the interventions described in other studies, the approximate duration of the teaching interventions of the current study was only two hours per day, due to the Greek Kindergarten characteristics. Gender balance was not achieved in the sample, as a different number of girls and boys participated in each group. Similar to other studies, the allocation of students to groups could not be done randomly since the intervention allowed by the kindergarten required working with intact groups formed according to criteria inherent to the kindergarten itself and independent of the study. Following the methodological criteria of this type of research design, measures were collected from all the preschoolers participated in the study (experimental and control group), before and after the intervention. The evaluation tests were carried out individually and each child spent approximately 30 min on both tests. The teaching intervention was conducted by the researchers, along with the preschool classroom teachers who provided support in class management.

Pre and post-test measurements were obtained for everyone in both groups. As there were no standardized measures for evaluating preschoolers CT skills, we adopted the procedure that described in the research work of Caballero-Gonzalez et al. (2019). Thus, we took into account the scientific works of Brennan & Resnick (2012), Strawhacker & Bers, (2018) and Bers, Flannery, Kazakoff, & Sullivan (2014). Similar to the work described by the Caballero-Gonzalez et al. (2019) we created an evaluation rubric and the CT skills were explored through the characteristics: sequences, patterns and debugging through problem solving challenges. Brennan & Resnick (2012) call computational concepts, which include concepts like sequencing and loops. As cited by Bers et al. (2019), Brennan and Resnick (2012) define a Computational Thinking Framework that matches the developmental ability of young children and includes: sequencing, repeats, conditionals, and debugging. According to Muñoz-Repiso & Caballero-González (2019) 'sequences' is the ability to sequence actions by responding to a challenge through programming activities while an 'action-instruction correspondence' is the ability to relate the instructions given to a robot with the action it performs. Additionally, 'debugging' is the ability to identify and correct existing errors in a programming sequence. Consequently, a total of 6 challenges were carried out in the evaluations. Each activity started with a short narrative and then the problem or challenge to be solved was raised. Two challenges were assigned for each of the explored characteristics. Following Caballero-Gonzalez et al (2019) guidelines each dimension of the questionnaire was evaluated through the resolution of two challenges posed to children.

The design of the evaluation questionnaire, for both, pre-test and post-test was designed using 6-likert scale type questions (Robbins & Heiberger, 2011) and was also based in the work of Bers at al. (2014). Bers at al. (2014, p. 149) suggest assessing learning outcomes after each activity, while a score of 4 or higher is defined as the target level of achievement. Similarly, in the present study the researchers assessed each child's level of understanding of selected core concepts as seen by successful application of the concepts in the robot or program and if needed, they also talked with children to gain more information about their work and understandings. The 6-point Likert scale reflected the following levels of competence: 5 Complete achievement of the goal, task, or understanding 4 Mostly complete achievement of the goal, task, or understanding; 3 Partially complete achievement of the goal, task, or understanding; 2 Very incomplete achievement of the goal, task, or understanding; 1 Did not complete the goal, task, or understanding; 0 Did not attempt/Other (Bers at al., 2014, p. 149).

## Data Analysis

All statistical analysis was performed using the Statistical Package for the Social Sciences (SPSS) version 23.0 (IBM® SPSS® Statistics, Armonk, USA) while a p-Value $< 0.05$ was selected as the critical value for statistical significance for all statistical tests. Prior to statistical analysis, the normality of data was checked with the Shapiro-Wilk test and the homoscedasticity of variance with the Levene's test. These tests are recommended for use in studies with samples smaller than 2000 elements. The data of the current study did not follow a normal distribution (less than .05). Thus, the nonparametric test Mann-Whitney U test was used to compare outcomes between two independent groups.

## Ethical Compliance

Before commencing the study, approval was obtained from the University of Crete Research Ethics Committee (REC) and kindergarten administrators. Participation in this study was on a voluntary basis. Additionally, all participants were assured of the anonymity and confidentiality of their responses. Parents of the preschoolers were also told that they could withdraw from the study at any time.

## RESULTS

As already mentioned, the potential of Computational Thinking skills has been explored through the three characteristics: sequences, patterns and debugging. The nonparametric Mann–Whitney U test can be used to determine if there was a significant difference between two conditions (Kent State University Libraries, 2017). Thus, in this study, this test was applied to determine if there were significant differences in the initial values of sequences, patterns and debugging as well as the complete CT test among the groups. These conclusions can range from simply stating whether the two populations differ through to determining if there are differences in medians between groups. According to the calculations that were made there was no significant difference between the experimental and control group in the values of each of the characteristics of CT test and the complete test as well (Sequences: $U=269.5$, $p=.872$, Patterns: $U=268$, $p=.84$, Debugging: $U= 270$, $p=.885$, Complete pretest: $U=266.5$, $p=.834$ (see Tables 1 and 2). We can then consider that the two groups were equivalent on the knowledge of the topic prior the teaching intervention.

*Table 1. Group statistics data pre-test (experimental and control group)*

| | Group | N | Mean Rank |
|---|---|---|---|
| Sequences | Bee_bot intervention | 24 | 23.73 |
| | Control group | 23 | 24.28 |
| Patterns | Bee_bot intervention | 24 | 23.67 |
| | Control group | 23 | 24.35 |
| Debugging | Bee_bot intervention | 24 | 23.75 |
| | Control group | 23 | 24.26 |
| Complete test | Bee_bot intervention | 24 | 23.60 |
| | Control group | 23 | 24.41 |

*Table 2. Values for Mann-Whitney U with data pre-test (experimental and control group)*

| CT characteristic evaluated | Mann-Whitney U | Wilcoxon W | Z | Asymp. Sig. (2-tailed) |
|---|---|---|---|---|
| Sequences | 269.500 | 569.500 | -.161 | .872 |
| Patterns | 268.000 | 568.000 | -.202 | .840 |
| Debugging | 270.000 | 570.000 | -.145 | .885 |
| Complete test | 266.500 | 566.500 | -.209 | .834 |

At the end of the teaching intervention, the same questionnaire was administered to the same participants by the same data collector with the purpose of comparing the responses of the first application (pre-test) with the application of the same questionnaire (post-test). Consequently, the nonparametric Mann–Whitney U test applied again to the values obtained by the members of both groups in the application of the post-test test. According to the calculations that were made there was no significant difference between the experimental and control group in the values of each of the characteristics of CT test and the complete test as well (Sequences: U=96, $p$= .000, Patterns: U=152.5, $p$=.002, Debugging: U=46.5, $p$=.000, Complete posttest: U= 29, $p$= .000 (see Tables 3 and 4). Interestingly, similar to the study of Caballero-Gonzalez et al. (2019) the present study also found that there was a small, but significant, difference between the members of the experimental and control groups, in the values associated with each of the characteristics of computational thinking and in the complete test.

*Table 3. Group statistics data posttest (experimental and control group)*

| | Group | N | Mean Rank |
|---|---|---|---|
| Sequences | Bee_bot intervention | 24 | 31.50 |
| | Control group | 23 | 16.17 |
| Patterns | Bee_bot intervention | 24 | 29.15 |
| | Control group | 23 | 18.63 |
| Debugging | Bee_bot intervention | 24 | 33.56 |
| | Control group | 23 | 14.02 |
| Complete test | Bee_bot intervention | 24 | 34.29 |
| | Control group | 23 | 13.26 |

*Table 4. Values for Mann-Whitney U with data posttest (experimental and control group)*

| CT characteristic evaluated | Mann-Whitney U | Wilcoxon W | Z | Asymp. Sig. (2-tailed) |
|---|---|---|---|---|
| Sequences | 96.000 | 372.000 | -4.368 | .000 |
| Patterns | 152.500 | 428.500 | -3.142 | .002 |
| Debugging | 46.500 | 322.500 | -5.147 | .000 |
| Complete test | 29.000 | 305.000 | -5.340 | .000 |

## DISCUSSION

Studies have shown that children who have been enrolled in playful programming activities with computational toys utilize CT skills as they participate in computational challenges (Hamilton et al., 2019). For instance, children while they create a sequence of instructions for a robot to follow, break down a complex problem into smaller parts and fix errors within their program use various aspects of CT competencies such as algorithmic thinking, problem decomposition and debugging among others.

This study similar to other studies (Bers, Flannery, Kazakoff, & Sullivan, 2014; Kazakoff, Sullivan & Bers, 2013; Sullivan & Bers, 2016) provides preliminary evidence that robotics kits specifically designed for young learners can be a useful and educational tool in early childhood classrooms. The children enjoyed learning how to use the Bee-Bot robots, because the activities that involved the robots complemented the students' cognitive load. Children not only have fun working with robotics, but research shows positive learning outcomes as well. This study agrees with Bers et al. (2019) findings that show positive implications for expanding this kind of learning environment, which relies on coding and computational thinking as playground, to other early childhood contexts. In the present study, in general, children observed the proposed problem, mentally built its solution, inserted the instructions and clicked on the robot button that starts the execution. Our findings echo earlier work as noted by Herdzina (2019) that found preschool and early elementary school-age children are able to learn fundamental coding skills over several training sessions. Similar to other studies, children showed difficulties in dealing with long sequences of instructions (5 or more) and with combinations of various different instructions (Gomes et al., 2018). As Gomes et al. (2018) hypothesize the data set is long enough to memorise them. In the present study was mentioned that the children often ended up missing one or more instructions. Of course, this does not necessarily mean that they did not know how to solve that challenge. Gomes et al. (2018) in their study mention that many children tried to memorise instructions and insert them all at once. Interesting enough is that other children inserted instructions one by one, mimicking with their fingers the action to be performed (Gomes et al., 2018). Raabe et al (2015) in their study with Bee-Bot robotic kit found that without proper instructions, preschoolers had difficulties in grasping the concept of turning and its representation (the physical button of an arrow pointing to the side), and in planning the sequence of actions. They suggest that activities engaging children's body and using physical materials, along with teacher mediation, must performed before children continuing with the interaction with Bee-Bot, to help children develop an understanding of the rotation function, and improving their capability of planning. Additionally, as Walsh & Campbell (2018) note learning to code a robot such as Bee-Bot without first manipulating 'code you can touch' it can be difficult for young children because it is too abstract. The children cannot 'see' the code once they have pushed the buttons on top of the Bee-Bot. They also cannot watch the Bee-Bot execute the coded sequence they entered alongside the symbols, nor understand how their coding was entered incorrectly if the Bee-Bot does not go where they wanted it to go.

During this study, through play-based learning, it was found that children can discover, create, improvise, and imagine as they play with the Bee-Bot robotics kit. Within their groups, they also collaborate in playful ways to challenge each other's thinking and build new extended understandings about CT concepts. The results obtained through this research coincide with other studies that propose the development of social skills and computational thinking using challenges of programming and robotics (Caballero-Gonzalez et al., 2019). As Walsh & Campbell (2018) recommend while children play with the robotic kits, educators should regularly prompt them to explain what they are doing and what they are thinking as they interweave the language of computer science. Children's talk must be supported and extended by educators to foster deeper understandings of the literacy practices of coding. When early years educators use design briefs and robots this way, they are creating a play-based learning context where young children begin to move from the familiar and concrete, to the unfamiliar and abstract as they learn to code the various robotic kits through fun, hands-on experiences (Walsh & Campbell, 2018). In the present study there were no found early gender gaps in motivation to pursue CT, coding and engineering. This is considered important as studies have found that girls may be affected by stereotypes about intellectual ability and STEM as early as 6 years of age (Bian, Leslie, & Cimpian, 2017).

In this study it was also noticed that children may need some time to understand the 'CLEAR' button, know how to move forwards and backwards and be able to turn left and right. They will also need enough time to experiment pressing the 'GO' button. In classroom conditions the educator must be ready to help children to show them how to work the Bee-Bot once they are ready for the robot to move. This is considered important as it was also found that whilst some children loved to play and discover how to make it work, many children got bored when Bee-Bot keeps doing the wrong thing. The motivation for the use of the tool emerges from the activity, the problem or challenge to be solved, or from the story that the children represent using the robot (González-González et al., 2019). For example, converting a robot into a character with a mission to fulfill or a role to embody engages children in different types of activities, offering them the opportunity to explore the possibilities of the environment (González-González et al., 2019).

Also, as Lee (2019) recommends for introducing coding to young children, it is important for educators to know the terms or words used in an official coding process and to present the terms or words in ways children understand. As directional words are often used in the coding process it is considered important for educators to help children become familiar with words such as move forward, move backward, turn left, turn right. Lee (2019) suggests for teachers, instead of saying 'Come here,' a teacher might say 'Come forward,' or instead of saying 'Sit by John,' a teacher might say 'Sit by John's left side' and point to that side. This provides a cue for young children about where to sit even though they do not yet know the directional words. An easy solution for educators is to display posters with directional words and arrow signals in the classroom to help children, to get familiar with the words and the directional signs as well as to learn sequential ordinal words (first, second, third, etc.) (Lee, 2019). Additionally, the 'CLEAR' button is very important when children engage with the Bee-Bot robotic kit. Again, teachers must be ready to explain children that the clear button helps to tell Bee-Bot to listen to new commands. This means that teachers must be self-motivated to learn to use robotic kits although they are generally novices when it comes to teaching with robotics as part of the curriculum (Bers et al., 2019). We can say that coding activities with robots should be appealing, skill-based, and intentionally used to see the benefits (Herdzina, 2019). Early childhood educators always should use their knowledge of child development and effective practices to carefully and intentionally select and use technology and media if and when it serves healthy development, learning, creativity, interactions with others, and relationships (NAEYC & Fred Rogers Center & Joint Position Statement, 2012).

## Limitations

The study had limitations, such as the derived from the developmental research in the complex nature of the educational practices and the people involved in the research. The teaching intervention was implemented by the research assistants and not by regular classroom teachers. Thus, as Sullivan & Bers (2016) note it may be possible that it is not the teaching intervention itself that led to students' development of their CT competency but having these experts in the classroom may have contributed as well. As Sullivan & Bers (2016) recommend future research must follow classroom teachers using innovative new tools such as the Bee-Bot robotics kit to ensure that they are not only developmentally appropriate for children to use, but easy for early childhood instructors to teach with as well. The validity of the study is also questionable due to the nature of the data analyzed as it was purely quantitative. But it was beyond the scope of this study to discuss more nuanced details such as the problem-solving strategies children employed on the different tasks or the styles of mistakes that children made on the programming tasks.

In fact, these types of details could allow teachers to create curricular activities and assessments that are appropriately challenging and engaging for children in kindergarten (Sullivan & Bers, 2016). Future research should look at the qualitative differences in children's programming ability and approaches to robotics and programming tasks. Finally, this study is limited by the relatively short duration between implementation and evaluation and the small sample size that prevented the use of parametric statistical testing, as non-parametric tests were used in the present study. As Caballero-Gonzalez et al. (2019) recommend the present study could be improved by implementing the activities in different geographical contexts and/or including other educational levels. This would allow to increase the certainty and quality of the results widening the scope and generalization of the conclusions.

## CONCLUSION

The last decade block-based programming environments and robotics systems that are developmentally appropriate for young children have been developed. As a result, robotic kits (Bee-Bot, Blue-Bot, Cubetto, Kibo, etc.) and coding applications (Code Karts, ScratchJr, etc.) are increasingly getting popular in all stages of K12 as well as in early education (Durak et al., 2019). These new technologies offer young children the possibility to develop and integrate knowledge about CT and STEM concepts (Bers, 2018a). International research on teaching coding to preschoolers has showed that children as young as age four can learn to correctly sequence a robot to perform a specific task and most are able to engage in developmentally-appropriate computer programming through game-like activities (Herdzina, 2019). Robots have appealing features and functions for children that able to maintain children's attention for a longer period of time, improving their performance, their ability to concentrate, and their cognitive flexibility (González-González et al., 2019). Coding practice involves various early mathematical and scientific skills and processes, including spatial sense, number sense, problem-solving skills, inquiry skills and reasoning skills (Papadakis, 2018b). In addition, coding involves communication skills (Lee, 2019). By using robots, children can enhance their working memory skills and learn to sequence increasingly complex programs and master all of KIBO's syntax rules (Bers et al., 2019).

In the present study we can claim that preschoolers adopted a computational perspective by ensuring they are developing an understanding about themselves as producers rather than just consumers of technology (Chalmers, 2018).

Due to increased need of young children engagement in computational thinking or critical problem-solving skills early years educators, are suddenly responsible for introducing the new literacy of coding to young children (Walsh & Campbell, 2018). Thus, early childhood educators must know and provide children with concrete experiences related to coding in ways children enjoy and understand (Lee, 2019). For early years educators to feel motivated and confident to use mobile technologies to teach coding as a literacy, they first need an introduction to the basics of coding (Papadakis, 2018c). However, many of these educators have had limited exposure to the idea that coding is a type of literacy (Alafodimos, Kalogiannakis, Papadakis, & Papachristos, 2009). This is because most early childhood teacher education programmes do not include the teaching of coding as part of literacy subjects and computer science subjects are generally only taught in upper primary and secondary schools (Walsh & Campbell, 2018). Additionally, this study highlights the need for a shift toward improving scholars understanding of how an instructional model and curriculum designed to enhance children's computational thinking (Sullivan & Bers, 2016) can be implemented in Greek kindergarten classrooms.

This study was a pilot and future work with larger sample sizes will be conducted to gather more widely generalizable results. Results from this and future work will inform the re-design of the robotics kit implementation and accompanying curriculum.

## FUNDING

The authors received no financial support for the research, authorship and/or publication of this article.

## DECLARATION OF CONFLICTING INTERESTS

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## ACKNOWLEDGMENT

We would like to express our sincere appreciation of the cooperation of the staff and students in the schools that participated in this work.

## REFERENCES

Alafodimos, C., Kalogiannakis, M., Papadakis, St., & Papachristos, D. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece. In D. Guralnick (Ed.), *Proceedings of the International Conference on E-Learning in the Workplace (ICELW-09)*. New York: Kaleidoscope Learning.

Angeli, C. (2019). Computational thinking education: Issues and challenges. *Computers in Human Behavior*. doi:10.1016/j.chb.2019.106185

Atmatzidou, S., Demetriadis, S., & Nika, P. (2018). How Does the Degree of Guidance Support Students' Metacognitive and Problem Solving Skills in Educational Robotics? *Journal of Science Education and Technology*, *27*(1), 70–85. doi:10.100710956-017-9709-x

Bers, M. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York: Teacher's College Press.

Bers, M. U. (2018a). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 8. doi:10.20897/ejsteme/3868

Bers, M. U. (2018b). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 2094-2102). IEEE. 10.1109/EDUCON.2018.8363498

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. doi:10.1016/j.compedu.2013.10.020

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130–145. doi:10.1016/j.compedu.2019.04.013

Bian, L., Leslie, S. J., & Cimpian, A. (2017). Gender stereotypes about intellectual ability emerge early and influence children's interests. *Science*, *355*(6323), 389–391. doi:10.1126cience.aah6524 PMID:28126816

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association*.

Caballero-Gonzalez, Y. A., Muñoz-Repiso, A. G. V., & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 19-23). ACM. 10.1145/3362789.3362874

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93–100. doi:10.1016/j.ijcci.2018.06.005

Çiftci, S., & Bildiren, A. (2019). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer Science Education*, *§§§*, 1–19.

Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational Thinking, Programming Self-Efficacy, Problem Solving and Experiences in the Programming Process Conducted with Robotic Activities. *Contemporary Educational Technology*, *10*(2), 173–197. doi:10.30935/cet.554493

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. *Computers in the Schools*, *33*(3), 169–186. doi:10.1080/07380569.2016.1216251

European Commission. (2018). *Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions on the Digital Education Action Plan*. Retrieved from https://ec.europa.eu

Full Scale. (2019). *Talent Shortage of Software Developers*. Retrieved from https://fullscale.io/talent-shortage-software-developers/

Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction*, *16*, 77–84. doi:10.1016/j.ijcci.2017.12.005

González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., & Infante-Moro, A. (2019). Computational Thinking and Down Syndrome: An Exploratory Study Using the KIBO Robot. *Informatics (MDPI)*, *6*(2), 25–45. doi:10.3390/informatics6020025

Govindarajan, M. (2019). *Families That Code Together Learn Together: Exploring Family-Oriented Programming in Early Childhood with ScratchJr and KIBO Robotics* (Doctoral dissertation). Tufts University.

Gribbons, B., & Herman, J. (1996). True and quasi-experimental designs. *Practical Assessment, Research & Evaluation*, *5*(1), 14.

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2019). An Emerging Technology Report on Computational Toys in Early Childhood. *Technology, Knowledge and Learning*, 1-12.

Herdzina, J. (2019). *Young children: What's code got to do with it? Guest blog by Sarah Pila*. Retrieved from http://teccenter.erikson.edu/what-were-reading/pila_coding/

Kalogiannakis, M., & Papadakis, St. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. *Proceeding of the 5th WSEAS/IASME International Conference on Engineering Education*, 305-311.

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). IGI Global. doi:10.4018/978-1-7998-1486-3.ch005

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*(4), 245–255. doi:10.100710643-012-0554-5

Kent State University Libraries. (2017). *SPSS tutorials: Independent samples t test.* Retrieved from https://libguides.library.kent.edu/SPSS/IndependentTTest

Lee, J. (2019). Coding in early childhood. *Contemporary Issues in Early Childhood*, 1463949119846541.

Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology*, *160*, 92–106. doi:10.1016/j.jecp.2017.03.013 PMID:28433822

Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to Develop Computational Thinking in Early Childhood Education. *Comunicar: Media Education Research Journal*, *27*(59), 63–72. doi:10.3916/C59-2019-06

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners. *The Asia-Pacific Education Researcher*, *28*(5), 1–11. doi:10.100740299-019-00438-4

National Association for the Education of Young Children, & Fred Rogers Center for Early Learning and Children's Media at Saint Vincent College. (2012). *Technology and interactive media as tools in early childhood programs serving children birth through age 8*. Retrieved from https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/topics/PS_technology_WEB.pdf

Orfanakis, V., & Papadakis, St. (2016). Teaching basic programming concepts to novice programmers in Secondary Education using Twitter, Python, Ardruino and a coffee machine. In *Hellenic Conference on Innovating STEM Education (HISTEM)*. University of Athens.

Papadakis, S. (2018a). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2018b). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2018c). Evaluating pre-service teachers' acceptance of mobile devices with regards to their age and gender: A case study in Greece. *International Journal of Mobile Learning and Organisation*, *12*(4), 336–352. doi:10.1504/IJMLO.2018.095130

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). IGI Global. doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, St., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In M. Kalogiannakis, D. Stavrou & P.G. Michaelidis (Eds.), *Proceedings of the 7th International Conference on Hands-on Science, Bridging the Science and Society Gap*, (pp. 235-240). Academic Press.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed.), *Early Childhood Education* (pp. 1–19). London, UK: IntechOpen Limited. doi:10.5772/intechopen.81714

Papadakis, S., & Kalogiannakis, M. (2020). A Research Synthesis of the Real Value of Self-Proclaimed Mobile Educational Applications for Young Children. In *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). IGI Global. doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. In Information Resources Management Association (IRMA) (Ed.), Early Childhood Development: Concepts, Methodologies, Tools, and Applications (pp. 797-819). IGI-Global. doi:10.4018/978-1-5225-7507-8.ch039

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2018). The effectiveness of computer and tablet assisted intervention in early childhood students' understanding of numbers. An empirical study conducted in Greece. *Education and Information Technologies*, *23*(5), 1849–1871. doi:10.100710639-018-9693-7

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *Int. J. Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Papadakis, S., Zaranis, N., & Kalogiannakis, M. (2019). Parental involvement and attitudes towards young Greek children's mobile usage. *International Journal of Child-Computer Interaction*, *22*, 100144. doi:10.1016/j.ijcci.2019.100144

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2019). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*. doi:10.1016/j.chb.2019.03.003

Raabe, A., Rodrigues, A. J., Santana, A. M., Vieira, M. V., Rosário, T., & Carneiro, A. C. R. (2015). Brinquedos de Programar na Educação Infantil: Um estudo de Caso. Anais do Workshop de Informática na Escola, 21(1), 42.

Robbins, N. B., & Heiberger, R. M. (2011). Plotting Likert and other rating scales. In *Proceedings of the 2011 Joint Statistical Meeting* (pp. 1058-1066). American Statistical Association.

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129–141. doi:10.1016/j.compedu.2016.03.003

Servoz, M. (2018). *The future of work? Work of the future! On how artificial intelligence, robotics and automation are transforming jobs and the economy in Europe*. Retrieved from https://ec.europa.eu/epsc/sites/epsc/files/ai-report_online-version.pdf

Strawhacker, A., & Bers, M. U. (2018). Promoting Positive Technological Development in a Kindergarten Makerspace: A Qualitative Case Study. *European Journal of STEM Education*, *3*(3), 9. doi:10.20897/ejsteme/3869

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20. doi:10.100710798-015-9304-5

Sullivan, A. A. (2016). *Breaking the STEM Stereotype: Investigating the use of robotics to change young children's gender stereotypes about technology and engineering* (Dissertation). Tufts University.

Vitanza, A., Rossetti, P., Mondada, F., & Trianni, V. (2019). Robot swarms as an educational tool: The Thymio's way. *International Journal of Advanced Robotic Systems*, *16*(1), 1729881418825186. doi:10.1177/1729881418825186

Walsh, C., & Campbell, C. (2018). Introducing Coding as a Literacy on Mobile Devices in the Early Years. In Mobile Technologies in Children's Language and Literacy: Innovative Pedagogy in Preschool and Primary Education (pp. 51-66). Emerald Publishing Limited. doi:10.1108/978-1-78714-879-620181004

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

## ADDITIONAL READING

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978–988. doi:10.1016/j.compedu.2011.10.006

Campbell, C., Speldewinde, C., Howitt, C., & MacDonald, A. (2018). STEM practice in the early years. Creative Education Journal Special Edition Preschool Education Research, 9(1), 11-25.

Demetriou, K., & Nikiforidou, Z. (2019). The relational space of educational technology: Early childhood students' views. *Global Studies of Childhood*, *9*(4), 290–305. doi:10.1177/2043610619881458

DePryck, K. (2016). From computational thinking to coding and back. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 27-29). ACM. 10.1145/3012430.3012492

Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2010* (pp. 4006-4014). Chesapeake, VA: AACE.

Instefjord, E. J., & Munthe, E. (2017). Educating digitally competent teachers: A study of integration of professional digital competence in teacher education. *Teaching and Teacher Education*, *67*, 37–45. doi:10.1016/j.tate.2017.05.016

Kalogiannakis, M., Nirgianaki, G.-M., & Papadakis, St. (2018). Teaching magnetism to preschool children: The effectiveness of picture story reading. *Early Childhood Education Journal*, *46*(5), 535–546. doi:10.100710643-017-0884-4

Kalogiannakis, M., & Papadakis, S. (2007, June). The dual form of further education of educators in ICT: technological and pedagogical training. In *C. Constantinou, Z. Zacharias & M. Papaevripidou, Proceedings of the 8th International Conference On Computer Based Learning in Science* (pp. 265-276).

Kalogiannakis, M., & Papadakis, St. (2017). Combining mobile technologies in environmental education: A Greek case study. *International Journal of Mobile Learning and Organisation*, *11*(2), 108–130. doi:10.1504/IJMLO.2017.084272

Kalogiannakis, M., & Papadakis, St. (2018). An Evaluation of Greek Educational Android Apps for Pre-Schoolers. In Finlayson, O., McLoughlin, E., Erduran, S., & Childs, P. (Eds.), Electronic Proceedings of the ESERA 2017 Conference. *Research, Practice and Collaboration in Science Education*, Part 4/Strand 4 (co-ed. Kalle Juuti & Eleni A. Kyza, Digital Resources for Science Teaching and Learning), (pp. 593-603). Dublin, Ireland: Dublin City University.

Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital Student Conference Platform Implementation: The case study of the "Research Project" course. *Open Education*, *12*(2), 5–23.

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, *10*(3), 235–250. doi:10.1504/IJTCS.2019.102760

Papadakis, St., Kalogiannakis, M., Sifaki, E., & Vidakis, N. (2018). Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University. *EAI (European Alliance for Innovation) Endorsed Transactions on Creative Technologies, 5*(16), e1, 1-12.

Papadakis, S. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

Psycharis, S. (2015). The Impact of Computational Experiment and Formative Assessment in Inquiry Based Teaching and Learning Approach in STEM Education. *Journal of Science Education and Technology*, *25*(2), 316–326. doi:10.100710956-015-9595-z

Vidakis, N., Barianos, A. K., Trampas, A. M., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (Vol. 1, pp. 36-43). 10.5220/0007810800360043

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

Yelland, N. (2011). Reconceptualising play and learning in the lives of young children. *Australasian Journal of Early Childhood*, *36*(2), 4–12. doi:10.1177/183693911103600202

## KEY TERMS AND DEFINITIONS

**Computational Thinking:** Is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could execute.

**Early Childhood Education:** Is a broad term used to describe any type of educational program that serves children in their preschool years, before they are old enough to enter kindergarten.

**Educational Robotics:** Is an interdisciplinary learning environment based on the use of robots and electronic components to enhance the development of skills and competencies in students.

**Preschool Education:** Is education that focuses on educating children from the ages of infancy until six years old.

**Preservice Teachers:** Students enrolled in an initial educator preparation program.

**Programming:** Lines of code that are written in a certain language that demands a logic of reasoning from the developers.

# Chapter 12
# Educational Robotics and Computational Thinking Development

**Timoleon Theofanellis**
*School of Pedagogical and Technological Education, Greece*

**Evagelia Voulgari**
https://orcid.org/0000-0003-4437-5907
*University of Thessaly, Greece*

**Savvas Tsolakis**
https://orcid.org/0000-0001-9259-5467
*University of Thessaly, Greece*

## ABSTRACT

*Computational thinking (CT) is a problem-solving process that refers to characteristics such as decomposition, abstraction, pattern recognition, and algorithms. This chapter focuses on educational robotics and their use in developing CT. Firstly, the importance of CT is analyzed along with the way it is applied in the classroom. It goes on discussing the way the introduction of educational robotic systems in education affect CT and the importance of the do-it-yourself philosophy. It presents two widely used educational robotic systems follows, Arduino and Lego EV3, along with examples of their relationship with CT development. The chapter finishes with a comparison of the two systems regarding the easiness and difficulties of using them.*

## INTRODUCTION

Computational thinking (CT) is a way to "solve problems, design systems, and even understand human behavior by making use of the concepts that are fundamental to computer science". CT also involves other areas, such as problem decomposition, data representation, and modeling. "Computational thinking is a fundamental skill for everyone, not just for computer scientists (Chen et al., 2017). According

DOI: 10.4018/978-1-7998-4576-8.ch012

to Wing, (2006) computational thinking should be added to every child's analytical ability on top of the core abilities reading, writing, and arithmetic. Chen et al. (2017) also believe that transfer is aligned with the interpretation of CT which is a fundamental skill. The interested parts computer scientists, cognitive researchers, and educators are still discussing are the nature, definition, and application of CT (Barr et al., 2011).

Educators may use CT components to build CT skills throughout the curriculum for all grade levels and contents. CT as a problem-solving process consists of (Barr et al., 2011):

- the formulation of problems so they can be solved by using computers and other tools
- the logical organization and analyzing of data
- representation of data using abstraction (models and simulations)
- automated solutions as a series of ordered steps which is algorithmic thinking
- identification, analysis, and implementation of feasible solutions focusing on an efficient and effective combination of steps and resources
- generalization and transfer of a problem-solving process to a wide series of questions

Digital construction and creation technologies, combined with appropriate learning methodologies (constructivism and constructionism), may contribute to learning experiences that promote creativity, critical thinking, teamwork, and problem-solving skills. Those skills are imperative for the organizations of the twenty-first century (Alimisis et al., 2019).

CT differs from critical thinking and mathematical thinking according to Barr et al. (2011) because:

- it is a particular combination of thinking skills which provide the basis of a new and effective way to solve problems
- it is tool-oriented
- it uses familiar problem-solving skills such as trial and error, iteration in previously impractical contexts but is made possible because they can be automated and implemented at greater speeds

It is important to offer all the types of thinking to students so that each will find the way of thinking that he/she is good at and improve the other ways that he/she is not doing so well. That way we provide opportunities for all students that have different intelligences (Gardner, 1983).

## BACKGROUND

There is great talk about CT and educational robotics. This chapter focuses on how these two subjects interact, on what educational robotics has to offer to CT and how this can be achieved. To support our case, we used the current bibliography and the authors experience in computer science teaching using educational robotics bearing in mind CT.

CT became popular as a term in 2006 from the article in Communications of the ACM, from the computer scientist Jeannette Wing's in which she manifested CT as a fundamental skill for all and not just for computer scientists (Wing, 2006). Officially the term was used much earlier from Papert (1980), but his definition did not mean the same thing as Wing's (2006) definition and further use (Vaidyanathan, 2016).

CT is to collect data and analyze it in any way so that you can understand the problem in a way to decompose (break it down) into simpler problems. That way you do not only solve a specific problem but by removing details and abstracting you search for patterns, so you end up solving problems that are not specific but of that type. By defining the steps to solve a problem (the algorithm) and if it is possible, to build a model to simulate, test and debug the solution (Vaidyanathan, 2016).

CT can be included comprehensively in education both in primary and junior high schools but not about learning to code, it should be about teaching students to survive and to succeed in the digital world. This is done by enabling students to solve problems using the CT strategy. This is the reason many curriculums chose the term "computational thinking" instead of "computer science". MIT's Scratch is a good example that explains CT by focusing on computational concepts, practices, and perspectives (Vaidyanathan, 2016).

CT is focusing on solving a specific problem, and not about designing and implementing it in code. It is important to offer every student the opportunity to create using technology and to be able to solve problems, using the powerful ideas included in CT. It is also imperative that students should switch from the consumption of technology to create and problem-solving (Vaidyanathan, 2016).

## WHY CT DEVELOPMENT IS IMPORTANT

The power of computational thinking is that it applies to all types of reasoning (Barr & Stephenson, 2011). According to Phillips (2009) "learning activities that allow students to discover and explain scientific relationships, predict events, and learn procedural skills will enable them to better understand these subjects, to predict behavior, and to build CT the power skills".

CT is important because computers and other digital tools extend the power of human thought as an essential part of everyday life and work. There is also a growing need to understand how, when, and where computers and other digital tools that offer similar capabilities can help humanity to find solutions to problems. Apart from that computers offer many communication tools that can help solve problems. For example, by searching or communicating with other people or using simulations. Programming as a process is a problem-solving process. Therefore, it is related to high levels thinking abilities such as problem-solving, logical and mathematical thinking, critical thinking and creative thinking (Korkmaz, 2016). Students learn many elements of the set of CT skills in a variety of disciplines, but educators have to ensure that all students have the opportunity to learn or at least have an overview of the complete set of skills so their combined power is available to them.

Apart from all the advantages mentioned above Miller et al. (2014) and Shell et al. (2017) support the idea that computational thinking and creative thinking help Computer Science students to improve their learning performance.

## CT IN THE CLASSROOM

CT is an analytical skill that all children should master to succeed in the digital era (Czerkawski, 2013). Chen et al., (2017) also valid CT as a valuable skill for learners of any discipline as it applies to everyday problem-solving activities, and other science, technology, engineering and mathematics (STEM) learning areas. That makes it popular among K-9 educators (Mannila et al., 2014).

Students should swift from just tool users as they have the opportunity to build the tools they use. CT uses a versatile set of concepts (abstraction, recursion, and iteration), to process and analyze data, and to even create artifacts (real or virtual). As it is a problem-solving methodology it can be automated, transferred, and applied across subjects (Barr & Stephenson, 2011).

Grover & Pea (2013) believe that educators and curriculum developers should focus on what they expect students should know or should improve, once they go through a curriculum that aims to develop CT. Of course, it is fundamentally important to find criteria on how the use of this curriculum can be evaluated. An example could be what Chen et al. (2017) when they used robotics programming and everyday reasoning to measure students' CT application.

CT in primary and secondary education can change the focus from applications and code to community and context. CT helps educators to broaden and deepen CT usage on a larger scale than ever before (Kafai, 2016).

Leonardo et al. (2018) found that educators' attitude in computer science is related to culturally responsive teaching (CRT) efficacy beliefs. Culture may be used to attract students to study CT skills and to prepare them for computer science studies is not fully exploited. Implementation of CRT is essential in preparing primary and secondary education students for STEM or Information and Communications Technologies (ICT) careers.

For CT to be more efficient both in strategies and characteristics, teachers and students should:

- Use appropriate computational vocabulary to describe a problem and its solution
- Accept any failed solution attempts and recognize that these failures can turn to a successful outcome
- Work cooperatively in teams, using:
  - decomposition - break problems down into smaller parts that may be more easily solved
  - abstraction - simplify from the concrete to the general as solutions are developed
  - negotiation – each group within the teamwork together to merge parts of the solution into the general solution
  - consensus-building - work to build group solidarity as part of one idea or solution

Students need to work in teams. Each team member should have a specific role that alternated in each lesson (e.g. circuit building, code writing, supervisory control) and should be able to explain his or her subject to the rest of the group (Li & Lam, 2013).

The idea is that the instructor presents a problem. The students have all the tools (structures for example if, for, etc.) needed to solve it and they are trying to solve it. When finished each team should present what they created, how their code works and what difficulties they encountered, also other possible solutions are discussed.

Grover & Pea (2013) support that there is no well agreed-upon theory of what CT looks like in practice and how it can be assessed and measured in generic ways. CT terminology provides the basis for analyzing about developing the appropriate curricula for teaching CT in schools.

## CT AND BLOCK-BASED PROGRAMMING ENVIRONMENTS

Computational Thinking is closely related to programming. But introductory programming courses often disappoint both students and teachers. One of the major factors of this is the traditional approach to teaching the fundamentals of programming. Traditional approaches are unable to provide students with an interesting and richly stimulating environment through which problems and concepts are the subjects of investigation creatively and enjoyably (Papadakis & Orfanakis, 2016).

Visual block-based programming environments can be used instead. Visual programing environments support the construction of programs through a drag-and-drop interface. This makes them appropriate coding tools for teaching novice programmers (Papadakis & Orfanakis, 2018). Visual block-based programming environments allow students to create their programs in ways that are more accessible than in textual programming environments. These environments address education that allows students to program without any concerns of syntax errors (errors in typing commands) found in traditional text-based languages (Papadakis et al, 2017). Additionally, in many blocks' languages, blocks belonging to similar collections are grouped by the same colors so users can easily recognize them. Block shapes guarantee that blocks can be connected only in syntactically meaningful ways (Papadakis et al., 2016).

App Inventor (AI) and Scratch as blocks-based programming environments were created for novices' programmers (Papadakis et al, 2017). Papadakis et al., (2016) studied the use of these two block-based programming environments by novice students. The results of the study showed that teaching programming using Scratch and AI contributed so that students significantly improved their performance on programming (Papadakis et al., 2016)

## CT AND ROBOTICS IN EDUCATION

There is an increased focus on technology in education and on engaging learners in constructionist practices with technology (Hughes et al., 2017). For example, when Fronza et al. (2017) tried to foster CT skills through Educational Robotics, CT was used to show that programming a robot is an accessible task. This can be used for coding purposes, but it is more important to also use it to teach students to interact and collaborate with peers to solve problems using a computational strategy (Hambrusch et al., 2009). Educational Robotics has the students' interest as the tools and strategies that are employed (e.g., visual programming languages and teamwork) simplify the learning process (Fronza et al., 2017).

Apart from gaining students' attention, Educational Robotics provides opportunities for teaching CT. Educational Robotics integrate areas, such as programming and engineering design (Shoop et al., 2016). Students work in groups and this establishes a powerful environment to foster CT practices and perspectives, like communicating with others and exchanging ideas (Fronza et al., 2017). CT is also used to show that a good computational strategy can promote the programming procedure (Hambrusch et al., 2009; Fronza et al. 2017). Robotics seems to engage students on a high level and gamification is becoming a trend. Some focus on integrating game mechanics to introduce an element of fun into classroom teaching which is leading to more participatory and more memorable classroom experiences (Bonderud, 2019).

Educational robotics offer educators an educational program that boosts students' imaginations. To encourage this, educators should offer students, the chance to experiment and improve their confidence and self-esteem while using robotics (Ruzzenente et al., 2012). The most common educational robotics systems are the Arduino platform and the Lego Mindstorms EV3 kit and these are presented in this chapter. They both employ a combination of software and hardware. There is similar hardware to Arduino's philosophy such as the Raspberry PI platform and the microbit (addressed to younger children). On the other hand, the Lego WeDo kit is the Lego equivalent kit for a younger audience, while Edison and Thymio use the Lego bricks for the constructive part.

Roscoe, et al. (2014) suggest that to work with robotics projects requires good knowledge of hardware and software and some basic knowledge of microcontrollers. The learning outcomes from working with robotics include:

- understanding basic electronic circuits and specific components (such as light-emitting diodes, and many sensors), these are taught in physics or technology
- work and familiarize with environments like the Arduino software, Scratch for Arduino (S4A, 2020) and LEGO MINDSTORMS Education EV3 software, these can be taught in the computer science curriculum.

At this point, we can say that if an educational organization chooses to teach educational robotics small changes and adaptations may help the implementation and make it easier as well as more profitable for the students.

## WHY DIY IS IMPORTANT?

DIY comes from Do-It-Yourself and refers to activity such as decorating or repairing your home, or making things by yourself, rather than having someone else to do it for you.

Students belong to the digital generation so educational electronics using today's trends should be used. Those are flexible to experiment for someone who is just getting started and does not possess yet any design skills. There are cheap to buy and as there is mainly freeware there are many applications and ideas with instruction on how to implement them.

When students make something, the created object is a demonstration of what they have learned to do and provide evidence of their learning. This allows talking about that object, to communicate about it, to tell a story about it in a way to learn. This learning strategy is one of the positive aspects of the maker movement to education that Dewey (1938), called "learning by doing" approach.

Teachers and curriculum specialists are becoming interested in integrating maker education into the curriculum. According to Gerstein, (2019), this is due to:

- the increase of DIY projects, a lot can be found in online communities e.g. Instructables (2020)
- the eager to provide all students with knowledge and skills to be able to pursue STEM (science, technology, engineering, and mathematics) and STEAM (STEM with the addition of arts) related careers
- the development of technologies including, block-based programs such as Scratch and microcontrollers like Makey Makeys, micro:bits, and Arduinos
- the ability of educators, industry professionals, and other people to share their ideas and projects online freely.

Involvement in DIY projects encourages leaving passive technology usage for some time and create something. There is a shift from being just users of technology to creators. This leads to a new era of innovation by hobbyists and professionals. Advanced computing skills open up a world of opportunities for developing new tools, toys, and experiments (Anderson, 2012). The completed DIY task gives great satisfaction, in a way that is unlike achieving a work task. Also, this involvement is a positive step towards simulating your brain to blossom with new ideas and creativity. At the same time, it increases your self-reliance, making you capable to face difficulties. These skills can be transferred to students and inspire them. Giving students the independence to problem-solving at a practical level is a blessing, and a brilliant bonding exercise (Heath, 2016).

The Constructivist “learning by making” methodology (Papert & Harel, 1991) is strongly related to the “do-it-yourself” (DIY) philosophy. The maker movement has emerged recently in education promising democratized access to opportunities for learning by making (Alimisis et al. 2017). The design of the 21st-century learning environment should engage students with learning tasks, hands-on activities and finally provide essential skills such as creativity, critical thinking, teamwork, and problem-solving (Schon et al., 2014).

Robotic technologies integrated with digital construction and DIY electronics emerge as unique and attractive teaching tools that keep students interested and motivated with hands-on activities and enable them to develop technological interest and competences. Educational robotics are sometimes prefabricated robots with ready-made code to program behaviors for the robots. Thus, robotics is conceived as a “black box” for young children who start playing or interacting with a robot without understanding “what’s inside” and how it works. The trend is to design, pilot and evaluate a digital project to make robots transparent for children to help them make their own robotic artifacts (Alimisis, 2019).

When working with robotics students work the parallel use of software and hardware helps reduce the esoteric nature of some computing aspects and allows participants to work where they have the most interest, without being forced into overly unfamiliar territory (Roscoe et al., 2014).

CT apart from good practice to solve problems and design systems are needed because computing expands beyond the desktop (for instance Arduino and electronic technologies) educational methods have to be extended as well. There are scientists such as Rode et al. (2015) that even argue that education should move from CT to computational making as an educational framework.

## ARDUINO

Rubio et al. (2013) have chosen the Arduino board as the hardware platform for the electronic component. Arduino is an open-source platform and has a huge user community who share their ideas and information, specific projects and solutions along with their usable documentation. The guided use and interactions within an online community have educational value (Severance, 2015).

The community of the makers (Anderson, 2012) and the Arduino’s affordable technology for a diverse range of projects that can suit any need, allows students to get hands-on and interact with technology in ways not feasible before. Educators should catch the opportunities available for new ways of learning both about computers and with computers (Roscoe et al., 2014).

While Arduino was originally conceived and designed to help in the creation of design prototypes with electronic components, it has the potential to bring a hardware element to teaching at all levels of computational thinking and computer science (Severance, 2015). Arduino can be used to actively engage high-school students in STEM education through hands-on projects based on the low-cost Arduino platform (Martin-Ramos et al., 2016).

The traditional separation of hardware and software make it much easier for everyone to build hardware in everyday usable projects. Building hardware is not just easy with microcontrollers; it's also chip, which lets a wide range of engineers solve problems using a combination of custom-developed hardware and software (Severance, 2015).

Many Maker and DIY movements have adopted Arduino, and it is used widely to introduce young students to technology simply and comprehensively. The use of hardware and software are combined to produce new projects. There are many web sites that one can download a software file (program), plug the board in, and after making the changes needed, have working hardware (Severance, 2015).

Students' motivation and learning can be increased using the Arduino board. Students are capable to work with Arduino using the fading scaffolding (Hao, 2016). They perceive it as a unique learning experience and seek for more laboratory devoted to Arduino (Rubio et al., 2013). Several researchers have described situations where students failed to solve a problem at an abstract level but succeeded in using tangible objects (Piaget, 1953). The use of multiple representations of the same knowledge can possibly explain this learning improvement (Ainsworth, 1999). Different representations offer the student alternative paths to knowledge and the student can choose the one that suits him better. Additionally, the availability of different representations might help their abstraction process (Ainsworth, 2004).

Additional skills are required for successful making, namely: aesthetics, creativity, constructing, visualizing multiple representations, and understanding materials. These skills are vital in addition to those of CT and should be included for STEAM education (Rode et al., 2015).

The use of the Arduino controller stimulates the students' curiosity, motivates them to engage in electronics and programming, and helps them to gain a practical understanding of the "abstract code" of the programming structures they are taught. On the other hand, it may create difficulties mainly in the construction of circuits as pupils have usually no previous experience. The available time to complete each activity may be proved short for some teams and the instructor should provide extra activities for the faster teams so they won't get bored.

## Case Study: The Blinking Led and The Fade

Arduino constructions give imminent feedback and the fastest way to get this feedback is via an LED. LEDs are simple devices that are in an on or off condition. However, they form the basis for advanced technologies such as LED TVs, projectors, or lasers (Amariei, 2015).

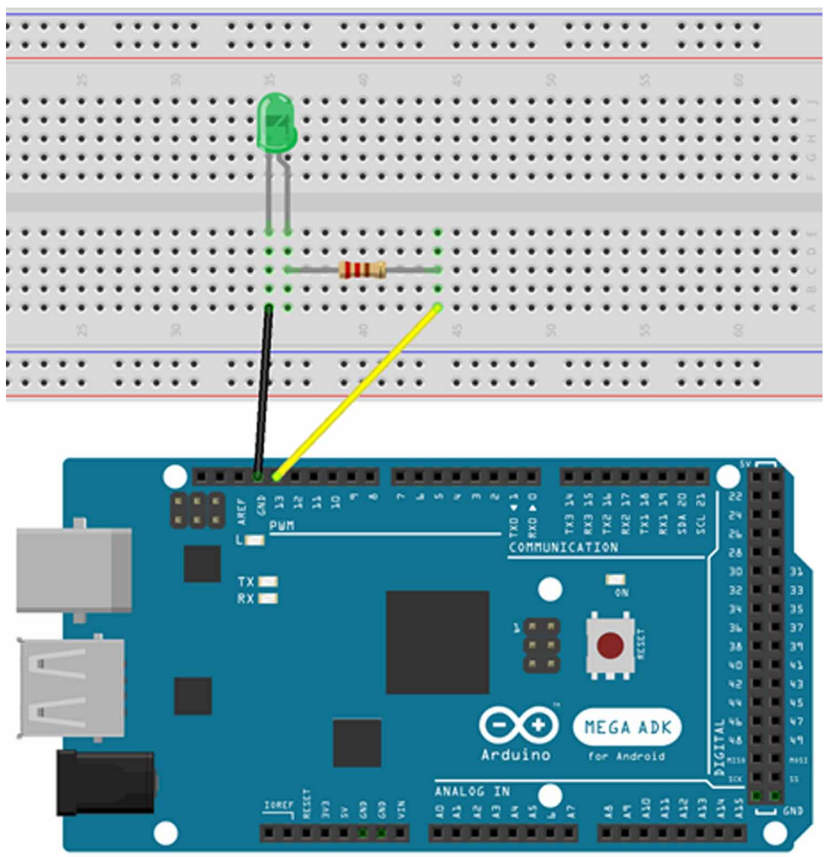
LED stands for Light Emitting Diode and, in its core, it's just a diode that emits light. LEDs are easy to find and inexpensive.

A blinking LED is an easy construction consisting of:

- an Arduino board that connects to a computer to program and have electric power to work
- a breadboard
- jumper wires
- a LED
- a 220 Ohm resistor

The resistor is mounted on the breadboard having one of its ends connected to a digital pin on the Arduino board using a jumper wire. The LED is also mounted on the breadboard with the anode (+) pin connected to the available pin on the resistor. The longer pin is the positive end (anode that needs to be connected to the positive battery pole) of the LED. The LED cathode (-) (the sorter pin) is connected to the Arduino ground (GND) using jumper wires. Figure 1 demonstrates this circuit.

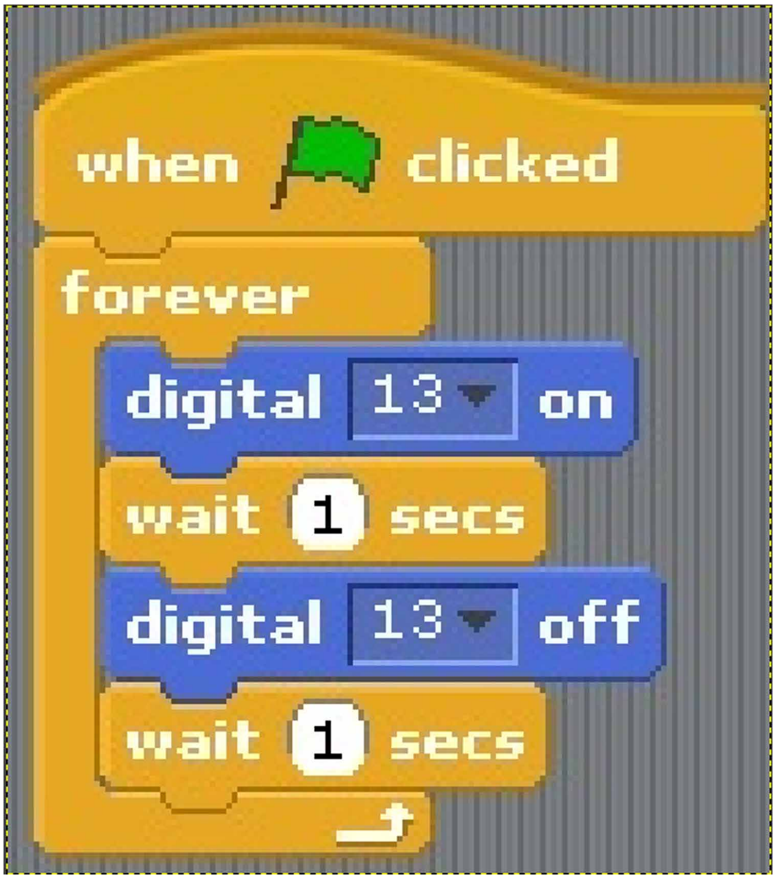
*Figure 1. Arduino circuit*




To program the Arduino there are many different programming environments and languages such as Arduino IDE (Arduino IDE, 2020), Scratch for Arduino (S4A, 2020), ArduBlock, Snap4Arduino or Python. There are graphical environments and easier to start for elementary school students. Arduino IDE is based on C language. Arduino IDE and Python are addressed to high school students and even to professionals in programming. But most of the projects that are found around the web are using Arduino IDE, so it is more useful to start and understand this language.

Below, figure 2, in the first column of the table, demonstrates the corresponding code written in in Scratch for Arduino (S4A, 2020) environment. The second column contains the code written using and the Arduino Software IDE (Arduino IDE, 2020). It is thus clear that Scratch for Arduino is more suitable for younger or novice students. The Arduino Software IDE can be used later in this group of children,

Figure 2. Code written in in Scratch for Arduino (S4A) environment




after they have gained some experience with the Arduino hardware and its programming, using Scratch for Arduino (see Figure 2).

// the setup function runs once when you press reset or power the board

```
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                    // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

When digital pin 13 is set to HIGH, the Arduino controller provides 5 V of electricity in the circuit. The electricity reaches the resistor, the LED and GND. When enough voltage and current is present, the LED is on. The resistor is used to limit the amount of current passing through the LED. Without it, the LED (or worse, the Arduino pin) may burn.

Whatever the software is used, students are introduced in the unconditional repetitive processes and the endless loop. It is the simplest example of the programming loop structure and the Arduino robotic construction helps the students realize how the code, in the loop statement, is executed again and again and forever until they unplug it from their computer.

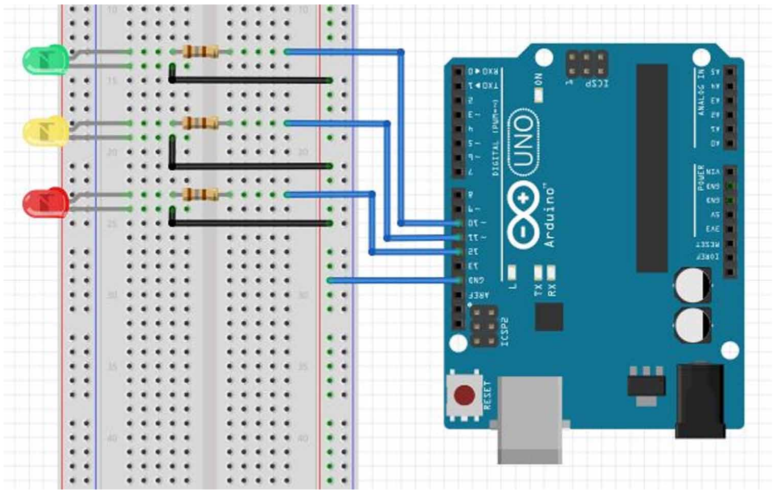
Students can be urged to increase or decrease the delay interval and notice the results on their robotic construction. Therefore, they can understand the time delay concepts. During the whole process, the educator should urge students to experiment safely so they can understand what is going on. They can also change the resistor value used and observe the outcomes.

## Extending the Fade Example

An extension of the above example is the use of three different LEDs to construct a traffic light. Two more LEDs of different colors, resistors and more wires will be used. A project can be implemented by using other materials to construct a traffic light. Figure 3 demonstrates the traffic light example circuit, and the corresponding program follows

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pins 10, 11, 12 as outputs.
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(10, HIGH);  // turn the green LED on (HIGH is the voltage lev-
el)
  delay(6000);                   // wait for a six second
  digitalWrite(10, LOW);   // turn the green LED off by making the voltage LOW
  digitalWrite(11, HIGH);  // turn the yellow LED on (as soon as the green LED
is off and without any delay)
  delay(3000);                   // wait for three second
  digitalWrite(11, LOW);   // turn the yellow LED off by making the voltage
LOW
  digitalWrite(12, HIGH);  // turn the red LED on (as soon as the yellow LED
is off and without any delay)
  delay(5000);                   // wait for five second
  digitalWrite(12, LOW);   // turn the red LED off by making the voltage LOW.
In the next loop the green LED is on without any delay
}
```

*Figure 3. The traffic light example circuit*




Students experiment by setting up the construction and writing down the code. To determine the order in which each LED will go on and off or where a delay is appropriate, they think of the real word traffic light and how it works.

A more complicated example could combine the above traffic light with a pedestrian traffic light. The challenge is that the student should synchronize the function of the two lights in parallel. The simplest implementation is without any changes to the software. By simply connecting two more LEDs one green (for the pedestrian) in parallel to the red LED that was for the cars and one red LED (for the pedestrian) in parallel to the green LED that was for the cars. That way it goes green for the cars it will directly go red for the pedestrians. So whenever the green light is on for the cars, the red LED is on for the pedestrian and vice versa. A possible question is what happens when the yellow LED is on?

A more sophisticated approach is to keep the pedestrian LEDs separate from the car's LEDs. This implies that one has to build two different circuits. At this point, the educator can ask the following questions to build the project incrementally. When should the pedestrian traffic light turn green and for how long? Should the traffic light become green as soon as the pedestrian traffic light turns red? Should we wait a bit more before it becomes green? Are these processes part of the same loop structure?

The final corresponding program is shown below.

```
#define PEDESTRIAN_GREEN         9
#define PEDESTRIAN_RED           10
#define CAR_GREEN                11
#define CAR_ORANGE       12
#define CAR_RED                  13
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pins 9-13 as an output.
  pinMode(PEDESTRIAN_GREEN, OUTPUT);
  pinMode(PEDESTRIAN_RED, OUTPUT);
```

```
  pinMode(CAR_GREEN, OUTPUT);
  pinMode(CAR_ORANGE, OUTPUT);
  pinMode(CAR_RED, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(CAR_RED, HIGH);   //When the traffic light is RED (pin 13)
  //The pedestrian traffic light should be GREEN
  digitalWrite(PEDESTRIAN_RED, LOW);
  digitalWrite(PEDESTRIAN_GREEN, HIGH);
  delay(5000);
  //The pedestrian traffic light turns RED while the traffic light is still 
RED
  //for five more seconds
  digitalWrite(PEDESTRIAN_GREEN, LOW);
  digitalWrite(PEDESTRIAN_RED, HIGH);
  delay(5000);
  //Finally the traffic light turns GREEN and ORANGE
  digitalWrite(CAR_RED, LOW);
  digitalWrite(CAR_GREEN, HIGH);
  delay(5000);
  digitalWrite(CAR_GREEN, LOW);
  digitalWrite(CAR_ORANGE, HIGH);
  delay(5000);
  digitalWrite(CAR_ORANGE, LOW);
}
```

These projects combine physics, electronics, and programming. Students learn to construct electric circuits (physics) and put programming commands in the right order. The use of the Arduino hardware implies direct results (visualization) of the code. All of the above examples show how educational robotics promote CT.

## The Fade

In the first Arduino example, the LED has two states: ON and OFF. But what if we want to adjust the brightness? How can we do that if we can only turn it ON or OFF?

A technique called Pulse Width Modulation (PWM), which is built into the Arduino. It allows to dim the LED with up to 256 settings. The same circuit as in the first example with the difference that the pin used to connect the LED is not digital pin 13 but a PWM pin. PWM pins are those having a ~ in front of the PIN number.

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by
```

```
// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  analogWrite(led, brightness);                   // set the brightness of pin
9
  brightness = brightness + fadeAmount;   // change the brightness for next
time through the loop
  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  delay(30);         // wait for 30 milliseconds to see the dimming effect
}
```

The fade example shows how to fade a LED connected on pin 9 using the analogWrite() function. The analogWrite() function uses PWM. On most Arduino, the PWM pins are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

The fade experiment combines mathematics, logic, and programming. A variable increases or decreases its value until it reaches an upper or lower threshold. A logical check is responsible to keep its value in a specified range. The program includes a sophisticated way to determine whether there should be an increase or a decrease in this amount. The Arduino construction visualizes the result of the ongoing iterative process. Using this example students are introduced in the conditional statements and in Boolean Algebra (Boolean algebra has been fundamental in the development of digital electronics). There are many projects that an educator may think or look for ready projects online that may capture their students' attention, interests or even needs. Educators have to align these to the curriculum and the students' abilities.

## Tinkercad

Students today are routinely becoming virtual tourists - entering simulated worlds to experience curriculum content or to have their real-world augmented with layers of information designed to hook into the curriculum and improve learning opportunities (McKean, 2017).

As most of the schools and universities have limited budgets there is an emphasis on open platform software such as Tinkercad (Kantrowitz, 2019) and consequently open-source electronic prototyping platforms such as Arduino. This leads to many on-line communities where people collaborate and exchange ideas, practices and of course project ideas and implementations.

Tinkercad is a cloud-based software that provides applications to learn the basics of designing the circuits and programming with Arduino (Autodesk, 2019). It can be used for creating and simulating an electric circuit. The interesting part is that the circuit can also include Arduino and its programming using either Ardublock or its original programming language that is similar to C. Thus, provides the environment for making digital circuits and controlling them with code. An option is to see the code and the blocks so this can be used as a transition between Ardublock and C. This transition has to be done because all the available projects on the web are using IDE (IDE, 2020) and very few are using Ardublock.

Alimisis & Loukatos, (2018) suggest that the software tools selected for educational purposes should meet most of the following characteristics:

- pedagogically meaningful
- runnable on the hardware environment
- reduced need for installation/update of software elements
- friendly user interface
- easy integration with the external hardware
- open source
- free or very low cost.

Liu et al. (2013) suggest that working with the virtual robots (Tinkecad is a virtual electronics environment) allowing students to learn more efficiently as the teacher and students were able to focus their time on the programming instead of the mechanical side.

## LEGO MINDSTORMS EV3

The goal of the Lego Mindstorms Robotics (EV3) platform is to provide a physical computing device that can be used to solve problems using motors and sensors in a block-based programming environment (Imberman et al., 2014).

The EV3 education kit is used to teach skills to students such as computational thinking, teamwork, problem-solving and programming (Chaudhary et al., 2016; Korkmaz, 2016). They observe that designing, constructing and programming robots is exciting for students and increases their engagement level. Hands-on assignments and tasks make learning both fun and challenging. They conclude that teaching system integration, creative and innovative design from components, parts, and connectors is easier and more effective for instructors with a robotics education kit than a lecture-based approach.

More skills include proportional reasoning as students calculate wheel rotation and approximate the distance the robot will travel once it is programmed (Grubbs, 2013).

EV3 is a popular system for developing robotic devices. (Brandt & Colton, 2008). It provides versatility, student appeal, all the necessary components for developing simple interfaces, including sensors and motors, and simplicity

EV3 projects could be to introduce curriculum elements such as (Brandt, 2008):

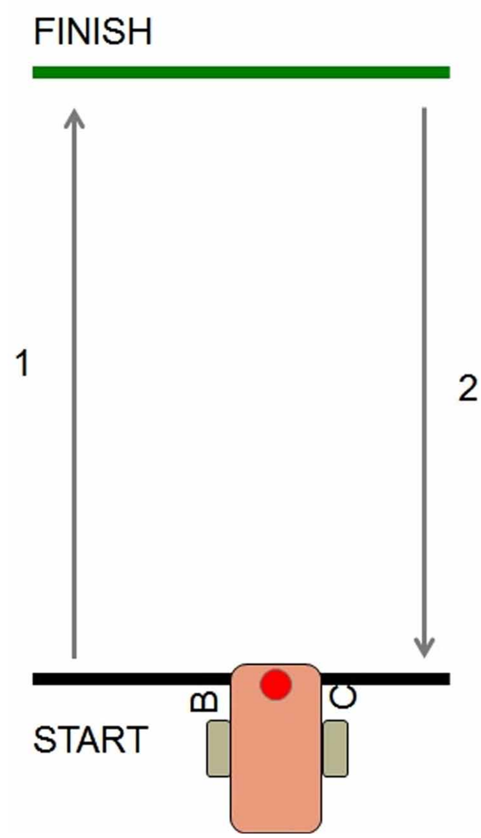
- Mechanical design
- Simple kinematics
- Programming concepts
- Physical principles

These curriculum elements could be applied at any teaching level, with the level of complexity determined by the experience and enthusiasm of the students.

## Case Study: From the Simple Mobile Robot to The Basic Line Following Approach

The simplest mobile EV3 robot is an easy construction consisting of a Lego EV3 brick, two large servo motors, a set of wheels, connector cables, beams, axles, bushes, and pins. The goal is to make the robot move forward or backward (Abdulhamid, 2018) as shown in Figure 4.

*Figure 4. A robot moves forward or backward*




The EV3 programming environment provides three modes that a motor can move, namely SECONDS, DEGREES or ROTATIONS and adjust duration or distance respectively. Students try different speed values and come up to some interesting conclusions; when you move in seconds your speed will matter. This is not the case if you move using degrees and rotations. On the other hand, wheel size affects degrees and rotations.

The next step is to make the robot turn for some degrees. Turning 90° doesn't mean that the robot turns 90 degrees if one just picks 90 degrees for rotations. Students have to experiment with different degrees values

*Figure 5. The three modes that a motor can move*

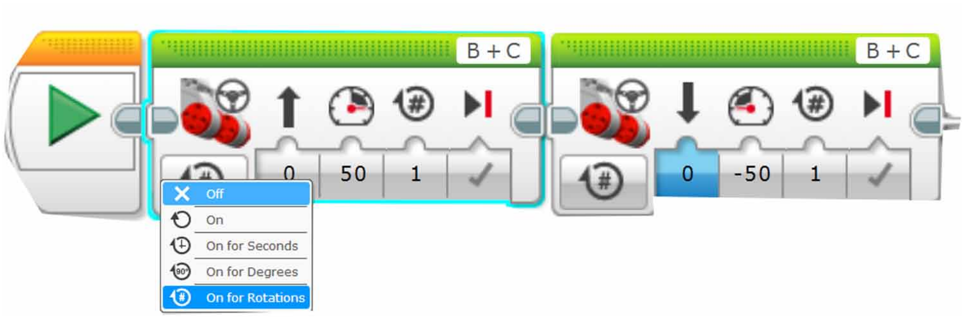
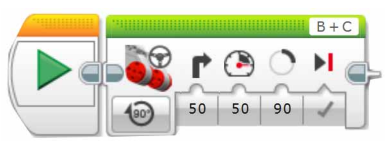


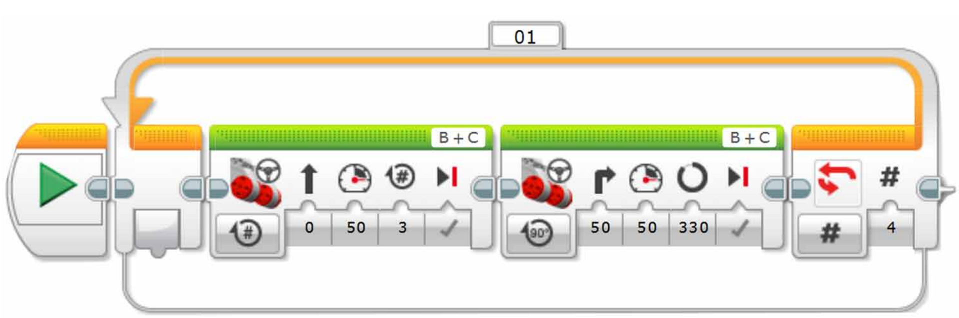

*Figure 6. The robot turns for some degrees*




Putting it all together.

The final challenge is to make the robot move in a square. This combines a forward movement and a turn four times. The student must realize that this is an iterative process and should use the loop structure to form the square instead of 'writing the same code' four times. They, therefore, find the formula of the repeating process.

*Figure 7. The robot moves in a square*




This last challenge engages students in the problem-solving process, as they are asked to decompose a problem into sub-tasks, to explore alternate solutions, and draw conclusions and how they tackle difficulty and complexity (Chaudhary et al., 2016).

Let's go a step forward. What if the robot should move on a specific straight line or even on a curved line? A human can move on any line successfully, but is it the case for a robot? If we draw a large straight line on the floor and use the initial program making the robot move forward for some seconds is it going to stay on the line up to the end? The answer is no. So, students should add intelligence to their robot.

A typical solution is the line following robot (LFR). A line following robot is a mobile robot that can follow a visible line on a surface consisting of contrasting colors. For an LFR to function effectively, it must demonstrate excellent line tracking control. This is achieved by having accurate and responsive control algorithms as well as high precision color sensor systems.

An LFR uses the same construction elements as the simple moving robot, consisting of a Lego EV3 brick, two large servo motors, a set of wheels, connector cables, beams, axles, bushes, and pins. In addition, a color sensor is mounted on it. The robot is intended to follow a black straight or curved line on a white surface. The servo motors are used to drive the two front wheels. A rear small castor wheel supports the robot. The mounted color sensor at the front end is used to identify the line. The color sensor is centered between the two front wheels. It is designed to follow an oval track made of black electrical tape on a white surface. The color sensor identifies the reflected color or light intensity

## Line Following Robot Algorithms

The "line following" algorithm works by using the color sensor (in reflected light intensity mode) to read the changes in the reflected light levels along the edge of a dark and light surface. The reflected light intensity is measured as a percentage from 0% (very low reflectivity) to 100% (very high reflectivity). More light is reflected from a white surface compared to the black surface. Depending on the light sensor value, the motors are directed internally to vary the speed through the change of the direction (in the programming environment)

In a program, white and black values are defined using a threshold value. The threshold is the average of the sensor value with the sensor on the black line and one found on the white area. Different measurements for black and white depending on factors such as the light level in the room, the robot's battery level, and the type of surface used.

The light sensor will read the light value. Then the robot can be programmed such that if the sensor sees black, which is when the sensor value is less than the threshold, the robot should turn right, else it should turn left. The basic line following approach is shown in the figure below and can be summarized as follows:

1. The robot will be started. It will then be set to move forward. It will be made to steer right until it detects the line edge.
2. Once the sensor sees black, the robot will continue to go forward while turning left gradually.
3. Whenever the sensor will see white (i.e. the robot leaving the line), the robot will turn to the right until the sensor finds black again. (Figure 9)
4. The sequence then will be repeated in a loop, unless the robot is stopped.

In the algorithm, the robot has to choose which way to turn when the color sensor sees a different color. The choice depends on what edge of the line it is following and not the initial color it sees. (Figure 10)

In the code below (Figure 11) the robot turns right if it black is detected. For the robot to follow the line it must be placed on the right of the black line in Figure 10.

The line follow algorithm is a typical CT example. Students must decide the robot behavior according to the reflected light intensity value of the light sensor. They analyze the situation in two cases. In the first, if the color sensor sees blacker (value < 34%) then the robot must turn right for some degrees (the first branch is executed). Otherwise, it must turn left (the second branch is executed)

Figure 8. Line following approaches

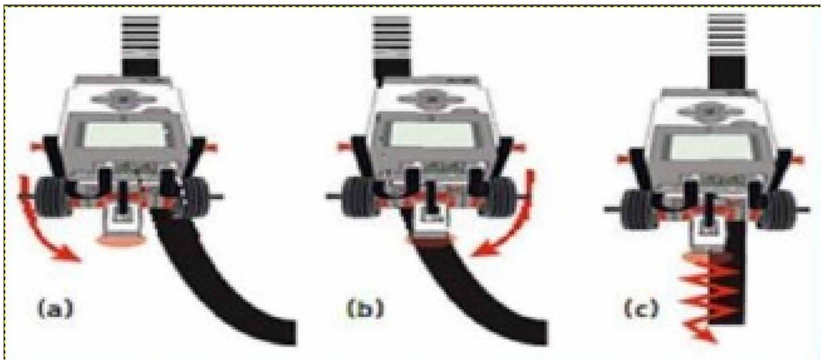



Figure 9. The robot must choose which way to turn when the color sensor sees black or white

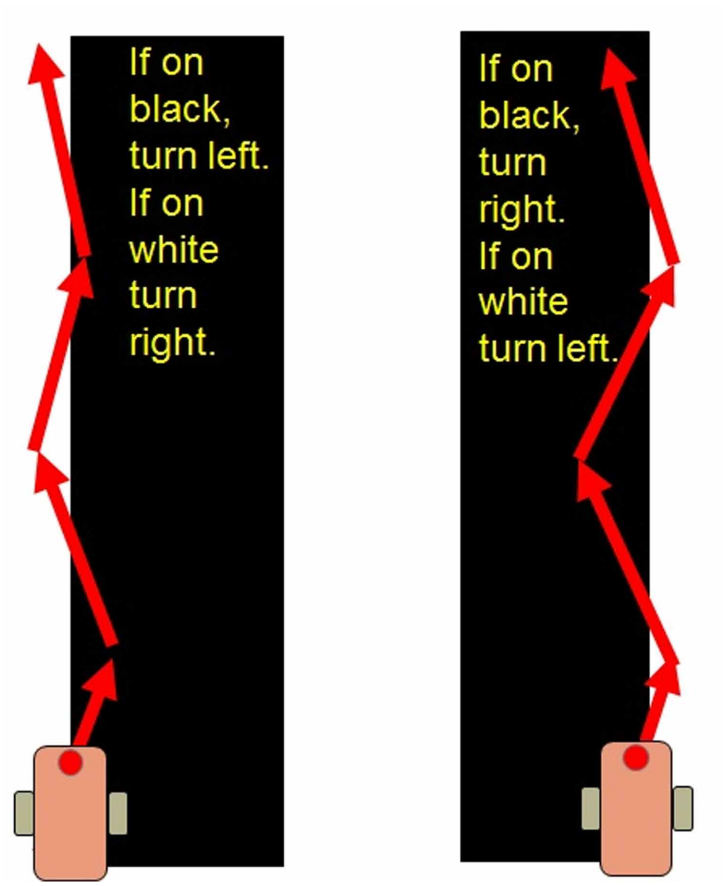


*Figure 10. Choose the line edge to start*

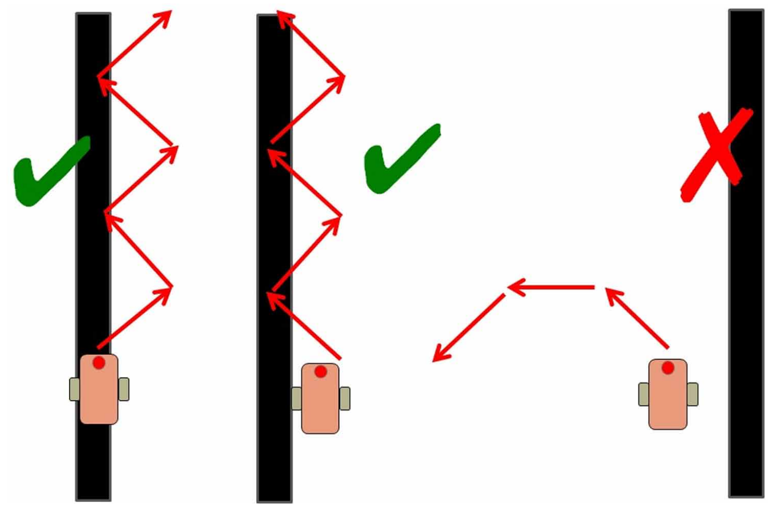
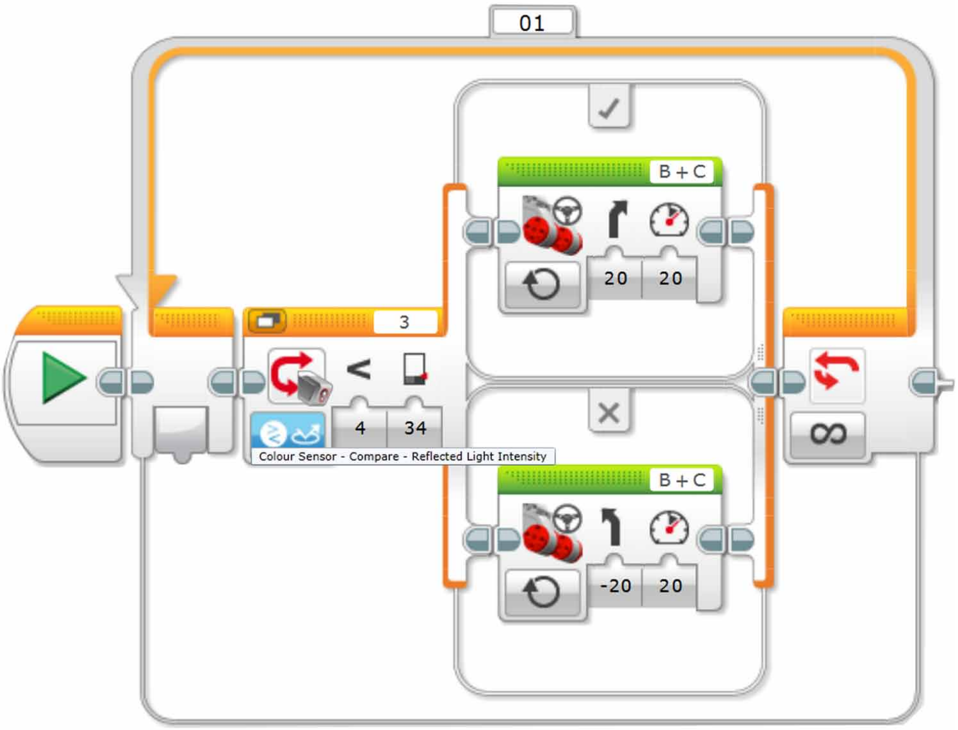


*Figure 11. The robot turns right if black is detected*

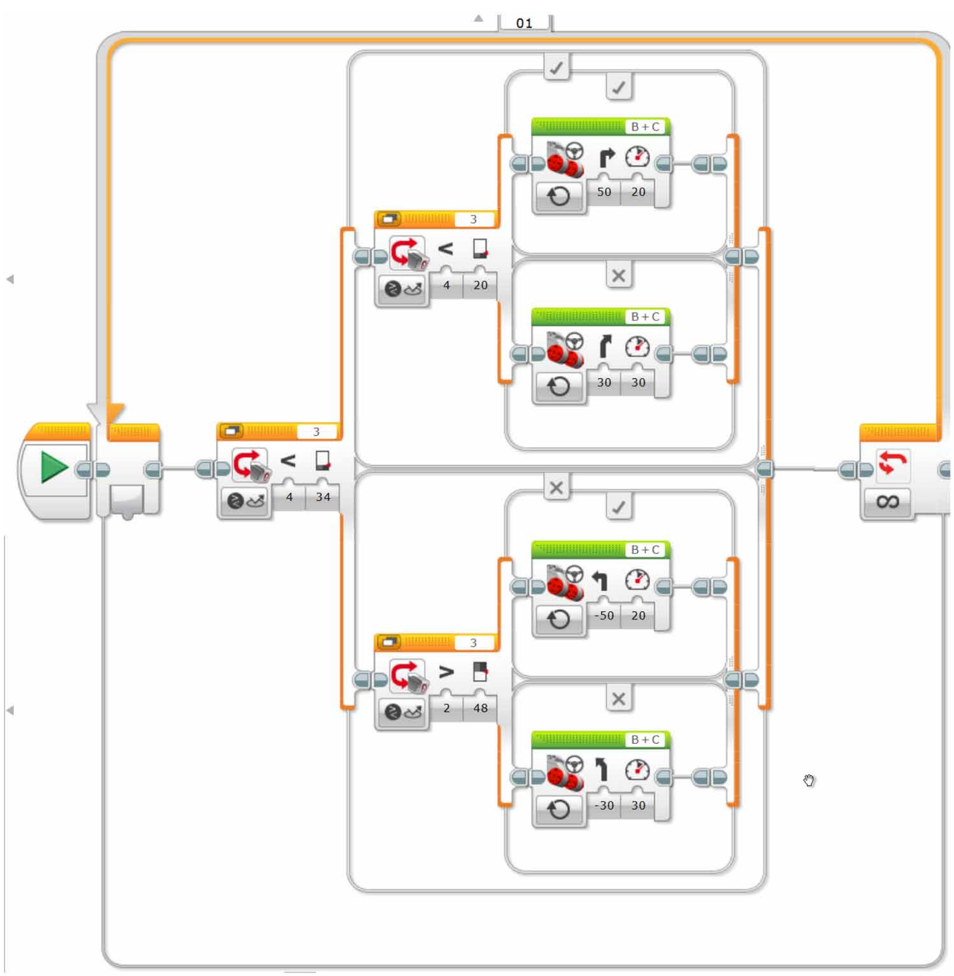


They also must realize that line follow happens on the edges and the program is tightly related to the edge they have chosen to place the robot initially.

The algorithm includes a decision-making process and an experimental phase where the threshold and the speed are determined. In this way, students are introduced to programming structures such as “if … else” and “loop” statements. They have to decide when the robot will stop by adding a stop condition in the loop structure such as stop the robot after some rotations of a motor. They also experiment in the threshold value, according to factors such as the light of the room, the degrees, and the speed to make their robot move quickly and smoothly without living the line. A more curved line demand different manipulation than a straight line.

A solution for a smoother line following in curved lines is to introduce more thresholds for the reflected light intensity value that the sensor reeds and therefore more cases for the angle and the speed of the turn.

*Figure 12. A more sophisticated algorithm*



A more sophisticated algorithm is the PID algorithm. The algorithm includes three parts

**P**roportional (error) that determines how bad is the situation now
**I**ntegral that determines whether the past fixes helped fix things
A **D**erivative that determines how the situation is changing

The PID control combines the error, integral and derivative values to decide how to steer the robot.

## ARDUINO VS LEGO MINDSTORMS EV3

At this point, we compare Arduino and Lego Mindstorms EV3 by presenting the advantages and disadvantages of each of the two systems that seem to be mainstream nowadays.

Arduino is an open-source electronics platform, easy-to-use hardware, and software. It is intended for designers, hobbyists, and anyone interested in creating interactive hardware and software projects. The basic system consists of a microcontroller and various peripheral interfaces that are programmed by an existing software platform that is widely used. The IDE is installed on a computer and then Arduino is programmed using the USB port (Arduino IDE, 2020). The most common type of Arduino is Uno but there are others like the Nano (for compact use) and the LilyPad (for wearable applications). Even from 2011, there was much literature relative to Arduino used in computer science and computer engineering (Jamieson, 2011).

The major benefits for using Arduino in an educational setting are:

- Ease of setup - plug and play
- Many examples for controlling peripherals – preloaded in the IDE
- Many open-source projects available online
- Works on any platform
- Low-cost hardware - build or purchase prebuilt
- Free software
- Low maintenance cost - microprocessors can be replaced at a very low cost
- Students can prototype quickly
- There is a great variety of sensors that can be used
- Can be used to build a wide range of constructions
- It can be programmed in several languages including C and block-based languages.

Programming in IDE (Arduino IDE, 2020) or other text-based environments requires the development of a certain mentality. Concretization makes it easy to minimize the negative influence of working with abstract concepts and to understand the underlying logic. Using the signals sent by sensors, Arduino is a physical programming platform to develop systems and robots which interact with the environment (Kunduracıoğlu, 2018).

On the other hand, Lego uses a block-based visual programming language, to overcome programming difficulties. Using this environment, the logic and fundamentals of programming can be taught easily. As a block-based programming environment, it provides a simple and more appealing environment than a text-based programming environment.

As far as the construction of a Lego robot is concerned, students are more or less familiar with it as they have used Lego or Lego-like bricks. As a result, building a Lego robot is amusing and attractive. The main drawback of Lego EV3 is that it is more expensive compared to Arduino. On the other hand, it is easier for a younger child to build a Lego robot compared to Arduino which implies knowledge of electronic circuits. Moreover, there is a limitation on the sensors provided by Lego which limits the variety of constructions. The drawback of building a more complicated Arduino construction is that one has to use electronic tools like soldering iron used to link cables, voltomer, etc.

The great variety of sensors that can be attached to Arduino lets students develop skills that are related to other technological fields such as electronics, mechanical engineering, and physics.

Finally, the organization of robotic competitions that involve either Arduino or Lego or even both systems is an extra benefit for children that participate as they work more efficiently and cooperate with each other to solve a real-world problem.

In conclusion, depending on the goals of the curriculum, the teachers and the students' needs and abilities both systems can be used to the benefit of the students.

## SOLUTIONS AND RECOMMENDATIONS

Educational robotics contributes to CT development by increasing and evolving students thinking skills. Therefore, their use in the curriculum is strongly recommended through all grade levels and content areas. Any type of robotic system and their corresponding programming environments can be used according to the students' needs, their level, skills, and interests the instructors' knowledge/experiences and last but not least the schools' financials.

## FUTURE RESEARCH DIRECTIONS

The use of educational robotics for increasing students' CT skills raises questions on how to assess students' CT development, how they adopt these skills into the real-life world and to what extent.

Another challenge is to set learning tasks for each domain, and of course, they have to be different for each age group.

## CONCLUSION

CT skills are like cooking techniques while robotics are the ingredients. Students are learning about the ingredients and the characteristics of each ingredient, and then they go on making specific guided recipes. Meanwhile, they learn to experiment, to ask questions, to try to find the answers, to collaborate with peers, to listen and to celebrate their victories. In the end, they are left alone to search, to experiment and to create as members of a group.

Educators use educational robotics to build CT skills across the curriculum through all grade levels and content areas, not only STEM fields. For the newcomers in the area this chapter will a be a useful asset as it offers guidelines and ideas. The robotic systems Arduino or Lego EV3 and others may contribute to formulating technological problems so students can solve them through DIY thus promoting their CT skills. Educational robotics used to build CT skills actively promote 21st-century skills. Students organized in groups analyze data, build models and simulations and find automating solutions as a series of ordered steps which is algorithmic thinking. In the process, they collaborate to implement solutions and they can transfer this problem-solving process to a wide variety of problems.

## REFERENCES

S4A. (2020). *Scratch for Arduino*. Retrieved from https://scratch.mit.edu/

Abdulhamid, M., & Mutheke, M. (2018). Design of Digital Control System for Line Following Robot. *Technological Engineering*, *15*(2), 48–52. doi:10.1515/teen-2018-0018

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education*, *33*(2), 131–152. doi:10.1016/S0360-1315(99)00029-9

Ainsworth, S., & Van Labeke, N. (2004). Multiple forms of dynamic representation. *Learning and Instruction*, *14*(3), 241–255. doi:10.1016/j.learninstruc.2004.06.002

Alimisis, D., Alimisi, R., Loukatos, D., & Zoulias, E. (2019). Introducing Maker Movement in Educational Robotics: Beyond Prefabricated Robots and "Black Boxes". In L. Daniela (Ed.), *Smart Learning with Educational Robotics*. Cham: Springer. doi:10.1007/978-3-030-19913-5_4

Alimisis, D., & Loukatos, D. (2018, June). STEM education post-graduate students' training in the eCraft2Learn ecosystem. In *Proceedings of the 2nd International Conference on Innovating STEM Education*, *Athen, Greece* (pp. 22-24). Academic Press.

Alimisis, D., Moro, M., & Menegatti, E. (Eds.). (2017). Educational Robotics in the Makers Era. Springer. doi:10.1007/978-3-319-55553-9

Amariei, C. (2015). *Arduino Development Cookbook*. Packt Publishing Ltd.

Anderson, C. (2012). *Makers: The New Industrial Revolution*. Crown Publishing.

Arduino, I. D. E. (2020). *Arduino web editor*. Retrieved from https://www.arduino.cc

Autodesk. (2019). *Tinkercad*. Retrieved from https://www.tinkercad.com/

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning and Leading with Technology*, *38*(6), 20–23.

Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What Is Involved and What Is the Role of the Computer Science Education Community? *ACM Inroads*, *2*(1), 48–54. doi:10.1145/1929887.1929905

Bonderud, D. (2019). *How K–12 Classrooms Can Benefit from Robotics*. Retrieved from https://edtechmagazine.com/k12/article/2019/09/how-k-12-classrooms-can-benefit-robotics-perfcon

Brandt, A. M., & Colton, M. B. (2008). *Toys in the classroom: LEGO MindStorms as an educational haptics platform. In 2008 symposium on haptic interfaces for virtual environment and teleoperator systems* (pp. 389–395). IEEE. doi:10.1109/HAPTICS.2008.4479982

Chaudhary, V., Agrawal, V., & Sureka, A. (2016). *An experimental study on the learning outcome of teaching elementary level children using Lego mindstorms EV3 robotics education kit.* arXiv preprint arXiv:1610.09610

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162–175. doi:10.1016/j.compedu.2017.03.001

Czerkawski, B. (2013). Instructional Design for Computational Thinking. In R. McBride & M. Searson (Eds.), *Proceedings of SITE 2013--Society for Information Technology & Teacher Education International Conference* (pp. 10-17). New Orleans, LA: Association for the Advancement of Computing in Education (AACE). Retrieved from https://www.learntechlib.org/primary/p/48062/

Dewey, J. (1986, September). Experience and education. In The Educational Forum (Vol. 50, No. 3, pp. 241-252). Taylor & Francis Group. doi:10.1080/00131728609335764

Fronza, I., El Ioini, N., & Corral, L. (2017). Leveraging Robot Programming to Foster Computational Thinking. *CSEDU 2017 - 9th International Conference on Computer Supported Education*, 109-116. 10.5220/0006310101090116

García-Peñalvo, F. J., & Cruz-Benito, J. (2016). Computational thinking in pre-university education. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*. 10.1145/3012430.3012490

Gardner, H. (2011). *Frames of mind: The theory of multiple intelligences*. Hachette UK.

Gerstein, J. (2019). *Learning in the Making: How to Plan, Execute, and Assess Powerful Makerspace Lessons*. ASCD.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Grubbs, M. (2013). Robotics Intrigue middle school students and build STEM skills. *Technology and Engineering Teacher*, *72*(6), 12–16.

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *SIGCSE Bulletin*, *41*(1), 183–187. doi:10.1145/1539024.1508931

Hao, S. (2016). *Effects of Faded Scaffolding in Computer-Based Instruction on Learners' Performance, Cognitive Load, and Test Anxiety*. Retrieved from http://purl.flvc.org/fsu/fd/FSU_2016SP_Hao_fsu_0071E_12940

Heath, O. (2016). *5 reasons why DIY is good for you*. Retrieved from https://www.housebeautiful.com/uk/renovate/diy/tips/a1131/why-diy-is-good-for-your-wellbeing/

Hughes, J., Gadanidis, G., & Yiu, C. (2017). Digital Making in Elementary Mathematics Education. *Digital Experiences in Mathematics Education*, *3*(2), 139–153. doi:10.100740751-016-0020-x

Imberman, S. P., Sturm, D., & Azhar, M. Q. (2014). Computational thinking: Expanding the toolkit. *Journal of Computing Sciences in Colleges*, *29*(6), 39–46.

Instructables, Yours for the making. (2020). Retrieved from https://www.instructables.com/

IWG. (2010). *Computational Thinking for Youth*. Newton, MA: Education Development Center, Inc.

Jamieson, P. (2011). Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat? In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Kafai, Y. (2016). From computational thinking to computational participation in K–12 education. *Communications of the ACM*, *59*(8), 26–27. doi:10.1145/2955114

Kafai, Y. B., Fields, D. A., & Searle, K. A. (2011, November). Everyday creativity in novice e-textile designs. In *Proceedings of the 8th ACM conference on Creativity and cognition* (pp. 353-354). 10.1145/2069618.2069692

Kafai, Y. B., Searle, K., Kaplan, E., Fields, D., Lee, E., & Lui, D. (2013, March). Cupcake cushions, scooby doo shirts, and soft boomboxes: e-textiles in high school to promote computational concepts, practices, and perceptions. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 311-316). 10.1145/2445196.2445291

Kantrowitz, A. (2019). *Maker Ed in the Art Studio: Preparing Pre-Service Artist/Educators to Integrate Maker Education into P-12 Art Studio/Classrooms*. Retrieved from: https://nyc2019.fablearn.org/wp-content/uploads/2019/04/FL2019_paper_109.pdf

Korkmaz, Ö. (2018). The effect of scratch-and lego mindstorms Ev3-Based programming activities on academic achievement, problem-solving skills and logical-mathematical thinking skills of students. MOJES. *Malaysian Online Journal of Educational Sciences*, *4*(3), 73–88.

Kunduracıoğlu, İ. (2018). Examining the interface of lego Mindstorms ev3 robot programming. *Journal of Educational Technology and Online Learning*, *1*(1), 28–46. doi:10.31681/jetol.372826

Lamprou, A., & Repenning, A. (n.d.). Computational Thinking ≠ Programming. *SI Magazine*. Retrieved from https://magazine.swissinformatics.org/en/computational-thinking-%E2%89%A0-programming/

Liu, A., Newsom, J., Schunn, C., & Shoop, R. (2013). Students Learn Programming Faster through Robotic Simulation. *Tech Directions*, *72*(8), 16–19.

Martín-Ramos, P., da Silva, M. M. L., Lopes, M. J., & Silva, M. R. (2016, November). Student2student: arduino project-based learning. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 79-84).

McKean, R. (2017). Heritage Perth and the mixed reality makeover. *Education Technology Solutions*, *78*, 44–46.

Miller, L. D., Soh, L. K., Chiriacescu, V., Ingraham, E., Shell, D. F., & Hazley, M. P. (2014, March). Integrating computational and creative thinking to improve learning and performance in CS1. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 475-480). 10.1145/2538862.2538940

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, S., Kalogiannakis, M., Zaranis, N., & Orfanakis, V. (2016). Using Scratch and App Inventor for teaching introductory programming in secondary education. A case study. *International Journal of Technology Enhanced Learning*, *8*(3-4), 217–233. doi:10.1504/IJTEL.2016.082317

Papadakis, S., & Orfanakis, V. (2016). The combined use of Lego Mindstorms NXT and App Inventor for teaching novice programmers. In *International Conference EduRobotics 2016* (pp. 193-204). Springer.

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning*, *10*(1-2), 44–72. doi:10.1504/IJTEL.2018.088333

Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. Basic Books.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1–11.

Phillips, P. (2009). *Computational thinking: A problem-solving tool for every classroom*. Retrieved from http://education.sdsc.edu/resources/CompThinking.pdf

Ruzzenente, M., Koo, M., Nielsen, K., Grespan, L., & Fiorini, P. (2012). A review of robotics kits for tertiary education. In *Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum* (pp. 153-162). Academic Press.

Schon, S., Ebner, M., & Kumar, S. (2014). The Maker Movement Implications from modern fabrication, new digital gadgets, and hacking for creative learning and teaching. eLearning Papers, 86-100.

Severance, C. (2015). Python for Everybody, University of Michigan, Coursera. Retrieved from https://www.coursera.org/specializations/python

Shell, D. F., Soh, L. K., Flanigan, A. E., Peteranetz, M. S., & Ingraham, E. (2017, March). Improving Students' Learning and Achievement in CS Classrooms through Computational Creativity Exercises that Integrate Computational and Creative Thinking. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 543-548). 10.1145/3017680.3017718

Shoop, R., Flot, J., Friez, T., Schunn, C., & Witherspoon, E. (2016). Can computational thinking practices betaught in robotics classrooms? In *International Technology and Engineering Education Conference*. Carnegie Mellon Robotics Academy/University of Pittsburgh.

Vaidyanathan, S. (2016). *What's the Difference Between Coding and Computational Thinking?* Retrieved from https://www.edsurge.com/news/2016-08-06-what-s-the-difference-between-coding-and-computational-thinking

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

## ADDITIONAL READING

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Orfanakis, V., & Papadakis, S. (2014). A new programming environment for teaching programming. A first acquaintance with Enchanting. In *Proceedings of the 2nd international virtual Scientific Conference Scieconf* (pp. 268-273).

Orfanakis, V., & Papadakis, S. (2016). Teaching basic programming concepts to novice programmers in secondary education using Twitter, Python, Ardruino and a coffee machine. In *Hellenic Conference on Innovating STEM Education (HISTEM)* (pp. 16-18).

Papadakis, S. (2018). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, *10*(3), 235–250. doi:10.1504/IJTCS.2019.102760

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, St., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in Preschool Education. A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187–202. doi:10.1504/IJMLO.2016.077867

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

## KEY TERMS AND DEFINITIONS

**Algorithm Designing Efficacy:** It is the ability to use algorithms by making logical inquiries knowing for which purpose the algorithms are used.

**Basic Programming Efficacy:** It is the ability to know and apply the basic concepts and stages of the programming process.

**Block-Based Coding Language:** A programming language that uses graphic elements as a means of providing visual cues to the user as to how and where commands may be used.

**Computational Thinking:** This concept is a thinking process which expresses active use of information and communication technologies' concepts in solution of complex problems.

**Novice Programmer:** A computer programmer who is not experienced at programming.

**Problem-Solving Efficacy:** It is the adequacy of performing a problem-solving process in a logical context using experiences.

**STEM:** The term STEM (science, technology, engineering, and mathematics) is an acronym used by those relevant to the educational method concerning the fields of science, technology, engineering, and mathematics.

**Text-Based Languages:** A programming language that does not involve graphical elements (blocks) as a main part of its programming language, but instead is mostly oriented around text.

# Chapter 13
# Exploring Preservice Teachers' Attitudes About the Usage of Educational Robotics in Preschool Education

**Stamatios Papadakis**
https://orcid.org/0000-0003-3184-1147
*Department of Preschool Education, University of Crete, Greece*

**Michail Kalogiannakis**
https://orcid.org/0000-0002-9124-2245
*Department of Preschool Education, University of Crete, Greece*

## ABSTRACT

*Educational robotics have become popular worldwide with a broad range of students, including preschoolers. Although the impact of robotics technology in classrooms has been extensively studied, less is known about preschool teachers' perceptions of how robotics technology impacts learning and its relation to use in the classroom. This is problematic since we know that teachers' perceptions have a great influence on their teaching practices. This study used survey data gathered from 102 students of the Department of Preschool Education in a University in Greece. A questionnaire developed by the researchers were used as data collection tool. At the end of the study, it was determined that preservice preschool teachers' attitudes about educational robotics usage in preschool classrooms were positive although they lack in relevant knowledge. These findings are discussed with respect to their educational implications.*

## INTRODUCTION

Over the past two decades, the current societies have been establishing their economic and social prosperity on the introduction, diffusion and effective use of Information and Communication Technologies (ICT) in all every-day and professional activities of their citizens (Bikos, Stamovlasis, & Tzifopoulos,

DOI: 10.4018/978-1-7998-4576-8.ch013

2018). As a result, the debate whether children should be interacting with and using digital technologies is now considered to be outdated (Walsh & Campbell, 2018; Papadakis & Kalogiannakis, 2020). New digital and interactive technologies offer young children the possibility to develop and integrate knowledge about computer science and engineering (Kalogiannakis & Papadakis, 2020).

Nowadays young age children can build their own robot, program it to do what they want, and decorate it with art supplies (Bers, 2018). Thus, in education sector especially, the question today evolves around how children will be enabled to use digital technology in a way that best benefits them (Demetriou & Nikiforidou, 2019; Papadakis, 2020).

The rapid developments of technology have affected the use of technological tools such as smart mobile devices, mobile applications, and robots in preschool education. Over the past 5 years, there has been a recent surge in the number of robotics kits, and relative toys for young children (Sullivan & Bers, 2019). For instance, STEM education toys, such as Beebot, and KIBO offering a variety of programming interfaces, are being marketed as tools to teach computer programming and problem-solving skills in preschoolers (Hamilton, Clarke-Midura, Shumway & Lee, 2019; Sullivan & Bers, 2019). This is considered important as Wing advices that computational thinking skills must be introduced from the early years of schooling in order to ensure a common and solid basis of understanding and applying these skills (Wing, 2006, p. 3720).

In general, children who engage in playful programming activities with computational toys utilize CT skills as they participate in computational challenges (Hamilton et al., 2019). Studies have found that even young children can be introduced to easy to use visual programming tools and robotics platforms (Chalmers, 2018). As early childhood is an important time for young children to grow, play, and explore the world they live in (Bers, 2018) robotics can be an effective way to introduce computational thinking as it involves students being able to systematically process tasks and developing the sequenced step by step coding commands needed to program a robot (González-González et al., 2019). The same researchers highlight the fact that educational robots promote superior cognitive functions, like executive functions involved in problem solving, such as reasoning and planning, in typically developing preschool children (González-González et al., 2019).

Although the potential positive impact of technological tools such as tablets and robotic kits on learning in schools has been widely acknowledged, more research is needed to explore in depth the teachers' perceptions of the impact of robotic kits in learning in preschool education (Papadakis & Kalogiannakis, 2010), and how these perceptions could inñuence the use of these tools in the learning process. For instance, Yelland (2011) highlights the fact that children often come to their early learning service with knowledge of technology that may be unfamiliar and even intimidating for their educators. On the contrary, lack of technical knowledge and support has been identified in relevant literature as a potential challenge for teachers implementing robotics in their classroom (Alafodimos, Kalogiannakis, Papadakis & Papachristos, 2009; Chalmers, 2018). As such, educators must think carefully and critically about how their beliefs and confidence with technology influence what they choose to integrate and how their choices can best align with children's social and cultural experiences (Papadakis, Kalogiannakis & Zaranis, 2019).

In consequence, the purpose of the present research is to explore the preservice teachers' perceptions of the impact of educational robotics in learning in preschool education, and how these perceptions could influence the use of these tools in the learning process. In this context, the rationale of this article is to explore how Greek early childhood students as future teachers bridge educational robotics to their views and attitudes on how this type of technology contributes to children's learning and development.

The research was done with 102 preschool students in the University of Crete, Greece. A questionnaire developed by the researchers was used as data collection tool. The gathered data was analyzed with descriptive statistics. At the end of the study, it was determined that preservice preschool teachers' attitudes about educational robotics usage in preschool classrooms were positive although they lack in relevant knowledge.

## LITERATURE REVIEW

Several studies have shown that applying technology products to education could enhance children's motivation and participation in learning and that learning with technology products improved cognition, constructive knowledge, problem solving, and language (Kalogiannakis, & Papadakis, 2008; Couse & Chen, 2010). As a result, governments around the globe have invested extensively in educational technology highlighting the growing link between technology and economic prosperity as a driver for transforming teaching and learning through digital education (Papadakis, 2018a). Educational technology is an umbrella term that includes a variety of technological tools and resources and their use through procedures and approaches that aim to enhance learning experiences in several different settings including formal and informal learning (Demetriou & Nikiforidou, 2019).

Additionally, the last years, studies have shown that the introduction of coding in the curriculum at an early age is a long-term investment in bridging the skills gap between the technology demands of the labour market and the availability of people to fill them (Depryck, 2016; Orfanakis & Papadakis, 2016). Researchers have argued that computer science is well-suited for early childhood education as it offers a learning environment where young children can engage in simple coding or early programming while having the opportunity to explore and experience problem-solving, computational thinking etc., when supported by well-designed and developmentally appropriate digital technologies (Papadakis, Kalogiannakis & Zaranis, 2018; Murcia, Campbell & Aranda, 2018).

One of the emerging resources to develop computer thinking, as well as students' own digital competence is educational robotics. Studies have proven that learning to code, particularly with tangible systems such as robots, can positively impact young children's ability to sequence correctly and debug/trouble-shoot (Chalmers, 2018). There is evidence that even pre-kindergarten students can learn computational thinking skills while working with robots as research has shown that children as young as three years old can build and program simple robotics projects (González-González et al., 2019). Learning to code robots provides young children with twenty-first century skills that include coding as a literacy practice. Through learning the literacy practice of coding, in much the same way they learn the literacy practices of reading and writing, young children also develop computational thinking skills where they engage in problem-solving (Walsh & Campbell, 2018).

Computational thinking and coding are becoming an integral part of K-12 education, with female students being underrepresented in such subjects. The proliferation of technological tools and programming environments offers the opportunity for creative coding activities for children and increases the need for appropriate instructional practices (Papavlasopoulou, Sharma, & Giannakos, 2019).

Because it is also widely believed that by learning to code, young children engage in computational thinking or critical problem-solving skills and habits (Bers, Flannery, Kazakoff, & Sullivan, 2014), the call for early years literacy teachers to also be teachers of coding, is beginning to take hold (Walsh & Campbell, 2018).

Educational robotics is a type of educational technologies that is more suited to an early year learning environment than others. A robot is a tangible object, with which you can interact with the environment through programmed instructions, also useful in Kindergarten, as a tool for the development of cognitive skills, through play, creativity, or the resolution of challenges (Bers, 2014). Robots have appealing features and functions for children that able to maintain children's attention for a longer period of time, improving their performance, their ability to concentrate, and their cognitive flexibility, but it has also been found that a robot alone is not motivational enough (González-González et al., 2019).

The use of robots is not only a motivator in the classroom, but for its technological features, allows the design of activities that promote both computational thinking, and skills related to scientific and mathematical skills such as social skills, collaborative and teamwork (Benitti, 2012; Papadakis & Kalogiannakis, 2019). For instance, researchers have observed young children becoming socially isolated when they are focused on a screen, yet when playing with tangible coding technologies such as Bee-Bot and Cubetto they collaborated and communicated with others as they coded the actions of the 'robot' (Murcia, Campbell, & Aranda, 2018).

Any effort to facilitate creativity in school settings must necessarily consider the role of the individuals, that is, the teachers, who are called to realise the goals specified by national curricula and educational programmes. Teachers' perceptions about the impact of educational technology in learning reñect their beliefs about how this technology inñuences learning processes (Ertmer, 2005; Papadakis, 2018b; Papadakis, Tousia & Polychronaki, 2018). Understanding teachers' perceptions of educational technology provides a means for promoting a more meaningful use of this technology in the classroom setting.

It is apparent through international research and practice that many early childhood educators lack confidence and knowledge about digital technologies and developmentally appropriate pedagogies for integrating them into the learning environment (Papadakis, 2018c). Educators often do not have a strong understanding of the digital or ICT skills needed to guide young children's play with tangible coding technologies (Campbell et al., 2018). However, educators with a positive attitude as well as high self-confidence towards technology are more likely to use technologies in the learning environment (Murcia, Campbell & Aranda, 2018; Murcia & Pelliccione, 2017). Early childhood teachers have a general lack of knowledge about how robotics technology can be implemented in the classroom (Chalmers, 2018). For early years educators to feel motivated and confident to use mobile technologies to teach coding as a literacy, they first need an introduction to the basics of coding. They also need information that explains how teaching coding as a literacy dovetails with play-based learning and teaching other forms of literacy aligned with local, state and national curriculum documents. (Walsh & Campbell, 2018).

Given the media and political rhetoric coupled with the increasing number of coding apps (Code Karts, Think & Learn Code-a-pillar Twist, ScratchJr, etc.) and robotic toys or robots for young children (WowWee Elmoji, Blue-Bot, Cubetto, Kibo, Alpha, etc.), early years educators, by default, are suddenly responsible for introducing the new literacy of coding to young children. (Walsh & Campbell, 2018).

Research suggests that coding experience enhances children's interest in knowledge and skills include engineering-science, technology, engineering, and mathematics (STEM) areas, and reduces gender-biased stereotypes associated with STEM careers (Metz, 2007). In fact, recent studies prove the positive effects of using coding in early childhood on children's attitudes, knowledge, and skills in various areas such as problem-solving and computational thinking (e.g. see Bers et al., 2014; Sullivan and Bers, 2016; Lee, 2019).

Several studies have reported relationships between demographic characteristics of teachers and their reported use of technologies; age, gender, race, education level, socio-economic status of students taught, years of teaching, years of technology use, specializations, and size of school were among the factors reported in key literature. Another key factor affecting the integration of digital technology is the technology-related training offered to teachers.

However, many of these educators have had limited exposure to the idea that coding is a type of literacy. This is because most early childhood teacher education programmes do not include the teaching of coding as part of literacy subjects and computer science subjects are generally only taught in upper primary and secondary schools (Walsh & Campbell, 2018)

Technology-related training plays a crucial role in developing teacher's competency with computer applications as well as influencing teachers' attitudes towards digital technology (Wozney, Venkatesh & Abrami, 2006)

## METHODOLOGY

### Purpose of the Study

The purpose of the present study was to examine student preservice teachers' beliefs about educational robotics.

### Participants

The participants were pre-service teachers (N=102), belonging to the department of Preschool Education at the University of Crete during the academic year 2018-2019. The sample corresponds to 5% of the students' population studying in School of Preschool Education. In addition, the students comprising our sample were in the last year of their university studies and had completed, or were completing, their school practicum at the time of the study. It is important to note that the nature and content of the studies and training of this study's participants is not part of this article.

Participants have had general teaching and learning sessions on the role of educational technology in early years, but this has not been included in the context of this study. Students participation was optional, voluntary and fully anonymous.

### Research Design

The study adopted a quantitative approach and was conducted in one stage that involved a survey, including both open and closed questions. The data were collected through a paper-and-pencil procedure, which lasted about 10 minutes. The questions focused on general views of the educational robotics, to gain an insight into the practices, perceptions and views implemented in a range of settings with these technological devices.

Below are sample questions included in the survey:

- What do you believe are the advantages of using educational robotics in the classroom?
- What do you believe are the problems of using educational robotics in the classroom?
- What factors or circumstances would allow you to use educational robotics in the classroom?

## Instrument

Student teachers' beliefs about educational robotics were assessed with a questionnaire that consisted of two parts. Part A included seven questions. All the questions were close ended, requiring students to define their socio-professional background, training experience about educational robotics, initial experiences with educational technology. Part B of the questionnaire included 15 questions. In their majority, the questions and the statements on this part of the questionnaire examined beliefs and perceptions about the introduction of educational robotics in the preschool classroom.

The development of scale included in this section has been done considering previous questionnaire and were informed by literature on learning impacts from robotics technology, reviewed in the theoretical background section. It includes items, which reñects a wide range of potential learning impacts resulting from using robotics technology and represents the teachers' perceived impact that using robotics technology plays in learning. A ðve-level ordinal rating scale, ranging from 1 "strongly disagree" to 5 "strongly agree" has been used. The participants were recruited through a number of calls via the virtual learning environments of their courses. The alpha coefficient of the internal reliability of the instrument was 0.89.

## Ethical Considerations

The study was approved by the University of Crete Ethics Committee and followed the university guidelines on ethical practice when researching with adults. Anonymity and confidentiality, gaining consent and ongoing assent, and the right to withdraw were the ethical principles adhered to. Rigorous procedures were followed to ensure confidentiality, anonymity and the voluntary nature of being involved in the study.

## Data Analysis

The data were analysed using IBM SPSS Statistics for Windows, Version 23.0.

# RESULTS

The sample of participants consisted of women. Only 5 (4.9%) of the students had taken part during their studies in any form of training programs related to the use of educational robotics. The rest of the students (97, 95.1%) had never the opportunity to take part in any training activity related to educational robotics. The asymmetry of the data regarding the gender and the training experience of the students surveyed has not allowed analysis of the relationship between gender and the answers obtained. Regarding their knowledge in scientific disciplines related to educational robotics such as programming, robotics as well as exact sciences the results of the participants answers are presenting in Table 1.

From the results we can conclude that students in general have no background knowledge in robotics as well as in programming. Their background knowledge seems little better in exact science. These results could be a result of their studies in university or in their background during the school years.

This finding is consistent with participants response regarding their knowledge of educational robotics. The majority of those interviewed stated that they had little (82.4%) or moderate knowledge (15.7%). In this study, only 2 of the students (2%) stated that they had reasonable knowledge about the field of educational robotics (see Table 2). Further analysis showed that the 2 students who declared that

*Table 1. Students background in programming, robotics, and exact science knowledge*

| Knowledge | | | | | | |
|---|---|---|---|---|---|---|
| | Exact science | | Programming | | Robotics | |
| | Frequency | Percent | Frequency | Percent | Frequency | Percent |
| Not at all | 22 | 21.6 | 51 | 50.0 | 96 | 94.1 |
| A bit | 61 | 59.8 | 39 | 38.2 | 6 | 5.9 |
| Average | 15 | 14.7 | 12 | 11.8 | - | - |
| Too much | 4 | 3.9 | - | - | - | - |

*Table 2. Educational robotics knowledge*

| Educational robotics knowledge | | |
|---|---|---|
| | Frequency | Percent |
| Not al all | 36 | 35.3 |
| Little | 48 | 47.1 |
| Average | 16 | 15.7 |
| Enough | 2 | 2.0 |

they had sufficient knowledge they had also participated in some type of training regarding educational robotics usage. Additionally, their training covered concepts such as introduction to robotics, sensors & commands, educational environments and robot programming. Finally, the students reported that the training they attended was offered by private educational centers.

The question regarding the introductory level of educational robotics revealed that the participants have no clear attitude. However, in absolute numbers, the introduction of educational robotics to classes larger than the kindergarten predominates in their responses while 15 students preferred not to answer this question (see Table 3).

It is noteworthy that interviewed students consider that kindergarten is not the appropriate stage for introducing educational robotics. Our findings are in contrast to some prior studies that have found that young children can be introduced to easy to use visual programming tools and robotics platforms that can help teachers focus on developing students' coding and computational thinking skills (Chalmers, 2018). This is problematic since we know that teachers' perceptions have a great iñuence on their teaching practices.

*Table 3. Introductory class of educational robotics*

| | Frequency | Percent |
|---|---|---|
| Kindergarten | 27 | 26.5 |
| A-B (Primary School) | 17 | 16.7 |
| C-F (Primary School) | 43 | 42.2 |
| Do not know | 15 | 14.7 |

All the students were asked to rate the factor or the factors that they believed would influence their future involvement with educational robotics in the preschool classroom. A number of factors exist from literature that affect directly or indirectly the wider use of ICT of pre-service and in-service teachers such as the lack of knowledge, the lack of infrastructure, etc. From students' answers derives that students consider as the main deterrent to introduction of educational robotics into their teaching daily practice the lack of infrastructure. The reason might be the lack of equipment that students faced during their practical experience in the school and classroom. The lack of relevant knowledge was the second most important factor. These results are to be expected given the extremely low levels of knowledge they declared in robotics, programming, and in the exact sciences. Their practical experience in the school may also influence the classification of the lack of interest of the administration as the third most important factor for non-using educational robotics in their future teaching. Factors such as the lack of personal interest and the lack of educational time classified very low in comparison with the other 3 predefined factors. This is very important considering that students, despite their lack of knowledge, have a positive attitude towards educational robotics. Thus, we can conclude that with a proper working environment and training or guidance they would be willing to adopt educational robotics in their daily teaching practice. In addition, in the question regarding their desire to take part in training robotics seminars 84 students (82.4%) stated that they would like to participate in this form of training in the near future. Only 18 students (17.6%) stated that they did not want to participate. Finally, students were asked which would be best suited for teaching robotics skills to kindergarten girls and boys. Their answers were evenly distributed among the 3 options. In absolute terms, however, students stated that they would prefer to teach robotics in kindergarten themselves with appropriate education and training to improve their practice (see Table 4).

*Table 4. Person responsible to introduce robotics in kindergarten*

| | Frequency | Percent |
|---|---|---|
| Informatic teacher | 33 | 32.4 |
| Specialist in collaboration with university/organization/community | 32 | 31.4 |
| A kindergarten teacher with knowledge in the domain | 37 | 36.3 |

Students were then asked their attitudes towards the use of educational robotics in their future teaching practice (see Table 5). Participants' responses (on a 5-point Likert scale from strongly disagree [1] to strongly agree [5]) were coded and analyzed.

From the students' answers, it appears that most students consider educational robotics usage in preschool classrooms as something positive in general. They specifically state that they feel comfortable with the idea of using educational robotics as a learning and teaching tool (M = 3.44), exciting to use educational robotics as a learning tool (M= 3.72), considering that educational robotics is a useful tool for educators (M= 3.84) as robots help teachers teach in more effective ways (M=3.60). They also consider that the use of educational robotics helps students understand concepts more effectively (M=3.50), as they allow them to express their thinking in many ways (M=3.55).

*Table 5. Descriptive statistics of learning impact of robotics technology in preschool classrooms*

| | M | SD |
|---|---|---|
| Q1. I feel comfortable with the idea of using educational robotics as a learning and teaching tool | 3.44 | .815 |
| Q2. The use of educational robotics in learning and teaching annoys me | 3.11 | 1.024 |
| Q3. Using educational robotics as a learning and teaching tool makes me skeptical | 3.34 | .939 |
| Q4. Using educational robotics as a learning tool excites me | 3.72 | .905 |
| Q5. Using educational robotics as a learning and teaching tool scares me | 2.55 | 1.040 |
| Q6. Educational robotics is a useful tool for educators | 3.84 | .741 |
| Q7. Educational robotics do not promote students learning because they are not easy to handle | 2.46 | 1.012 |
| Q8. Educational robotics help students understand concepts more effectively | 3.50 | .830 |
| Q9. Educational robotics help students learn because they allow them to express their thinking in many ways | 3.55 | .828 |
| Q11. Educational robotics help teachers teach in more effective ways | 3.60 | .836 |
| Q12. Educational robotics do not promote effective teaching because they present technical problems | 2.62 | .821 |

Even in questions with negative connotation about educational robotics, students' answers confirm their positive attitude toward educational robotics as their answers to these questions scored the lowest (Q5, Q7, Q12). For instance, students consider that robots do not promote students learning because they are not easy to handle (2.46) and robots do not promote effective teaching because they present technical problems (M=2.62) or that the usage of educational robotics as a learning and teaching tool scares them (min = 1, max =5). However, it is worth pointing out that in 2 questions respondents' answers do not go along with their positive attitude towards the use of educational robotics in their future teaching practice. In particular, they state that the use of educational robotics in learning and teaching annoys me (M=3.11) and the use of educational robotics as a learning and teaching tool makes me skeptical (M=3.34). Perhaps, the negative connotation in the respondents' answers goes hand in hand with their lack of knowledge of concepts relative to programming, Computational Thinking in general as well as their lack of training in educational robotics.

Finally, the specificity of the sample does not allow their answers analysis to go beyond descriptive statistics. Specifically, the majority of students (95.1%) stated that they had never had the opportunity to participate in an educational robotics training while the sample consisted almost entirely of women (98%). Most of them reported that they had no or little knowledge (82.4%) in the exact sciences, programming (88.2%) and robotics (100%).

## LIMITATIONS AND FUTURE DIRECTIONS

Despite the clear findings, the present research has a number of limitations originating predominantly from its exploratory character. The sample is not large enough and representative of the population and, consequently, conclusions need be corroborated by conducting new researches. Since it is the first endeavor on this matter and the sample was not representative of the Greek pre-service teachers the results should be treated with caution as far as generability issues is concerned. Second, despite the fact that this work provides interesting data about these emergent research topics, the independent factors affecting the preservice perceptions regarding are not limited to those examined in the present work; there is a plethora of variables, cultural or individual differences, playing an important role in the digital divide and they are worth investigating.

## CONCLUSION - DISCUSSION

We live in an era where technology develops and expands into all aspects of our life (Marsh, 2016), as it forms an integral part of everyday life both of adults and children (Demetriou & Nikiforidou, 2019). Scholars and educators welcome the availability and diversity of computational devices, including programming tools like Dash and Dot, Robot Turtles, and Cubetto, yet much can be done to integrate these devices appropriately and effectively within classroom settings and K-12 curriculum (Hamilton et al., 2019).

Educational technology and effective integration of technology in early childhood education is an emerging, complex field of necessary interest (Demetriou & Nikiforidou, 2019). The developmentally use of digital technologies in early childhood can support young children's development of problem-solving and computational thinking. (Murcia, Campbell & Aranda, 2018). This integration can be done through the use of programmable robotic devices and the application of project-based learning methodologies (González-González et al., 2019).

The range of new technologies designed specifically for young children has grown significantly in recent years making it easier to find tools that engage young children in technology and engineering initiatives (Sullivan & Bers, 2019). Several companies have developed various educational robotics kits to teach robotics to students from kindergarten to K-12 (Papadakis, Kalogiannakis, Orfanakis & Zaranis, 2019). Thus, a wide spectrum of computationally-themed manipulatives and toys, ranging from digital versions to completely screen-less options, are available to teach young learners (Hamilton et al., 2019). In their majority these programmable robotic devices serve as learning tools to ensure fun, hands-on activities in an attractive learning environment and promote students' interest and curiosity (Eguchi, 2010).

According the European Commission the innovation in education systems, is most effective and sustainable when embraced by well-trained teachers and embedded in clear teaching goals (European Commission, 2018). The rapid adoption of new technologies in education necessitates teachers learning about the technology at the same time as being able to integrate developmentally appropriate activities in the classroom (Chalmers, 2018). In order for teachers to teach successfully with programming tools, such as robots, they need to understand the technology, the computational skills they are teaching with the technology, and how to apply teaching practices relevant to teaching the skills (Chalmers, 2018).

In this respect, early childhood students' views and attitudes regarding educational technology matter.

As future practitioners, their professional practice will depend on their beliefs, literacy, training, but also on the availability and accessibility of the technological devices in their classroom (Instefjord & Munthe, 2017). Even though the sample of this study was relatively small, and this is one of the limitations of the present study, participants similar to other studies provided aspects of their perceptions about educational robotics, by unpicking how this technological environment can promote children's learning and development (Demetriou & Nikiforidou, 2019).

One of the main themes that emerged focused on the teachers' concerns about their own knowledge about the robotics technology. Practicing teachers and teacher candidates at universities need to acquire the knowledge and skills of new technologies. According to NAEYC, in addition to curricular alignment, early childhood educators should be provided with necessary resources, support, and training in order to realize effective technology integration (Hamilton et al., 2019; NAEYC, 2012).

To foster deeper understanding of technology as a pedagogical tool, educators need opportunities to engage in reflective discourse that thoroughly examine the theories and philosophies that justify its value (Palaiologou, 2016). With this approach, technology integration is more likely to be pedagogically sound as well as socially and culturally appropriate. Hence well-designed professional learning is needed to assist early childhood educators to build technological content knowledge and age appropriate pedagogical practices aligned with a play–based philosophy (Murcia et al., 2018).

The present study, aiming to explore the beliefs of educational robotics among pre-service teachers, suggests that the contemporary Greek pre-service teachers could be considered educational robotics illiterate. The majority of them although recognize the benefits of educational robotics, in their majority they are not prepared to use robots in their future daily teaching practice. In view of our results, a series of recommendations can be made to improve the teaching of educational technology in Early Childhood Education classrooms.

It is necessary that the preservice teachers receive adequate education from the education departments to deal with the scientific content, so that preservice teachers acquire a suitable scientific literacy, vital for them as future teachers. Effective teacher education and support is needed for the use of "developmentally appropriate robotics" and to build an understanding about appropriate pedagogical approaches. Through targeted activities and courses preservice teachers' can build their confidence, pedagogical capabilities, technological knowledge, and develop an understanding of how to incorporate computational thinking (Chalmers, 2018).

## FUNDING

The authors received no financial support for the research, authorship and/or publication of this chapter.

## DISCLOSURE STATEMENT

No potential conflict of interest was reported by the authors.

## ACKNOWLEDGMENT

We would like to express our sincere appreciation of the cooperation of the staff in the schools that participated in this work.

## REFERENCES

Alafodimos, C., Kalogiannakis, M., Papadakis, St., & Papachristos, D. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece, In D. Guralnick (Ed.), *Proceedings of the International Conference on E-Learning in the Workplace (ICELW-09)*. New York: Kaleidoscope Learning.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978–988. doi:10.1016/j.compedu.2011.10.006

Bers, M. U. (2014). Tangible Kindergarten. *The Go-To Guide for Engineering Curricula PreK-5: Choosing and Using the Best Instructional Materials for Your Students, 133*, 145.

Bers, M. U. (2018). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 2094-2102). IEEE. 10.1109/EDUCON.2018.8363498

Bikos, K., Stamovlasis, D., & Tzifopoulos, M. (2018). Dimensions of digital divide and relationships with social factors: A study of Greek pre-service teachers. *Themes in eLearning, 11*(1), 23-34.

Campbell, C., Speldewinde, C., Howitt, C., & MacDonald, A. (2018). STEM practice in the early years. Creative Education Journal, 9(1), 11-25.

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93–100. doi:10.1016/j.ijcci.2018.06.005

Couse, L. J., & Chen, D. W. (2010). A tablet computer for young children? Exploring its viability for early childhood education. *Journal of Research on Technology in Education*, *43*(1), 75–96. doi:10.1080/15391523.2010.10782562

Demetriou, K., & Nikiforidou, Z. (2019). The relational space of educational technology: Early childhood students' views. *Global Studies of Childhood*, *9*(4), 290–305. doi:10.1177/2043610619881458

DePryck, K. (2016). From computational thinking to coding and back. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 27-29). New York: ACM. 10.1145/3012430.3012492

Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2010* (pp. 4006-4014). Chesapeake, VA: AACE.

Ertmer, P. A. (2005). Teacher pedagogical beliefs: The final frontier in our quest for technology integration? *Educational Technology Research and Development*, *53*(4), 25–39. doi:10.1007/BF02504683

European Commission. (2018). *Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions on the Digital Education Action Plan*. Retrieved from https://ec.europa.eu

González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., & Infante-Moro, A. (2019). Computational Thinking and Down Syndrome: An Exploratory Study Using the KIBO Robot. *Informatics (MDPI)*, *6*(2), 25. doi:10.3390/informatics6020025

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2019). An Emerging Technology Report on Computational Toys in Early Childhood. *Technology, Knowledge and Learning*, 1-12.

Instefjord, E. J., & Munthe, E. (2017). Educating digitally competent teachers: A study of integration of professional digital competence in teacher education. *Teaching and Teacher Education*, *67*, 37–45. doi:10.1016/j.tate.2017.05.016

Kalogiannakis, M., & Papadakis, St. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. *Proceeding of the 5th WSEAS/IASME International Conference on Engineering Education*, 305-311.

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). IGI Global. doi:10.4018/978-1-7998-1486-3.ch005

Marsh, J. A. (2016). The digital literacy skills and competences of children of pre-school age. *Media Education: Studi, Ricerche*. *Buone Practice*, *7*(2), 197–214.

Murcia, K., Campbell, C., & Aranda, G. (2018). Trends in Early Childhood Education Practice and Professional Learning with Digital Technologies. *Pedagogika*, *68*(3), 249–264. doi:10.14712/23362189.2018.858

Murcia, K., & Pelliccione, L. (2017). Early childhood iSTEM: Playing as learning with digital technology toys. *Digitising Early Childhood International Conference*.

NAEYC. (2012). *Position statement of the National Association for the Education of Young Children and the Fred Rogers Center for Early Learning and Children's Media at Saint Vincent College*. Retrieved from https:// www.naeyc.org/resources/topics/technology-and-media

Orfanakis, V., & Papadakis, St. (2016). Teaching basic programming concepts to novice programmers in Secondary Education using Twitter, Python, Ardruino and a coffee machine. *Hellenic Conference on Innovating STEM Education (HISTEM)*.

Palaiologou, I. (2016). Children under five and digital technologies: Implications for early years pedagogy. *European Early Childhood Education Research Journal*, *24*(1), 5–24. doi:10.1080/1350293X.2014.929876

Papadakis, S. (2018a). Evaluating pre-service teachers' acceptance of mobile devices with regards to their age and gender: A case study in Greece. *International Journal of Mobile Learning and Organisation*, *12*(4), 336–352. doi:10.1504/IJMLO.2018.095130

Papadakis, S. (2018b). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2018c). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). IGI Global. doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, St., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In M. Kalogiannakis, D. Stavrou & P.G. Michaelidis (Eds.), *Proceedings of the 7th International Conference on Hands-on Science, Bridging the Science and Society Gap*, (pp. 235-240). Academic Press.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed.), *Early Childhood Education* (pp. 1–19). London, UK: IntechOpen Limited. doi:10.5772/intechopen.81714

Papadakis, S., & Kalogiannakis, M. (2020). A Research Synthesis of the Real Value of Self-Proclaimed Mobile Educational Applications for Young Children. In *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). IGI Global. doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. In Information Resources Management Association (IRMA) (Ed.), Early Childhood Development: Concepts, Methodologies, Tools, and Applications (pp. 797-819). IGI-Global. doi:10.4018/978-1-5225-7507-8.ch039

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2018). The effectiveness of computer and tablet assisted intervention in early childhood students' understanding of numbers. An empirical study conducted in Greece. *Education and Information Technologies*, *23*(5), 1849–1871. doi:10.100710639-018-9693-7

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *Int. J. Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Papadakis, S., Zaranis, N., & Kalogiannakis, M. (2019). Parental involvement and attitudes towards young Greek children's mobile usage. *International Journal of Child-Computer Interaction*, *22*, 100144. doi:10.1016/j.ijcci.2019.100144

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2018). How do you feel about learning to code? Investigating the effect of children's attitudes towards coding using eye-tracking. *International Journal of Child-Computer Interaction*, *17*, 50–60. doi:10.1016/j.ijcci.2018.01.004

Sullivan, A., & Bers, M. U. (2019). Investigating the use of robotics to increase girls' interest in engineering during early elementary school. *International Journal of Technology and Design Education*, *29*(5), 1033–1051. doi:10.100710798-018-9483-y

Walsh, C., & Campbell, C. (2018). Introducing Coding as a Literacy on Mobile Devices in the Early Years. In G. Oakley (Ed.), *Mobile Technologies in Children's Language and Literacy: Innovative Pedagogy in Preschool and Primary Education* (pp. 51–66). Emerald Publishing Limited. doi:10.1108/978-1-78714-879-620181004

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Wozney, L., Venkatesh, V., & Abrami, P. (2006). Implementing computer technologies: Teachers' perceptions and practices. *Journal of Technology and Teacher Education*, *14*(1), 173–207.

Yelland, N. (2011). Reconceptualising play and learning in the lives of young children. *Australasian Journal of Early Childhood*, *36*(2), 4–12. doi:10.1177/183693911103600202

## ADDITIONAL READING

Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York, NY: Teachers College Press.

Jimoyiannis, A., & Komis, V. (2007). Examining Teacher's beliefs about ICT in Education. *Teacher Development*, *11*(2), 149–173. doi:10.1080/13664530701414779

Kalogiannakis, M., Nirgianaki, G.-M., & Papadakis, St. (2018). Teaching magnetism to preschool children: The effectiveness of picture story reading. *Early Childhood Education Journal*, *46*(5), 535–546. doi:10.100710643-017-0884-4

Kalogiannakis, M., & Papadakis, S. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In *C. Constantinou, Z. Zacharias & M. Papaevripidou, Proceedings of the 8th International Conference On Computer Based Learning in Science* (pp. 265-276).

Kalogiannakis, M., & Papadakis, St. (2017). Combining mobile technologies in environmental education: A Greek case study. *International Journal of Mobile Learning and Organisation*, *11*(2), 108–130. doi:10.1504/IJMLO.2017.084272

Kalogiannakis, M., & Papadakis, St. (2018). An Evaluation of Greek Educational Android Apps for Pre-Schoolers. In Finlayson, O., McLoughlin, E., Erduran, S., & Childs, P. (Eds.), Electronic Proceedings of the ESERA 2017 Conference. *Research, Practice and Collaboration in Science Education*, Part 4/Strand 4 (co-ed. Kalle Juuti & Eleni A. Kyza, Digital Resources for Science Teaching and Learning), (pp. 593-603). Dublin, Ireland: Dublin City University.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255. doi:10.100710643-012-0554-5

Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital Student Conference Platform Implementation: The case study of the "Research Project" course. *Open Education*, *12*(2), 5–23.

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, *10*(3), 235–250. doi:10.1504/IJTCS.2019.102760

Papadakis, St., Kalogiannakis, M., Sifaki, E., & Vidakis, N. (2018). Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University. *EAI (European Alliance for Innovation) Endorsed Transactions on Creative Technologies, 5*(16), e1, 1-12.

Papadakis, S. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

Psycharis, S. (2015). The Impact of Computational Experiment and Formative Assessment in Inquiry Based Teaching and Learning Approach in STEM Education. *Journal of Science Education and Technology*, *25*(2), 316–326. doi:10.100710956-015-9595-z

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20. doi:10.100710798-015-9304-5

Vidakis, N., Barianos, A. K., Trampas, A. M., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (Vol. 1, pp. 36-43). 10.5220/0007810800360043

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

## KEY TERMS AND DEFINITIONS

**Early Childhood Education:** Is a broad term used to describe any type of educational program that serves children in their preschool years, before they are old enough to enter kindergarten.

**Educational Robotics:** Is an interdisciplinary learning environment based on the use of robots and electronic components to enhance the development of skills and competencies in students.

**Preschool Education:** Is education that focuses on educating children from the ages of infancy until six years old.

**Preservice Teachers:** Students enrolled in an initial educator preparation program.

**Programming:** Lines of code that are written in a certain language that demands a logic of reasoning from the developers.

**Teacher Perceptions:** The thoughts or mental images which teachers have about their professional activities and their students, which are shaped by their background knowledge and life experiences and influence their professional behavior.

# Chapter 14
# Future STEMist Join Forces

**Vasiliki Psaridou**
https://orcid.org/0000-0001-6674-2400
*Minority School of Dokos, Greece*

**Marina Molla**
https://orcid.org/0000-0001-8535-3494
*9th Primary School of Komotini, Greece*

## ABSTRACT

*The aim of this educational school project was to inspire young children around STEMs and for STEM professions to empower them to cope with stereotypes in this field and discrimination in the professional field to work together in favour of integration and diversity (in and outside the school). That is why the effort to build a ramp for disabled people was chosen as a single topic. This project is a collaborative project involving two minority primary schools and a Turkish high school. The topic was covered by a cross-curricular approach to STEM. STEM professionals visited one of the schools, where they presented their profession and received questions from students via teleconference. A workshop of engineering followed. Students used traditional and modern engineering tools to make measurements. The data they collected from their measurements were processed in a mathematics lab, where they designed a ramp for the school.*

## INTRODUCTION

"FUTURE STEMists JOIN FORCES" was a STEM inspired collaborative project between different educators and different cultures, who have worked methodically to achieve it. Three schools participated, two Minority Primary Schools from Greece and Turkey's Ahmet Tanner Kysalı Gymnasium.

It is noted that the Minority Schools of Thrace are schools attended by Muslim students from Thrace and the curriculum of these schools is bilingual. Greek and Turkish languages are taught equivalently. The curriculum follows the articles of the Treaty of Lausanne1 and the Educational Protocols that followed the Treaty. The aim of the project was to inspire young children to STEMs generally and with STEM professions particularly, to empower them to cope with the stereotypes and discrimination that exist in the field and to work for inclusion and diversity.

DOI: 10.4018/978-1-7998-4576-8.ch014

## LITERATURE REVIEW

The acronym STEM (Science Technology Engineering Mathematics) was created in the 90s by the National Science Foundation and is used as a generic title for any action, policy, program, or practice that comprises one or more STEM fields (Bybee, 2010). And while two decades have passed since the introduction of the acronym, its definition remains a source of ambiguity for professionals, especially in the field of education (Sanders, 2009). The definitions range from simple references to the four separate branches of the acronym to part-time or all-encompassing educational methodology to a complete cross-curricular approach to STEM education (Scientix Report, 2018).

The value of the interdisciplinary approach to STEM education is addressed in 29 reports (Rosicka, 2016). Students often fail to see how their subject of studies applies to their everyday lives and how it connects to STEM career options because math and science subjects are disconnected from other subject matter and the real world, (AAAS, 2001). Even though these students rely on science and technology in their everyday lives by using smartphones, computers, televisions, medicines and everyday products, they don't understand the underlying connections to math and science (NRC, 2012).

So while it is necessary to learn skills from individual STEM fields of knowledge, research demonstrates the benefits of the STEM thematic approach, which includes improved problem solving skills, increased motivation and improved results in mathematics and science (Rosicka, 2016; Blackley & Howell, 2015; Becker & Park, 2011; English & King, 2015). Helping students see the connections between math and science and future career opportunities is a critical aim of the STEM pipeline (De Coito, 2014). A thematic approach helps students understand not only what they are learning, but also why and how they can apply what they learn (Rosicka, 2016; Everett, Imbrie & Morgan, 2000; Hanover Research, 2012).

In addition, it is generally accepted that the underrepresentation of STEMs in the education system, combined with a "negative image", is due to the difficulty of engaging curious young minds in the first grades, especially in elementary school. Inadequate preparation for STEM in the first grades ultimately plays a role in secondary education options and ultimately in higher education and vocational choices (De Coito, 2014). Motivating interest in math and science requires improved teaching strategies in the classroom and opportunities within and outside the classroom to demonstrate linkages between math and science, real-world applications, and future careers (Singh, Granville, & Dika, 2002; Tella, 2007).

This has attracted a great deal of interest in the teaching of science in elementary school as teachers, researchers and policy makers have united around the view of the fundamental importance of the role of science teaching in elementary school for later success in education. STEM (Duschl, Schweingruber, & Shouse, 2007).

"Children born in the last fifteen years were born holding a mobile or a tablet" is an expression we hear from many (parents, teachers, experts and non-educators). For this reason the EU considered necessary the introduction of ICT into education initiated by the relevant European Council resolution of 24-9-1983, agreed with the proposals of the 1990s Community institutions on the use of multimedia educational software and teacher training, and dominated by the European leaders' binding decision in Lisbon in the spring of 2000 to shape school knowledge through the use and exploitation of ICT in the educational process.

## ICT IN EDUCATION

Recent research has shown that the way and frequency of ICT use by the Greek educational community depends on many factors. Teacher individual characteristics, such as computer self-efficacy (Paraskeva, Bouta & Papagianni, 2008), attitudes toward technology (Bullock, 2004), and gender of the teacher appear to be related to use of ICT in teaching (Schoretsanitou & Vekiri, 2010). As pointed out by Lavidas, Komis, & Gialamas (2012), many times children surpass adults in their modern technological knowledge even before attending kindergarten. Consequently, according to Prensky (2001), young children can be described as "digital natives" since they are growing up in the digital world.

In July 2014, European Commissioner Androulla Vassiliou wrote a letter to EU Education Ministers urging them to promote children's opportunity to develop basic computer coding skills in school. In this letter, jointly signed by Neelie Kroes, Vice-President for the Digital Agenda, Vassiliou states, "coding is the literacy of today. Each and every interaction with computers is governed by a code. Programming is fundamental to the understanding of a hyper-connected world" (Barshay, 2014).In the same letter, she states, "coding skills are part of the solution to youth unemployment and create a growing skills gap in the ICT sector, in which we expect to see a shortage of 900.000 ICT practitioners by the end of 2020" (European Commission, 2014 as cited in Papadakis et al. 2016). Computer science (CS) develops students' computational and critical thinking skills, thus enabling them, becoming active creators, rather than being passive consumers. This fundamental knowledge is demanded to educate pupils for the 21st century, irrespective of their ultimate field of work or occupation (Code.org, 2015 as cited in Papadakis et al. 2016).

There is a long, rich history of attempts and strategies to ease the process of learning to program and make programming engaging and accessible to a broader population (Kelleher and Pausch, 2007). Some of these strategies include: a) applying pair-programming and collaborative learning b) using themes that are attractive to students as multimedia, game and/or animation approach c) using visualised programming to introduce core concepts before more advanced and in-depth courses (Dekhan et al., 2013).

Research has shown that children who participate in classroom programming, even if they are short-term, show significant improvement in fundamental skills such as literacy and math skills, maximizing their kinesthetic experiences. In addition, they gain rich learning experiences with positive results in developing their social and communication skills (Kazakoff & Bers, 2012; Resnick, 2006).

## ICT AND MINORITY EDUCATION

The modern school, as a crossroads of cultures and a place of intercultural encounter, is called upon to exploit the coexistence of cultures for the benefit of society and the individual. The presence of diverse cultures in a society automatically raises the question of how to educate them in order to effectively integrate them into the incoming society while preserving their particular cultural characteristics. Modern technology can help achieve this. Unfortunately, ICTs are not included in the Primary Minority Schools Curriculum as a separate course.

In essence (Shiakka, 2017), it is, according to Mavrommatis (2004), a two-way language program in which the two languages are not only taught as language courses but are also used to teach individual subjects. As children's academic and social behavior is multifactorial, the teacher must provide empowering learning experiences, recognize the need for culturally appropriate teaching for children from different cultural environments, and use culturally appropriate strategies for students and finally feel sufficient to teach all children effectively, because in this case, the school performance of children from different cultural backgrounds are likely to show significant improvement (Tucker et.al., 2005).

It is therefore only up to the will, ability and determination of each teacher (both Greek and minority) to incorporate them into their daily teaching. Teachers must make the best efforts possible to create as many opportunities as they can to bring children from different backgrounds together (Hughes & Donnelly, 2006).

Teachers work within accepted policy frames and are expected to harmonize their teaching with the objectives and goals promoted in educational policy documents. They are frequently faced with demanding situations when trying to find ways to provide opportunities for children from non-majority linguistic backgrounds to fully realize their potentials. These challenges become even more complex in contexts where different ethnic communities (typically coinciding with different languages) experience a high level of mistrust and social distance, and protracted ethnic or religious conflict (Dryden-Peterson, 2012).

Research in an international context suggests that such divisions in education can have a detrimental impact on the individual level (i.e. students from the minority group not being able to integrate into the wider community, where the language of the majority group prevails), social cohesion (e.g. Gallagher, 2010; Barbiery, Vrgova & Bliznakovski, 2013), and could subsequently replicate themselves in the other domains within broader society (Smith, 2010)

## STEM STEREOTYPES AND DISCRIMINATION

Other Research has shown that women are underrepresented in science, technology, engineering and mathematics (STEM) majors and careers in most industrialized countries around the world (Blickenstaff, 2005; Papadakis, Tousia, & Polychronaki, 2018). Especially for occupations in technology and engineering (Blickenstaff, 2005; European Commission, 2015). In the past decades the proportion of females in these fields remains constant at approximately 25% in the EU. One of the reasons females avoid STEM subjects, lies in the negative and stereotyped perception(s) of these subjects (Engeser et al., 2008; Schuster & Martiny, 2017). The persistence of gendered paths in career choices has recently been reflected in the current Global Gender Gap Report of the World Economic Forum (WEF), which states that on average men are underrepresented in the fields of education, health and welfare whereas women are underrepresented in the STEM fields (WEF, 2017).

Many females in the Western world still believe in the stereotype that professions and subjects in STEM are “male” domains (Nosek et al., 2009). According to Dresel, Schober, and Ziegler they often apply these stereotypes to the assessment of their own abilities in STEM (in Ertl, Luttenberger & Paechter, 2017). Stereotypical classifications of professions and subjects have strong implications for females as they impair learning and prevent females from fulfilling their full potential (Ertl et al., 2017). Stereotypes lower one’s self-assessment and sense of competence, i.e., a person’s self-concept (Marsh & Scalas, 2011). They even have an impact on career choices (Engeser et al., 2008; Schuster and Martiny, 2017).

Additionally, according to Funk and Parker there are wide gaps between men and women working in STEM jobs when it comes to perceptions of fair treatment for women at work and experiences of workplace discrimination. Women in STEM jobs are much more likely than their male colleagues to report discrimination at work because of their gender and to consider discrimination a major reason that more women are not working in STEM (Funk & Parker, 2018).

Furthermore, the results of the PISA studies also point out the necessity to overcome gender gaps and support females' interest in STEM subjects (OECD, 2016). Apart from gender disparities, racial disparities are also prevalent in STEM professions (Young, Young & Paufler, 2017). Projections suggest that the proportion of underrepresented people of color in science and engineering would need to triple to match their proportions in the U.S. population (Schneider, Judy, & Mazuca, 2012).Taking in consideration that the development of the academic self-concept begins in infancy and unfolds its most significant impact(s) after primary school (Senler and Sungur, 2009) it is at most crucial to support and foster young childrens' interest in STEM in school with an inclusive STEM education that promotes diversity in STEM fields.

STEM success for all begins in the classroom (Young, Young & Paufler, 2017). Participation in authentic applications of STEM through projects promotes interest in science and mathematics careers (Rukavina, Zuvic-Butorac, Ledic, Milotic, & Jurdana-Sepic, 2012).

In addition, according to Quinn and Lyons (2011) there is no difference in science engagement between boys and girls and the suggestion that the science gender gap is a reflection of perceptions rather than ability (Knezek, Christensen, & Tyler-Wood, 2011). Hence, it is of great importance that girls should be exposed to experiences that explicitly and tangentially reflect their cultural funds of knowledge (Young, Young & Paufler, 2017). This can be accomplished by soliciting female engineers as guest speakers or mentors to whom girls can relate. Interactions with positive female role models in the scientific community can encourage girls to pursue their interest in STEM at a university level (Austin & Sax, 2006; National Research Council (NRC), 2006).

## METHODOLOGY

"Future STEMists Join Forces" was a short-term collaborative project that lasted three (3) months. Three schools collaborated: two Minority Primary Schools from Greece, the Minority Primary School of Dokos and the 2nd Minority Primary School of Komotini and a Highschool from Turkey, Ahmet Taner Kışlalı Gymnasium. Five (5) teachers and three (3) classes, that consisted of sixty-two (62) 3rd and 4th graders in total.

### Pedagogical Methods

It was implemented with a thematic and interdisciplinary approach to STEM. The theme for the project was that STEM inspired students would join their forces and work for inclusion and diversity in their School. Students identified the lack of a ramp in their schools as a serious problem that excludes students with disabilities2 from School and Education. Students' and Teachers' collaborative work was meaningful, it related to a real-world problem, students applied their STEM knowledge to solve it and contributed to a greater cause, that of making their School accessible3 to all the children.

### Curricular Integration

The actions of the project were designed and integrated collaboratively by the Teachers in the Curriculums of the participating Schools and were aligned with their aims and goals, in specifically: Science, Technology, Engineering, Mathematics, Language, English and Art. In the Minority Primary Schools of Greece, they were integrated in the Curriculum of the Greek and Turkish Language program.

## Procedure

The following actions were selected to integrate all four STEM fields:

- Engineering: STEM Professionals, an Architect and a Civil Engineer visited the schools. An Engineering workshop was organized by STEM professionals.
- Mathematics: A Mathematics workshop for students was organized by teachers.
- Technology: a workshop on introducing students to programming by experimenting with Scratch.
- Science, in the field of Natural Sciences (Medicine): students became aware of the difficulties that people with disabilities face in everyday life.

Teachers visited the local hospital because a new ramp was built there recently. They took some photos of it and talked to some doctors about the benefits this new ramp has to the disabled people. In order to inform the local community about the current situation in schools, students (with the help of teachers) took pictures of the entrance and stairs to the school and after introducing Scratch and experimenting with it, they made a presentation about the lack of a ramp in our schools and the action taken by students and teachers.

ICT was treated in the classroom as a learning tool (Zaranis & Kalogiannakis, 2011a). For the students, ICT is a means for achieving familiarity with new technologies as well as being a tool of investigation, communication and understanding across the full range of the curriculum.

The Lieberman, Bates and So (2009a) report that several studies have shown that digital media can introduce children to abstract concepts that were previously considered too advanced for their age. In her research, Yelland (2005) has shown that activities entailing the use of digital media, within the school environment, facilitate collaborative learning for young children and the development of logical thinking while reinforcing their ability to solve problems. Digital learning activities may encourage children to work together. They have been found to be more effective than traditional learning activities (Zaranis, 2011; Zaranis & Kalogiannakis, 2011a).

Weebly was then used to create the following website https://futurestemists.weebly.com/ for the project, where we published alls activities, while teachers visited a local non-profit organization called “PERPATO”4, which helps people with disabilities and their families to adapt and solve everyday problems. They interviewed a Social Worker and related paper materials were taken, which were presented to the class. A discussion followed that led to an invitation of “PERPATO in our schools.

The Minority Primary School of Dokos was visited by a female architect and a Civil Engineer and presented their profession. The presentation was watched through SKYPE by their project partners. In the end, students from all three schools had the opportunity to ask both Engineers and the woman in particular about the difficulties a woman might encounter in this area.

It was followed by an Engineering workshop by STEM professionals for students in the school yard of Minority School of Dokos, which was repeated at the 2nd Minority Primary School in Komotini. Students used traditional tools such as distance measurement and surface computing to prepare their own classroom ramp, as well as modern engineering tools, like lasers, an engineer’s safety helmet and many others. All the children were very excited, and more questions were asked during the process.

For many years, the central theme for scientific debate was the formulation of a comprehensive mathematical theory towards interpreting phenomena related to the teaching and learning of mathematics. Such a theory aimed to provide educators with the ability to help students understand mathematics as a subject and as a tool for solving everyday problems (Zaranis, Kalogiannakis, Papadakis, 2013).

According to Freudenthal (1983), mathematics is a human activity and therefore it must constitute a human value, must be close to reality of fact, be close to children and be related to society. The central idea of the Realistic Mathematical Education is when saying, "I know math" really means "I know how to do math". The student becomes able to easily handle the mathematical language, to solve and construct problems, but mainly, to recognize mathematical concepts within specific situations. The term "Realistic Mathematics" refers to mathematics, which relate to problems of the real world as well as to phenomena, which appear in our everyday life.

On the other hand, in non-computer assisted teaching methods, children try to reinforce their mathematical skills using recitation techniques or complete, practical sets of exercises. Such traditional methods lack immediate feedback and reduce pupils' interest in learning (Panagiotakopoulos, Sarris, & Koleza, 2013; Papadakis, Kalogiannakis, & Zaranis, 2016).

To avoid the above mentioned, the data collected from the students were used in a mathematics lab in classrooms. Students were divided into groups and with pencils, geometry tools and papers sketched the ramps and learned about triangles, squares, tables and different angles and applied their knowledge of the ramps. They then uploaded the data to the computer. Learning and applying Mathematical and Geometric knowledge made sense.

Compared to the traditional thematic teaching method, results showed that computer-assisted learning may significantly enhance the development of mathematical skills and the cultivation of a deeper perceptual ability for the pupils (Zaranis, 2011; Zaranis & Kalogiannakis, 2011b).

Technology is not a panacea. It is not the hardware or the software, but the combined use of ICT with the pedagogical approach that has the potential to make a significant contribution to young children's mathematical achievement. At this point elementary teachers' beliefs about mathematics must become the focus of extensive research, as it is widely accepted that they have a significant impact on what gets taught, how it gets taught and what gets learnt in classrooms (Xenofontos, 2014). The teacher, being the prevailing source concerning the content of mathematical knowledge, is considered to be the factor that determines the epistemological level of the development of mathematical concepts in the classroom. (Papadakis, Kalogiannakis, Zaranis, 2016).

Finally, the students used their imagination and designed in groups what their school yard would look like after constructing a ramp and prepared a proposal to the Mayors on the need to build a ramp at school. These proposals were afterwards handed to the Mayors. At the end of the school year the students decided to present "Future STEMists join forces" to the whole school and parents.

## ICT PROJECT TOOLS

Digital activities are considered to be particularly effective when they are designed to examine a specific problem or to teach a specific skill. For example, encouraging learning in the thematic areas of the curriculum such as mathematics, natural sciences and language where the specific objectives can be determined and when it is selectively developed within a context relevant to the learning activity and the specific target (Johnson et al., 2011).

Lieberman et al., (2009b) distinguish the quality of digital learning activities in the following categories:

- Well-designed activities—provide powerful interactive experiences which can enhance the learning of young children, fostering skills development, as well as their healthy development;
- Poorly designed activities—simple sedentary activities that contribute little in children's learning, skills development or their healthy development while potentially associated with obesity and poor physical condition;
- Very poorly designed activities—can potentially cause considerable damage to children either through strengthening the aggressive or antisocial behaviour, approbating ethnic or transgender stereotypes and promoting bad eating standards.

More specific benefits are identified in the following sectors:

- Learning—Digital activities can provide considerable educational services to children. Comparative studies have suggested that well-designed educational activities provide potentially more motivation and result in encouraging learning compared to traditional teaching methods (Swing & Anderson, 2008);
- Cognitive skills—Using digital activities, children learn cognitive skills through repetition, as relevant studies have found improvements in operating memory, spatial ability, visual attention, etc. (Thorell, Lindqvist, Bergman, Bohlin, & Klingberg, 2009).
- The most powerful benefits of using ICT are the enhancement of a higher level of thinking and the development of mathematical skills such as classification, counting and number recognition (Lieberman, Bates, & So, 2009a)
- First, we created a website where we put the outcomes of our efforts. For this purpose, we selected the Weebly tool, which enables the teacher to create a web page and import photos, videos, Word files, Powerpoint as well as additional web-2 tools. The environment is easily adapted to the requirements and interests of the teacher. It's easy to use. From the left bar you select the type you want to import with the drag and drop method. The templates are many and varied and can be edited to a better representation of the data.
- The e-books were presented using the Issuu tool, which enables the teacher to create relatively easily and quickly an e-book (data needs to be converted to a pdf file), which can be inserted with Embed Code into web site configuration. It appears as a book that one can scroll through and read. Students were excited about the way the Issue tool works so we used it more than once.
- Programming and computer science began as concepts from a very long time ago, in 1940, the Swiss computer science professor von Neumann laid the foundations of computer science for the first time, which remain in force to this day. According to the Greek Wikipedia, the definition of Programming is "all the processes of writing a computer program to get things done or solve a given problem. Programming also involves checking the program to verify its accuracy and preparing instructions for a computer to perform the tasks specified in the program specifications" (Barbopoulos, 2015). In other words, each program consists of a sequence of instructions given by a computer user to perform a specific task, while the process of writing these instructions is called programming (digitalschool.minedu.gov.gr, 2015).

According to Dufoyer (1988) learning programming can cause seven changes in students' cognitive system:

- Strictness in thought, precision of expression, conscious need to clarify actions.
- Accessing and understanding general concepts such as process, variable, function, transformation (directly related to mathematical education).
- Using heuristics and methodology: planning, searching for similar cases, solving with parts analysis.
- Learning error-seeking techniques, which can be transferred to other non-programming areas.
- Providing the general idea of building the solution in the form of small processes or incremental parts, which can be used interconnected to build the solution to complex problems.
- Expand awareness and knowledge on problem solving techniques.
- Expand and develop the use of comparative methods for the multiplicity of ways to achieve a given objective.

It has been observed that many children involved in programming have acquired future skills by appropriately structuring their personality. This knowledge has a dual benefit for the child. From a psychological point of view, the child is provided with confidence about his / her abilities in using new technologies and software while helping him / her to interact with his / her external environment giving him/her the potential and creativity. In addition, the child learns to interact with the social community, to express his / her ideas, to listen, to accept and to analyze other children's ideas. In addition to the above, the child learns to organize his thinking better. Through his/her involvement with programming, the child learns to follow step by step whatever method he/she chooses to solve a problem while becoming even more observant with the environment around him/her (parentbook.gr, 2015).

In an attempt to increase interest in computer science (CS), there has been much effort in developing tools and activities as preliminary learning materials in schools and universities (Papadakis et al., 2016). There are many ways to teach programming to children. Depending on the age of the child we can offer different possibilities and different ways out. Schoolchildren (of the upper classes of 4th - 6th grade) can learn robotics and do many difficult tasks. Although we initially doubted whether and how we would use a coding tool, we ended up using Scratch.

Scratch is one of the most popular programming environments for novices and young people (used by millions of children worldwide, most commonly in out-of-school contexts) (Benton et al., 2017). Scratch is an interpretive dynamic visual programming language based on Squeak implemented. Being dynamic, it allows code changes even during the execution of programs. It is designed to teach programming concepts to children and adolescents and to enable them to create video and music games. The ability for a student to learn how to use programming is an important part of reading and writing skills in today's society. Scratch programming learning teaches users important strategies to solve problems, design projects, and communicate (Economou, 2010).

There are many advantages of 'blocks' languages, like Scratch. They eliminate the need for typing complete programming statements. Blocks are selected from drawers – menus of related blocks (e.g., math blocks, control blocks) – so it is not necessary for novices to remember or type their names (Turbak et al., 2014a). Additionally, in many blocks' languages, blocks belonging to different collections are often distinguished by different colours so users can easily recognise them. Block shapes guarantee that blocks can be connected only in syntactically meaningful ways (Papadakis et al., 2014; Turbak et al., 2014b). This is the most important dimension for blocks languages, whose primary purpose is to reduce syntactic errors in text-based programming. The blocks also contain other useful visual information that is not necessarily readily available in textual programming languages (Chadha, 2014). With these programming environments, novices can focus on learning the concepts, thinking, and problem-solving skills associated with CS principles rather than being hindered by the frustrations of syntax errors that differ in each language (Papadakis et al., 2014; Chadha, 2014).

Scratch is suitable for children 8 and older. With Scratch, someone develops his ‘fluency skills’ namely his ability to ‘think creatively, reason systematically, and work collaboratively’. Moreover, young people working with Scratch have the opportunity to face significant mathematical and computational issues, like conditions, parallelism, variables and other (Resnick, 2012). Scratch is available free of charge and has been translated into more than 50 languages (Resnick, 2013). In addition, Scratch software is often redistributed by school systems and educational organisations as One Laptop per Child (OLPC) (Maloney et al., 2010). Concepts that are often difficult for novices are easier to understand in Scratch because less is hidden. For example, variables can be made visible, helping the user to understand immediately the effect of operations such as clearing or incrementing that variable (Wolz et al., 2009). Their proposed framework includes three key dimensions:

- Computational concepts,
- Computational practices, and
- Computational perspectives (Papadakis, et al.,2017)

As a visual, block-based programming language and environment for children allows users to create interactive 2D games and animations or interactive stories on a blank canvas (Garcia & De la Rosa, 2016).

We chose not to go deep into learning many commands but only specific and more appropriate for their age. Moreover, the internet gives more opportunities for culturally diverse students to use both the mother tongue and the official language of the host country; students, for example, can improve their reading, writing and spelling or text formatting skills and enrich their vocabulary. Finally, virtual and online environments can be designed to promote problem solving, live interaction and creative collaboration.

Another tool for video presentation, collage, images and short film creation is Kizoa. In Kizoa the teacher can import their own photos and create their own presentation or choose from ready-made templates. There is also the ability (which we used) to import entire video, edit audio and music, and easily integrate it into the under-development webpage.

A similar web-2 tool is the Powtoon. With this tool the teacher can create animated presentations using the tool’s existing slides but can additionally use his own photos that accept edits and text additions. There is built-in music, but it is possible to incorporate mp3 music as well as external recording. It can be downloaded either on youtube or another website.

SKYPE was chosen for the simultaneous communication of the schools. SKYPE is a tool with which schools and teachers can connect and chat at the same time, attend presentations, exchange files. With the help of the camera they can communicate using picture and sound at the same time. Students from different countries and with different languages and cultures can connect and exchange ideas and ideas.

## DISCUSSION

To assess the project the teachers decided to create a rubric which was filled once during the project to give feedback to the teachers about the objectives of the project and at the end of it. Furthermore, a portfolio about the project was also created for each School. In addition, the project was evaluated by the students through their drawings where they expressed their feelings about the project artistically and their initiative to make a final presentation about the project at the end of the school year to the whole school and parents.

The combination of the aforementioned forms of assessment as well as the discussions between the teachers and in the plenary of the classroom during the project and at the end of it led to the conclusion that “Future STEMists Join Forces” was a successful project that fulfilled its objectives. Both teachers and students were affected positively by the project and enjoyed the whole process.

In particular, students were obviously engaged and highly motivated during the whole process. They implemented their knowledge in a real-life problem which had meaning to them. They took initiatives to write a letter to the Mayor to propose the construction of a school ramp in their school in order to include children with disabilities and to present it to their whole school community. Furthermore, it was noticed that the students’ attitudes towards STEM and STEM professions were affected positively.

Finally, the teachers enjoyed the positive impact of the project on their students and of course on themselves. Teachers shared and learnt from each during their collaboration in the development of the project’s actions and the implementation of it. It is noted that their collaboration is continuing in new interdisciplinary STEM projects.

## PROJECT LIMITATIONS AND SUGGESTION

Designing and integrating a STEM inspired project with an interdisciplinary and thematic approach and integrating it in the Curriculum requires teacher training and teacher mentoring in order for the teachers to feel confident to implement such project. It is noted that in this project three of the teacher partners were experienced and had received training as Scientix Ambassadors and Teacher Leader in Coding.

## ACKNOWLEDGMENT

A very important contribution to its creation and implementation was the eTwinning Teacher Training Program, through which we learned and used most of the web-2 tools of the program. This program participated in the European STEM ALLIANCE competition, presented at MOOC’s TeachMeet, “STEM is Everywhere” offered by Scientix at the European Schoolnet Academy, and presented at the 10th Pan-Hellenic ICT Teachers Conference “Use of Information and Communication Technologies in Teaching Practice” in Rhodes, 12-14 April 2019. This is an extended version of the aforementioned presentation.

Particular acknowledgments to Mr. Mehmet Bùlbù[1], Mr. Ali Molla Amet Ali and Ms. Zeynep Kalinci for their collaboration and the implementation of “Future STEMist Join Forces”.

## REFERENCES


American Association for the Advancement of Science (AAAS). (2001). *American Association for the Advancement of Science: Project 2061*. Washington, DC: AAAS.

Austin, H., & Sax, L. (1996). Developing scientific talent in undergraduate women. In C. Davis, A. Ginorio, C. Hollenhead, B. Lazarus, P. Rayman, & ... (Eds.), *The equity equation: Fostering the advancement of women in the sciences, mathematics, and engineering*. San Francisco, CA: Jossey-Bass.

Barbiery, S., Vrgova, R., & Bliznakovski, J. (2013). *Overcoming of the ethnic segregation: how public schools to be integrated in Macedonia and Bosnia and Herzegovina, Why not*. Skopje: Sarajevo and Institute for Democracy Societas Civilis.

Barbopoulos, G. (2015). *Programming tools and syllabuses for preschool and school children*. School of Technology, Department of Informatics Engineering, Antirrio.

Barshay, J. (2014). *Getting Boys – and Girls – Interested in Computer Coding*. http://educationbythenumbers.org/content/girls-computer-coding_1691/31

Becker, K., & Park, K. (2011). Integrative approaches among science, technology, engineering, and mathematics (STEM) subjects on students' learning: A meta-analysis. *Journal of STEM Education*, *12*(5), 23–37.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging Primary Programming and Mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, *3*(2), 115–138. doi:10.100740751-017-0028-x

Blackley, S., & Howell, J. (2015). A STEM narrative: 15 years in the making. *The Australian Journal of Teacher Education*, *40*(7), 102–112. doi:10.14221/ajte.2015v40n7.8

Blickenstaff, J. C. (2005). Women and science careers: Leaky pipeline or gender filter? *Gender and Education*, *17*(4), 369–386. doi:10.1080/09540250500145072

Bullock, D. (2004). Moving from theory to practice: An examination of the factors that preservice teachers encounter as they attempt to gain experience teaching with technology during field placement experiences. *Journal of Technology and Teacher Education*, *12*(2), 211–237.

Bybee, R. (2010). Advancing STEM Education: A 2020 Vision. *Technology and Engineering Teacher*, *70*(1), 30–35.

Chadha, K. (2014). *Improving the Usability of App Inventor through Conversion between Blocks and Text* (Honors thesis). Wellesley College.

Constitution of Greece, Resolution of the Z Revision Chamber of the Greeks, April 6th, 2001

De Coito, I. (2014). Focusing on science, technology, engineering, and mathematics (STEM) in the 21st century. *Ontario Professional Surveyor, 57*(1), 34-36. Retrieved from https://www.krcmar.ca/resource-articles/2014_Winter_Focusing%20on%20STEM_0.pdf

Dekhan, S., Xu, X., & Tsoi, M. Y. (2013). Mobile app development to increase student engagement and problem-solving skills. *Journal of Information Systems Education*, *24*(4), 299.

Dresel, M., Schober, B., & Ziegler, A. (2007). Golem und Pygmalion. Scheitert die Chancengleichheit von Mädchen im mathematisch-naturwissenschaftlich-technischen Bereich am geschlechtsstereotypen Denken der Eltern? In Erwartungen in Himmelblau und Rosarot. Effekte, Determinanten und Konsequenzen von Geschlechterdifferenzen in der Schule. Weinheim: Juventa.

Dryden-Peterson, S. (2016). Refugee education in countries of first asylum: Breaking open the black box of pre-resettlement experiences. *Theory and Research in Education*, *14*(2), 131–148. doi:10.1177/1477878515622703

Dufoyer, J.-P. (1988). *Informatique, éducation et psychologie de l'enfant*. Paris: PUF.

Duschl, R., Schweingruber, H., & Shouse, A. (2007). *Taking Science to School: Learning and Teaching Science in Grades K-8*. Washington, DC: National Academies Press.

Economou, B. (2011). *The Scratch Programming Language*. Retrieved from https://economu.wordpress.com/h-language-programming-scratch /

Engeser, S., Limbert, N., & Kehr, H. (2008). *Abschlussbericht zur Untersuchung Studienwahl Informatik*. Retrieved from: http://www.psy.wi.tum.de/Docs/Studienwahl_Informatik-Abschlussbericht.pdf

English, L., & King, D. (2015). STEM learning through engineering design: Fourth-grade students' investigations in aerospace. *International Journal of STEM Education*, *2*(1), 14. doi:10.118640594-015-0027-7

Ertl, B., Luttenberger, S., & Paechter, M. (2017). The Impact of Gender Stereotypes on the Self-Concept of Female Students in STEM Subjects with an Under-Representation of Females. *Frontiers in Psychology*, *8*, 703. doi:10.3389/fpsyg.2017.00703 PMID:28567022

European Commission. (2015). *She Figures 2015*. Retrieved from: https://ec.europa.eu/research/swafs/pdf/pub_gender_equality/she_figures_2015-final.pdf

Everett, L., Imbrie, P., & Morgan, J. (2000). Integrated curricula: Purpose and design. *Journal of Engineering Education*, *89*(2), 167–175. doi:10.1002/j.2168-9830.2000.tb00511.x

Freudenthal, H. (1983). *Didactical phenomenology of mathematical structures*. Dordrecht: Reidel.

Funk, C., & Parker, K. (2018). Women and Men Often at Odds over Workplace Equit. Pew Research Center.

Gallagher, T. (2010). *Key Issues in Coexistence and Education*. Coexistence International at Brandeis University.

Garcia, P. G. F., & De la Rosa, F. (2016). RoBlock–Web App for Programming Learning. International. *Journal of Emerging Technologies in Learning*, *11*(12), 45–53. doi:10.3991/ijet.v11i12.6004

Hughes, J., & Donnelly, C. (2006). *Contact as a mechanism for promoting better relations in integrated schools in Northern Ireland and bilingual/bi-national schools in Israel*. Paper presented at IACM 18TH Annual Conference.

Johnson, L., Smith, R., Willis, H., Levine, A., & Haywood, K. (2011). *The Horizon Report* (2011 Edition). Austin, TX: The New Media Consortium.

Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergar-ten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, *21*(4), 371–391.

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, *50*(7), 58–64. doi:10.1145/1272516.1272540

Lavidas, K., Komis, V., & Gialamas, V. (2012). Spreadsheets as cognitive tools: A study of the impact of spreadsheets on problem solving of math story problems. *Education and Information Technologies*, *18*(1), 113–129. doi:10.100710639-011-9174-8

Lieberman, D., Bates, C., & So, J. (2009a). Young children's learning with digital media. *Computers in the Schools*, *26*(4), 271–283. doi:10.1080/07380560903360194

Lieberman, D., Fisk, C. M., & Biely, E. (2009b). Digital games for young children ages three to six: From research to design. *Computers in the Schools*, *26*(4), 299–313. doi:10.1080/07380560903360178

Marsh, H. W., & Scalas, L. F. (2011). Self-concept in learning: reciprocal effects model between academic self-concept and academic achievement. In S. Järvela (Ed.), *Social and Emotional Aspects of Learning* (pp. 191–197). Amsterdam: Elsevier.

Mavrommatis, G. (2004). *The children of Calcantza. Education, poverty and social exclusion in a Muslim community of Thrace*. Athens: Metaixmio.

National Research Council. (2006). *To recruit and advance: Women students and faculty in science and engineering*. Washington, DC: National Academies Press.

National Research Council. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. Washington, DC: National Academies Press.

Nosek, B. A., Smyth, F. L., Sriram, N., Lindner, N. M., Devos, T., Ayala, A., ... Greenwald, A. G. (2009). National differences in gender-science stereotypes predict national sex differences in science and math achievement. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(26), 10593–10597. doi:10.1073/pnas.0809921106 PMID:19549876

OECD. (2016). *PISA 2015: PISA Results in Focus*. Paris: OECD Publishing.

Panagiotakopoulos, C. T., Sarris, M. E., & Koleza, E. G. (2013). Playing with numbers: development issues and evaluation results of a computer game for primary school students. In T. Sobh & K. Elleithy (Eds.), *Emerging trends in computing, informatics, systems sciences, and engineering* (Vol. 151, pp. 263–275). New York: Springer. doi:10.1007/978-1-4614-3558-7_22

Papadakis, S. (2018). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*. ACM. 10.1145/2643604.2643613

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Comparing tablets and PCs in teaching mathematics: An attempt to improve mathematics competence in early childhood education. *Preschool & Primary Education*, *4*(2), 241–253. doi:10.12681/ppej.8779

Papadakis, S., Kalogiannakis, M., Zaranis, N., & Orfanakis, V. (2016). Using Scratch and App Inventor for teaching introductory programming in secondary education. A case study. *International Journal of Technology Enhanced Learning*, *8*(3/4), 217–233. doi:10.1504/IJTEL.2016.082317

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Paraskeva, F., Bouta, H., & Papagianni, A. (2008). Individual characteristics and computer self-efficacy in secondary education teachers to integrate technology in educational practice. *Computers & Education*, *50*(3), 1084–1091. doi:10.1016/j.compedu.2006.10.006

Psaridou, V. (2015). *Accessibility Planning Interventions for Visually Impaired Greek Legislation*. Democritus University of Thrace.

Quinn, F., & Lyons, F. (2011). High school students' perceptions of school science and science careers: A critical look at a critical issue. *Science Education International*, *22*(4), 225–238.

Resnick, M. (2006). Computer as paintbrush: Technology, play, and the creative society. Academic Press.

Resnick, M. (2012). Mother's Day, warrior cats, and digital fluency, stories for the scratch online community. Constructionism 2012 Educational Technology Lab, 52–58.

Resnick, M. (2013). Coding for all: interest-driven trajectories to computational fluency. Proposal to the National Science Foundation.

Rosicka, C. (2016). Translating STEM education research into practice. Australian Council for Educational Research.

Rukavina, S., Zuvic-Butorac, M., Ledic, J., Milotic, B., & Jurdana-Sepic, R. (2012). Developing positive attitude towards science and mathematics through motivational classroom experiences. *Science Education International*, *23*(1), 6–19.

Sanders, M. (2009). STEM, STEM Education, STEMmania. *Technology Teacher*, *68*(4), 20–27.

Schneider, B., Judy, J., & Mazuca, C. (2012). Boosting STEM interest in high school. *Phi Delta Kappan*, *94*(1), 62–65. doi:10.1177/003172171209400112

Schoretsanitou, P., & Vekiri, I. (2010). ICT Integration in Education: Factors for Predicting the Educational Crisis. In A. Tzimogiannis (Ed.), *Proceedings of the 7th Panhellenic Conference with International Participation "ICT in Education"*, (*vol. 2*, pp. 617-624). Korinthos.

Schuster, C., & Martiny, S. E. (2017). Not feeling good in STEM: Effects of stereotype activation and anticipated affect on women's career aspirations. *Sex Roles*, *76*(1-2), 40–55. doi:10.100711199-016-0665-3

Scientix Observatory Report. (2018). *STEM – Education policies in Europe*. Retrieved from http://www.scientix.eu/documents/10137/782005/Scientix_Texas-Instruments_STEM-policies-October-2018.pdf/d56db8e4-cef1-4480-a420-1107bae513d5

Senler, B., & Sungur, S. (2009). Parental influences on students' self-concept, task value beliefs, and achievement in science. *The Spanish Journal of Psychology*, *12*(1), 106–117. doi:10.1017/S1138741600001529 PMID:19476224

Shiakka, F. (2017). *Minority education and social exclusion. Teacher perceptions in minority schools in Xanthi*. Thessaloniki: Aristotelio University of Thessaloniki.

Singh, K., Granville, M., & Kida, S. (2002). Mathematics and Science Achievement: Effects of Motivation, Interest, and Academic Engagement. *The Journal of Educational Research*, *95*(6), 323–332. doi:10.1080/00220670209596607

Smith, A. (2010). *The influence of education on conflict and peace building. The influence of education on conflict and peace building, Background paper prepared for the Education for All Global Monitoring Report 2011 The Hidden Crisis: Armed conflict and education*. Paris: UNESCO.

Tella, A. (2007). The Impact of Motivation on Student's Academic and Learning Outcomes in Mathematics among Secondary School Students in Nigeria. *Eurasia Journal of Mathematics, Science, &. Technology (Elmsford, N.Y.)*, *3*(2), 149–156.

Thorell, B., Lindqvist, S., Bergman, S., Bohlin, G., & Klingberg, T. (2009). Training and transfer effects of executive functions in pre-school children. *Developmental Science*, *12*(1), 106–113. doi:10.1111/j.1467-7687.2008.00745.x PMID:19120418

Tucker, C. M., Porter, T., Reinke, W. M., Herman, K. C., Ivery, P., Mack, C. E., & Jackson, E. S. (2005). Promoting Teacher Efficacy for Working with Culturally Diverse Students. *Preventing School Failure*, *50*(1), 29–34. doi:10.3200/PSFL.50.1.29-34

Turbak, F., Sherman, M., Martin, F., Wolber, D., & Pokress, S. C. (2014a). Events-first programming in app inventor. *Consortium for Computing Sciences in Colleges Northeastern Region Conference*, 81–89.

Turbak, F., Wolber, D., & Medlock-Walton, P. (2014b). The design of naming features in app inventor 2. *IEEE Symposium on Visual Languages and Human-Centric Computing (VLHCC 2014)*. 10.1109/VLHCC.2014.6883034

Wolz, U., Leitner, H. H., Malan, D. J., & Maloney, J. (2009). *Starting with scratch in CS1. ACM SIGCSE Bulletin, 41, 223–227.*

World Economic Forum. (2017). *The Global Gender Gap Report. WEF*. Retrieved from: http://hdl.voced.edu.au/10707/349201

Xenofontos, C. (2014). The cultural dimensions of prospective mathematics teachers' beliefs: Insights from Cyprus and England. *Preschool & Primary Education*, *2*, 3–16. doi:10.12681/ppej.85

Yelland, N. (2005). The future is now: A review of the literature on the use of computers in early childhood education (1994-2004). *AACE Journal*, *13*, 201–232.

Young, J. L., Young, J. R., & Paufler, N. A. (2017). Out of School and into STEM: Supporting Girls of Color Through Culturally Relevant Enrichment. *Journal of Interdisciplinary Teacher Leadership*, *1*(2), 2–7.

Zaranis, N. (2011). The influence of ICT on the numeracy achievement of Greek kindergarten children, In A. Moreira, M. J. Loureiro, A. Balula, F. Nogueira, L. Pombo, L. Pedro, & P. Almeida (Eds.), *Proceedings of the 61st International Council for Educational Media and the XIII International Symposium on Computers in Education (ICEM&SIIE'2011) Joint Conference* (pp. 390-399). University of Aveiro.

Zaranis, N., & Kalogiannakis, M. (2011a). The use of ICT in preschool education for science teaching with the Van Hiele theory, In M. F. Costa, B. V. Dorrío, & S. Divjak (Eds.), *Proceedings of the 8th International Conference on Hands-on Science* (pp. 21-27). University of Ljubljana.

Zaranis, N., & Kalogiannakis, M. (2011b). Greek primary students' attitudes towards the use of ICT for teaching natural sciences, In M. F. Costa, B. V. Dorrío, & S. Divjak (Eds.), *Proceedings of the 8th International Conference on Hands-on Science* (pp. 50-55). University of Ljubljana.

Zaranis, N., Kalogiannakis, M., & Papadakis, S. (2013). Using Mobile Devices for Teaching Realistic Mathematics in Kindergarten Education. *Creative Education*, *4*(7A1), 1-10.

## ADDITIONAL READING

Kalogiannakis, M., & Papadakis, (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. *training, 6*, 2.

Kalogiannakis, M., Nirgianaki, G.-M., & Papadakis, St. (2018). Teaching magnetism to preschool children: The effectiveness of picture story reading. *Early Childhood Education Journal*, *46*(5), 535–546. doi:10.100710643-017-0884-4

Kalogiannakis, M., & Papadakis, S. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In *C. Constantinou, Z. Zacharias & M. Papaevripidou, Proceedings of the 8th International Conference On Computer Based Learning in Science* (pp. 265-276).

Kalogiannakis, M., & Papadakis, St. (2017). Combining mobile technologies in environmental education: A Greek case study. *International Journal of Mobile Learning and Organisation*, *11*(2), 108–130. doi:10.1504/IJMLO.2017.084272

Kalogiannakis, M., & Papadakis, St. (2018). An Evaluation of Greek Educational Android Apps for Pre-Schoolers. In Finlayson, O., McLoughlin, E., Erduran, S., & Childs, P. (Eds.), Electronic Proceedings of the ESERA 2017 Conference. *Research, Practice and Collaboration in Science Education*, Part 4/Strand 4 (co-ed. Kalle Juuti & Eleni A. Kyza, Digital Resources for Science Teaching and Learning), (pp. 593-603). Dublin, Ireland: Dublin City University.

Kalogiannakis, M., Vassilakis, K., Alafodimos, C., Papadakis, S., Papachristos, D., & Zafeiri, E. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece. *International Journal of Advanced Corporate Learning*, *2*(4), 15–20. doi:10.3991/ijac.v2i4.981

Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital Student Conference Platform Implementation: The case study of the "Research Project" course. *Open Education*, *12*(2), 5–23.

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In *HSci2010 7th International Conference Hands-on Science Bridging the Science and Society Gap*.

Papadakis, S., & Kalogiannakis, M. (2018). Evaluating a course for teaching advanced programming concepts with scratch to preservice kindergarten teachers: a case study in Greece. In *Early Childhood Education*. IntechOpen.

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, *10*(3), 235–250. doi:10.1504/IJTCS.2019.102760

Papadakis, St., Kalogiannakis, M., Sifaki, E., & Vidakis, N. (2018). Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University. *EAI (European Alliance for Innovation) Endorsed Transactions on Creative Technologies, 5*(16), e1, 1-12.

Papadakis, S. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning*, *10*(1-2), 44–72. doi:10.1504/IJTEL.2018.088333

Vidakis, N., Barianos, A. K., Trampas, A. M., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (Vol. 1, pp. 36-43). 10.5220/0007810800360043

## KEY TERMS AND DEFINITIONS

**Computational Thinking:** Is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could execute.

**Early Childhood Education:** Is a broad term used to describe any type of educational program that serves children in their preschool years, before they are old enough to enter kindergarten.

**Educational Robotics:** Is an interdisciplinary learning environment based on the use of robots and electronic components to enhance the development of skills and competencies in students.

**Preschool Education:** Is education that focuses on educating children from the ages of infancy until six years old.

**Preservice Teachers:** Students enrolled in an initial educator preparation program.

**Programming:** Lines of code that are written in a certain language that demands a logic of reasoning from the developers.

## ENDNOTES

1 After the Asia Minor Catastrophe and according to the Lausanne Treaty that was signed on the 24th of July of 1923 the Greek Orthodox population of Constantinople, Imbros and Tenedos and the Muslim population of Western Thrace were considered as non-exchangeable populations with the hope to become a uniting bridge of peace and friendship in the future between Greece and the Republic of Turkey.

2 People with Disabilities according to the World Health Organization's International Classification are people with motor disabilities, people with visual and hearing impairments, people with perception and speech problems (e.g. paraplegic, quadriplegic, people with cerebral palsy, blind people, deaf, mentally retarded, etc.). They are estimated to make up about 10% of the country's population. Article 21 (6) of the Constitution of Greece (2001 revision) states that "persons with disabilities have the right to enjoy measures ensuring their autonomy, professional integration and participation in the social and political life of the country". Consequently, any act or situation that is against the autonomy and the possibility of people with disabilities to participate in social activities violates a constitutionally guaranteed right and establishes the right to appeal to Greek justice. It goes without saying that the lack of access to infrastructure, services and goods is an unconstitutional act and creates a right of action.

3 "Accessibility" means the characteristic of the environment, which allows all persons - without discrimination on the basis of sex, age and other characteristics (body shape, strength, perception, nationality, etc.) - to have access to it, and can independently, safely and easily access and use the infrastructure, as well as the services (conventional and electronic) and the goods available in the environment. The term 'accessibility' therefore refers not only to infrastructure but also to services and goods. At the same time, in addition to physical access, it also refers to functionality, but also to the ability to communicate and inform, and determines the degree of autonomy and security of the individual in relation to the environment (physical, structured and / or electronic). We are referring to infrastructures, services, equipment, goods accessible to the individual (Psaridou, 2015). Prerequisite: accessibility to all sectors (e.g. environment and urban planning, transport, health and welfare, research and technological development, education, work, safety and hygiene etc.) and at

all levels (local, regional, central, European, international), (Disability Trade Union Training, 2013). Accessibility is the key to resolving problems, because equal opportunities are provided for people with some form of disability, for equal treatment and for dignity in living for all. “Accessibility ultimately ensures a common level of reference for all, eliminating the disadvantages of creating a disadvantaged social structure capable of serving only part of its members. An accessible environment removes barriers, allows everyone to work on an equal footing and redefines the rights and obligations of citizens with disabilities.” (Disability Trade Union Training, 2013).

4 The Association of People with Disabilities and friends of Rodopi Prefecture, “Perpato” was founded in 2002 from a group of friends of People with Disabilities (PWD), their families and other friends. Today “Perpato” includes around 300 members with disabilities and 150 active volunteers. Since 2005 it has been training students of the Department of Social Administration (Faculty of Social Work-Democritus University of Thrace). In 2003 “Perpato” Association was incorporated into the National Registry of Private non-profit Operators Sector that provide social care services. It is also incorporated in the Special Registry of the Volunteer Non-Profit Organizations of the Greek Ministry of Health and Social Services (as it was called at that time). In addition, the Greek National Centre of Social Solidarity in 2011 accredited Perpato for being an official provider of social care services in the private non-profit sector of the Greek Ministry of Health and Social Solidarity. Moreover, in August 2011 Perpato was accredited with management competence type B’ and C’ according to the ELOT 1420 model for the following applied field: planning and implementation of co-financed projects (public supply and service contracts), implementation of specific activities that use the same means as above.

Section 4

# Implementation of Digital Technologies

# Chapter 15
# The Role of Digital Fabrication in Today's Society

**Tandra Lea Tyler-Wood**
*University of North Texas, USA*

## ABSTRACT

*Digital fabrication and the "maker movement" can play a major role in bringing computational technology into the 21st century classroom. Digital fabrication is defined as the process of translating a digital design developed on a computer into a physical object or any process for producing/printing a three-dimensional (3D) object. The maker movement is a platform for today's futuristic artisans, craftsmen, designers and developers to create, craft, and develop leading ideas and products. Digital fabrication and "making" could provide a new platform for bringing powerful ideas and meaningful tools to students. Digital fabrication has the potential to be "the ultimate construction kit." Digital fabrication has strong ties to the maker movement. Maker spaces provide students with safe areas that allow students to safely use digital fabrication to make, build, and share their creations. This chapter will look at the role that digital fabrication can play in incorporating computational technology into the K-12 classroom.*

## INTRODUCTION

Digital fabrication and "making" will provide a new platform for bringing powerful ideas and meaningful tools into today's classrooms (Blikstein, 2013). Digital fabrication has the potential to be "the ultimate construction kit" (Briones, 2019). With strong ties to the maker movement, digital fabrication perpetuates innovation. Maker spaces support innovation and provide students with safe areas to use digital fabrication to make, build, and share creations" (Blikstein, 2013, p. 6).

DOI: 10.4018/978-1-7998-4576-8.ch015

## BACKGROUND

Computational technology "involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. This chapter will look at the role that digital fabrication can play in incorporating computational technology into the k-12 classroom. Although digital fabrication has been available for the last 15 years (Cutcher-Gershenfeld, Gershenfeld, & Gershenfeld, 2018) until the last few years the cost of the technology made classroom use probative. Currently, many schools have the technology, however, integrating digital fabrication into the k- 12 classroom has presented a challenge. Innovative opportunities for incorporating digital fabrication into the classroom have been explored by various educators (Bull & Garofalo, 2009; Stansell, 2016). Educators have trouble making the new technology "fit" into what they are required to teach. There appears to be a strong relationship between engineering curriculum and digital fabrication however few countries actually employ and implement engineering curriculum in the k-12 setting. Because of a lack on integrated curriculum, digital fabricators are often used for creative opportunities for exploration with little connection to required classroom curriculum. Cutcher-Gershenfeld, Gershenfeld, and Gershenfeld (2018) believe that digital fabrication has the potential to change our day-to-day environment and even redefine the concept of work. Digital fabrication builds on two earlier digital revolutions, digital computation and communication. These two factors are key components of computational technology and hold the basis for connecting computational technology into digital fabrication. Digital fabrication holds the potential to be an innovative technology, just as innovative as the Internet was a decade ago. To make the most of future workplace opportunities, clearly students need the opportunity to learn and work with digital fabrication in a purposeful way.

## MAIN FOCUS OF THE CHAPTER

With the potential to provide infinite creative experiences for k-12 students, digital fabrication involves the conversion of a digital design into a physical object through a computer-controlled fabrication system. Personal digital fabrication makes designing and producing objects feasible. The opportunities for creative projects are boundless. At initial introduction, most students are enthralled, making, phone holders, whistles, name tags, bracelets, and other objects. Projects spring off the digital fabricator as students begin to view themselves as "creators" and "makers." Students can design and create objects with Computer Aided Design (CAD) software or even download an existing model from the massive number of Internet shape libraries that are emerging to house models. At one time, digital fabrication required industrial plants for computer-aided design and manufacturing (CAD/CAM). But today, desktop fabrication systems make these technologies available to schools and the general public. For example, the Cornell College of Engineering developed a 3D fabrication system, Fab@Home available for home users. Digital fabrication and 'making' could be a new and major chapter in a process of providing powerful ideas, literacies, and expressive tools for learners. Today, the range of accepted disciplinary knowledge associated with digital fabrication has expanded to include not only programming, but also engineering, design (Astrachan, Hambrusch, Peckham, & Settle, 2009; Yasar & Landau, 2003), mathematics (Bull & Garfola, 2009, Stansel), and language arts (Tillman, Kjellstrom, Smith & Yoder, 2011). There is a need to determine the impact that digital fabrication may play on: students' learning in STEM, students' attitudes towards STEM and students' interest in a STEM career. It is yet to be determined the role that digital fabrication will play in 21st century STEM job markets as well as the general job market. Digital fabrication is so new that we don't have a "blueprint" for predicting how this technology will change our workplaces.

The coming digital fabrication revolution offers opportunities for integrating STEM content into K- 12 schools in an appealing way. Classroom engineering engages students at many levels. It involves abstract concepts and visualization, but it also involves problem-solving in the physical world. Students can create tools in response to authentic challenges. Educators can use this capability to encourage scientific thinking and the development of authentic skills including engineering skills.

Affordable digital fabrication kits are just beginning to emerge making personal fabrication kits available for schools, less than $2,000 for a basic fabricator. Placing digital fabrication into the schools has become a relatively easy task. However, the existing curriculum in schools is already filled to capacity as a result of forces associated with an increasing emphasis on meeting minimal standards and high-stakes testing. Consequently, there is little room for the addition of any new content. Realistically, the only way to incorporate educational innovations is to address existing curricular standards through their use. Fortunately, many potential links to existing math, science, and technology standards are embedded in digital fabrication. Further, connecting science, technology, engineering, and mathematics (STEM) topics through digital fabrication increases the likelihood that these subject areas will be learned in a meaningful context.

Constructivism is the theory that students learn by individually or socially transforming information (Slavin, 1997). In line with constructivist epistemology, meaningful 21st century student learning occurs when teaching and content connect to relevant experiences. By keeping teaching and content as critical components of the learning process, educators can explore the critical integration of technologies within their curricula that will prepare students for the challenges of the 21st century economy and related career fields (Borko, Whitcomb, & Liston, 2008; Harris, Mishra, & Koehler, 2009).

Digital fabrication technologies can play a major role in preparing today's students for tomorrow's jobs. In this chapter, we will:

- Define digital fabrication
- Provide examples of the uses of digital fabrication
- Link digital fabrication to existing educational standards
- Establish the strong links between computational technology and digital fabrication
- Provide examples of the effective use of digital fabrication
- Establish links between digital fabrication and tomorrow's job market
- Examine the future for digital fabrication.

## DIGITAL FABRICATION DEFINED

Digital fabrication is an additive technology. In its simplest form, a computer sends a signal to a printer that is capable of printing in layers. The printer lays down layer upon layer of a malleable substance that hardens into an object. Using this technology can accomplish remarkable printing feats such as printing multi-dimensional objects. Users can accomplish innovative tasks such as printing text inside an orb. Printers can have more than one nozzle. Each nozzle can project a different substance making multi-material and intricate designs possible.

The process of turning data files into "printed objects" is becoming more and more sophisticated. Children in schools can print out pencil holders. Industry can print "parts" as needed decreasing "production downtime." Digital fabrication has the potential to revolutionize our everyday life.

The advent of digital fabrication poses a societal dilemma. How will we live, play, learn, and work when we have the capability of printing-off a vast array of objects? Clearly, digital fabrication can provide convenience, the opportunity for unlimited expression as well as the power to accomplish both good and evil tasks. Soon, it may be possible to call your computer on your way home from work and have your dinner printed for your arrival at home. However, a quick internet search will yield plans for printing-off a gun. As the prices of three-dimensional fabricators (3D printers) decreases, fabrication opportunities become more available for the masses, the regulation of this new, imposing technology will prove a formidable task.

Gershenfeld defines digital fabrication as "*an evolving suite of capabilities that turn data into things and things into data. Many years of research remain to complete this vision, but the revolution is already well under way. The collective challenge is to answer the central question it poses: How will we live, learn, work, and play when anyone can make almost anything, anywhere?*"

## COMPUTER-AIDED DESIGN (CAD)

Computer-aided design (CAD) software refers to a type of software program used by makers, designers and engineers to create two-dimensional and three-dimensional models. Once created these CAD models can be uploaded onto a computer to make a physical representation of the design. CAD has all but replaced the pencil and paper design, t-squares and protractors used by past generations of designers. The previous process called manual drafting has been updated to CAD.

Two-dimensional (2D) CAD models provide the layouts, dimensions, and general information used in product development and design. Many industries to include aerospace, architecture, automotive, civil engineering, interior design, landscaping and even fashion use CAD programs to develop 2D designs.

Three-dimensional (3D) CAD models provide greater detail when compared to 2D programs. Instead of simply displaying length and width, a third dimension is added to the model resulting in a product that can depict an actual 3D object. 3D CAD programs aren't typically used in tasks such as designing a floor plan. However, industries such as the aerospace and automotive industries rely heavily on 3D models.

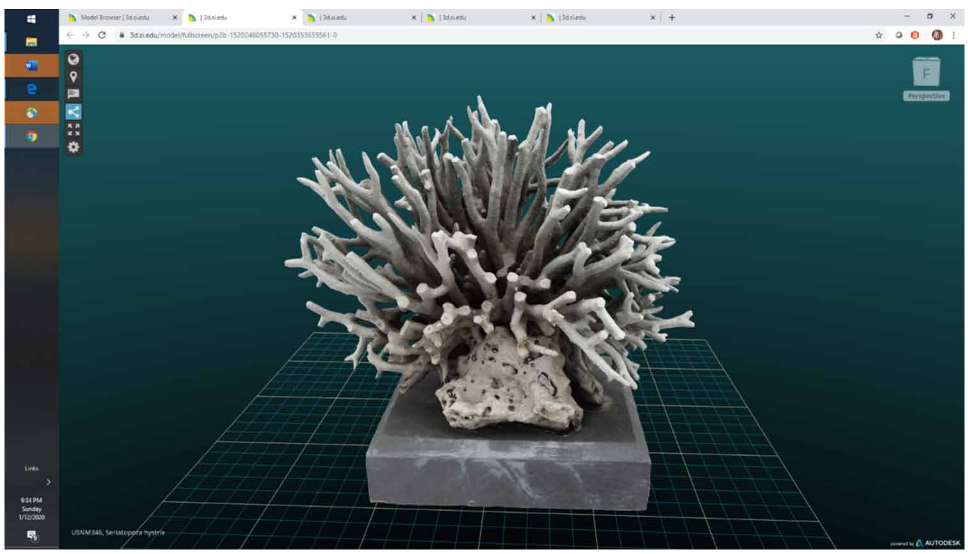
CAD software can be used to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing. CAD products often take the form of electronic files for printing, machining, or other manufacturing operations.

## SCANNING

The ability to scan an object provides new opportunities for digital fabrication. It is possible to capture existing objects into the computer so they can be modified, enhanced, or simply re-created using digital fabrication. This process allows individuals to duplicate art or other objects of interests that might be difficult to design particularly for an amateur or novice to digital fabrication. Libraries of objects that are available for download are provided by places such as the Smithsonian Institute. This "object housing service" is particularly useful in the case of artwork or other unique formations that could not otherwise be available for demonstrations or classroom use.

The picture below depicts a 3D scan of a coral skeleton available on the Smithsonian website (Smithsonian, retrieved 1/20/2020) (see Image 1). When alive, this colony was covered in the coral animal's soft tissue, made up of individual polyps, which look like tiny sea anemones. These polyps secrete aragonite, a type of calcium carbonate (CaCO3) assembled from ions in the ocean. In essence, corals convert seawater into stone, thereby building their skeletons as well as a valuable reef habitat for other sea life. Coral reefs, although they cover less than one percent of the global ocean, provide food and homes for about a quarter of all marine species on earth. These coral specimens were collected by the scientist Dr. James Dwight Dana on the United States Exploring Expedition of 1838-1842.

*Figure 1. A 3D scan of a coral skeleton available on the Smithsonian website*




Advanced CAD programs have reconstructed ancient artifacts such as mummified remains in Egypt. In addition, CT scans and a stereolithographic system, researchers at the University of Dundee have to printed King Richard III's skull into solid form, re-creating what this monarch looked like in life (Smithsonian, retrieved 1/20/2020).

Early 3D capturing systems relied on a probe that contacted the printed object at many different locations, defining a "point cloud" around the object's shape to define its basic geometry. The point cloud is then filled in with greater detail as the scanner measures finer points between the original markers. These systems are still used in machinery analysis and other durable environments. More recent scanners use illumination from lasers or structured light — projections that measure the distance from the camera to different parts of an object, so there is no risk of harm to the object under investigation from the contact points of the scanner.

Coupled with software on a computer, this structured-light scanner can build a 3D model from repeated measurements of an object's surface structure as the scanner is simply waved above an object of interest.

## MAKERSPACES

Makerspaces are places that are designed to inspire creativity. A makerspace is not simply a space for constructing. Makerspaces provide a space to think, be creative, and learn a skill. Makerspaces provide an opportunity for creators to meet and share ideas. Such spaces encourage creativity and the use of higher-level thinking skills. Virtual making, such as coding and using online software tools, may be part of a maker program, too. A makerspace is a place of inspiration. Makerspaces engage individuals and encourage creativity and through the creative process empower individuals to develop new and innovative projects. Makerspaces embody the spirit of tinkering, creating, adapting failing, and learning from failure to move the creative process forward. Makerspaces encourages the participant not to just make an object, but to experiment and to see failure as a necessary and critical part of the creative process. Makerspaces allow users to experience failure as a necessary step towards success. Makerspaces encourage persistence while providing users with the physical materials and mental support to solve real life problems. These life skills will hopefully extend beyond the maker environment and provide valuable lifelong tools for success. Helping developers learn to persist in the face of failure is crucial to their ability to succeed in the real world and to solve real-world problems. Overall, makerspaces encourage creativity, inventiveness, sharing, learner choices, persistence, real-world problem solving, exploration, and project-based learning.

## MAKER MOVEMENT

Makerspaces encourage students to understand how things work, to experiment, invent and redesign things through multiple iterations, to democratize and understand processes of engineering, science, and innovation. The Maker Movement with its roots in informal or project-based learning (PBL) is now supported in general society as well as many schools. O'Brian (2020) indicates that project based learning is an instructional methodology that encourages students to learn and apply knowledge and skills through an engaging experience. PBL presents opportunities for deeper learning in-context and for the development of important skills tied to college and career readiness. Successful implementation of Maker Spaces in schools requires acquisition of the technology, technology training on using items such as the 3D printer and laser cutter, and curriculum modifications that allow for utilization of the new technology.

The Maker Movement refers broadly to a community of inventors, artists, craftsmen, engineers, and tinkerers who experiment with new and used materials to create physical objects (Martin, 2015). Humans have always been "makers" who create, build, and invent. The current

"maker movement" adds a layer of "new technologies" such as laves, 2D cutters and 3D printers. The maker movement has been defined as a "new industrial revolution" (Anderson, 2012), and is distinguished from previous making activities by three elements: digital resources, online sharing of designs, and common design standards that enable sharing and replication (Halverson and Sheridan, 2014). While the Maker Movement originated and has mostly been based among adults in informal settings, increasing numbers of K-12 schools are now experimenting with Maker Spaces designed to support science, technology, engineering, and math (STEM) experiences (Martin, 2015).

Fontichiaro (2016), describes many variations in what constitutes a makerspace:

*A professional makerspace likely feels and looks like the merger between a computer lab, a wood shop, and an auto repair bay. In a classroom, it could be the addition of science kits to the curriculum, plus a few bookshelves of tinkering kits and materials to keep kids engaged during down time. In a public library, it might be the addition of new programming related to STEM and/or a set of circulating science or art equipment. In a school library, it might range from centers designed for mental relaxation to cardboard boxes awaiting transformation to digital fabrication tools like a 3-D printer (p. 49).*

However, the uniting concept is that makers are given the opportunity to create things in an informal and unstructured environment. Often, makerspaces are found in libraries, but these spaces can be implemented in classrooms, in open areas and commons. One impact of the maker movement is the breaking of barriers between disciplines through the utilization of various tools and techniques and applying these tools and techniques for fun, creativity, and the arts (Smith et al. 2014). A makerspace is not simply a room with supplies and materials. Makerspaces encourage users to be producers of new technology, products and visions.

Currently, in most schools, the curriculum of each subject area is full, leaving little room for introduction of additional content. In an era of accountability, teachers and schools are evaluated on the basis of scores on high stakes tests. A sustainable approach to innovation on a wide scale in schools must advance existing educational standards and objectives rather than replace them. Reworking the curriculum to reflect new technologies is a challenge. A curriculum package that incorporates teacher training, existing standards, and new technology would facilitate the acceptance of the Maker Movement within the schools.

## TEACHING SCIENCE AND ENGINEERING THROUGH 3D-PRINTED HISTORICAL RECONSTRUCTIONS

Clearly, advanced manufacturing technologies are transforming the design process. The Society of Manufacturing Engineering identified 3D technology as a key Innovation that Could Change Engineering. In Virginia, a report, Developing an Advanced Manufacturing Workforce for Virginia, USA (2013), found that thousands of jobs related to advanced manufacturing will emerge in the region. That report identified a critical gap in skilled workers to fill these needs as well as a need to develop these workforce skills.

In 2014, The Smithsonian Institute collaborated with the Laboratory School for Advanced Manufacturing (Lab School) at the University of Virginia to digitize key inventions. Within its collections, The Smithsonian Institute holds original inventions such as the telephone, telegraph, and early electric motors. These digitized renditions of critical, noteworthy, inventions enable students to reproduce digitized models provided through the Smithsonian web site (http://3d.si.edu). Using the models, students have the opportunity to reconstruct working replicas of the inventions using advanced manufacturing technologies such as 3D printing. The curriculum developed through this project enabled the use of 3D printing in middle school physical science classes. This integration of coursework provided an early experience with engineering design skills. The science principles underlying the inventions were leveraged when the inventions were integrated into labs in science class to teach foundational knowledge.

An example of an invention kit developed through the project is the Solenoid Invention Kit curriculum (https://www.maketolearn.org/). The curriculum contains five lab activities, two make activities, and one invent activity. These lessons are scaffolded to lead the learners from a baseline toward full mastery of the content and processes involved. Each lab activity is guided by essential questions and teaches key concepts and skills that are utilized in the make activities. The culmination of the unit is the invent activity, where previous learning is utilized in a new and creative way. Through this project, students reconstructed a historical invention (solenoid) with pedagogical support involving four phases to promote student learning:

- Eliciting students' knowledge of phenomena related to classic inventions about related science and engineering concepts underlying the inventions;
- Introduction of new knowledge that may support, expand, or even conflict with students' ideas through the use of pivotal cases grounded in historic inventions;
- Comparison of students' science and engineering concepts with observed phenomena, to determine how students' existing ideas may conflict with, or extend the new ideas related to newly introduced phenomena;
- Reflecting and refining students' science and engineering knowledge to address gaps and discrepancies between anticipated results and actual outcomes, thus, allowing students to successfully reconstruct and potentially modify the historical invention. Smithsonian (http://3d.si.edu)

A pedagogical method known as "reverse engineering," was utilized. This process involves deconstruction of an existing mechanism to understand underlying ideas.

The teaching process associated with reverse engineering resulted in students developing more refined understandings of science and engineering concepts. The goal for the students was not to create an exact physical replica, but to reinterpret and reinvent the device using technology available in the makerspace. Students who participated in the project demonstrated not only a greater understanding of engineering concepts but also demonstrated a greater affinity towards a STEM career.

## TRANSMEDIA BOOKS: CURRICULUM OPTIONS FOR 3D PRINTING

Stansell (2016) used transmedia books to provide 3D learning activities in the middle school classroom. Transmedia books use a storyline to introduce issues that the reader can solve using technology applications. This project based learning approach allows students to utilize technology in real-life scenarios. Stansell's storyline requires students' to develop a plan to bring water to a farm in crisis. Among other activities, students can use CAD and 3D printing to design and manufacture a cistern for holding water. Other potential products include designing and manufacturing a desalination apparatus to convert nearby saltwater into palatable water. Project based learning supports student engagement in meaningful problem-solving of real life situations to develop skills for success that help students transfer knowledge and express that knowledge (Tseng, Chang, Lou, & Chen, 2011). CAD programing and 3D printing can provide tools to support project-based learning. Each unit in the transmedia book has a unique project to problem-solve in the field of agricultural engineering. The final unit incorporates a cooperative learning experience where students have an opportunity to display their knowledge of CAD programming and 3D printing. Students are able to physically make a product based on the meaningful application of knowledge that they have constructed from their peers and various resources through questioning, understanding, and finally developing solutions to a problem.

Project based learning supports student engagement in meaningful problem-solving of real life situations to develop skills for success that will help them transfer knowledge and express that knowledge (Tseng, Chang, Lou, & Chen, 2011). Each unit in the intervention book has a unique project to problem-solve in the field of agricultural engineering. The final unit was written to be completed with an optional capstone project of developing a solution through software that will go from digital to being fabricated with a 3D printer.

Using the transmedia book, students learn through the research and development of creating STEM based project solutions. The 3D printer is a very realistic tool supporting project-based learning. Students can physically make a solution-based product specific to the meaningful application of knowledge acquired from their own research and interaction with peers as well as the utilization of innovative resources such as 3D printing.

## MEDICAL APPLICATIONS OF 3D PRINTING

Major gains have been made in the medical applications of 3D printing since the first prosthetics were introduced in 2000 (Ventola, 2014). Supported by open source repositories such as NIH's 3D Print Exchange (3dprint.nih.gov), the health care industry is changing quickly as customized, cost effective opportunities support tissue and organ fabrication, prosthetic production and pharmacological advances, and assistive technology development. Providing an opportunity for school aged children to experience such technologies can provide a pathway for much needed 21st century jobs.

## PROSTHETICS AND TISSUE PRODUCTION

Dental implants were among the first medical applications of 3D printing (Ventola, 2014), but production of prosthetics has rapidly expanded to artificial limbs, knee and hip implants, and even skulls. Materials vary by prosthetic function and have ranged from titanium and other metals to plastics, silicon, and live cells (Karagöz, 2014). Artificial noses and ears, bones and cartilage, heart valves, and skin grafts have all supplemented the organ donor shortage through reproduction in 3D printers (Ventola, 2014).

## ASSISTIVE TECHNOLOGY INVENTIONS

Maker fair competitions such as MakerBot's Assistive Technology Challenge (Scott, 2015) continually encourage the development of new and vibrant assistive technology options. Recent designs include controls for wheelchairs, ergonomic asthma inhalers, tactile graphic maps and image-to-heat conversion for the visually impaired (Buehler, 2015), magnetic shoe closures, a mouth-operated mouse, a printable wheelchair, and wheel chair ramps (Scott, 2016). As innovative 3D medical designs continue to be created, access to personalized medicine is becoming a reality for the future.

## E-NABLE

Frankie Flood is a strong member and supporter of the e-NABLE Community. E-NABLE "unites over 4000+ individuals and thinkers in over 30 countries who are putting aside their religious and political differences, views on gender roles and more to work together on designs for free 3D printed assistive devices for children and adults with upper limb differences" (Flood, 2016). e-NABLE is comprised of "out of the box thinkers" who are forming groups within their universities, trade schools and community colleges. As educators actively engage their students with the e-NABLE project, they not only support the development of problem solving and communication skills, but also provide innovative opportunities for student growth. The ability of young people to receive services as well as interact within the project opens the door for future collaborative experiences.

## FAB-LABS

To keep up with what people are learning in the labs, the fab lab network has launched the Fab Academy (Gershenfeld, 2008). Children working in remote fab labs have progressed so far beyond any local educational opportunities that they would have to travel far away to an advanced institution to continue their studies. To prevent such brain drains, the Fab Academy has linked local labs together into a global campus. Along with access to tools, students who go to these labs are surrounded by peers to learn from and have local mentors to guide them. They participate in interactive global video lectures and share projects and instructional materials online.

## ART AND 3D PRINTING

Artists are known for embracing innovation. Throughout history, we've seen art adapt to changes in technology. Clearly, there is a place for 3D printing within the arts. With a 3D printer, an artist has a new medium for inspiration and production. 3D technology has been introduced to almost every branch of the art world, and the end products are astounding. Examples of innovative uses of 3D printing in the arts follows.

The most obvious use of 3D printing technology occurs in the visual arts. 3D printed structures and sculptures can be found virtually anywhere. 3D printing gives artists the freedom to create complex structures that would otherwise be almost impossible, or extremely time-consuming to make. 3D printing puts the power of creation into the hands of the artists. With tools such as CAD design and scanning, artists, including student artists are empowered to create. Because artists don't need specialized skills to physically make the 3D pieces, new innovative types of form and dimension can be achieved.

Musician have found ways to embrace 3D printing. Through 3D printing, musicians can see their instruments and tools built more elaborately and quickly than previously available. The ready availability of innovative instruments can validate and inspire school-aged students.

Dance is also using 3D printing. For dancers on pointe, shoes are critical pieces of equipment that allow for the expression of the art form. Dancers often must wear pointe shoes for hours on end each day. Ill-fitting shoes can cause bleeding and bruising. One artist recognized this problem and found a way to improve the world of ballet as we know it. Neeman (2019) had dancers scan their feet to create a perfectly fitted shoe made of printed elastomeric polymer. This material is three times stronger than current shoes on the market. This type of innovation could provide better "equipment" for individual dancers and enhance their ability to optimize their art.

## SOLUTIONS AND RECOMMENDATION

We stand at beginning of the Digital Fabrication, Maker Movement. The technology, opportunities for collaboration and possibilities for new innovation can definitely be a part of our future. It is important to provide these "tools" to all future makers. Collaborative efforts and shared libraries as espoused by futurist such as Flood are critical to making the most of 3D technology. Worldwide efforts to use 3D technology should be shared and lauded. We all have so much to gain.

## CONCLUSION

Digital fabrication definitely has the potential to change the future. With the advent of this new technology, where will we be 10 years from now? Where will this new technology be even one year from now? Digital fabrication can and probably will make a dramatic change in our everyday life. In this chapter we've looked at the role digital fabrication can play in industry, medicine, education and the arts. Who's to say what field this new technology will impact next? It seems clear that those that have the opportunity to engage with digital fabrication and supportive programs such as CAD and scanning will have an advantage. These individuals have a "jump-start" on creating future innovations. Not all individuals, countries or cultures will have equal access to this wonderful technology. Clearly, it becomes important to make sure that everyone has an opportunity to learn and invent. It is critical for society to use digital fabrication for the good of all. That's what we should be concerned about as we think about the future. We mentioned there is potential for misuse of 3D technology but when we look at some of the things that have already been created such as devices that assist individuals with disabilities, truly we can see that digital fabrication has the potential to make the world a better place. The responsible use of this tool is critical as we move forward.

## ACKNOWLEDGMENT

Funding for this project was provided by the National Science Foundation (NSF1510289).

## REFERENCES

Anderson, C. (2012). *Makers: The new Industrial Revolution*. New York: Crown.

Astrachan, O., Hambrusch, S., Peckham, J., & Settle, A. (2009). The present and future of computational thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education* (SIGCSE '09). ACM. https://doi.acm.org/10.1145/1508865.1509053

Berry, R. Q., Bull, G., Browning, C., Thomas, C. D., Starkweather, G., & Aylor, J. (2010). Use of digital fabrication to incorporate engineering design principles in elementary mathematics education. *Contemporary Issues in Technology & Teacher Education*, *10*(2), 167–172.

Blikstein, P. (2013). Digital fabrication and 'making' in education: The democratization of invention. In FabLabs: Of Machines, Makers and Inventors, 4, 1-21.

Borko, H., Whitcomb, J., & Liston, D. (2008). An Education President for the 21st Century. *Journal of Teacher Education*, *59*(3), 207–211. doi:10.1177/0022487108317803

Briones-Garcia, S. J. G. (2019). *Diverse Learners in the Classroom: Students with Special Needs enrolled in Science, Technology, Engineering, and Mathematics (STEM) Texas Public Classrooms* (Dissertation). University of North Texas.

Bull, G., & Garofalo, J. (2009). Personal fabrication systems: From bits to atoms. *Learning and Leading with Technology*, *36*(7), 10–12.

CurriculumS. (n.d.). https://www.maketolearn.org/inventions-kits/motor/activity-1-2/

Cutcher-Gershenfeld, J., Gershenfeld, A., & Gershenfeld, N. (2018). *Digital fabrication and the future of work, High Tech Futures*. Retrieved from http://cba.mit.edu/docs/papers/19.01.POW.pdf

Ferrini-Mundy, J., Khargonekar, P., & Kurose, J. K. (2016). *Dear colleague letter*. Washington, DC: U.S. National Science Foundation. Retrieved from https://www.nsf.gov/pubs/2015/nsf15086.jsp

Fontichiaro, K. (2016). Help! My principal says I need to start a makerspace in my elementary library. *Teacher Librarian*, *44*(1), 49–51.

Gershenfeld, N. (2008). *Fab: the coming revolution on your desktop--from personal computers to personal fabrication*. Basic Books.

Halverson, E. R., & Sheridan, K. (2014). The maker movement in education. *Harvard Educational Review*, *84*(4), 495–504. doi:10.17763/haer.84.4.34j1g68140382063

Harris, J., Mishra, P., & Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education*, *41*(4), 393–416. doi:10.1080/15391523.2009.10782536

Karagöz, Z. (2016). *The robotel Türkiye project*. Retrieved February 27, 2017, from http://enablingthefuture.org/2016/12/05/enablingturkey/

Lood, F. (2017). *Assistive technology*. Retrieved 28, 2017), from https://www.frankieflood.com/studio.html

Martin, L. (2015). The promise of the maker movement for education. *Journal of Pre-College Engineering Education Research, 5*(1), 4.

Neeman, H. (2019). Retrieved January 20, 2020, from https://www.dezeen.com/2018/08/10/p-rouette-3d-printed-ballet-shoe-hadar-neeman-graduate-design/

O'Brian, M. (2020). Retrieved March 3, 2020, from https://www.definedstem.com/blog/what-is-project-based-learning/

Rainwater, B. (2016). *How the maker movement is revitalizing industry in American cities*. Retrieved from https://www.fastcompany.com/3057349/how-the-maker-movement-is-revitalizing-industry-in-american-cities

Scopigno, R., Cignoni, P., Pietroni, N., Callieri, M., & Dellepiane, M. (2017). Digital fabrication techniques for cultural heritage: A survey. *Computer Graphics Forum*, *36*(1), 6–21. doi:10.1111/cgf.12781

Scott, C. (2015). *MakerBot announces winners of 3D printed Assistive Technology Challenge*. Retrieved February 27, 2017, from https://3dprint.com/111643/makerbot-assist-tech-winners/

Scott, C. (2016). *A look at some small 3D printed devices that make a huge difference in the lives of the disabled*. Retrieved February 27, 2017, from https://3dprint.com/139584/3d-printed-assistive-technology/

Slavin, R. E. (1997). *Educational psychology: Theory and practice* (5th ed.). Boston: Allyn & Bacon.

Smith, S., Tillman, D., Mishra, P., Slykhuis, D., Alexander, C., Henriksen, D., & Goodman, A. (2014). Building multidisciplinary connections: Intersections of content, creativity, and digital fabrication technologies. In M. Searson & M. Ochoa (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014* (pp. 2506-2510). Chesapeake, VA: Association for the Advancement of Computing in Education (AACE).

Smithsonian. (n.d.). https://www.smithsonianmag.com/smithsonian-institution/ten-smithsonian-artifacts-you-can-3-d-print-180973649/

Stansell, A. (2016). *Transmedia STEM Intervention Book in Middle School for Educational Change* (Dissertation). University of North Texas Libraries, UNT Digital Library. https://digital.library.unt.edu

Tillman, D., Kjellstrom, W., Smith, S., & Yoder, E. (2011). Digital fabrication scaffolds for developing preservice elementary teachers' mathematics pedagogy. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference* (892-897). Chesapeake, VA: AACE.

Tseng, K.-H., Chang, C.-C., Lou, S.-J., & Chen, W.-P. (2011). Attitudes towards science, technology, engineering and mathematics (STEM) in a project-based learning (PjBL) environment. *International Journal of Technology and Design Education*, *23*(1), 87–102. doi:10.100710798-011-9160-x

Tyler-Wood, T., Cockerham, D., & Johnson, K. R. (2018). Implementing New Technology in a Middle School Curriculum: A Rural Perspective. *Smart Learning Environments*, *5*(22), 22. doi:10.118640561-018-0073-y

Ventola, C. L. (2014). Medical applications for 3D printing current and projected uses. *P&T*, *39*(20), 704–711. PMID:25336867

Yasar, O., & Landau, R. H. (2003). Elements of computational science and engineering education. *SIAM Review*, *45*(4), 787–805. doi:10.1137/S0036144502408075

## ADDITIONAL READING

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Orfanakis, V., Papadakis, St., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital Student Conference Platform Implementation: The case study of the "Research Project" course. *Open Education*, *12*(2), 5–23.

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In *HSci2010 7th International Conference Hands-on Science Bridging the Science and Society Gap*.

Papadakis, S., & Kalogiannakis, M. (2018). Evaluating a course for teaching advanced programming concepts with scratch to preservice kindergarten teachers: a case study in Greece. In *Early Childhood Education*. IntechOpen.

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, *10*(3), 235–250. doi:10.1504/IJTCS.2019.102760

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Vidakis, N., Barianos, A. K., Trampas, A. M., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (Vol. 1, pp. 36-43). 10.5220/0007810800360043

## KEY TERMS AND DEFINITIONS

**Computational Tools:** Are the implemented techniques in computers to solve problems by either stepwise, repeated, and iterative solution methods; also known as in-silico methods.

**Computer-Aided Design:** Is the use of computers to aid in the creation, modification, analysis, or optimization of a design.

**Creativity:** Is the act of turning new and imaginative ideas into reality. It is characterised by the ability to perceive the world in new ways, to find hidden patterns, to make connections between unrelated phenomena, and to generate solutions.

**Digital Fabrication:** Is a design and manufacturing workflow where digital data directly drives manufacturing equipment to form various part geometries.

**Project-Based Learning (PBL):** Is a teaching method in which students learn by actively engaging in real-world and personally meaningful projects.

# Chapter 16
# The CCAP Project:
## Using 3D Technologies to Support Teaching Scenarios of History

**Panagiotis Angelopoulos**
*Ministry of Education, Greece*

**Efthalia Solomou**
*Ministry of Education, Greece*

**Alexandros Balatsoukas**
*University of Groningen, The Netherlands*

## ABSTRACT

*The "CCAP" project is an effort to teach in an interdisciplinary way both the teaching subjects of History (the trip of Columbus to discover America) and Informatics (3D modelling and printing). Students of B grade from the Junior High School of Vrilissia (age 13), on a voluntarily basis, separated into groups of 4-6, have created in a 3D design environment instruments used by Columbus during its trip to America, astrolabes, compasses, caravels, etc., as were taught during the subject of History and according to the description of the instruments given by the school book and other resources. The instruments were eventually printed out using the 3D printer in the computer lab. Part of the program was supported through the school's curriculum hours, and part of the program had to be implemented out of school hours. After the completion of the project, students responded to a questionnaire prepared by the teachers in a Google form format. The most important results of this questionnaire are discussed in this work.*

## INTRODUCTION

Mader & Dertien (2016), notice that Tinkering as a method in students' teaching and learning is characterized by a playful, experimental, iterative style of engagement, in which makers are continually reassessing their goals, exploring new paths, and imagining new possibilities. They also argue that not only engineering and design can profit from tinkering, but also science. In this Chapter we argue that

DOI: 10.4018/978-1-7998-4576-8.ch016

Tinkering can also supports not only pure sciences but also social sciences, classical studies and the majority of cognitive objects taught in the classroom.

Artifacts serve as a way to interpret a story. People define themselves by their creations and actions and students can examine works to define a story, a period, an important event in history. Simultaneously, teachers use these experiences to get the students talking and thinking about the topic. Following Columbus and his trip to discover America, students can freely express themselves and produce their own creations and proceed to the construction of maritime objects used at that period, through the utilization of digital 3D technology.

Also, Papadakis & Kalogiannakis (2017) notice that "there are concerns among researchers and education professionals that students in our classrooms are bored, unmotivated and disengaged from school. One of the reasons is that old teaching methods are no longer beneficial to the students. On the other hand, game-based learning can improve learning motivation of students. Compared with traditional lectures, digital game-based approaches can indeed produce better learning effects, which underscore the need to develop appropriate instructional materials.

The terms Computational Thinking, educational artifact and making in learning was already known through the work of Seymour Papert (Papert, 1980) and his constructionism theory that supports that the experience and process of building something physical or digital provides the context for developing understanding and learning. According to Psycharis (2017) CT is the ability to: a) develop computational abstractions of real-world problems; and b) design, develop, refine and reason about computation artefacts. The Computational Thinking is now recognized as a concept that encompasses the pervasiveness of Computer Science constructs and problem-solving strategies such as abstraction at different hierarchical levels, algorithmic thinking, automation, decomposition, modeling, patterns, recursion, scale, and symbolic representations.

Recent works have provided frameworks and examples for incorporating computational thinking across different subject areas and others relate Computational Thinking to the larger context of learners informally engage in as makers and creators (Voogt et al.,2015). In this work, by using 3D technologies in formal and informal school activities, there is an effort to study how CT skills can be developed in students in disciplines other than programming and Computer Science and how Tinkering, making of Artifacts can contribute in teaching and learning. We have used an interdisciplinary approach and focus on the learning results and students' opinion about the followed teaching approach described below.

Although, there are some recent studies providing ideas for investigating the effectiveness of educational CAD tools and curricular scaffolds designed specifically for K-12 students for supporting integrated STEM learning anchored in the design process (Dasgupta, Magana, Vieira, 2019; Ng & Chan, 2019) the literature in connecting the use of 3D Technologies and development of Computational Thinking skills is poor at the moment. On the other hand, 3D technologies can be a supporting field concerning the development of STEM education and the deriving skills like problem solving, use of patterns, use of abstraction and algorithmic thinking. From this point of view, the wide use of 3D technologies in education can play an important role in supporting integrated STEM teaching and in developing the grow of acute demand for skills that meet economic challenges (Kelley & Knowles, 2016). Many education systems around the world are engaged in promoting STEM skills and pursuing educational reforms. Recent reforms in USA (such as Next Generation Science Standards) advocate for purposefully integrating STEM by providing deeper connections among the STEM domains locating and teaching intersections for STEM integration as STEM subjects often are taught disconnected from the arts, creativity, and design (Hoachlander & Yanofsky, 2011), while STEM subjects are completely disconnected from non-STEM subjects. Even if there is a rich literature background for integrated STEM teaching, interdisciplinary teaching approaches between STEM and non-STEM subjects (History, Languages, Social and Political Sciences, etc) do not exist. Although, same benefits and literature suggestions from integrated STEM teaching can be found and applied in all integrated teaching efforts including STEM and non-STEM subjects.

Sanders (2009) described integrated STEM education as "approaches that explore teaching and learning between/among any two or more of the STEM subject areas, and/or between a STEM subject and one or more other school subjects". Sanders suggests that outcomes for learning at least one of the other STEM subjects should be purposely designed in a course—such as a math or science learning outcome in a technology or engineering class (Sanders, 2009). Moore et al. (2014) defined integrated STEM education as "an effort to combine some or all of the four disciplines of science, technology, engineering, and mathematics into one class, unit, or lesson that is based on connections between the subjects and real-world problems"

In the context of the statistical analysis and the evaluation of the program, we have to mention that the purpose of this research is to realize the impact of the 3d technologies to the students. The participants of this research were students second class students (14 years old). The sample of 40 students is composed by 20 males and 20 females, who after completing the 3d design and printing answered a questionnaire is pretty small. It is very important to mention that the questionnaire uses mainly Linkert scale (Derrick & White, 2017), which is a reliable and convenient way to measure the opinions of the participants (students). The questionnaire includes a series of questions providing 5 balanced responses from which the students could answer. Therefore, the collected data are ordinal-scale and the distribution of the sample is unknown, so only non-parametric methods can be used, given the fact that using a distribution-free test lets us assume that the data that we analyze (analyzed population) does not need to follow a certain distribution or follow specific assumptions, or parameters, such as the normal distribution, which is necessary to make use of parametric tests such as t-tests or ANOVA.

The non-parametric statistics, (Corder & Foreman, 2014) are used generally on ordinal data, which are variables that have ordered categories and the distances between the categories cannot be counted or are unknown, such as "Strongly Agree", "Agree" "Neutral" "Disagree" "Strongly Disagree", or scale 1 to 5 with 1 being "Strongly Agree" and 5 "Strongly Disagree", or nominal data, such as "male" and "female". Even though the non-parametric statistics need to meet less assumptions than parametric statistics, they are less powerful than them, which practically means that they could not show an existent relationship between two variables or categories, while in fact one exists. The Spearman's Rho is a non-parametric test that we use generally to calculate whether an association between two variables is strong or not. If the $r=1$, then this means that we have a perfect positive correlation. In this case if the one variable is increased, so does the second. If the value is $r=-1$, then we have a perfect negative correlation, where when one variable is increased, the other is decreased. The Spearman Rho test is used in our case to explore whether there is an association between our continuous and discrete ordinal variables. By confirming statistically that there is a positive or negative correlation between the rank values of two of our variables, we prove that we can proceed to the realization of the other two tests, such as Mann-Whittney U test and Wilcoxon signed-rank test.

The non-parametric tests that will be used will be the Mann-Whitney U test, which could be considered an alternative of the independent t-test, and the Wilcoxon signed-rank test, which is also an alternative for the paired-samples t-test. The Mann-Whitney U test is a non-parametric test which is used in case that the researcher compares two independent samples and the dependent variable is of ordinal measurement level, without assuming a normal distribution. For example, if we want to compare the difference of the scaled opinions (non-countable) between two independent samples –male-female(gender), we would use the Mann-Whittney U test. The Mann-Whitney test is constructed upon some assumptions. Generally, the observations of both groups must be independent of each other. We should also assume that the measurements of the sample are at least of ordinal level, the distributions of populations under the H0 (null hypothesis) are equal and the distributions of the Ha (alternative hypothesis) are not equal. In order to observe whether there are statistically significant results amongst the opinions of 40 students, using as a criterion the gender –male-female, we will use the Mann-Whittney U test.

The Wilcoxon signed-rank test is also used in this study. This test is used when there are two conditions or dependent samples (f.e. impression of people pre and post one change). At least one ordinal measurement level is required so that the Wilcoxon signed-rank test can take place. For example, if we want to compare the impression of the same group of workers before and after an educative seminar, we use the Wilcoxon signed-rank test. According to the assumptions of the realization of these tests, the data should come from the same population and be paired, which means that the researchers will test the same sample twice. In addition, the dependent variable is of ordinal level of measurement. The observations should also be independent, which means that the subject should on the one hand not be related one to another and that the pairs of scores should not be repeated but come from different people. Lastly, this test is based on the medians of the samples, which means that the distribution of the difference in scores is symmetrical. In order to understand how the impression of the students regarding the class of History and Informatics has improved, we needed to compare the mean of their impression. To compare a non-countable measure, such as impression the non-parametric Wilcoxon signed-rank test was used.

## THEORETICAL FRAMEWORK

The "CCAP" project tackles objectives relevant to secondary education (students of B Grade Junior High School, ages 13-14 years old), looking to increase the interest of the pupils in STEM disciplines and History and contribute to tackling the issue of underachievement in ICT and History. The program aimed to expose pupils and their teachers to integrated approaches leading to the development of key competences necessary to shape the future towards 2030; integrate Computational Thinking (Bocconi et al, 2016) and STEM thinking actions and methodologies in Classical disciplines to augment interest in STEM education and vice-versa, focus on competencies with an emphasis on learning through science by linking science with other subjects and disciplines (European Union, 2015), strengthen teachers and pupils to understand and assume the role of each discipline in the development of 21st century skills and the added value of their integrated contribution and to improve students and teachers' competences by supporting them to become innovative and work collaboratively.

The aim of the project was to support teaching between STEM and Classical subjects, co-create and implement integrated models inspired by STEM and classical disciplines and to investigate the behavior of students through a school program based in both formal and non-formal educational approaches.

Same approach can be easily applied from the first grades of primary education to the last grades of Secondary Education and include cognitive subjects such as History, Languages, Arts, Social subjects and STEM subjects, depending on the students' skills to handle a computer and applications of modeling and three-dimensional design. The overall objective of "CCAP" is to increase students' proficiency in the key competences necessary to shape the future towards 2030 (Nemorin & Selwyn, 2016) by integrating

STEM disciplines, social sciences, humanities, arts, and entrepreneurial disciplines (Classical disciplines) as broader educational goals are acknowledged and supported by modern educational systems

According to the OECD's PISA 2015 (OECD, 2015), which tested over than a half of million 15-year-old students in 72 countries and economies in science, reading, mathematics and collaborative problem solving, only 12 out of the 72 countries and economies assessed have improved their science performance over this period. For this reason, there is a need for innovative approaches, (Nistor et al, 2019) to increasing the motivation of pupils towards both STEM and classical subjects and for offering teacher training on new ways of introducing innovation and new teaching approaches (EU, 2019). Students interested in classical disciplines do not use, computational thinking, practice and methodologies based on the understanding of STEM subjects. Also, students with interest in STEM subjects are less interested in classical studies which lessens their chances of developing comprehensive knowledge and literacy on issues such as history, philosophy, languages, religions, etc. The "CCAP" project is an effort to teach in an interdisciplinary way (Berninger, 2015). both the teaching subjects of History (America's discovery by C. Columbus) and Informatics (3d modelling and printing), that took place in the 1st Junior High School of Vrilissia, during the school year 2017-2018, trying to include a number of characteristics that describe Computational Thinking, such as logically ordering and analyzing data and ideas, creating solutions using a series of ordered logical steps, and attitudes, such as the ability to confidently deal with complexity and open-ended problems and to work in groups and collaborate. CCAP stands for Caravel, Compass, Astrolabe, Portolan "tools" used by C. Columbus to his trip to discover America.

Youth, even if they considered to be very good at their subject of studies do not seem to have an overall perception of issues not related directly to their studies. Connecting STEM and Classical subjects (Informatics and History in our case) is vital in ensuring that the general knowledge sector will benefit of much needed new talent in its various fields, and that students think and act not only like scientists-experts in their field, but also as integrated personalities, weighing evidence to draw conclusions, and learning how to navigate the claims bombarding us in our everyday lives.

The "CCAP" program proposes to:

- Facilitate learning and cultivate attitudes and approaches related to problem solving process that includes a number of characteristics present in Computational Thinking, such as logically ordering and analyzing data and creating solutions using a series of ordered steps (or algorithms), and dispositions, such as the ability to confidently deal with complexity and open-ended problems. Students became researchers and producers by discovering information for the navigational instruments of the Columbus era and then by designing and printing their own creations.
- Support interdisciplinary teaching (James, 2019) between STEM and Classical subjects' teachers in the school. The project aims to create a community of practice for STEM and Classical teachers inside the school, on how to teach their subjects supporting each other and provide input and methodologies to their subjects' activities and teaching from STEM to Classical subjects and vice versa.
- Co-create and implement integrated models inspired by STEM and classical disciplines, increasing relevance while supporting the development of Computational Thinking related skills like solving problems, designing systems, and understanding human behavior (Wing, 2006) and of 21st skills in students and school progress.

- CT needs to be integrated in the curriculum in a way that does not actually widen the gap between those students who are already interested in technology and those who are not (Balanskat, 2018).

Its aim is also to investigate the behavior of students through a school program based in both formal and non-formal educational approaches. Part of the program is supported through the school's curriculum part of the program has to be implemented out of school hours. Particularly, its objectives are to monitor

- the acquisition of knowledge and skills related to both involved subjects.
- the acquisition of Computational Thinking skills (Decomposition, Pattern recognition, Pattern generalization and abstraction, develop algorithmic thinking etc).
- students' participation and performance in both subjects.
- the need and enthusiasm of students to attend an interdisciplinary program based on creativity and co-operation.

## METHOD

### Implementation

Cognitive objects: History, Informatics.
Addressing class: Second Grade of Junior High School (age 14-15)
Chapter: 7th, Histrory Student HandBook.
Duration: 2 teaching two hours in classroom
2 teaching hours in the Computer Laboratory
4 preparatory teaching hours in the Computer Laboratory
Access to the Computer Lab for one day every week for two months after the school time.
Number of students: 40 volunteers among the 103 students of B Grade
Space/Room that the classes take place: Computer Laboratory and Classroom.
Printer Used: Ultimaker 2+ that had been granted for two years to the 1st Junior High School of Vrilissia by GFOSS (www.gfoss.eu)

3D applications used the Tinkercad web 3d modelling tool (https://www.tinkercad.com) and the 3d printing software cura (https://ultimaker.com/software/ultimaker-cura

This teaching scenario effectively uses 3D modeling and printing techniques in order to support the level of learning and comprehension of C. Columbus trip to discover America.

Following a series of new for the epoch technical instruments: the compass, the astrolabe, which was an instrument of latitude determination based on the observation of stars, and the portolan, which were nautical chart and, the caravel a new type of ship, combined capacity, speed and safety, students can freely express themselves and produce their own creations, and proceed to the construction of these technical instruments through the use of digital 3D technology.

The general idea of this scenario, that is considered part of the cognitive context of multilingualism and multiculturalism, is to give a chance to the students to:

- Enjoy through one of the most famous trips ever
- Design the instruments by experimenting with 3D tools, concepts and shapes of Euclidean geometry and stereometry,
- Print 3D models by experimenting with the properties, attributes and parameters of a 3D printing tool (cura)
- Acquire Computational Thinking skills (Decomposition, Pattern recognition, Pattern generalization and abstraction, Algorithmic thinking)
- Acquire skills in 3D modelling and printing, following historical descriptions of these instruments.
- Acquire soft skills as Teamwork, Communication Skills, Problem Solving Skills, Flexibility/Adaptability, Interpersonal Skills etc
- Students become creators of their own personal instruments and ships

## Cross-Linking: History and Informatics

As to the cognitive object of History, realization of the lesson involves elements from the textbook and other resources centered approach and the recruitment method and of the comprehension of the learning objectives by using the critical analysis of discourse to analyze the text.

Learning activities: It is expected from students to:

- Understand the European Civilization during C. Columbus era– physical and spiritual.
- Debate analogies and differentiations from the contemporary epoch between the previous, current and future civilizations.

The multicultural - cooperative approach is connected with the roles that students are going to take during the two hours of guidance in class and concern:

- The participation of all the students, especially of these that are considered “frails/weak” or these that have been born in another country.
- Discover the different ways that other civilizations, nations and races construct and decorate their navigation instruments.

As to the cognitive object of INFORMATICS, students had to capture and model the navigation instruments and the ships, in 3D display (TeachThought Staff, 2015). The free offered in the web application of Tinkercad (www.tinkercad.com). was used to design their models and to transfer them in a 3D printable format. Consequently, they transfer the models and experiment in the 3d printing software “cura”, looking forward to create the ultimate printing models of the navigation instruments and ships.

Students are expected to be able to:

- Search for free 3d sketch/design software.
- Understand basic functions and tools of 3D sketch/design.
- Experiment and understand the behavior of stereometric and euclidean shapes when used in 3D modelling.
- Research advanced functions and tools of 3d modelling/design. Moreover, they can try to practice them in the design of the instruments and ships.

- Learn about the materials and the applications of 3D printing.
- Create their own shapes when necessary
- Get familiarized to the program, the properties and the parameters of the printing software.
- Produce open educational resources (cc licensing their models)

*Figure 1. Astrolab model in solid view*

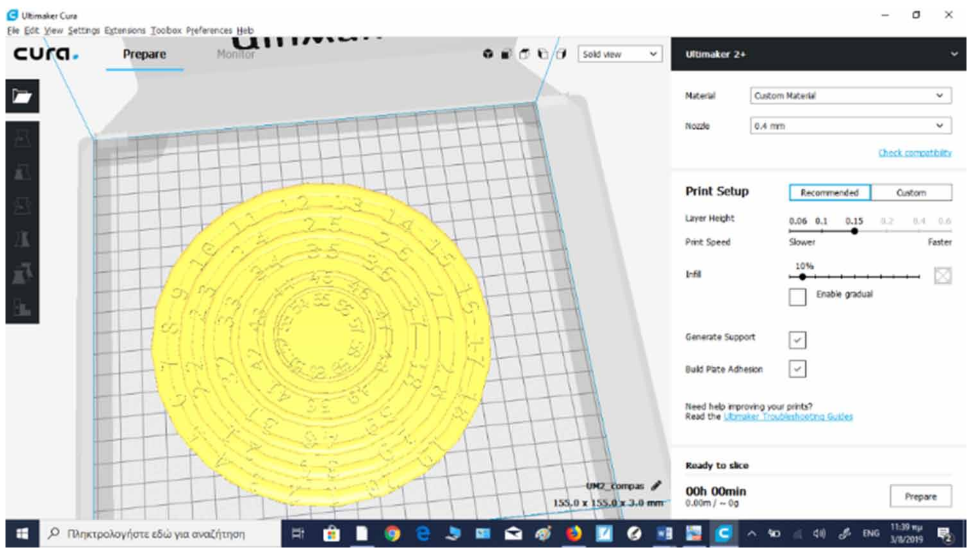
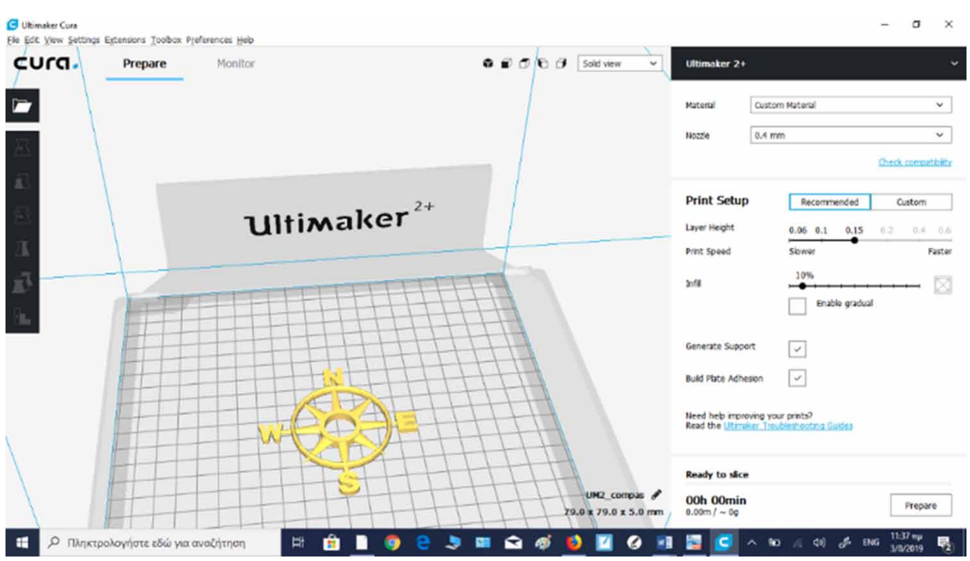


*Figure 2. Compass model in solid view*



## Means of Teaching-Materials-Conditions

School handbook, web resources, application tutorials, office suite, Internet browser, headphones/speakers, Interactive White Board system, 3D design software, 3D printing software, printing material PLA and/or ABS, 3D printer were the main means of teaching and materials used to implement the program.

The teaching of the equivalent chapter from the school handbook and other historical resources and the information of the students for the way of realization of the class objectives is pre-required.

First of all, the students had to read the hand-book chapter and Look for other resources, concerning the trip of C. Columbus to discover America. Secondly, with specific questions (about the ships used to cross the Atlantic, the navigation instruments, the maps used) in mind about the meaning of each one, they will search for keywords in that text. After, they will identify and separate all the necessary instruments and components that made this journey feasible. Then, they will return to the text to locate specific information and passages related to each component.

After identifying all the instruments that are referred to the chapter, they will write down all the information that can be found in the text, under the guidance of their teacher. Then they will proceed to the next step by designing and printing all the components (compasses, astrolabes, caravels, stars and the coast, sea and the waves etc).

With the assistance of the teacher, students in teams should be able to

- Navigate to the internet in order to, locate information relevant with the navigation instruments, the maps and the ships used in that period, locate information and tutorials about special modelling techniques needed for designing the above instruments, transfer the data to the 3D printer's software and setting the proper parameters for the final printing and finally printing the instruments and the ships.
- The students in the context of the Informatics class have already been familiarized (four teaching hours) with the usage of 3d design and 3d typing applications. Although, after they study the description of the instruments and ships two more hours are devoted for the construction of them for research-organization-modification of the information that have been collected already from the History lesson. They are going to start designing the instruments and the ships based on the knowledge acquired by the History subject and their own web search.
- Finalizing the models, export them to an .stl file, configuring the parameters of the 3D printing software, export them to an .gcode file and start the printing.
- The design and printing process continue after the end of the teaching hour for the time needed.

## RESTRICTIONS

The experiment of using an interdisciplinary approach to teach a certain Chapter of History (the trip of Columbus to discover America) together with the utilization of 3D printing technologies was according to students' opinion successful, interesting, fun and fruitful. The study was designed to determine the effect of the addition of 3d design and printing to the effectiveness and performance of the students during the class. The second aim of this study was to investigate the effects of this innovative technology on the impression that the students have on the classes that it was based and their cooperation. The most obvious finding to emerge from this study is that the impression of the students, in both cases, in

Figure 3. Parts of a Caravel model in Solid view

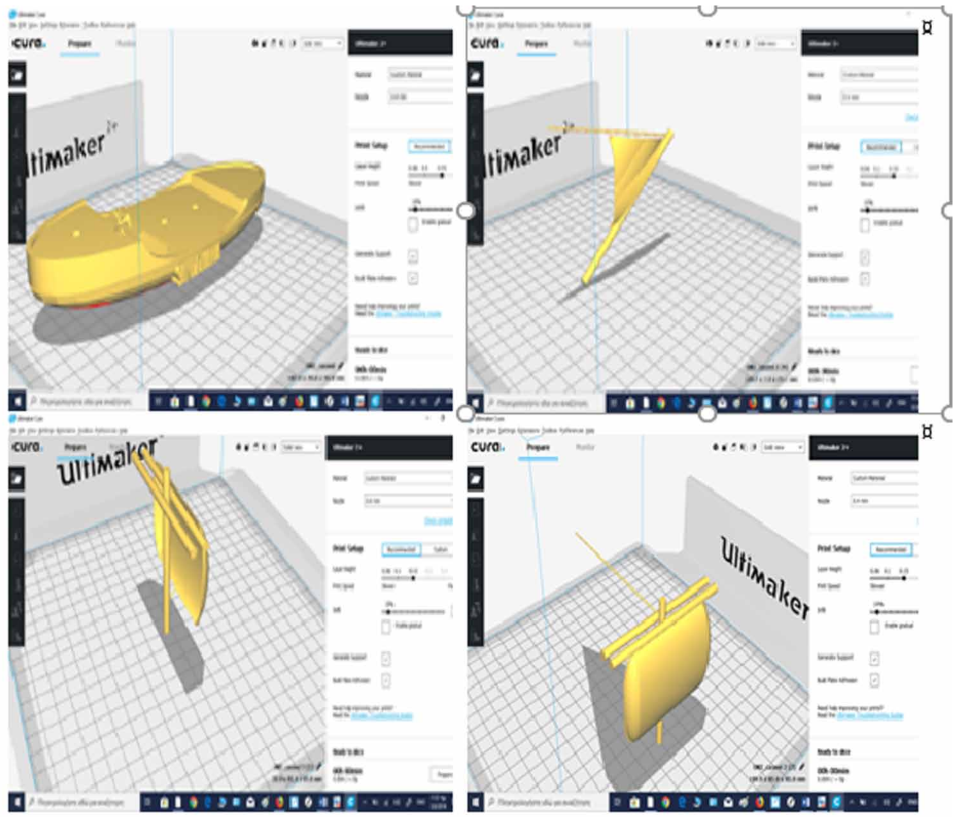



Figure 4. Astrolab Model in Solid and x-ray view

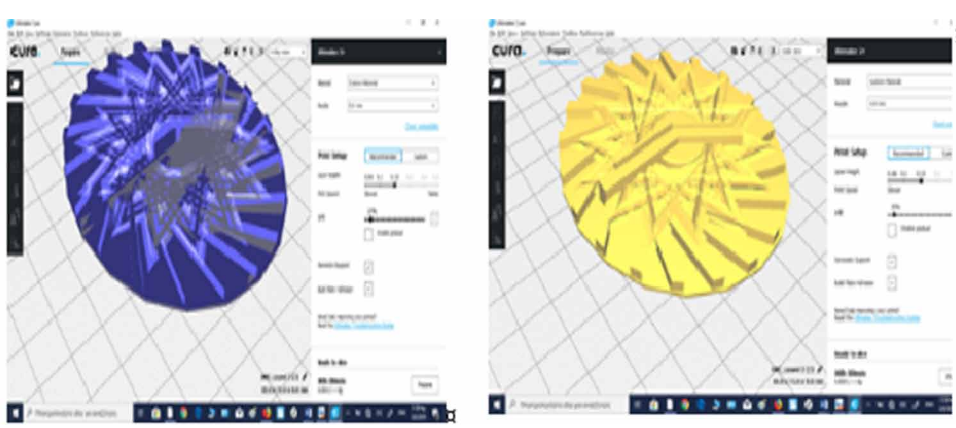
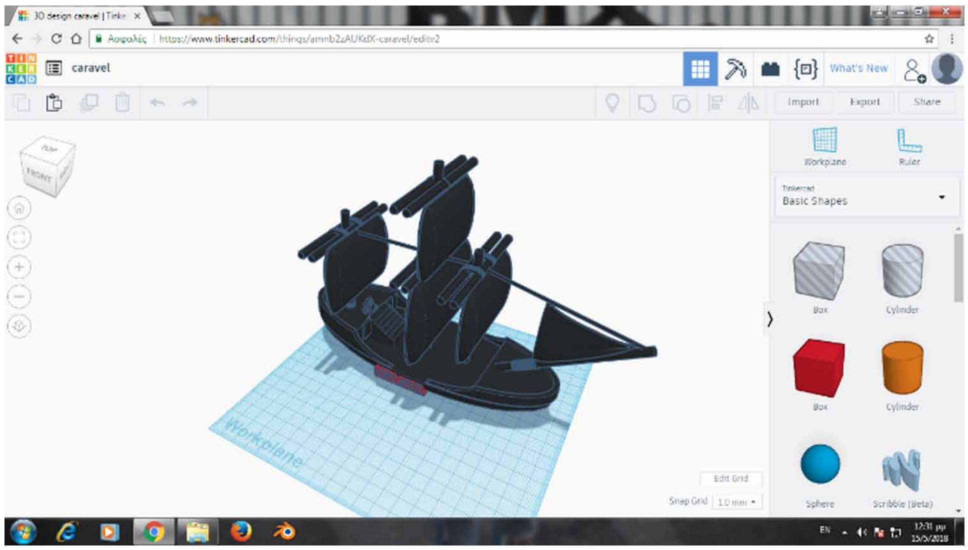


*Figure 5. A Caravel model designed in Tinkercad*




the class of History and CS, improved significantly. This study has also found that generally there is not statistically significant difference between students that have different gender. However, the sample of the study was small, which indicates that this study could be repeated on a bigger, statistically more consolidated sample. Only 40% (forty out of one hundred and tree) of B Grade students volunteered to participate. According to previous assessments they were students with moderate performance across all disciplines and exceptionally, some of them showed a particular calling in either the sciences or theoretical disciplines. Also, even if the computer lab was available during and after school, the total time dedicated to the program was limited, especially in the context of History and due to the need to follow the curriculum. Students had to spend time for retrieving information, designing and printing the objects for many weeks after the completion of the equivalent book chapter. It is obvious that this approach cannot be implemented to a large scale without drastic changes in the curriculum and the way of teaching.

As said before one source of weakness in this study, which could have affected the measurements of the Wilcoxon and Mann-Whitney U test is the number of the sample and another the limited hours that were dedicated to the project. In addition, in this project the performance of the students and their potential familiarity with technology is not examined or explored in any way. Although the current study is based on a small sample of participants, the findings suggest that we could support the idea that this study could strengthen the idea of the addition of the 3d design and printing to the classes of History and Informatics in the Secondary School. The present study also lays the groundwork for future research into the addition of the 3d design and printing to other courses as well.

## Project Evaluation

After the completion of the project students were asked to respond to an online post-implementation survey, developed by teachers in a google form format. Analytical presentation and evaluation of the questionnaire are outside the scope of this work; however, it is worth noting some of the students' answers. The questionnaire was anonymous and was completed by the students after a discussion in classroom where its main aspects were analyzed and the students were asked to complete it accurately and consistently. It consisted of a few demographic questions about sex, classroom, previous school etc., a few five-point scale question about the students' opinion for the cognitive subjects involved (History and Informatics), before and after the program's implementation in the classroom, a few five-point rating scale questions to evaluate the learning outcomes and the students' opinion about if and how much they think they have benefited from the application of the cross-cutting scenario and a two open-ended questions about the positive and the negative parts of the program according to students feeling. All forty students, twenty male students and 20 female students completed the questionnaire by answering all the questions in the questionnaire.

The encouraging results were announced in the teachers' club and were a source of inspiration for the implementation of cross-curricular programs with STEM and non-STEM subjects.

Students participating in the project had also to answer a second questionnaire. This questionnaire concerned students participating in different 3D technologies activities taken place in school and sought to investigate students' views on whether they consider that their involvement with 3D technology during the school year has developed their computational thinking skills and abilities. The questionnaire consisted of 8 rating scale questions concerning CT skills developed and student's thoughts about 3D technologies, 5 closed-ended questions about the cognitive subject of Informatics and about the student's opinion for 3D technologies, 2 open-ended question about what they liked or not during their occupation with 3D technologies and 2 demographic questions concerning sex and professional orientation. 94 out 150 students responded to this questionnaire. As the questionnaire was anonymous the answer of the forty students participated in the CCAP project cannot be extracted and discussed. Although, a few answers about the development of Computational Thinking skills will be mentioned in the following.

## RESULTS AND STATISTICS

In this section the purpose of the CCAP and the results of some answers will be discussed. First of all, it is needed to mention that the purpose of this activity was dual; On the one hand students had to conceptualize the tools used by Columbus, understand them, design and print them and on the other to communicate and cooperate with their fellow students and teachers. In addition, in order to process and later combine the information from two seemingly irrelevant fields, they had to think "interdisciplinary" and "computationally". Lastly, simple statistical analysis was used to prove that after the implementation of the 3D printing the motivation of the students rose up regarding the classes of History and Technology.

Analysis on the questionnaire showed that the use of 3D improved the understanding of the usage of 3D technologies in learning and in daily life. Furthermore, the majority of the students responded that the 3D printing activity assisted in conveying the knowledge from the field of History to Technology and vice versa. In addition, 82,5% admitted that via the knowledge of the historic use of these tools and their creation using 3D technology they understood their importance in daily life. Moreover, the 77,5% of the students emphasized that the fact of the representation of the tools made them comprehend how people were using them in the respective Modern Era.

Interestingly, this interdisciplinary activity underlined one of the pillars of computational thinking, the cooperation of students. The most surprising result of this questionnaire was that students admitted in a percentage of 70% that they coordinated harmonically with their professors. Not to mention that in an even greater percentage, 82,5%, they admitted that they had an effective and symphonic coordination with their classmates.

*Figure 6. Impression of the class of History before the program*

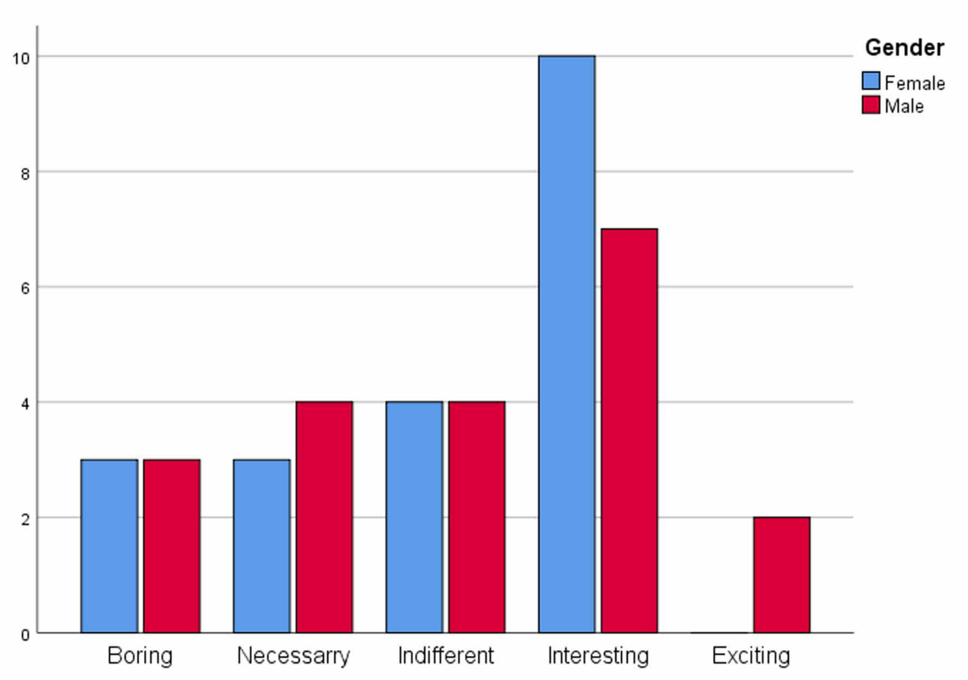



*Figure 7. Impression of the class of History before the program Mann-Whitney test*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_History_before3d | Male | 20 | 20,48 | 409,50 |
| | Female | 20 | 20,53 | 410,50 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Impresion_History_before3d |
|---|---|
| Mann-Whitney U | 199,500 |
| Wilcoxon W | 409,500 |
| Z | -,014 |
| Asymp. Sig. (2-tailed) | ,989 |
| Exact Sig. [2*(1-tailed Sig.)] | ,989[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

According to the graphical representation of the impression of the students (N=40) for the class of History before the inclusion of the program in the class there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Interesting" (42.5%). On average, the students found the class close to indifferent (m=3.05, sd=1.197), which means that a big number of students found the class boring, necessary or indifferent, as we can see in the descriptive statistics. From this data, it can be concluded that the impression for the class of History before the implementation of the program of the males was not statistically significant than this of the females (U (39) = 199.5, N1=20, N2=20, p= .989).

According to the graphical representation of the impression of the students (N=40) for the class of History after the inclusion of the 3d printing in the class there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Interesting" (60%). On average, the students found the class close to interesting (m=4.22, sd=0.8), which means that a big number of students found the class boring, necessary or indifferent, as we can see in the descriptive statistics. From this data, it can be concluded that the impression for the class of Homer after the implementation of the 3d printing of the males was not statistically significant than this of the females (U (39) = 199.5, N1=20, N2=20, p= .239 two-tailed). Therefore, we do not reject the null hypothesis (H0).

According to the graphical representation of the impression of the students (N=40) for the class of Informatics before the addition of the 3d printing in the class there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Interesting" (55%). On average, the students found the class close to indifferent (=3.28, sd=1.339), which means that a big number of students found the class boring, necessary or indifferent, as we can see in the descriptive statistics. From this data, it can be concluded that the impression for the class of Informatics before the implementation of the 3d printing of the males was not statistically significant than this of the females (U (39) = 164.5, N1=20, N2=20, p= .290). Therefore, we do not reject the null hypothesis (H0).

According to the graphical representation of the impression of the students (N=40) for the class of Informatics after the use of the 3d printing in the class there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Exciting" (47.5%). On average, the students found the class close to interesting (m=4.05, sd=1.28). From this data, it can be concluded that the impression for the class of Informatics after the implementation of the 3d printing of the males was likely not statistically significant than this of the females (U (39) = 146, N1=20, N2=20, p= .113). Therefore, we do not reject the null hypothesis (H0).

According to the graphical representation of the impression of the students (N=40) of the teaching impression it is evident that the data in the following graph are right-skewed. The most common answer amongst the students was "Interesting" (67.5%). On average, the students found the class close to interesting (m=3.83, sd=0.844). There is no evidence to support a difference between the ratings of understanding impression of the teaching experience between the two genders. (U (39) = 160, N1=20, N2=20, p= .289). Therefore, we do not reject the null hypothesis (H0).

According to the graphical representation of the transmition of knowledge (N=40) via the of the 3d printing in the class there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Neither agree or disagree" (45%). The mean of the impression of the students lies between neither agree or disagree and agree (m=3.48, sd=1.086). There is no evidence to support a difference between the ratings of transmission of knowledge via the 3d design and printing between the two genders. (U (39) = 184.5, N1=20, N2=20, p= .656). Therefore, we do not reject the null hypothesis (H0).

*Figure 8. Impression of the class of History after the program*

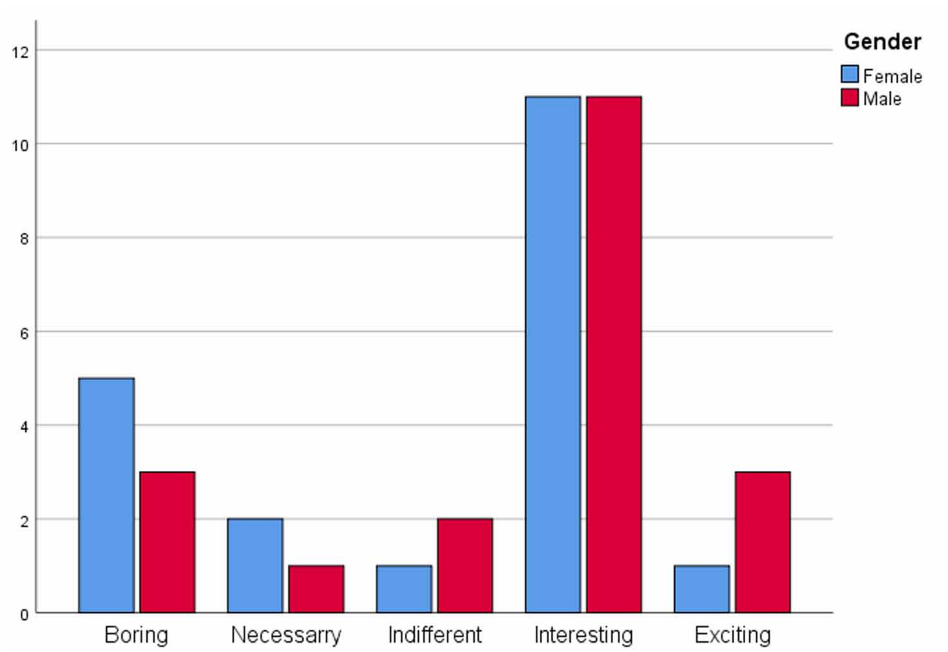



*Figure 9. Impression of the class of History after the program Mann-Whitney test*

**Mann-Whitney Test**

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_History_after3d | Male | 20 | 22,38 | 447,50 |
| | Female | 20 | 18,63 | 372,50 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Impresion_History_after3d |
|---|---|
| Mann-Whitney U | 162,500 |
| Wilcoxon W | 372,500 |
| Z | -1,178 |
| Asymp. Sig. (2-tailed) | ,239 |
| Exact Sig. [2*(1-tailed Sig.)] | ,314[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

According to the graphical representation of the understanding of the students (N=40) there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Agree" (32.5%). On average, the students found the class close to interesting (m=4.05, sd=1.109). There is no evidence to support a difference between the ratings of understanding importance of 3d between the two genders. (U (39)= 198, N1=20, N2=20, p= .955). Therefore, we do not reject the null hypothesis (H0). This means that the way that males and females perceive the 3d design does not vary statistically.

According to the graphical representation of the understanding of the lives of people in Columbus world by the students (N=40) there are some worth noting results. The data in the following graph are right-skewed. The most common answer amongst the students was "Agree" (35%). On average, the students found the class close to Neither agree or disagree (m=3.4, sd=1.277). There is no evidence to support a difference between the ratings of understanding the lives of people between the two genders. (U (39) = 178, N1=20, N2=20, p= .538). Therefore, we do not reject the null hypothesis (H0). This means that the way that males and females perceive the lives of the people in the ancient world does not vary statistically.

According to the graphical representation of the cooperation of the students with their classmates (N=40) the data in the following graph are right-skewed. The most common answer amongst the students was "Totally agree" (60%). On average, the students found the class close to Agree (m=4.22, sd=1.23). There is no evidence to support a difference between the ratings of the cooperation amongst the students between the two genders. (U (39) = 149, N1=20, N2=20, p= .116). Therefore, we do not reject the null hypothesis (H0).

According to the graphical representation of the cooperation of the students with their professors (N=40) the data in the graph above are right-skewed. The most common answer amongst the students was "Totally agree" (55%). On average, the students found the class close to Agree (m=3.98, sd=1.387). There is no evidence to support a difference between the ratings of the cooperation with the professors between the two genders. (U (39) = 156, N1=20, N2=20, p= .191). Therefore, we do not reject the null hypothesis (H0).

According to a non-parametric Spearman's Rho test the possibility that this relation is positive r=0.378 and statistically important is very high p=0.016(correlation is significant at the a=0.05 (0.05 level)). Therefore, we can proceed with the non-parametric Wilcoxon test, to check whether we accept or not the null hypothesis (H0) or the alternative (Ha). "A Wilcoxon Signed-Ranks Test indicated that post-test ranks, the impression of the students for the class of History, were statistically significantly higher than pre-test ranks, impression of students for the class of History. (Z = -4.529, p< .000.)". Therefore, it is safe to reject the null hypothesis (H0) that supports that there is no effect of the 3d printing regarding to the impression that the students have for the class of History.

According to a non-parametric Spearman's Rho test the possibility that this relation is positive r=0.573 and statistically important is very high p<0.001 (correlation is significant at the a=0.01 (0.01 level)). Therefore, we can proceed with the non-parametric Wilcoxon test, to check whether we accept or not the null hypothesis (H0) or the alternative (Ha). A Wilcoxon Signed-Ranks Test indicated that post-test ranks, the impression of the students for the class of IT, were statistically significantly higher than pre-test ranks, impression of students for the class of IT. (Z = -30748, p< .000.)". Therefore, it is safe to reject the null hypothesis (H0) that supports that there is no effect of the 3d printing regarding to the impression that the students have for the class of IT.

*Figure 10. Impression of the class of Informatics before the program*

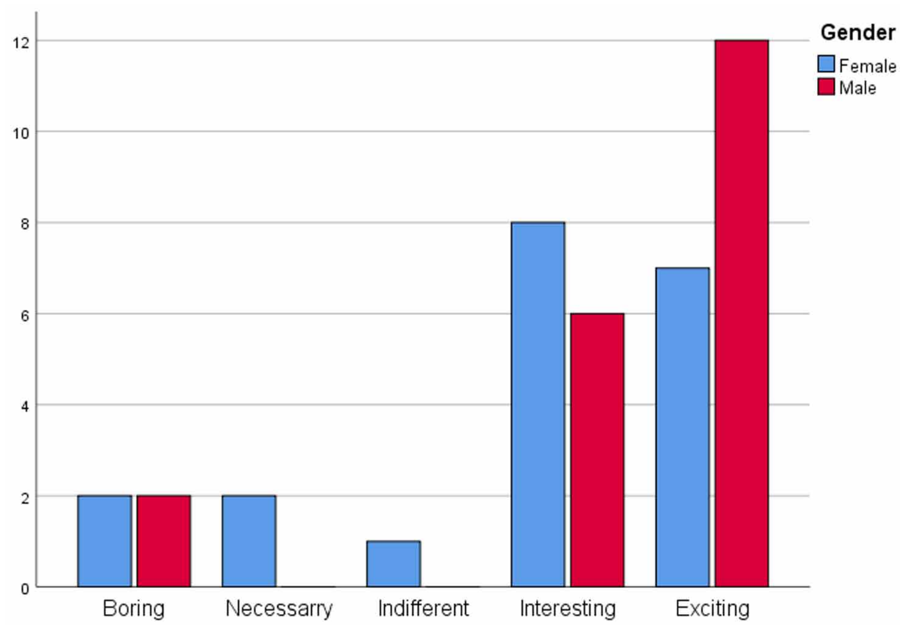



*Figure 11. Impression of the class of Informatics before the program Mann-Whitney test*

**Mann-Whitney Test**

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_IT_before3d | Male | 20 | 22,28 | 445,50 |
| | Female | 20 | 18,73 | 374,50 |
| | Total | 40 | | |

**Test Statistics**[a]

| | Impresion_IT_before3d |
|---|---|
| Mann-Whitney U | 164,500 |
| Wilcoxon W | 374,500 |
| Z | -1,058 |
| Asymp. Sig. (2-tailed) | ,290 |
| Exact Sig. [2*(1-tailed Sig.)] | ,341[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 12. Impression of the class of Informatics after the program*

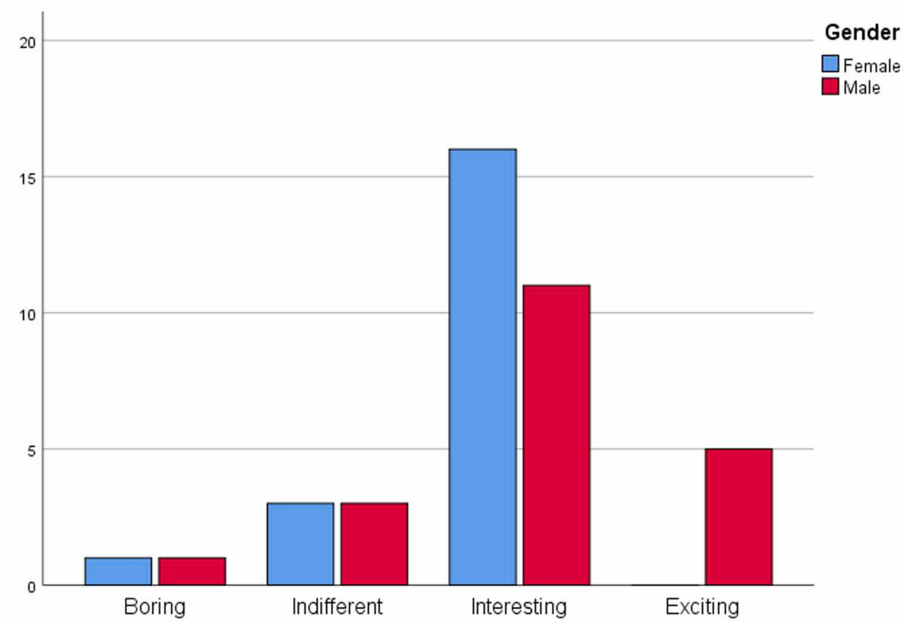



*Figure 13. Impression of the class of Informatics after the program Mann-Whitney test*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_IT_after3d | Male | 20 | 23,20 | 464,00 |
| | Female | 20 | 17,80 | 356,00 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Impresion_IT_after3d |
|---|---|
| Mann-Whitney U | 146,000 |
| Wilcoxon W | 356,000 |
| Z | -1,585 |
| Asymp. Sig. (2-tailed) | ,113 |
| Exact Sig. [2*(1-tailed Sig.)] | ,149[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 14. Impression of the teaching experience through 3d*

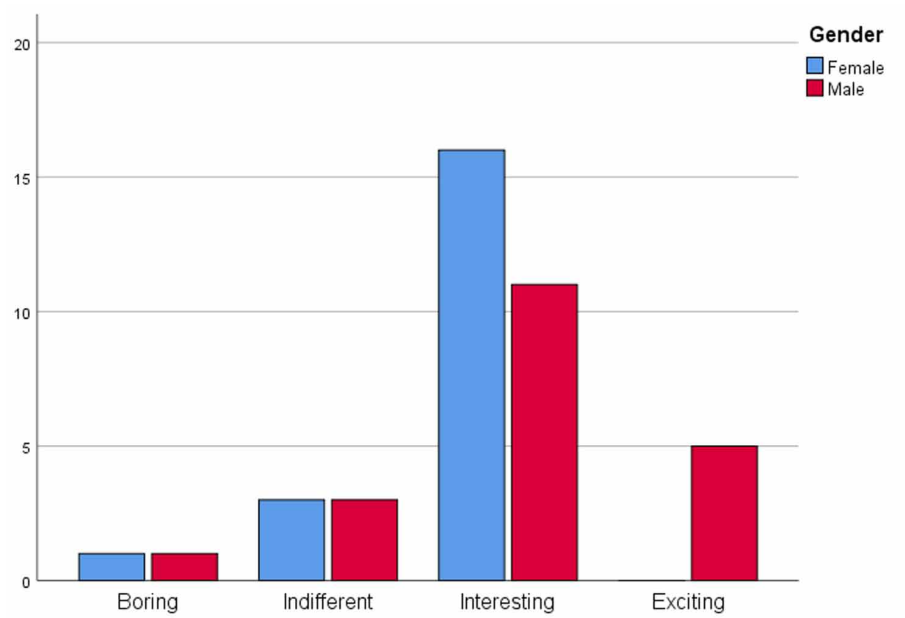



*Figure 15. Impression of the teaching experience through 3d Mann-Whitney test*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_teaching_experience | Male | 20 | 22,50 | 450,00 |
| | Female | 20 | 18,50 | 370,00 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Impresion_teaching_experience |
|---|---|
| Mann-Whitney U | 160,000 |
| Wilcoxon W | 370,000 |
| Z | -1,305 |
| Asymp. Sig. (2-tailed) | ,192 |
| Exact Sig. [2*(1-tailed Sig.)] | ,289[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 16. Transmit of knowledge via 3d*

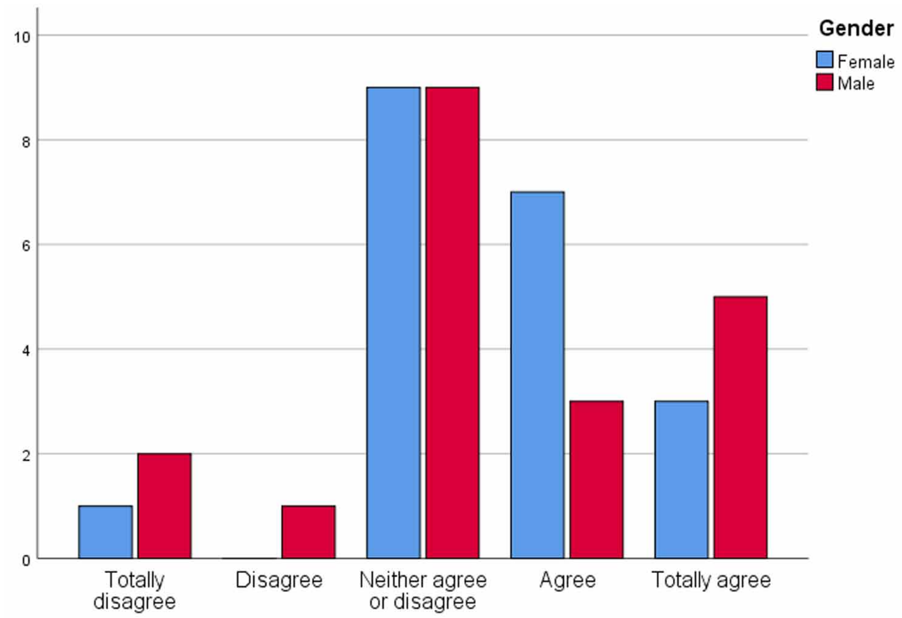



*Figure 17. Transmit of knowledge via 3d Mann- Whitney test*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Transmit_knowledge | Male | 20 | 19,73 | 394,50 |
| | Female | 20 | 21,28 | 425,50 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Transmit_knowledge |
|---|---|
| Mann-Whitney U | 184,500 |
| Wilcoxon W | 394,500 |
| Z | -,446 |
| Asymp. Sig. (2-tailed) | ,656 |
| Exact Sig. [2*(1-tailed Sig.)] | ,678[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 18. Understanding the importance of 3d*

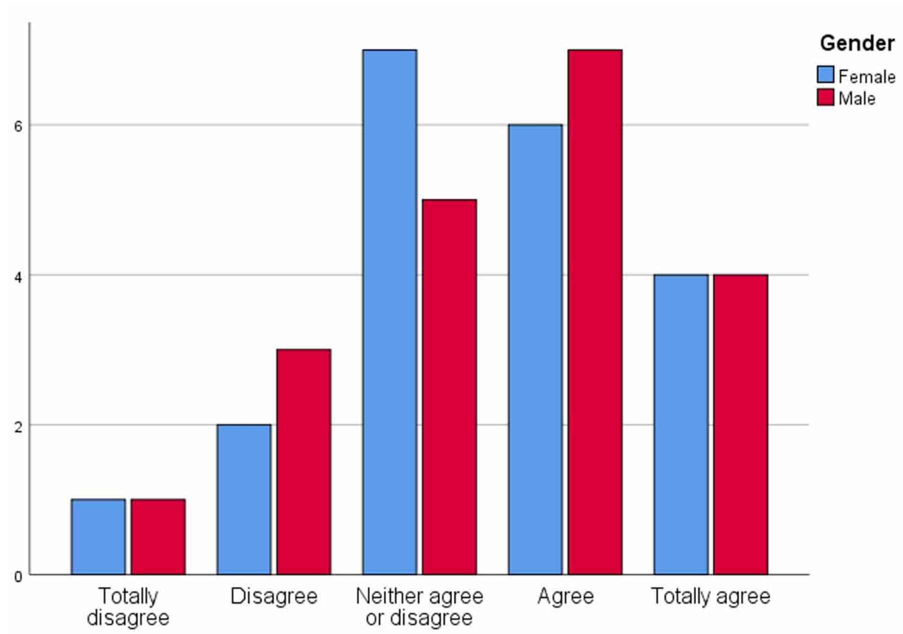



*Figure 19. Understanding the importance of 3d Mann- Whitney test*

**Mann-Whitney Test**

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Understand_importance | Male | 20 | 20,60 | 412,00 |
| | Female | 20 | 20,40 | 408,00 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Understand_importance |
|---|---|
| Mann-Whitney U | 198,000 |
| Wilcoxon W | 408,000 |
| Z | -,056 |
| Asymp. Sig. (2-tailed) | ,955 |
| Exact Sig. [2*(1-tailed Sig.)] | ,968[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 20. Understanding the lives of people at Columbus era via 3d Mann- Whitney test*

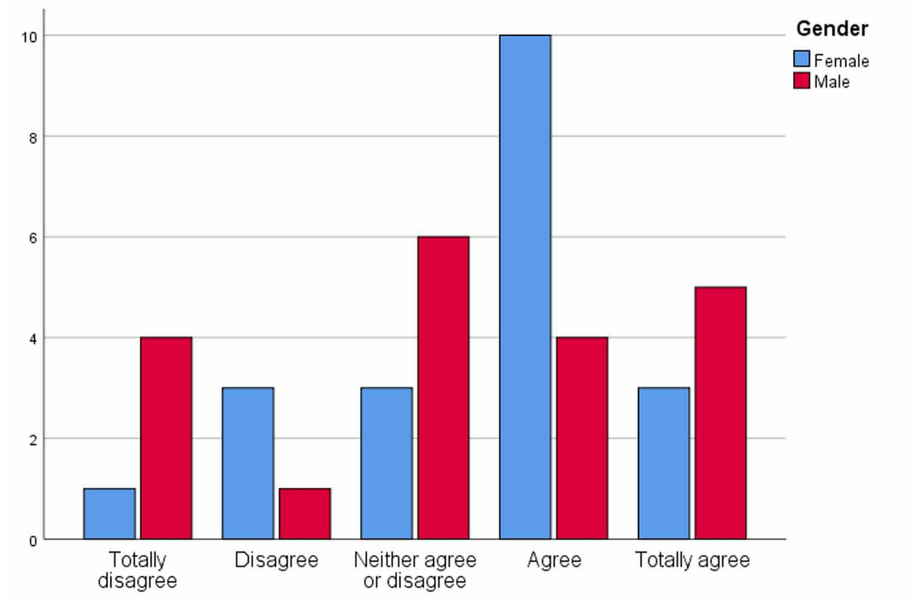



*Figure 21. Understanding the lives of people at Columbus era via 3d Mann- Whitney test*

## Mann-Whitney Test

### Ranks

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Understand_lives_people | Male | 20 | 19,40 | 388,00 |
| | Female | 20 | 21,60 | 432,00 |
| | Total | 40 | | |

### Test Statistics[a]

| | Understand_lives_people |
|---|---|
| Mann-Whitney U | 178,000 |
| Wilcoxon W | 388,000 |
| Z | -,615 |
| Asymp. Sig. (2-tailed) | ,538 |
| Exact Sig. [2*(1-tailed Sig.)] | ,565[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 22. Cooperation with classmates*

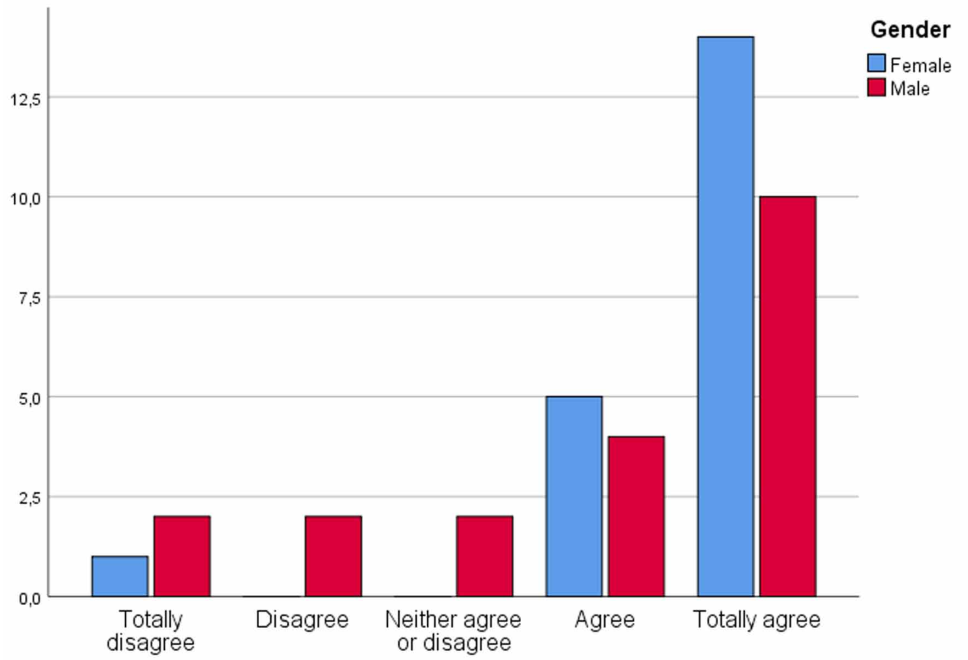



*Figure 23. Cooperation with classmates Mann- Whitney test*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Cooperation_with_classmates | Male | 20 | 17,95 | 359,00 |
| | Female | 20 | 23,05 | 461,00 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Cooperation_with_classmates |
|---|---|
| Mann-Whitney U | 149,000 |
| Wilcoxon W | 359,000 |
| Z | -1,570 |
| Asymp. Sig. (2-tailed) | ,116 |
| Exact Sig. [2*(1-tailed Sig.)] | ,174[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 24. Cooperation with teachers Mann- Whitney test*

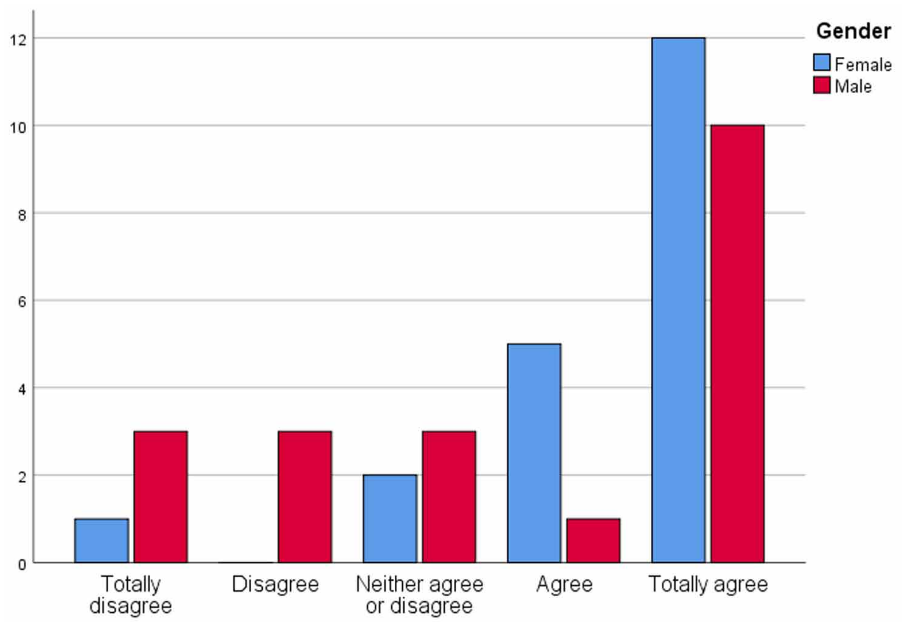



*Figure 25. Cooperation with teachers Mann- Whitney test*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Cooperate_with_professors | Male | 20 | 18,30 | 366,00 |
| | Female | 20 | 22,70 | 454,00 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Cooperate_with_professors |
|---|---|
| Mann-Whitney U | 156,000 |
| Wilcoxon W | 366,000 |
| Z | -1,308 |
| Asymp. Sig. (2-tailed) | ,191 |
| Exact Sig. [2*(1-tailed Sig.)] | ,242[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 26. Comparison of the impression at the class of History before and after the inclusion of the 3d - Correlations*

**Correlations**

| | | | Impresion_History_before3d | Impresion_History_after3d |
|---|---|---|---|---|
| Spearman's rho | Impresion_History_before3d | Correlation Coefficient | 1,000 | ,378* |
| | | Sig. (2-tailed) | . | ,016 |
| | | N | 40 | 40 |
| | Impresion_History_after3d | Correlation Coefficient | ,378* | 1,000 |
| | | Sig. (2-tailed) | ,016 | . |
| | | N | 40 | 40 |

*. Correlation is significant at the 0.05 level (2-tailed).

*Figure 27. Comparison of the impression at the class of History before and after the inclusion of the 3d – Wilcoxon test*

**Descriptive Statistics**

| | N | Mean | Std. Deviation | Minimum | Maximum | Percentiles 25th | Percentiles 50th (Median) |
|---|---|---|---|---|---|---|---|
| Impresion_History_before3d | 40 | 3,05 | 1,197 | 1 | 5 | 2,00 | 3,00 |
| Impresion_History_after3d | 40 | 4,22 | ,800 | 1 | 5 | 4,00 | 4,00 |

## Wilcoxon Signed Ranks Test

**Ranks**

| | | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_History_after3d - Impresion_History_before3d | Negative Ranks | 0[a] | ,00 | ,00 |
| | Positive Ranks | 26[b] | 13,50 | 351,00 |
| | Ties | 14[c] | | |
| | Total | 40 | | |

a. Impresion_History_after3d < Impresion_History_before3d

b. Impresion_History_after3d > Impresion_History_before3d

c. Impresion_History_after3d = Impresion_History_before3d

**Test Statistics[a]**

| | Impresion_History_after3d - Impresion_History_before3d |
|---|---|
| Z | -4,529[b] |
| Asymp. Sig. (2-tailed) | ,000 |

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

*Figure 28. Comparison of the impression at the class of Informatics before and after the inclusion of 3d - Correlations*

**Nonparametric Correlations**

**Correlations**

| | | | Impresion_IT _before3d | Impresion_IT _after3d |
|---|---|---|---|---|
| Spearman's rho | Impresion_IT_before3d | Correlation Coefficient | 1,000 | ,573** |
| | | Sig. (2-tailed) | . | ,000 |
| | | N | 40 | 40 |
| | Impresion_IT_after3d | Correlation Coefficient | ,573** | 1,000 |
| | | Sig. (2-tailed) | ,000 | . |
| | | N | 40 | 40 |

**. Correlation is significant at the 0.01 level (2-tailed).

*Figure 29. Comparison of the impression at the class of Informatics before and after the inclusion of 3d- Wilcoxon test*

**Descriptive Statistics**

| | N | Mean | Std. Deviation | Minimum | Maximum |
|---|---|---|---|---|---|
| Impresion_IT_before3d | 40 | 3,28 | 1,339 | 1 | 5 |
| Impresion_IT_after3d | 40 | 4,05 | 1,280 | 1 | 5 |

**Wilcoxon Signed Ranks Test**

**Ranks**

| | | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impresion_IT_after3d - Impresion_IT_before3d | Negative Ranks | 2[a] | 14,75 | 29,50 |
| | Positive Ranks | 23[b] | 12,85 | 295,50 |
| | Ties | 15[c] | | |
| | Total | 40 | | |

a. Impresion_IT_after3d < Impresion_IT_before3d

b. Impresion_IT_after3d > Impresion_IT_before3d

c. Impresion_IT_after3d = Impresion_IT_before3d

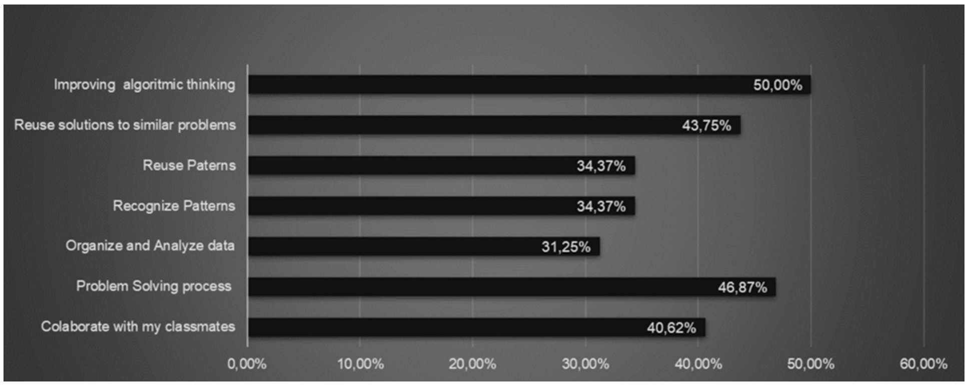
**Test Statistics[a]**

| | Impresion_IT _after3d - Impresion_IT _before3d |
|---|---|
| Z | -3,748[b] |
| Asymp. Sig. (2-tailed) | ,000 |

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

*Figure 30. Students Impression related to Computational Thinking Skills.*




As noticed above, students participating in CCAP project answered an extensive questionnaire created to investigate how 3D orienting technologies contribute to the improvement of Computational Thinking related skills, which was answered by 95 students of all grades participating in 3d activities during the school year, i.e. in Informatics subject, in interdisciplinary programs organized by teachers, and other STEM and environmental activities. The answers of the students participating in CCAP project were isolated and shown in the graph above. 32 out of 40 participated students answered the questionnaire. Every student had to choose 3 out of the 6 available choices. The 50% of the students believes that they improved their algorithmic thinking skills, while the 46,9% that improved his problem-solving skills by breaking a problem into more manageable ones. The recognition and the reuse of patterns in other similar simulations is followed by the one third of the students. A 40,62% of the students believe that they improved their collaboration attitude against their classmates while only the 31,25% of the students considers that their skill in organizing and analyzing data was improved. As the sample is considered to be quite small, it would be interesting to follow learners for a longer period of time to find out whether motivation affects their Computational Thinking skills in these two classes or whether this improved impression over these two classes does not affect the learning process.

## DISCUSSION

In this section the results will be discussed and the earlier analyzed theory, focusing on the computational thinking, will be linked with them. Previous studies evaluating computational thinking observed that one of the fundamentals is the process of conceptualization. The majority of students stated that not only were they able to conceptualize, but to think, as Wing supports, the use of these tools to navigate. Additionally, children understood in a high percentage how to make use of these tools in today's life. Plus, it is important to mention that the motivation of the students for the corresponding fields rose up. The importance of the benefits gained for students when they create their own artifacts must also be mentioned. Artifacts are important because they comprise primary evidence for students to understand what happened in the past, to study deeper, the civilization, the technology and the use these tools in Colobus era, over the course of history and to compare the significance and the difficulties faced during that trip comparing to the technologies used today. From the point of view of the cognitive subject of

Informatics, students have come into contact with a very promising field of technology that promises to play an important role in many areas of future human technological and scientific activity, focusing on the environment, health, engineering and applied sciences.

In the article of Wing it is noticed that the combination of several fields is necessary to first imagine and later understand in a deeper level not limiting the conceptualization to the real world. Most of the students did understand the use of the 3D technology and its importance and went a step further, by sharing their ideas with their schoolmates and teachers. Sharing the ideas with everyone, they achieved the essence of computational thinking. Firstly, to process the knowledge and not "compute" and secondly to spread it to everyone, at least amongst the class.

At this point the criteria for approaching the overall effort must be mentioned. To the authors point of view, these criteria enforced students to deal with the study of navigation tools, their design and printing. The challenge was real. It involved an authentic challenge grounded in compelling societal, economic, and environmental issues that affected people's lives and communities during C. Columbus era but also today. Students must be able to relate to the challenge. If students do not care about the challenge, their buy-in will be limited. This needs to be a significant challenge that students care about. Even if constructing navigation tools was not a daily problem in their own life or community, teachers decided to proceed with a group of volunteers who decided to commit themselves with an unfamiliar challenge as studying and creating navigation tools of another era. The challenge was feasible. For a STEM challenge to be successful, students should have access to the resources, knowledge, and skills they need to solve the problem—and the scope of the problem should be affordable and manageable by the students. Designing navigation tools according to simple scientific and engineering standards was a realistic goal. The challenge allowed for multiple acceptable approaches and solutions. Solutions and proposals by students never were considered as an issue with a single, predetermined approach and "right" or "wrong" answer. Each team of students could choose a different approach for creating their tools and worked out on several different solutions Students had an active role in choosing the challenge. Although teachers came up with examples and suggestions from global History in order to help students understand the challenge, the final choice of the history chapter chosen was left to students. Their active involvement paved the way for their engagement and active participation to the learning process.

Also, as in many other studies found in literature concerning integrated STEM teaching i.e (Schauble et al. 1995), where students learned science through design and showed an improved understanding of experimentation, interviews with the students before and after the activity revealed that they learned history through design and showed an improved understanding of the use of tools in navigation in modern and in C. Columbus era. Even if, in recent years, development of Computational Thinking skills is related with concepts as coding, programming, algorithmic thinking and have been promoted by educational systems as fundamental skills for pupil's education (Papadakis, Kalogiannakis, Orfanakis, Zaranis, 2017), programming is considered by students as a mysterious and complex procedure which requires specialized technical training and education ((Papadakis, Kalogiannakis, Orfanakis, Zaranis, 2014). 3D technologies provide a fertile land of experimentation for developing Computational Thinking skills without demanding special knowledge and effort by the students and teachers and simulate a more game-based teaching and learning experience in fields related to STEM education as limited water supply, land usage, the coexistence of animals and humans, the effect of wildfires on a local community, energy consumption and energy sources, climate change, overpopulation, health & well-being, agriculture methods and their impact, lack of resources, access to scientific and technological breakthroughs etc.

In terms of statistics and tests used in this case, we know that for a medium effect size we probably need about 85 participants per sample for a power of 0.8. In this case, the number of the participants are 40, which consist a small group of participants. The power of the tests is small to medium, given the fact that the effect size (d = M1 - M2 / σpooled) for the class of History is 0.498 and for the Informatics 0.28. As for the methodology of the study, we would maybe want to know more about the familiarity of the participants with the classes of IT, History and their relation with 3d and generally digital technologies and innovation. It seems that the incorporation of 3d technologies has a greater effect to the class of History, compared to the Informatics class. Thus, it would be interesting to expand the 3d application to a variety of theoretical and practical lessons taught in schools in order to observe whether there is initially an improvement on the impression of the students regarding other classes and long term if there is an amelioration of their performance on them.

This study assumes that students have a better impression and are highly and more motivated to participate to the classes of Informatics and History Iliad if 3d technologies consist part of the classes. It would thus be interesting to follow learners for a longer period of time to find out whether motivation affects their performance and learning in these two classes or whether this improved impression over these two classes does not affect the learning process.

## CONCLUSION

According to the statistic test done, two observations are unambiguous. Firstly, that students' motivation for the classes of History and Technology that were related to the CCAP activity upraised. Secondly, it is proved that this phenomenon was not affected by the gender, which means that males and females displayed statistically higher motivation for both these classes.

Also, the implementation of the 3D improved the understanding of the concepts in the field of history on one hand and the implementation of them using technological tools on the other. Further understanding of these tools was accomplished not only in today's but in Columbus era as well. The use of the 3D printing also, as seen in the discussion, improved the communication amongst students and teachers. The high rates of concordance cause optimism for the future of the relation amongst students and teachers. Results indicate that students learned to co-participate in the design process and revealed practices of experimentation; practices usually present in informed designers with good knowledge of the subject. During the design process while analyzing the problem space, generating ideas, and evaluating solutions, they developed better understanding of the relationships between the navigation tools and their use and underlying science concepts, used various sources of information and graphical representations embedded in the available literature to inform their design decisions.

If the debate is to be moved forward, a bigger group and more time spent is necessary and a questionnaire during the 3D activities would also give us more insight. Further work is needed to fully understand if the improvement of the impression and the skills of the children regarding the classes is transient or not, if this improvement influences their performance of the children on these classes and if their Computational Thinking skills have been developed. Further research should be undertaken to explore how this action improves the relations of the children with technology before and after the experiment as well.

If STEM activities provided are not systemically connected to the curriculum, they will be far from being adopted by most schools and teachers. Another issue is that even if creativity and innovation are in the spotlight for formal and/or informal education it's still difficult to measure the impact of these programs and therefore their overall efficiency. The future task should be to establish a standard for evaluate these processes, so that the positive outcomes of these experimental or innovative programs could be integrated in a structured national programme more connected to STEM curricula and development of Computational Thinking skills.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to GFOSS (www.gfoss.eu) for providing the Ultimaker 2+ 3D printer for two school years, at the 1st High School of Vrilissia, for implementing this project.

## REFERENCES

Balanskat, A., Engelhardt, K., & Licht, A. H. (2018). *Strategies to include computational thinking in school curricula in Norway and Sweden*. Retrieved from http://www.eun.org/documents/411753/817341/Computational_thinking_report_2018.pdf/4d3d6fa0-dedd-4b62-a201-a26bf4dfd3a0

Berninger, V. (2015). *Interdisciplinary frameworks for schools: Best professional practices for serving the needs of all students*. Washington, DC: American Psychological Association. doi:10.1037/14437-000

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education –Implications for policy and practice*. EUR 28295EN; European Commission. Retrieved from https://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf

Corder, G. W., & Foreman, D. I. (2014). Nonparametric Statistics: A Step-by-Step Approach. *Wiley. Cura Tutorial.* Retrieved by https://all3dp.com/1/cura-tutorial-software-slicer-cura-3d/

Dasgupta, C., Magana, A. J., & Vieira, C. (2019). Investigating the affordances of a CAD enabled learning environment for promoting integrated STEM learning. *Computers & Education*, *129*, 122–142. doi:10.1016/j.compedu.2018.10.014

Derrick, B., & White, P. (2017). Comparing Two Samples from an Individual Likert Question. *International Journal of Mathematics and Statistics*, *18*(3), 1–13.

Hoachlander & Yanofsky. (2011, March). Making STEM Real (2011). *Educational Leadership*, *68*(6), 60–65.

James, I., & Novak, I. J. (2019). *Re-educating the Educators: Collaborative 3D Printing Education. In Interdisciplinary and International Perspectives on 3D Printing in Education* (pp. 28–49). IGI Global.

Mader, A., & Dertien, E. (2016). Tinkering as a Method for Academic Teaching. *International Conference on Engineering and Product Design Education*. Aalborg University.

Moore, T., Stohlmann, M., Wang, H., Tank, K., Glancy, A., & Roehrig, G. (2014). Implementation and integration of engineering in K-12 STEM education. In S. Purzer, J. Strobel, & M. Cardella (Eds.), *Engineering in Pre-College Settings: Synthesizing Research, Policy, and Practices* (pp. 35–60). West Lafayette: Purdue University Press. doi:10.2307/j.ctt6wq7bh.7

Nemorin, S., & Selwyn, N. (2016). Making the best of it? Exploring the realities of 3D printing in school. *Research Papers in Education*, *32*, 1–18. doi:10.1080/02671522.2016.1225802

Ng, O.-L., & Chan, T. (2019). Learning as Making: Using 3D computer-aided design to enhance the learning of shape and space in STEM-integrated ways. *British Journal of Educational Technology*, *50*(1), 294–308. doi:10.1111/bjet.12643

Nistor, A. (2019). *Bringing Research into the Classroom – The Citizen Science approach in schools.* Scientix Observatory Report.

OECD. (2015). Excellence and Equity in Education. *PISA 2015 Results, 1*. Available at: https://www.oecd.org/education/pisa-2015-results-volume-i-9789264266490-en.htm

Papadakis, St., & Kalogiannakis, M. (2017). *Using Gamification for Supporting an Introductory Programming Course. The Case of ClassCraft in a Secondary Education Classroom. 2nd EAI International Conference on Design, Learning & Innovation*, Heraklion, Crete, Greece.

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice Programming Environments. Scratch & App Inventor: a first comparison. *IDEE '14: Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*, 1–7. 10.1145/2643604.2643613

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Psycharis, S., & Kotzampasaki, E. (2017). A Didactic Scenario for Implementation of Computational Thinking using Inquiry Game Learning. *Proceedings of the 2017 International Conference on Education and E-Learning*, 26–29. 10.1145/3160908.3160918

Sanders, M. (2009). STEM, STEM education, STEMmania. *Technology Teacher*, *68*(4), 20–26.

Schauble, L., Glaser, R., Duschl, R. A., Schulze, S., & John, J. (1995). Students' Understanding of the Objectives and Procedures of Experimentation in the Science Classroom. *Journal of the Learning Sciences*, *4*(2), 131–166. doi:10.120715327809jls0402_1

TeachThought Staff. (2015). *10 ways 3D printing can be used in education*. Retrieved from https://www.teachthought.com/technology/10-ways-3d-printing-can-be-used-in-education/

Todd, R. (2016). A conceptual framework for integrated STEM education. *International Journal of STEM Education*, *3*(1), 11. doi:10.118640594-016-0046-z

TutorialT. (n.d.). Retrieved from https://www.tinkercad.com

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–36. doi:10.1145/1118178.1118215

## ADDITIONAL READING

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch005

Nistor, A., Gras-Velazquez, A., Billon, N. & Mihai, G. (2018). Science, Technology, Engineering and Mathematics Education Practices in Europe. *Scientix Observatory Report*.

Papadakis, S. (2018). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). Hershey, PA: IGI Global; doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. In Early Childhood Development: Concepts, Methodologies, Tools, and Applications (pp. 797-819). IGI Global. doi:10.4018/978-1-5225-7507-8.ch039

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning*, *10*(1-2), 44–72. doi:10.1504/IJTEL.2018.088333

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

## KEY TERMS AND DEFINITIONS

**3D Technologies:** Refers to a variety of technologies that provide a real-life 3D visual appearance that is displayed in print-in a computer.

**Computational Thinking:** A novel problem-solving approach that enhances the interpolation of digital technologies with human ideas.

**Informatics:** A term that is used mainly in Europe (Computer Science in the USA) for the cognitive subject of teaching computing in primary and secondary education. It includes issues about software and hardware, programming, robotics, applications, etc.

**Programming:** The process of planning, scheduling, or performing of a program for a device (such as a computer).

**STEM:** The term STEM (science, technology, engineering, and mathematics) is an acronym used by those relevant to the educational method concerning the fields of science, technology, engineering, and mathematics.

# Chapter 17
# The Use of 3D Technologies to Support Computational Thinking in STEM Education

**Panagiotis Angelopoulos**
*Ministry of Education, Greece*

**Alexandros Balatsoukas**
*University of Groningen, The Netherlands*

**Adina Nistor**
*European Schoolnet, Belgium*

## ABSTRACT

*Computational thinking (CT) is increasingly emerging as a thinking skill to support the development of 21st century skills such as critical thinking, creativity, collaboration, or technology literacy, essential for students to become successful in an increasingly complex society. Educators are always looking for new strategies for developing these skills in students. Three-dimensional (3D) printing and scanning technologies are sufficiently mature and economically accessible to be used at the school level. By using 3D technologies, students explore, invent, discover, and engage in real problems and situations. This study explores the use of 3D printing technologies in a secondary school in Athens over the course of two school years. The study investigates if 3D technologies can support the development of CT skills in students.*

## INTRODUCTION

According to the Organization for Economic Cooperation and Development's (OECD) Programme for International Student Assessment (PISA, 2018) 2018 report, which tested more than half a million 15-year-old students in 79 countries (including OECD countries) and economies in reading, mathematics and science, "*about 20% of students across OECD countries perform below Level 2, considered the baseline level of proficiency in science (at Level 2, students can draw on their knowledge of basic science content*

DOI: 10.4018/978-1-7998-4576-8.ch017

*and procedures to identify an appropriate explanation, interpret data, and identify the question being addressed in a simple experiment) while n the majority of countries with comparable data, students' performance in science remained essentially unchanged since 2006. Only 8% of students across OECD countries are top performers in science*". In the report it is also noted that "*Experiments and hands-on activities can be inspiring and can help students develop a conceptual understanding of scientific ideas and transferable skills, such as critical thinking*". Also, it is noted that "*On average across OECD countries, mean performance in reading, mathematics and science remained stable between 2015 and 2018*". In the case of Greece, the mean performance remains almost the same recording a slight drop overall.

For this reason, there is an emerging need for innovative teaching approaches, to increase student motivation towards science, technology, engineering and mathematics (STEM) subjects, to develop computational thinking skills and to offer a more attractive school environment. However, what constitutes an attractive school environment has changed over the years. Thirty years ago, it would have been sufficient if classrooms were equipped with maps, plaster anatomical models and basic laboratory instruments, as most students had not access to them at home. Today, homes are better technologically equipped than schools in many ways. Although 3D technologies have become reasonably affordable in recent years, 3D printers and / or scanners are not gadgets that can be found at home and their use at schools creates an attractive learning, exploration and collaboration environment that fuels students' imagination. At the same time it is a means for students to become more productive, creative and enthusiastic in both STEM and traditional learning objects.

People define themselves by their creations and actions and students can examine works to define a story, a period, an important event in history. Simultaneously, teachers use these experiences to get the students talking and thinking about the topic. Following stories, and important events, students can freely express themselves and produce their own creations and proceed to the construction of objects used at those periods, through the utilization of digital 3D technology.

Also, Papadakis & Kalogiannakis (2017) notice that "there are concerns among researchers and education professionals that students in our classrooms are bored, unmotivated and disengaged from school. One of the reasons is that old teaching methods are no longer beneficial to the students. On the other hand, game-based learning can improve learning motivation of students. Compared with traditional lectures, digital game-based approaches can indeed produce better learning effects, which underscore the need to develop appropriate instructional materials.

## THEORETICAL FRAMEWORK

The term Computational Thinking was already known through the work of Seymour Papert (1980), while many recent works refer to both the historical course of computational thought and its spiritual roots. Although, it seems that there is not yet a commonly agreed definition of CT, Nardelli E. (2019) notices that after the widely cited communications viewpoint by Jeannette Wing an extensive discussion opened with hundreds of subsequent works and papers analyzing the expression and arguing about what Computational Thinking is. In this work we argue that occupation with 3D printing technologies helped students to develop skills considering that Computational Thinking (CT) is a problem-solving process that incorporates a number of features and processes, delivering questions and problems to students in a way that allows to use a computer and other tools and procedures to solve them, such as logically organizing and analysing data, system design and understanding of human behavior, utilizing the fundamental concepts of computer science (Wing, 2006). We also consider that Computational thinking consists of four parts:

- Decomposition – breaking a problem into smaller more manageable ones
- Pattern recognition in the sense of making connections with similar problems and conditions
- Generalizations and removal of patterns and
- Algorithm design by constructing a finite sequence of determined steps to solve a problem.

Experience has shown that these skills may or may not be innate to a student. In any case, these skills can be evolved and developed through teaching and learning approaches. OECD 2018 report considers that it is necessary to work on solutions to complex social and environmental problems and all citizens, not just future scientists and engineers, need to be willing and able to confront science-related dilemmas and problems. For this reason, it is important to promote an attractive, fun, positive and inclusive image of science at schools. Science has to be nurtured among students by exposing them early to high-quality science environments. By using three-dimensional (3D) printing and scanning technologies, students explore, invent, discover, and engage in real problems and situations. It must also be noticed that these technologies are sufficiently mature and economically accessible to be used at the school level.

Another discussion that has been arisen is either to integrate Computational Thinking as part of a taught subject or transversally across the curriculum (Balanskat, Engelhardt, Licht, 2018). The authors argue noticing that the decision depends on the aim of implementing Informatics in the curriculum. For instance, the main goal to teach Informatics is the pupils to learn coding, digital applications etc, skills strongly connected to CT's principals, while in teaching other subjects where the goal is to learn about History, Ancient Greek, Philosophy, languages etc, then CT becomes a tool to achieve this goal. In this work the authors examine both ways, teaching CT principals through the cognitive subject of Informatics and in an interdisciplinary approach, Informatics with other subjects, aiming to increase the interest and the performance of students in all involved subjects.

Additionally, Bocconi et al (2016), in their study "Developing Computational Thinking in Compulsory Education" notice that despite the widespread interest of Computational Thinking and related concepts, successful CT integration in compulsory education still faces unresolved issues and challenges. The report discusses the most significant Computational Thinking developments for compulsory education in Europe comprising an extensive desk research among the European Ministries of Education and initiatives taken by European countries to integrate CT in their curricula. It is related to actions of teacher training, of pedagogical approaches, of proper learning tools that can make CT activities accessible and to assessment of CT concepts and practices for achieving an effective integration of CT in compulsory education.

Also, a number of recent works have provided frameworks and examples for incorporating CT across different subject areas and others relate Computational Thinking to the larger context of learners informally engage in as makers and creators (Voogt et al.,2015). In this work, by using 3D technologies in formal and informal school activities, there is an effort to examine how CT skills can be developed in students through disciplines other than programming and Computer Science.Although, there are some recent studies providing ideas for investigating the effectiveness of educational CAD tools and curricular scaffolds designed specifically for K-12 students for supporting integrated STEM learning anchored in the design process (Dasgupta, Magana, Vieira, 2019; Ng & Chan, 2019) the literature in connecting the use of 3D Technologies and development of Computational Thinking skills is poor at the moment.

## THE CONTEXT OF THIS STUDY

This chapter examines the impact on school students from the use of a 3D printer, in various ways, either within the curriculum, or beyond school hours, during two school years 2017/18 and 2018/19. At the 1st Junior high school of Vrilissia, a suburb north of Athens, a 3D printer was provided by GFOSS, the Open Technologies Alliance, (a non-profit organization founded in 2008, with 35 Universities and Research Centre as current shareholders) for the school years 2017-18 and 2018-19.

3D printing was used in three different contexts:

a) In the context of the Informatics cognitive subject students of the first grade (12-13 years old) were taught how to design and print out their own creations using the school's 3D printer.
b) In the context of an interdisciplinary didactic approach between the cognitive subject of Informatics and Ancient Greek and History. Specifically, Informatics and Omer's Odyssey in the first Junior High School grade (ages 12-13 years old), Informatics and Omer's Iliad, and Informatics and History in the second Junior High School grade (ages 13-14 years old). Students' involvement in these activities was optional. In the case of first grade students worked independently while in the case of second grade students worked in groups of four to six. The teaching scenarios effectively used 3D modelling and printing techniques in order to support the level of learning and comprehension of the cognitive subjects of Ancient Greek (Odyssey and Iliad) and History. Following Homer's description, and Hephaestus, the Greek god of fire and technology, students of second grade worked in groups to design and print out "the shield of Achilles". In the case of Homer's Odyssey students were supported to design and print the raft that Odysseus constructed to arrive to his destination, Ithaca. Finally, in the case of History, students of second grade worked in groups to design and print out instruments (compasses, astrolabes, caravels etc) used during the trip of C. Columbus to discover America (CCAP project).
c) Outside the curriculum, 3D technologies were used by students to support different phases of environmental and STEM school programs, such as the STEM competition "Formula1 in Schools" or the environmental programs "Trip to Olympus, the mountain of Gods" and "Psiloritis: The Divine Mountain of the Coexistence of Ecosystems and Sensations". These programs have taken place outside the school hours and involvement in these activities was also optional.

By using 3D printing technologies, students were offered opportunities to:

- develop computational and interdisciplinary thinking skills and attitudes, as a result of using 3D printing technologies in a cross-disciplinary setting. Skills like problem solving processes incorporating problem decomposition, problem formulation, algorithmic expression and generalisation. Attitudes like the ability to confidently deal with complexity and open-ended problems, collaboration to reach the target, making mistakes, determination to find the solution and effort to think out of the box.
- design, by experimenting with 3D tools, concepts and shapes of Euclidean geometry and stereometry,
- print a 3D model by experimenting with the properties, attributes and parameters of a 3D printing tool (cura),

- acquire skills in 3D modelling and printing, following the cognitive subjects' descriptions and steps, constructing the required models (i.e. shield, raft, compasses, astrolabes etc.),
- engage in learning about epic poems, history, use of technology and in other non-formal school activities,
- learn, in a non-traditional way, about the meaning of the Achilles shield, which is considered to be the hymn of life, the meaning of Odysseus's journey back to Ithaca, and about the trip of Columbus to discover America.
- become creators of their own constructions.

Following the use of 3D printing technologies in the various activities previously described, participating students were asked to respond to online post-implementation surveys, developed by teachers. While a detailed analysis of the data collection instruments used is beyond the scope of this work, a presentation is provided in the following. The main discussion will focus on the students' answers. The questionnaire was filled in the classroom after detailed discussion with students about issues such as skills and competences needed in the 21st century, the value of thinking, the digital world, Computational Thinking definitions, approaches and literature reviews, STEM education, the value of knowledge and of multidimensional education and training.

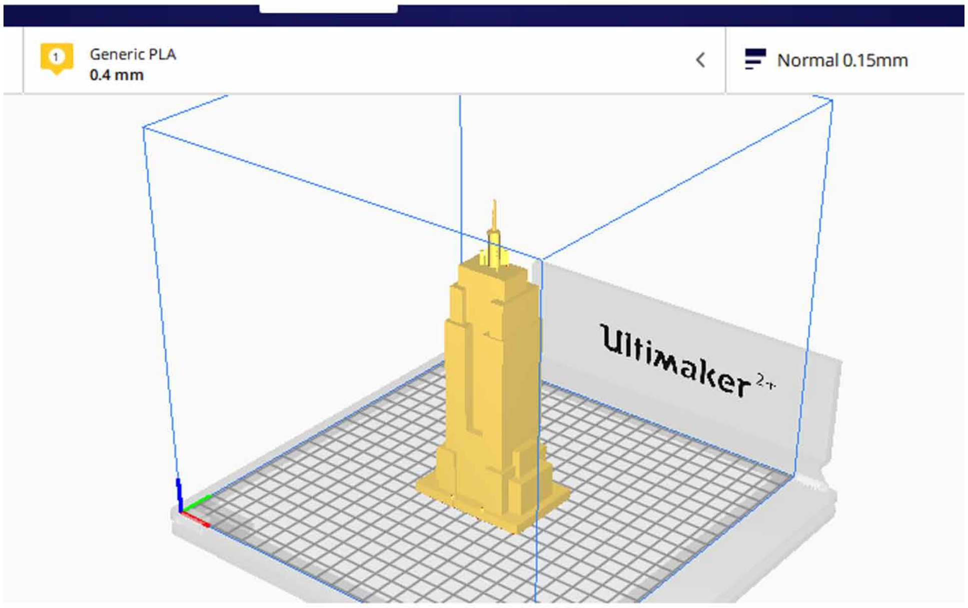
## IMPLEMENTATION

### Informatics

Cognitive subject: Informatics.
Addressing class: First Class of Junior High School (age 12-13).
Duration: 4 months' period, 1 teaching and working hour per week.
Number of students: 98 students, 8 classes of 12-13 students.
Space/Room that the classes take place: Computer Laboratory.
Printer used: Ultimaker 2+
Software used: Tinkercad, a free, easy-to-use application for 3D design, electronics, and coding and the Ultimaker Cura (version 3.1), a free, easy-to-use *3D* printing software application for printing.
Means of teaching materials and conditions: web resources, application tutorials (Tinkercad tutorial, 2017; Cura tutorial, 2017), office suite, Internet browser, headphones/speakers, Interactive White Board system, 3D design software, 3D printing software, printing material PLA and/or ABS, 3D printer were the main means of teaching and materials used to implement the program.

The overall objective of 3D modelling and printing in classroom, was to familiarize students with 3D technologies and to increase students' proficiency in key competencies and in Computational Thinking competencies, necessary to shape the future towards 2030 (Nemorin & Selwyn, 2016). 3D modeling blends advanced technology with creative thinking and problem-solving skills in a fast-paced and detail-oriented environment, skills strongly related to the development of Computational Thinking in students. 3D modelling and printing in the classroom was expected to develop the core components of Computational Thinking which as BBC (2018) outlines, these components consists of decomposition, pattern recognition, abstraction, and algorithms. Decomposition invites students to break down complex problems into smaller, simpler problems. Pattern recognition guides students to make connections between similar problems and experience. Abstraction invites students to identify important information while ignoring unrelated or irrelevant details. Lastly, students use algorithms when they design simple steps to solve problems.

*Figure 1. Designed and printable object by first grade students*




Students create their personal account in Tinkercad and get familiar (with teachers' guidance) with the design techniques needed to create their objects using the design software. They start with simple creations and gradually proceed to more complex ones. They are given assignments (homework) and they are taught how to use the application's repository and how to create their own Open Educational Recourses by licensing appropriately their own creations. The licenses offered in Tinkercad are from Creative Commons licensing suite. Subsequently, they learn how to export their drawings to a printable format. Also, they learn how to use and customize the most important functions of the printing software application (CURA) and to transfer the object's file to the printer. Finally, the get familiar with the printer's functions in order to 3D print out their creations.

Some of the student's creations licenced properly with a Creative Commons licence can be found in Appendix 1.

For a four-month period students had to create simple and complex objects (buildings, cups, trains, the Parthenon etc. and their own freehand drawings). During this procedure they had to analyse and draw their objects by breaking the initial idea into smaller more manageable parts and by assembling them to create the complete object. They had to identify patterns they used in the construction of identical parts of the overall construction, such as wheels, columns, doors, windows, roofs etc. They realized that experience gained by the previous constructions could be transferred to the future ones. Finally, every student developed his/her own hierarchy of steps that had to take, in order to create more complex objects.

## Interdisciplinary Approach

The cross-linking lessons between Informatics and other subjects, was an effort to teach in an interdisciplinary way the teaching subjects of Informatics, Ancient Greek and History. It was decided to experiment with cross-linking lessons between STEM cognitive subjects and other subjects taught at school such as History and Ancient Greek for the following reasons a) teachers were looking for innovative approaches to increase motivation of pupils towards both STEM and classical subjects, b) students interested in classical subjects do not use scientific thinking, computational thinking, practice and methodologies based on the understanding of STEM subjects and c) students with interest in STEM subjects are less interested in classical studies, lessening their chances to develop comprehensive knowledge and literacy on issues such as history, philosophy, languages, religions, etc.

c) In the case of Informatics and Odyssey students of first grade (12-13 years old) working independently, created in a 3D design environment models of the raft, that Odysseus had constructed at Calypso's island, to return to his homeland, Ithaca. The design of the raft followed the description given by Homer in his poem "Odyssey", where the poet describes in detail the raft that Odysseus constructed to leave the island (Dougherty, 2001).

In the case of Informatics and Iliad, volunteer students of second grade (13-14 years old), separated into groups, created in a 3D design environment the "Shield of Achilles". The design of the shield followed the description given by Homer in his poem "Iliad" (Rhapsody S, verses 478-608; Myrsiades, 1987).

In the case of Informatics and History, the same students of second grade (ages 13-14), on a voluntarily basis, separated into groups, created in a 3D design environment, instruments used by Columbus during its trip to America, astrolabes, compasses, caravels etc. The design of the instruments followed descriptions given in the handbook and other digital resources.

*Figure 2. Printable raft designed by second grade students.*

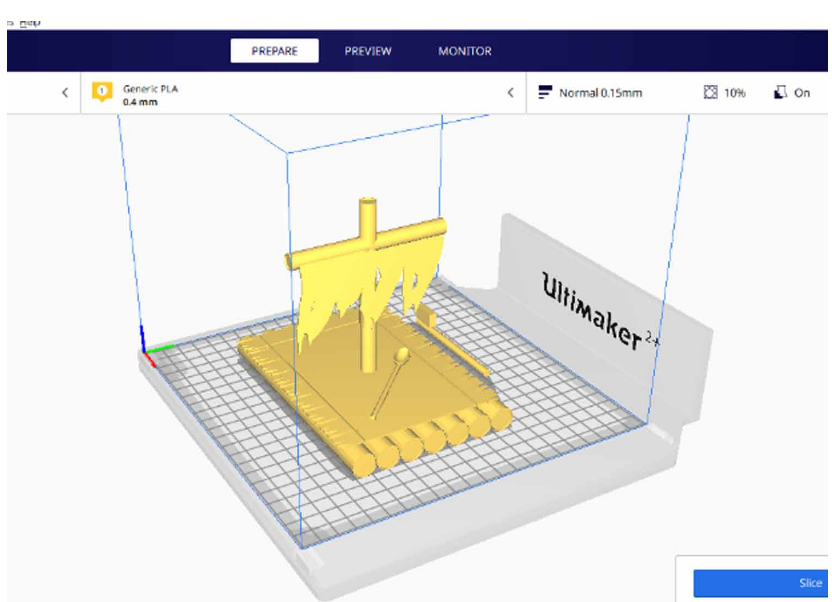
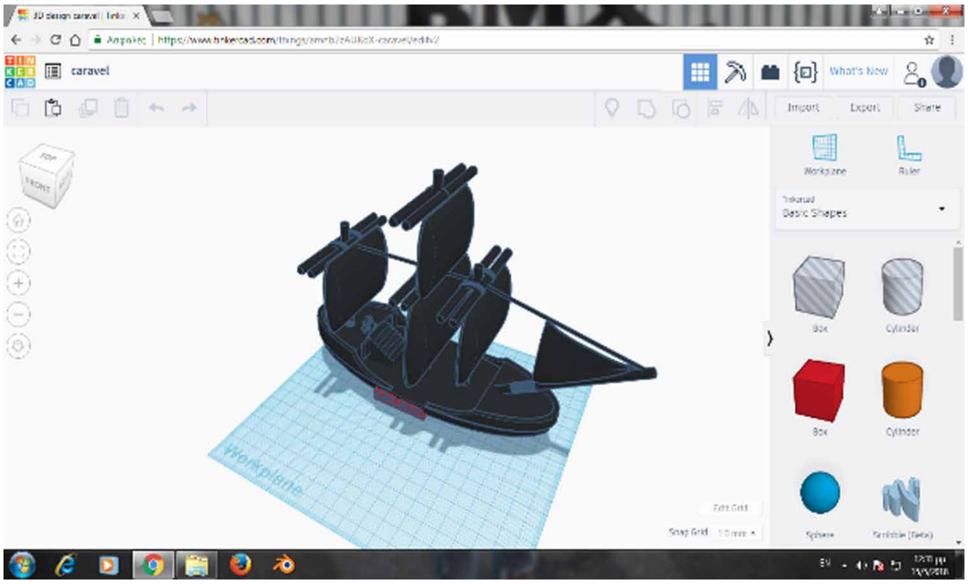


*Figure 3. Caravel designed by second grade students*



Part of these programs were implemented through the school's curriculum hours, part of the programs had to be implemented out of school hours. After the completion of the project students responded to a questionnaire prepared by the teachers in a google form format. The most important results of this questionnaire are discussed in the following.

The implementation of the program has shown that students' interest in both Informatics subject and the cognitive subjects of Ancient Greek (Odyssey and Iliad) and History, was highly increased. In particular, enhancing interest was much greater for weak students in all involved subjects Homer's Iliad and Odyssey, History and Informatics.

Teachers tried to draw attention to the dual purpose of the 3D printing activities. On one hand, students had to conceptualize the tools and objects they were studying, in order to understand their use, to design and print them. On the other hand, they had to communicate and cooperate with their fellow students and their teacher. In addition, in order to process and later combine the information from two seemingly unrelated fields, they had to think 'interdisciplinary' and 'computationally.

Students in groups, had to capture and model the navigation instruments and the ships, in 3D display (TeachThought Staff, 2015). They had to collaborate, analyze and draw their objects by breaking the initial idea into smaller more manageable parts and after by assembling them to create the complete object. They had to identify patterns they used for the construction of identical parts of the overall construction, parts of the shield, parts of the raft and the navigation instruments etc. They realized that experience gained by the previous constructions could be transferred to the future ones. Finally, every group developed his own hierarchy of steps that had to take in order to create more complex objects. As in real life they had to work as creating the real objects and face real conditions, difficulties, delays and setbacks.

As told above, the involved cognitive objects were Ancient Greek (Odyssey and Iliad), History and Informatics. The addressing grades were the first and second grade of Junior High School and the material in study came from the students' handbook and other web resources. The implementation of the projects was supported within two school hours in the Ancient Greek and History classroom and four preparatory hours and two teaching hours in the Computer Laboratory. Also, students had access to the Computer Lab for one day every week, for two months, after the school time. 10 out the 105 students of the first grade and 40 out the 103 students of second grade participated in a volunteer basis. The printer used was an Ultimaker 2+ granted for two years to the school by GFOSS (www.gfoss.eu), the Open Technologies Alliance, a non-profit organization founded in 2008 from 31 Hellenic Universities and Research Centers. For designing purposes, the 3D application used was Tinkercad. Tinkercad is a web 3D modelling tool (https://www.tinkercad.com). For printing purposes, CURA a 3D printing software was used (https://ultimaker.com/software/ultimaker-cura).

As to the cognitive objects of Odyssey, Iliad and History, realization of the lesson involves elements from the textbook and other resources. Teaching was based on students centered approach and the recruitment method of the comprehension of the learning objectives, by using the critical analysis of discourse to analyze the text. As to the cognitive subject of Informatics students had to analyze, to understand, to capture and model the objects for printing.

Students were expected to understand the type of civilization during Homer's and C. Columbus era– physical and spiritual, debate analogies and differentiations from previous, current and future civilizations.

At these point it is essential to notice that History and Ancient Language teachers, were familiar with the core concepts of computational thinking and Informatics and wanted to investigate how implementing computational thinking ideas through 3D technologies can positively influence students' performance in their own field of study.

After the completion of these projects the 40 second grade participated students were asked to respond to an online post-implementation survey, developed by teachers in a google form format. Students' answers will be discussed in the following.

## 3D Technologies Support School Programs

Having the opportunity to use 3D technologies at school, students participating to Environmental and STEM programs used 3D modelling and printing facilities to support their participation. Teams of students from the first, second and third grade (ages 12-15) used the 3D printer and 3D software to design and print objects related to their participation.

### The Ecomobility Program-Competition

ECOMOBILITY asks students to investigate problems of their local area (municipality, neighborhood, village, island) created by traffic patterns and conditions prevailing in their city. Students identify deficiencies and problems that affect the quality of life at school and the city. They think about solutions and suggestions that will greatly help their region and their school. Students during the school year observe and record problems in their city and finally propose, to local authorities, solutions to address the problems. Students can contact scientists and organizations to get informed and suggest possible solutions to municipal authorities. Many cities in Greece in the last 15 years have adopted solutions proposed by students who participated in the program. A small team of 3 first grade students, in order to support their participation in the competition decided to model and to print out a miniature of the central square of their city.

*Figure 4. Square designed by first grade students.*


*Figure 5. Rrintable square by first grade students*

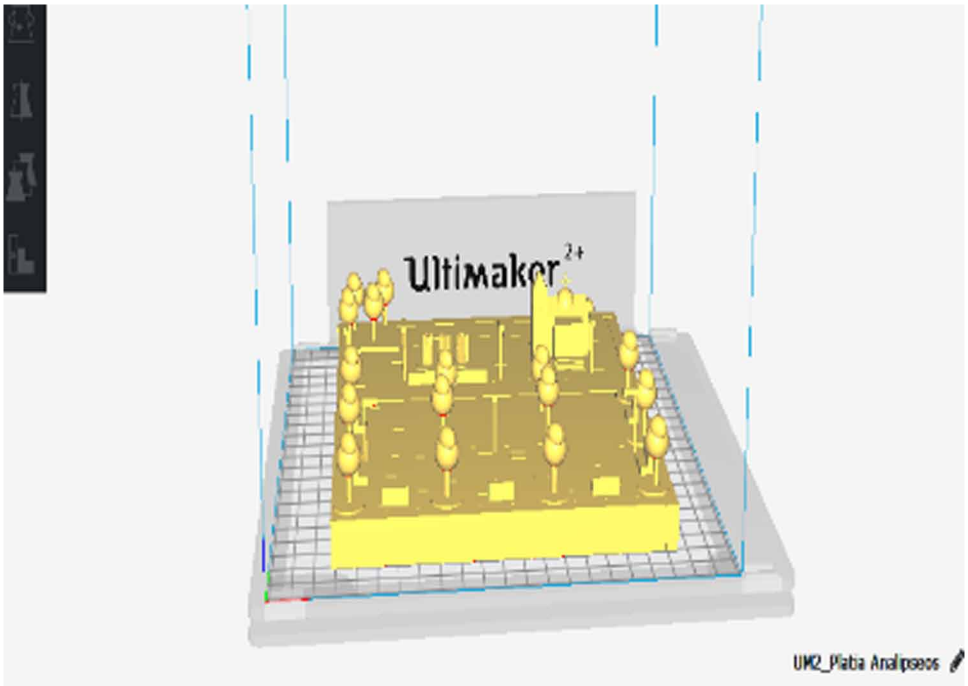


*Figure 6. Printed and painted square by first grade students*

## The "F1 in Schools" Competition

During the school years 2017-18 and 2018-19, students from all grades participated to the international competition of "Formula 1 in schools".

F1 in Schools is the largest technology competition in the world and one of the most comprehensive educational programs to enhance students' interest in STEM (Sciences, Technologies, Engineering and Mathematics).

With the "F1 in Schools" Program students are educated on the following topics:

- Physics and its various applications,
- New Information and Communication Technologies,
- Mechanical Engineering,
- Applied Mathematics.

At the same time, they understand concepts such as:

- Aerodynamics of Objects,
- Design and Production of Industrial Products,
- Materials Engineering
- Results Analysis
- Testing.

In the context of the competition students had to design and print parts of the car and then to assemble them with the main part of the car in order to build the racing cars.

Students from all grades collaborated harmonically to create their car according to the competition's restrictions. The objects that students designed and printed were wheels, wings, spoilers, axons and experimental models of the cars.

*Figure 7. Printed car designed by third grade students.*

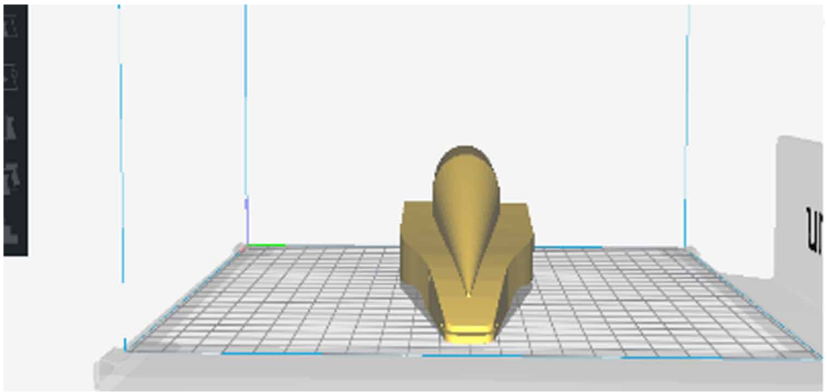


*Figure 8. Printed front spoiler designed by third grade students.*

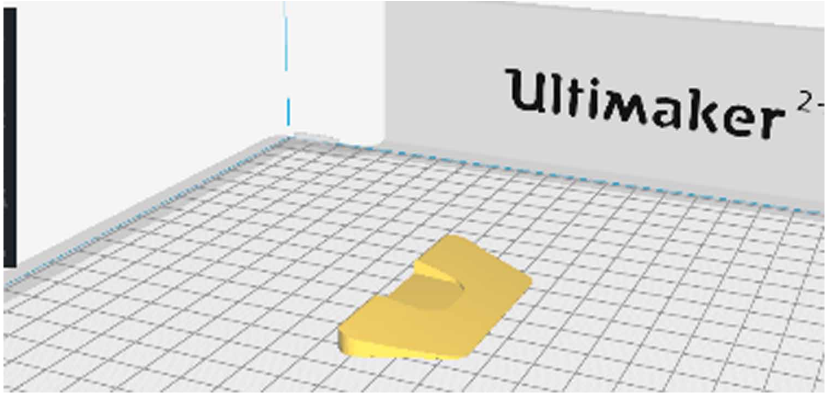



*Figure 9. Printed back spoilers designed by third grade students*

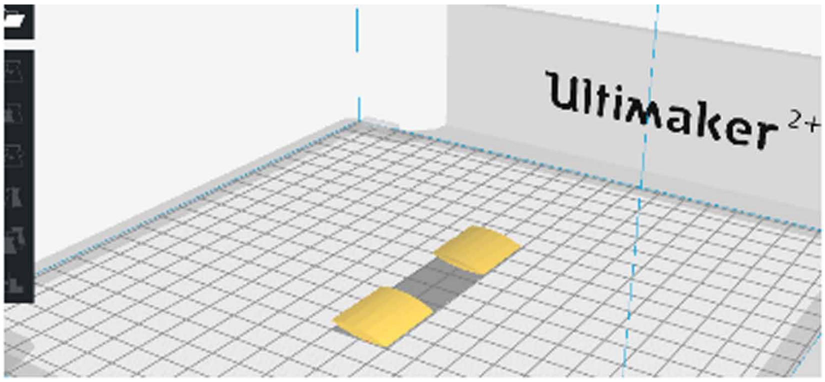
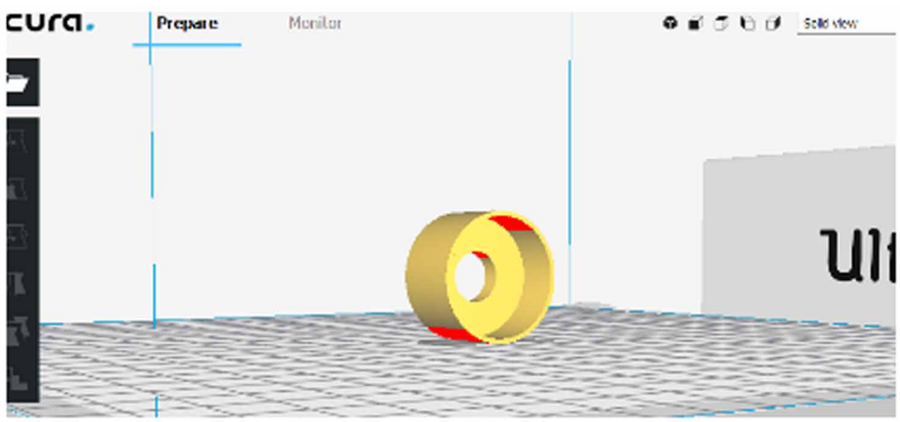


*Figure 10. Printed wheel designed by third grade students*



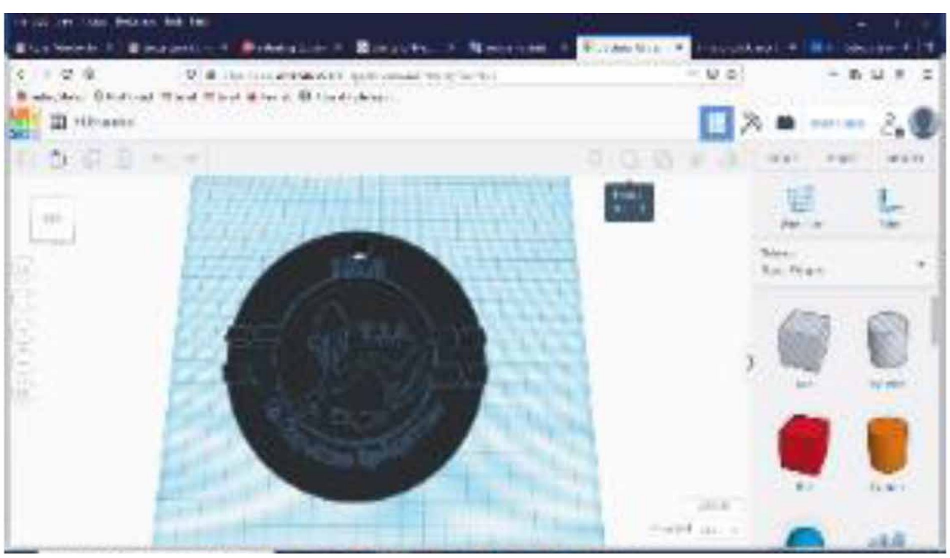

This STEM project related to 3D printing technologies revealed the connection that STEM has with the development of skills related to Computational Thinking and Computer Science. There are two approaches for STEM education integration: the content integration and the context integration. These different approaches allow teachers flexibility on how they integrate STEM in their classrooms (Psycharis S., 2018). In this content integration approach, engineering design is included by designing a prototype according to the scientific concepts and by asking questions about the material, the aerodynamic shape, and length and the slope of the wings and of the spoilers. A full understanding of a racing F1 car design also involves developing and applying physics concepts related to aerodynamics, the generated forces (resistance, friction etc.), the mathematical concepts related to geometry, rotation, and the wheel's system. Students had to deal with complex problems in designing and assembling the final cars. In many cases had to break down complex problems into smaller simpler and more manageable. During the two years, of dealing with the problem of designing and constructing a competitive car, students made connections between similar problems and gained experiences. At the same time, they had to identify important information by ignoring unrelated or irrelevant details. At the end of this trip they were confident to describe in an algorithmic way the steps to be completed, as well as the methodology, leading to the development of a competitive car, a legacy for the next group of students who would like to take part in the competition.

## Environmental Programs

Students from the third grade (14-15 yeyars old) during the school year 2017-18 participated to the environmental project "*Psiloritis: The Divine Mountain of the Coexistence of Ecosystems and Sensations*" and during the school year of 2018-19 participated to the environmental project "*Trip to Olympus, the mountain of Gods*". In both programs they had to travel to the mountain of Psiloritis-Crete and the mountain of Olympus-Thessaly, to meet other students from all around the country and to participate into meetings discussing environmental problems. In order to support their participation a small group of students decided to use 3D technologies to design and print out emblems, souvenirs, keychains etc. Apart from design and print restrictions, they had to deal with problems like converting a 2D image into a 3D object and time management by printing numerus objects in limited time.

*Figure 11. 2D designed Emplem*


*Figure 12. 3D designed Emplem.*



## LIMITATIONS

The experiment of using an interdisciplinary approach to teach Informatics and other disciplines together with the utilization of 3D printing technologies was according to students' opinion successful, interesting, fun and fruitful. The study was designed to determine the effect of the addition of 3D design and printing to the effectiveness and performance of the students. The second aim of this study was to investigate the effects of 3D technologies on the impression that the students form for the cognitive

*Figure 13. 3D printed Emblem.*


subjects that were involved. The most obvious finding to emerge from this study is that the impression of the students, for all cognitive subjects, improved significantly. This study has also found that generally there is not statistically significant difference between male and female students. However, the sample of the study was small, which indicates that this study could be repeated on a bigger, statistically more consolidated sample. Also, even if the computer lab was available during and after school, the total time dedicated to the programs was limited, especially in the context of the theoretical subjects and due to the need to follow the curriculum. Students had to spend time for retrieving information, designing and printing the objects for many weeks after the completion of the equivalent book chapter. It is obvious that this approach cannot be implemented to a large scale without drastic changes in the curriculum and the way of teaching.

As said before one source of weakness in this study, which could have affected the measurements of the Wilcoxon and Mann-Whitney U test is the number of the sample. Another weakness is the limited hours that were dedicated to the project. In addition, the performance of the students and their potential familiarity with technology is not examined or explored in any way. Although the current study is based on a small sample of participants, the findings suggest that we can strengthen the idea of the addition of the 3D design and printing to the classes of Ancient Greek, History and Informatics in the Secondary School. The present study also lays the groundwork for future research into the addition of the 3D design and printing to other cognitive subjects.

## EVALUATION

Almost 150 students participated to the 3D activities described above. All students of the first grade participated through the compulsory subject of Informatics and almost 50 students, mainly from the second grade (and a few from the third grade), participated on a voluntarily basis.

At the end of the school year students responded to questionnaires prepared by the teachers in a google form format. The questionnaire (Q1) was answered by all participating students and sought to investigate students' views, on whether they consider, that their involvement with 3D technology, during the school year, has developed their computational thinking skills.

As 3D technologies were extensively used at school during these 2 years' period, students' opinion about the cognitive subject of Informatics was also investigated, as well as the students' thoughts about their field of study after school, in order to identify a possible link between the profession students would like to pursue and their performance in digital subjects. A question about their expectations for the subject of Informatics was also launched. The questionnaire consisted of 8 Likert five-point scale questions concerning CT skills developed and student's thoughts about 3D technologies, 5 closed-ended questions about the cognitive subject of Informatics and the student's opinion for 3D technologies, 2 open-ended question about what they liked or not during their occupation with 3D technologies and 2 demographic questions concerning sex and professional orientation. 94 out 150 students responded to the questionnaire. The questionnaire in Greek can be found at http://bit.ly/2mugsxw.

A second questionnaire (Q2) prepared by the teachers, was also launched in a google form, to investigate student's thoughts about teaching and learning through an interdisciplinary approach. This questionnaire was aimed exclusively at the second's grade students who voluntarily participated in the programs of "Shield of Achilles" and "CCAP". The questionnaire consisted of 5 rating scale questions concerning student's impression, before and after the implementation of the programs, about the involved subjects (Informatics, History, Ancient Greek), 5 Likert five-point scale questions about the student's opinion concerning learning results, 3 open-ended question about the qualitative and quantitative extension of the programs and 2 demographic questions concerning sex and their class. 40 out 40 participated students responded to the questionnaire. The questionnaire in Greek can be found at http://bit.ly/2vSmMUz.

## RESULTS AND DISCUSSION

In the context of the evaluation of the program based on statistical analysis, we have to mention that the purpose of this research is to realize the impact of the 3D technologies to the students and if 3D technologies contribute to the development of Computational Thinking skills. The sample of this research were students from first grade (12-13 years old), following the curriculum of the cognitive subject of Informatics and students from second and third grade (13-15) participating in in-school and out-school programs, all using 3D technologies. The sample of 94 students is composed by 53 males and 41 females, who answered a questionnaire (Q1) after finishing their activities with 3D technologies offered at school. It is very important to mention that the questionnaire uses mainly Linkert scale (Derrick, B; White, P 2017), which is a reliable and convenient way to measure the opinions of the participants (students). The questionnaire includes a series of questions providing 5 balanced responses from which the students could choose. Therefore, the collected data are ordinal-scale and the distribution of the sample is unknown, so only non-parametric methods can be used, given the fact that using a distribution-free test lets us assume that the data that we analyze (analyzed population) does not need to follow a certain distribution or follow specific assumptions, or parameters, such as the normal distribution, which is necessary to make use of parametric tests such as t-tests or ANOVA.

Additionally, a second questionnaire (Q2) was answered only by the 40 volunteer students of second grade participating in the construction of the "shield of Achilles" and the CCAP project. Purpose of this research was mainly to mesure the impact of the 3d technologies to the students in terms of learning and teaching methodology.

The non-parametric statistics, (Corder & Foreman, 2014) are used generally on ordinal data, which are variables that have ordered categories and the distances between the categories cannot be counted or are unknown, such as "Strongly Agree", "Agree" "Neutral" "Disagree" "Strongly Disagree", or scale 1 to 5 with 1 being "Strongly Agree" and 5 "Strongly Disagree", or nominal data, such as "male" and "female".

The non-parametric tests used were Mann-Whitney U test, which could be considered an alternative of the independent t-test, and the Wilcoxon signed-rank test, which is also an alternative for the paired-samples t-test.

The evaluation of the interdisciplinary teaching of Informatics- History and Informatics - Ancient Greek, had a dual goal; On one hand to investigate if students had conceptualized tools and descriptions from the handbook and other resources, understood them, designed them and printed them and on the other hand to examine if they communicated and cooperated with their fellow students and teachers. In addition, in order to process and later combine the information from two seemingly irrelevant fields, they had to think "interdisciplinary" and "computationally". Lastly, simple analysis was used to prove that after the implementation of the 3D printing the motivation of the students rose up regarding the classes of History, Ancient Greek and Informatics.

Analysis on the questionnaire showed that the use of 3D improved the understanding of the use of 3D technologies in learning and in daily life. Furthermore, the majority of the students responded that the 3D printing activity assisted in conveying the knowledge from the field of History and Ancient Greek to Technology and vice versa. In addition, 82,5% admitted that the knowledge of the historic use of these tools and their activities using 3D technology, helped them to understand their importance in daily life. Moreover, the 77,5% of the students emphasized that the representation of the tools helped them comprehend how people were using them in the respective era. 77% of students were impressed because they understood the cultural and technological environment of the of the corresponding era. 72.5% believe that some school subjects need to be taught interdisciplinary, together with the cognitive subject of Informatics, while 17.5% of students believe that all school subjects should be taught in this way. Even if students experienced only the interdisciplinary teaching approach between Informatics and classical cognitive subjects, the majority of them believe that knowledge, concepts and skills that derive from the closely link of Informatics and STEM cognitive subjects are better developed. Finally, it should be noted that there was no significant difference between the responses of boys and girls.

Interestingly, this interdisciplinary activity revealed the development of one of the pillars of computational thinking, the cooperation between classmates and teachers. A very positive result of this questionnaire was that students admitted, in a percentage of 70%, that they coordinated harmonically with their teachers. Not to mention, that in an even greater percentage, 82,5%, they admitted that they had an effective and symphonic coordination with their classmates.

According to the graphical representation of the cooperation of the students with their classmates ($N=40$) the data in Figure 6 graph is right-skewed. The most common answer amongst the students was "Totally agree" (60%). On average, the students found the class close to Agree ($m=4.22$, $sd=1.23$). There is no evidence to support a difference between the ratings of the cooperation amongst the students between the two genders. ($U(39)= 149$, $N1=20$, $N2=20$, $p= .116$). Therefore, we do not reject the null hypothesis (H0).

*Figure 14. Cooperation with classmates*

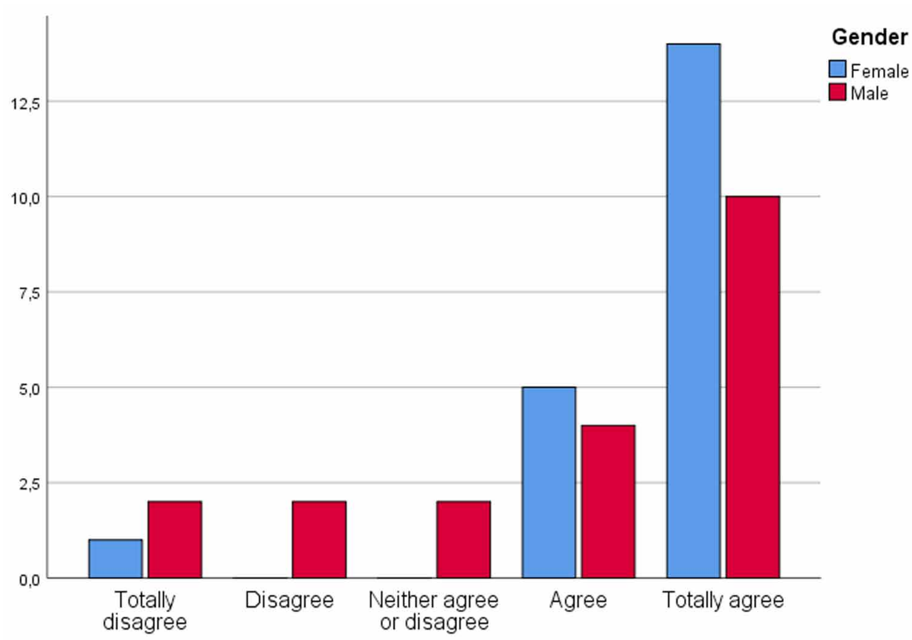
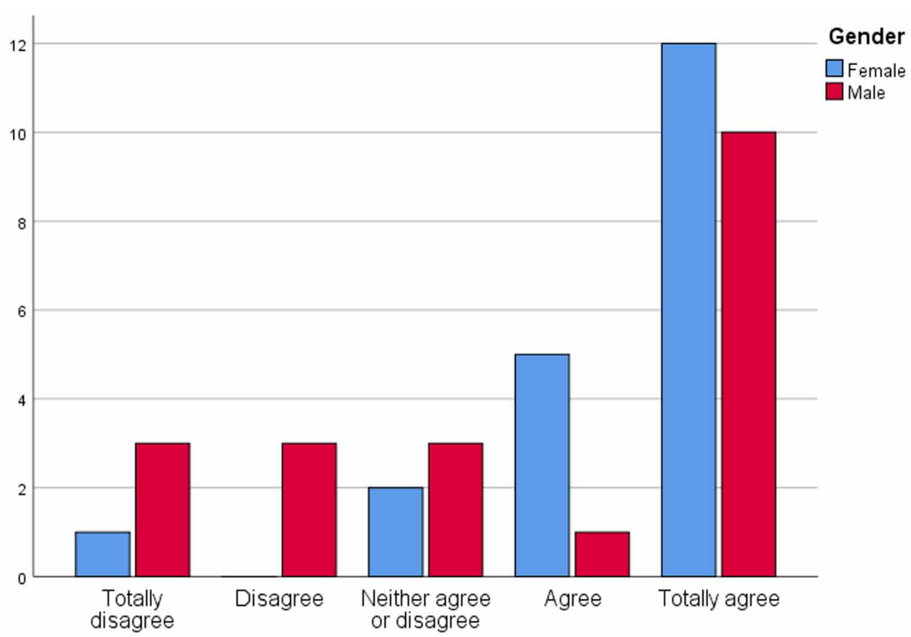



*Figure 15. Cooperation with teachers.*



According to the graphical representation of the cooperation of the students with their professors (N=40) the data in Figure 7 graph is right-skewed. The most common answer amongst the students was "Totally agree" (55%). On average, the students found the class close to Agree (m=3.98, sd=1.387). There is no evidence to support a difference between the ratings of the cooperation with the professors between the two genders. (U (39)= 156, N1=20, N2=20, p= .191). Therefore, we do not reject the null hypothesis (H0).

The questionnaire Q1, answered by the students off all grades participating in 3D activities, produced very interesting pedagogical data, which have been categorized, analyzed, and illustrated statistically and graphically. Students' perceptions of developing Computational Thinking skills, through 3D activities, were revealed through the questionnaire, while is important to examine the students' first choices on the development of a particular skill and the differences that appeared between male and female students.

Students were asked Figure 8, to answer hierarchically, which of the CT skills considered that were developed, through their occupation with 3D activities. The question was not compulsory and the students could choose from not to answer, up to all possible answers hierarchically. Males favorite choice was the development of the problem solving skill in the sense of breaking down complex problems into smaller, simpler problems. Females favorite choice was the development of the ability to think algorithmically

*Figure 16. Cooperation with teachers, Mann- Whitney test.*

**Mann-Whitney Test**

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Cooperate_with_professors | Male | 20 | 18,30 | 366,00 |
| | Female | 20 | 22,70 | 454,00 |
| | Total | 40 | | |

**Test Statistics[a]**

| | Cooperate_with_professors |
|---|---|
| Mann-Whitney U | 156,000 |
| Wilcoxon W | 366,000 |
| Z | -1,308 |
| Asymp. Sig. (2-tailed) | ,191 |
| Exact Sig. [2*(1-tailed Sig.)] | ,242[b] |

a. Grouping Variable: Gender

b. Not corrected for ties.

*Figure 17. Students' opinion in developing CT skills.*

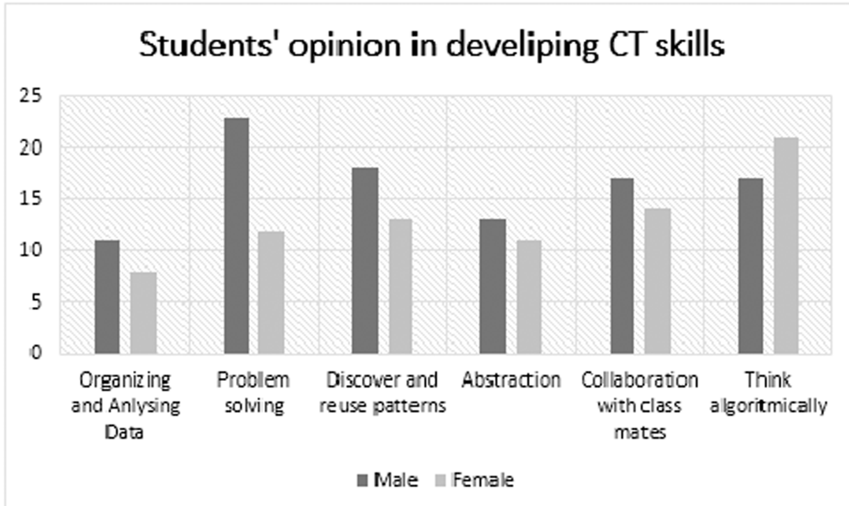
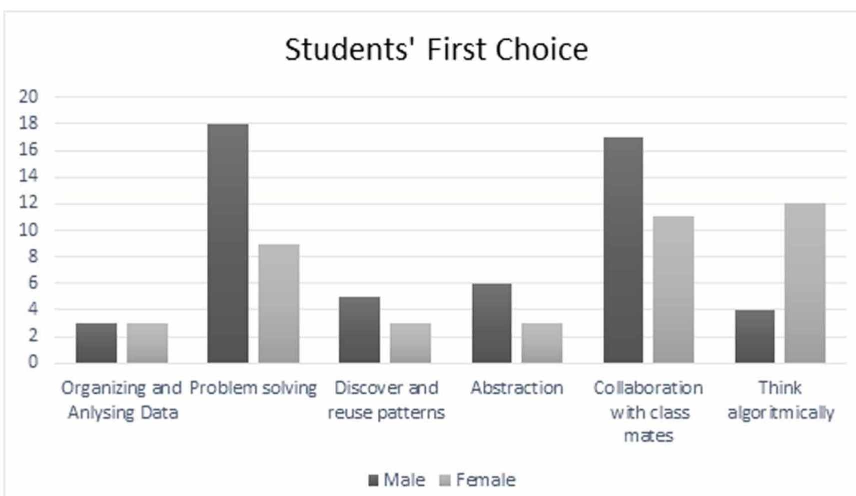



*Figure 18. Students' first choice.*




Males' first choice between the available answers was the Problem solving ability while Females considered that Thinking Algorithmically was developed through their occupation with 3D printing activities. It is important to notice that "collaboration with classmates" achieved a high score between both boys and girls.

The following, deal with some characteristics of the sample of students in terms of a) the impression they had for the cognitive subject of Informatics before and after their first year in junior high school, b) the impression they developed for 3D technologies, c) if they would like to continue working next year with 3D technologies or to deal with other areas of Informatics, d) what did the students like most, when they were involved with 3D design and e) their thoughts for future studies at university level.

*Figure 19. Students' opinion about the cognitive subject of Informatics before High School.*

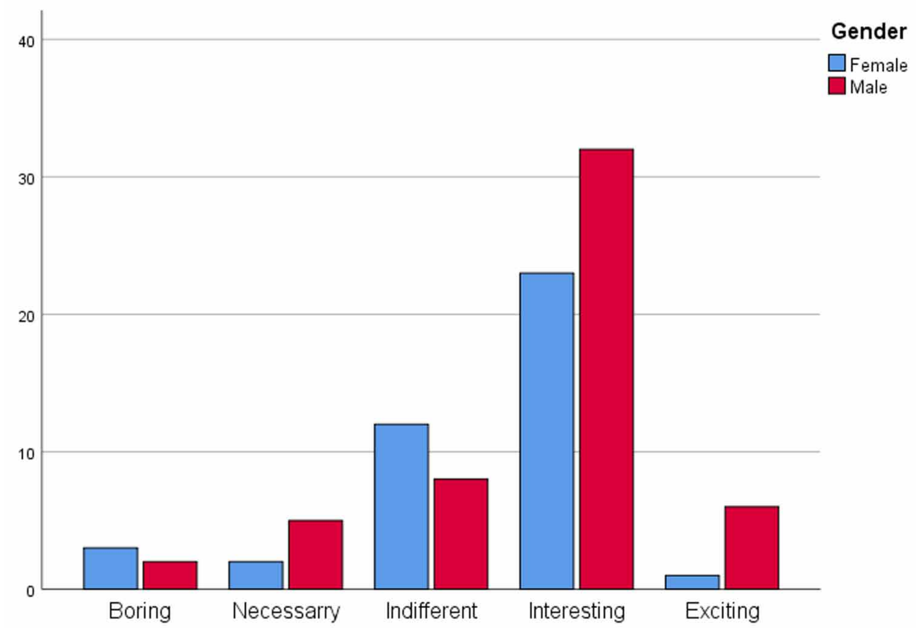



*Figure 20. Students' opinion about the cognitive subject of Informatics before High School, Mann-Whitney test.*

**Mann-Whitney Test**

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impression_IT_beforeHS | Female | 41 | 43,12 | 1768,00 |
| | Male | 53 | 50,89 | 2697,00 |
| | Total | 94 | | |

**Test Statistics[a]**

| | Impression_IT_beforeHS |
|---|---|
| Mann-Whitney U | 907,000 |
| Wilcoxon W | 1768,000 |
| Z | -1,541 |
| Asymp. Sig. (2-tailed) | ,123 |

a. Grouping Variable: Gender

According to the graphical representation of the impression of the students (N=94) regarding the Informatics cognitive subject before and after the first grade of secondary school, the data in both graphs is right-skewed. The most common answer amongst the students was "Interesting" 58.5% before, 61,7% after respectively. On average, before high school students found the subject, between indifferent and interesting (m=3.55, sd=0.935) and after one year in high school, the students found the subject interesting (m=3.74, sd=0.854).

From the Mann-Whitney test, it can be concluded that the difference in impression of the males (N=53) and the females (N=41) regarding the 3D before and after the first grade of secondary school was not statistically significant (U (93) = 907, N1=53, N2=41, p= .123) and (U (93) = 1006.5, N1=53, N2=41, p= .484) respectively, therefore we do not reject the null hypothesis (H0).

According to the graphical representation of the current impression of the students (N=94) concerning Informatics cognitive subject, the data in the following graph is also right-skewed. The most common answer amongst the students was "Interesting" (41,5%). On average, the students found the class interesting (m=3.80, sd=1,092). From this data, it can be concluded that the difference in the current impression of the males (N=53) and the females (N=41) regarding their impression for 3D technologies, we can conclude was not statistically significant (U (91) = 1041.5, N1=51, N2=41, p= .974), therefore we do not reject the null hypothesis (H0).

According to a non-parametric Spearman's Rho test the possibility that this relation is positive r=0.400 and statistically important is very high p=0.00(correlation is significant at the a=0.05 (0.01 level according to SPSS)). Therefore, we can proceed with the non-parametric Wilcoxon test, to check whether we accept or not the null hypothesis (H0) or the alternative (Ha). A Wilcoxon Signed-Ranks Test indicated that post-test ranks, the impression of the students for the subject of Informatics, were not statistically significantly higher than pre-test ranks, in other words their impression for Informatics before their first year of Secondary. (Z = -1.808, p< .071.)". However, the Wilcoxon Signed-Ranks Test is very close to the 5% limit and given the fact that it is a non-parametric test that could neglect some important factors, it would be very important to reexamine this specific aspect of this sector. Summarizing, it is not safe to reject the null hypothesis (H0).

Students were also asked to express their opinion about what they prefere to be taught during the cognitive subject of Informatics. They had to chose among 3D technologies, robotics, coding, the context of the book and other. They were also asked to express their opinion about the next years material they would like to be taught during the Informatics class. The prevailing view for the current year has been teaching and engaging with 3D technologies, while for the following year the views are shifting and the preferable subjects proposed for teaching are robotics and programming. The answers for the second year are balanced between boys and girls while Mann-Whitney test for the current year showed that there is a difference between boys and girls opinion about the teaching topic.

According to a Mann-Whitney test, it can be concluded that the difference in the preference, of parts of the processes, of the 3D application between males (N=53) and females (N=41) was statistically significant (U (93) = 832 N1=53, N2=41, p= .042), therefore we could reject the null hypothesis (H0) and probably accept the alternative hypothesis (Ha), while according to a Mann-Whitney test for the next year teaching material, it can be concluded that the difference in the preference of what should be including the next year between males (N=53) and females (N=41) was not statistically significant (U (93) = 1006.5, N1=52, N2=41, p= .524), therefore we do not reject the null hypothesis (H0).

*Figure 21. Students' opinion about the cognitive subject Informatics after one year in High School.*

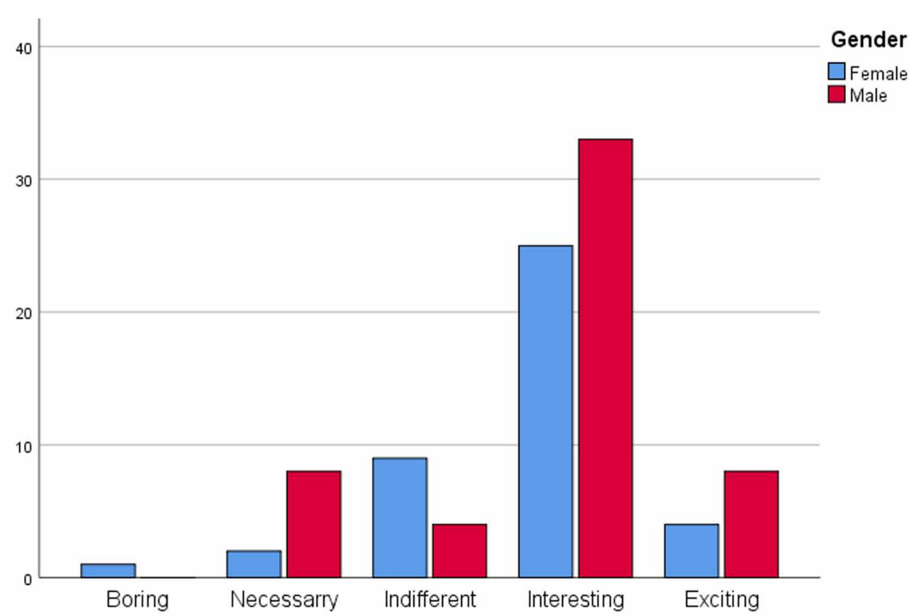



*Figure 22. Students' opinion about the cognitive subject Informatics after one year in High School, Mann-Whitney test.*

## Mann-Whitney Test

**Ranks**

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impression_IT_after_Class1 | Female | 41 | 45,55 | 1867,50 |
| | Male | 53 | 49,01 | 2597,50 |
| | Total | 94 | | |

**Test Statistics[a]**

| | Impression_IT_after_Class1 |
|---|---|
| Mann-Whitney U | 1006,500 |
| Wilcoxon W | 1867,500 |
| Z | -,700 |
| Asymp. Sig. (2-tailed) | ,484 |

a. Grouping Variable: Gender

*Figure 23. Impression of students for 3D technologies.*

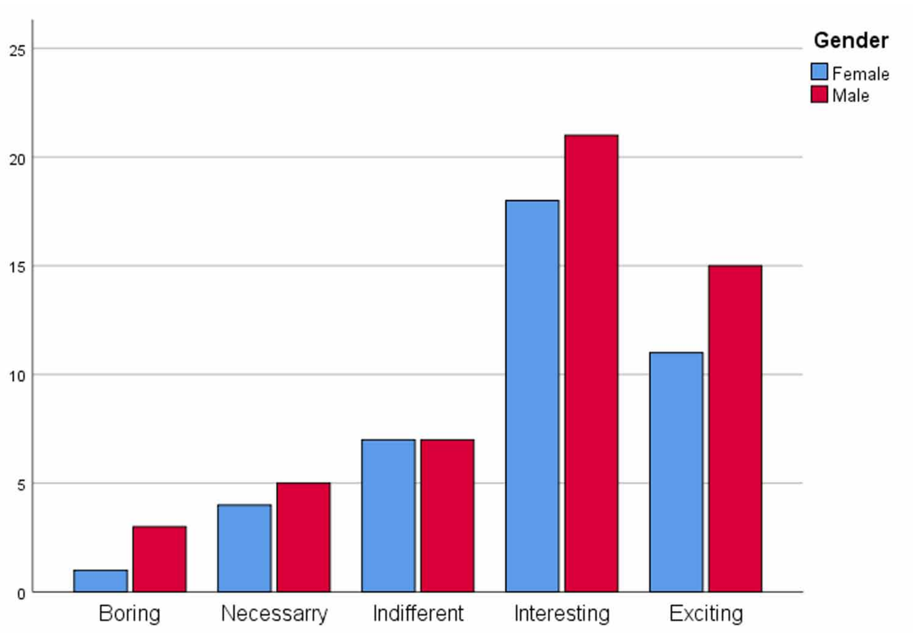



*Figure 24. Impression of students for 3D technologies, Mann-Whitney test.*

## Mann-Whitney Test

### Ranks

| | Gender | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Current_impression_3D | Female | 41 | 46,40 | 1902,50 |
| | Male | 51 | 46,58 | 2375,50 |
| | Total | 92 | | |

### Test Statistics[a]

| | Current_impression_3D |
|---|---|
| Mann-Whitney U | 1041,500 |
| Wilcoxon W | 1902,500 |
| Z | -,033 |
| Asymp. Sig. (2-tailed) | ,974 |

a. Grouping Variable: Gender

*Figure 25. Wilcoxon test on impression of Informatics subject before and after the occupation with 3D technologies.*

**Descriptive Statistics**

| | N | Mean | Std. Deviation | Minimum | Maximum | Percentiles 25th | 50th (Median) |
|---|---|---|---|---|---|---|---|
| Impression_IT_beforeHS | 94 | 3,55 | ,935 | 1 | 5 | 3,00 | 4,00 |
| Impression_IT_after_Class1 | 94 | 3,74 | ,854 | 1 | 5 | 3,00 | 4,00 |

**Wilcoxon Signed Ranks Test**

**Ranks**

| | | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| Impression_IT_after_Class1 - Impression_IT_beforeHS | Negative Ranks | 15[a] | 22,07 | 331,00 |
| | Positive Ranks | 28[b] | 21,96 | 615,00 |
| | Ties | 51[c] | | |
| | Total | 94 | | |

a. Impression_IT_after_Class1 < Impression_IT_beforeHS

b. Impression_IT_after_Class1 > Impression_IT_beforeHS

c. Impression_IT_after_Class1 = Impression_IT_beforeHS

**Test Statistics[a]**

| | Impression_IT_after_Class1 - Impression_IT_beforeHS |
|---|---|
| Z | -1,808[b] |
| Asymp. Sig. (2-tailed) | ,071 |

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

According to the graphical representation of the opinions of the students (N=94) concerning the difficulty of the use and the application of 3D technology, the data in Figure 17 is optimistic. The most common answer amongst the students was that the 3D technology seemed to be of low difficulty (37.2%) and in general only the 6.4% found it difficult. According to the equivalent Mann-Whitney Test, it can be concluded that the difference between the opinions of males (N=53) and females (N=41) was not statistically significant (U (93) = 1082, N1=53, N2=41, p= .971), therefore we do not reject the null hypothesis (H0).

According to the graphical representation of the opinions of the students (N=94) concerning the assistance of 3D to the development of computational thinking skills, the data in Figure 17a is distributed equally. According to the graphical representation of the contribution of the 3D to the discovery of new motives (N=94), the data in the corresponding graph, Image 17c are slightly right-skewed. Most of the students disagreed to its assistance. We also observe that the variety of the answers is big (27,7%, m=2.67, sd=1.223). According to the graphical representation of the impression of the students (N=94) concerning the contribution of the 3D to discern the similarities and differences into similar challenges, the data in Figure 17d graph are slightly left-skewed. The most common answer amongst the students was "Disagree" (28.7%). On average, according to the students' opinion the 3D class regarding this sector was negative (m=2.89, sd=1.205). When students asked to answer in separate questions if 3D activities develop their CT skills, problem solving, pattern recognition, abstraction and algorithmic thinking were rather closer to believe that their CT skills were not really developed through their occupation with 3D technologies. Graphs show a slight difference between boys and girls in some cases. It has also to be under consideration that the majority of the sample consisted of first grade students participating compulsorily to the 3D activities in the context of Informatics class. Although, 17% of the students believe that their problem solving skill was developed, 26,1% of the students believe that their ability to recognize patterns was improved, 33% of the students agree that their ability to move to generalizations was improved while the 34% of the students believes that they improve their ability to think algorithmically.

*Figure 26. Students preferences about the teaching topic in the context of the subject of Informatics.*

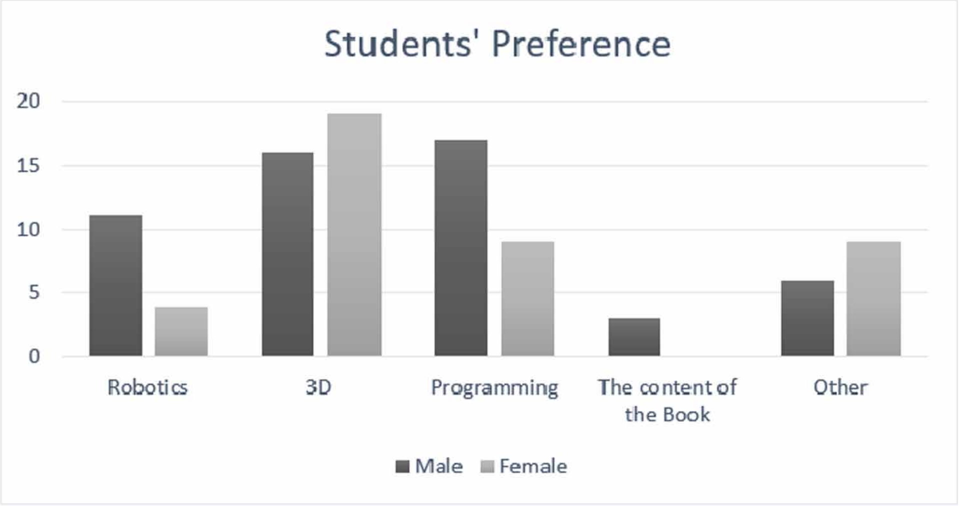



*Figure 27. Students preferences about the next year's teaching topic in the context of Informatics.*

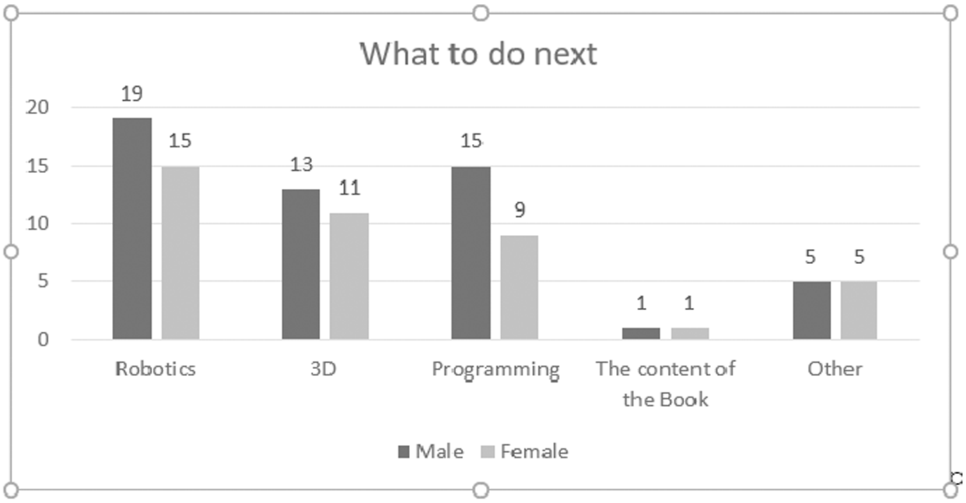


*Figure 28. Difficulty of the use and the application of 3D technology.*

**Statistics**

Opinion_of_students_on3D

| | | |
|---|---|---|
| N | Valid | 94 |
| | Missing | 0 |
| Mean | | 2,26 |
| Median | | 2,00 |
| Mode | | 2 |
| Std. Deviation | | ,903 |
| Variance | | ,816 |
| Skewness | | ,273 |
| Std. Error of Skewness | | ,249 |
| Kurtosis | | -,252 |
| Std. Error of Kurtosis | | ,493 |
| Range | | 4 |
| Minimum | | 1 |
| Maximum | | 5 |
| Sum | | 212 |

**Opinion_of_students_on3D**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | Easy | 21 | 22,3 | 22,3 | 22,3 |
| | Low Difficulty | 35 | 37,2 | 37,2 | 59,6 |
| | Normal difficulty | 32 | 34,0 | 34,0 | 93,6 |
| | High Difficulty | 5 | 5,3 | 5,3 | 98,9 |
| | Very high difficulty | 1 | 1,1 | 1,1 | 100,0 |
| | Total | 94 | 100,0 | 100,0 | |

## CONCLUSION

The aim of the projects was to support teaching of STEM and Classical subjects, co-create and implement integrated models inspired by STEM and classical disciplines and to investigate the behavior of students through a school program based in both formal and non-formal educational approaches.

Same approach can be easily applied from the first grades of primary education to the last grades of secondary education and include almost all cognitive subjects according to the students' skills to handle a computer and applications of modeling and three-dimensional design. The overall objective of engaging with 3D technologies was to increase students' proficiency in the key competences necessary to shape the future towards 2030 (Nemorin & Selwyn, 2016) by integrating STEM disciplines, social sciences, humanities, arts, and entrepreneurial disciplines (Classical disciplines) as broader educational goals are acknowledged and supported by modern educational systems.

*Figure 29. Development of the skill of algorithmic thinking.*

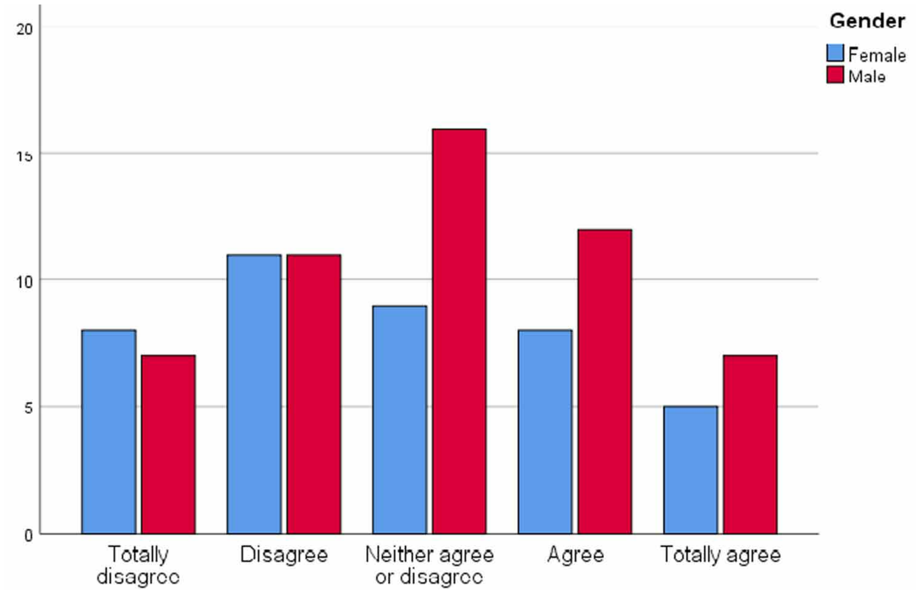



*Figure 30. Development of the skill of problem solving.*

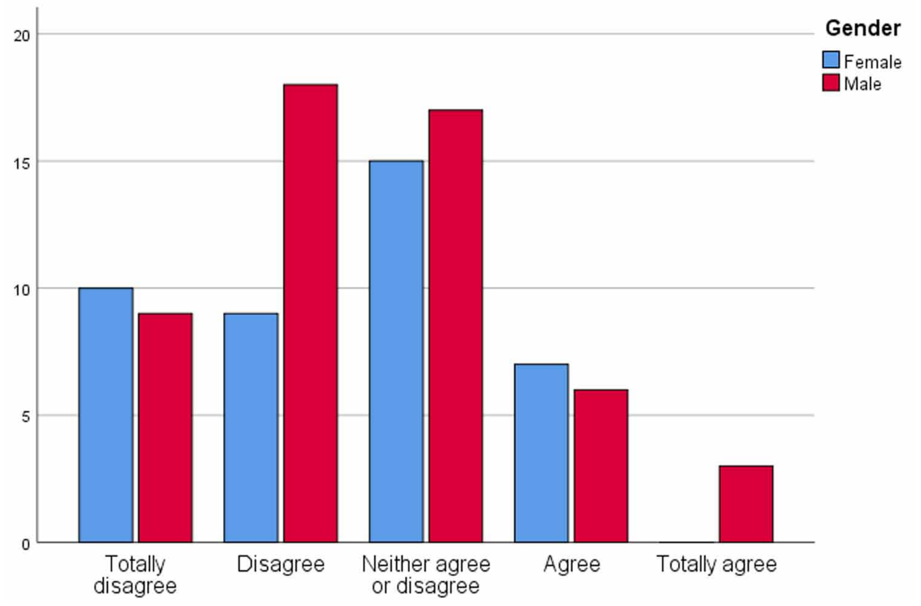


*Figure 31. Development of the skill to discover and use of motives.*

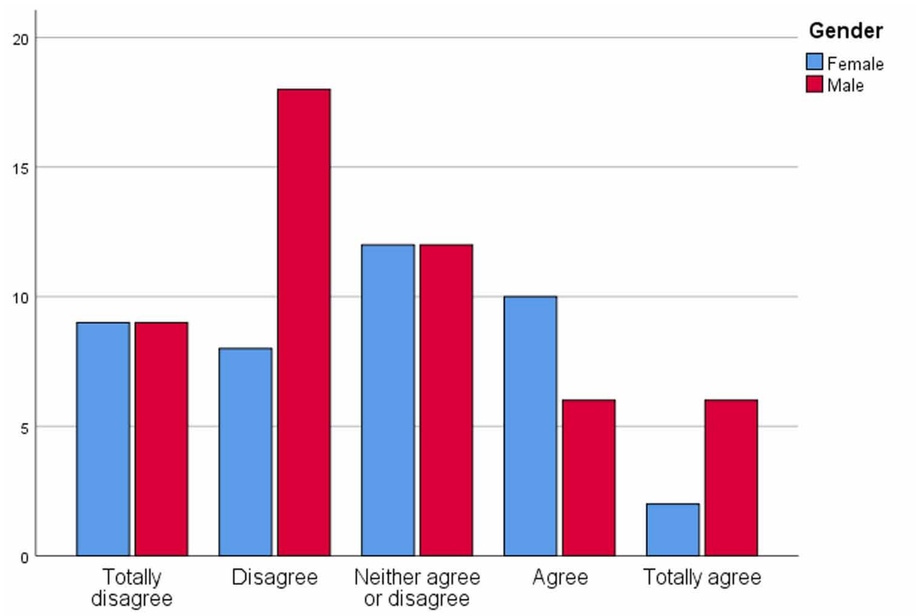
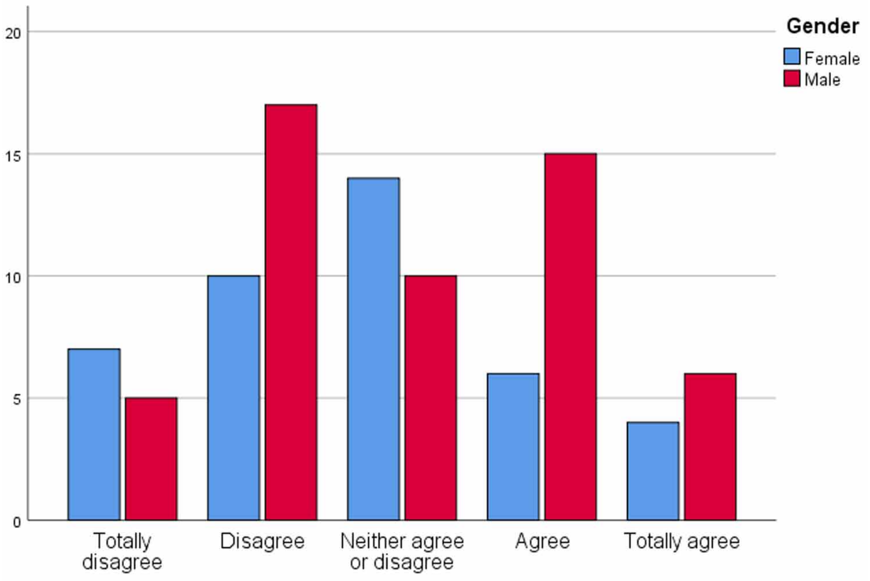



*Figure 32. Development of the skill of abstraction according to students.*



Wing J. (2008) argued that CT is a universal skill and attitude that complements thinking in mathematics and engineering with a focus on designing systems that help to solve complex problems that humans face (Wing 2008). Also Psycharis S. (2018) notices that there is a lot of discussion about the skills, attitudes, competences and practices that they should be included in CT and that there is also a discussion if CT is a problem-solving process. Also, in the literature many skills and practices claim to be part of the Computational Thinking suite skills i.e. optimization, association reuse, processing of information, symbol systems and representations, debugging. error detection, recursion etc.

The dominating word in Computational Thinking is "Thinking". There is not Computational Thinking if Thinking does not exist. Thinking and its derivatives, critical thinking, computational thinking, logical thinking, algorithmic thinking it is a necessity and the related skills have to be developed by early childhood because they are crucial for growing responsible citizens, with democratic values and eventually for giving pupils the necessary supplies for developing skills and practices needed in modern world. Teachers have to look always for strategies and exciting ways to integrate Thinking into their classrooms, have to try different technologies and pedagogic approaches in the classroom, improve students' engagement with learning, and create more active students in the context of the learning process. Three dimensional (3D) printing and scanning technologies are quite mature and of affordable cost. They can be used at school level. The use of 3D technologies is an approach where students explore, invent, discover, and engage in real problems and situations and it is turning to be an excited tool for developing CT skills among the pupils.

Students believe that the experiment of using 3D technologies was fan, successful, interesting, didactic and impressive. Through 3D designing, printing and scanning activities, they come in contact with a modern, promising technology not easily found outside school. They also believed that they worked in an innovative learning environment that helps promoting active learning among them. The experiment with 3D technologies introduced them to innovative learning through learning activities incorporating pedagogy for active learning, key competences and technology-enhanced learning. It was an environment for practice and reflection, but also meetings and discussions about the role of school and attractive ways of teaching.

## FUTURE RESEARCH DIRECTIONS

If the debate is to be moved forward, a bigger group and more time spent is necessary and a questionnaire during the 3D activities would also give us more insight. Further work is needed to fully understand if the improvement of the impression and the skills of the children regarding the classes is transient or not, if this improvement influences their performance of the children on these classes and if their Computational Thinking skills have been developed. Further research should be undertaken to explore how this action improves the relations of the children with technology before and after the experiment as well. The relation between students' thoughts for their future profession and students' CT skills could be explored, as well as the motivation of boys and girls related to their needs for developing particular skills. Another categorization is the students age and the compulsory or volunteer occupation with the taught subjects.

## ACKNOWLEDGMENT

The authors would like to express their gratitude to GFOSS (www.gfoss.eu) for providing the Ultimaker 2+ 3D printer for two school years, at the 1st High School of Vrilissia, for implementing this project.

## REFERENCES

Balanskat, A., Engelhardt, K., & Licht, A. H. (2018). Strategies to include computational thinking in school curricula in Norway and Sweden. *European Schoolnet's Study Visit. European Schoolnet.* retrieved by http://www.eun.org/documents/411753/817341/Computational_thinking_report_2018pdf/4d3d6fa0-dedd-4b62-a201-a26bf4dfd3a0

BBC. (n.d.). *Introduction to computational thinking*. Retrieved from https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education –Implications for policy and practice. *EUR*, *28295EN*. Advance online publication. doi:10.2791/792158

Corder, G. W., & Foreman, D. I. (2014). *Nonparametric Statistics: A Step-by-Step Approach*. Wiley.

Cura Tutorial. (2019). https://all3dp.com/1/cura-tutorial-software-slicer-cura-3d/

Dasgupta, C., Magana, A. J., & Vieira, C. (2019). Investigating the affordances of a CAD enabled learning environment for promoting integrated STEM learning. *Computers & Education*, *129*, 122–142. doi:10.1016/j.compedu.2018.10.014

Derrick, B., & White, P. (2017). Comparing Two Samples from an Individual Likert Question. *International Journal of Mathematics and Statistics*, *18*(3), 1–13.

Dougherty, C. (2001). *The Raft of Odysseus: The Ethnographic Imagination of Homer's Odyssey*. Oxford University Press.

Mader, A., & Dertien, E. (2016). Tinkering as a Method for Academic Teaching. *International Conference on Engineering and Product Design Education.* Aalborg University.

Myrsiades, K. (1987). *Approaches to teaching Homer's Iliad and Odyssey*. Modern Language Association of America.

Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM*, *62*(2), 32–35. doi:10.1145/3231587

Nemorin, S., & Selwyn, N. (2016). Making the best of it? Exploring the realities of 3D printing in school. *Research Papers in Education*, *32*, 1–18. doi:10.1080/02671522.2016.1225802

OECD. (2015). *Excellence and Equity in Education*. Retrieved from https://www.oecd.org/education/pisa-2015-results-volume-i-9789264266490-en.htm

OECD. (2018). *Excellence and Equity in Education*. Retrieved from https://www.oecd-ilibrary.org/education/pisa-2018-results-volume-i_5f07c754-en

Papadakis, St., & Kalogiannakis, M. (2017). *Using Gamification for Supporting an Introductory Programming Course. The Case of ClassCraft in a Secondary Education Classroom. 2nd EAI International Conference on Design, Learning & Innovation*, Heraklion, Crete, Greece.

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice Programming Environments. Scratch & App Inventor: a first comparison. *IDEE '14: Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*, 1–7. 10.1145/2643604.2643613

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.

Psycharis, S. (2018). STEAM in Education: A Literature review on the role of Computational Thinking, Engineering Epistemology and Computational Science. Computational STEAM Pedagogy (CSP). *Scientific Culture*, *4*(2), 51–72.

TeachThought Staff. (2015). *10 ways 3D printing can be used in education*. https://www.teachthought.com/technology/10-ways-3d-printing-can-be-used-in-education//

Tinkercad Tutorial. (2019). https://www.tinkercad.com

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–36. doi:10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

## ADDITIONAL READING

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). IGI Global., doi:10.4018/978-1-7998-1486-3.ch005

Kharkhurin, A. (2012). *Multilingualism and creativity (Bilingual education and bilingualism)*. Channel View Publications. doi:10.21832/9781847697967

Orfanakis, V., Papadakis, S., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital student conference platform implementation: The case study of the "research project" course. *Open Journal*, *12*(2), 5–23. doi:10.12681/jode.10871

Papadakis, S. (2018). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). IGI Global. doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In *HSci2010 7th International Conference Hands-on Science Bridging the Science and Society Gap*.

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, *10*(3), 235–250. doi:10.1504/IJTCS.2019.102760

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In S. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). IGI Global., doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Selena Nemorin, S., & Selwyn, N. (2016). Making the best of it? Exploring the realities of 3D printing in school. *Research Papers in Education*, *32*(5), 1–18.

# APPENDIX

*Table 1. Students creations and designs cc licensed*

| | |
|---|---|
| https://www.tinkercad.com/things/3a2D7vjxPpD | https://www.tinkercad.com/things/1bQIkpLZkqA |
| https://www.tinkercad.com/things/bYxV84w8B6n | https://www.tinkercad.com/things/gzg8jH20wXd |
| https://www.tinkercad.com/things/0eSZ4iwFTHz | https://www.tinkercad.com/things/3VQyAgMDoxP |
| https://www.tinkercad.com/things/e7wgMuXCH1D | https://www.tinkercad.com/things/hve0O6gDV6x |
| https://www.tinkercad.com/things/5uTvPINy6EJ | https://www.tinkercad.com/things/jmJYEIcTDgV |
| https://www.tinkercad.com/things/gVvo7NzLV80 | https://www.tinkercad.com/things/7cd1mvqYBpt |
| https://www.tinkercad.com/things/lot67cJmbHM | https://www.tinkercad.com/things/cznkHtKok8q |
| https://www.tinkercad.com/things/0JbTaEbz1aM | https://www.tinkercad.com/things/8oggtbpXNgS |
| https://www.tinkercad.com/things/5XRjpdwz5P8 | https://www.tinkercad.com/things/hAf7Xl9BKZH |
| https://www.tinkercad.com/things/lNXAVXcezXb | https://www.tinkercad.com/things/117Dr3n8bPc |
| https://www.tinkercad.com/things/cU9pPHUBe5V | https://www.tinkercad.com/things/0qhGAuR6xAH |
| https://www.tinkercad.com/things/ieo7YGFEaR2 | https://www.tinkercad.com/things/7nHDe9YByZ6 |
| https://www.tinkercad.com/things/jVf26xTrXKJ | https://www.tinkercad.com/things/aKxlP7j3eTH |
| https://www.tinkercad.com/things/3rG3hd97vbB | https://www.tinkercad.com/things/9cYuWGT8fim |
| https://www.tinkercad.com/things/ecLYb47Q5MK | https://www.tinkercad.com/things/ell2MCVm88R |
| https://www.tinkercad.com/things/bVMz7mwes1E | https://www.tinkercad.com/things/5Kx2TsHEB1u |
| https://www.tinkercad.com/things/dP7vGDR5uh7 | https://www.tinkercad.com/things/6zOx7kCLjkO |
| https://www.tinkercad.com/things/1EP1oLlaSRl | https://www.tinkercad.com/things/7myy44TB2Gp |
| https://www.tinkercad.com/things/eo5tIaM7Eo3 | https://www.tinkercard.com/things/5Kx2TsHEB1u |
| https://www.tinkercad.com/things/cDPw6eiPGEY | https://www.tinkercad.com/things/lk59xAWclsV |
| https://www.tinkercad.com/things/1CAGbpwkIg3 | https://www.tinkercad.com/things/0mzM4LPSy9G |
| https://www.tinkercad.com/things/2gplc90f5SA | https://www.tinkercad.com/things/ebThOGUK8lo |
| https://www.tinkercad.com/things/dSNpqX4SATG | https://www.tinkercad.com/things/4q1Zqm9DpwP |
| https://www.tinkercad.com/things/gtJYZKcTmkJ | https://www.tinkercad.com/things/eBMPc8OmtmH |
| https://www.tinkercad.com/things/1MIOK8SBvBS | https://www.tinkercad.com/things/5MydbxAoQ4T |
| https://www.tinkercad.com/things/d4181LItT8p | https://www.tinkercad.com/things/j3xctwYZfQD |
| https://www.tinkercad.com/things/7PoS6QkDoNc | https://www.tinkercad.com/things/jUz8huUjrZZ |
| https://www.tinkercad.com/things/4qlEoAafhaB | https://www.tinkercad.com/things/jfDcoqGRivC |
| https://www.tinkercad.com/things/dOzSOn1PSHQ | https://www.tinkercad.com/things/5V4EIeIsCwN |
| https://www.tinkercad.com/things/f1zeptK40w0 | https://www.tinkercad.com/things/3BsXdHjTckm |
| https://www.tinkercad.com/things/jgswOJxCGcD | https://www.tinkercad.com/things/2p8cjpjfKpE |
| https://www.tinkercad.com/things/cwkywkSjmKc | https://www.tinkercad.com/things/iddv5hkhITZ |
| https://www.tinkercad.com/things/cPOB4dzXidA | https://www.tinkercad.com/things/1cl5Zry5rbk |
| https://www.tinkercad.com/things/6EJYrDqqRQf | https://www.tinkercad.com/things/0mtjh07qkJg |
| https://www.tinkercad.com/things/6mdE62UE0XQ | https://www.tinkercad.com/things/j38dQeL4kLq |
| https://www.tinkercad.com/things/eehurGqHuQL | https://www.tinkercad.com/things/gIA6GACAXMC |

*continues on following page*

*Table 1. Continued*

| | |
|---|---|
| https://www.tinkercad.com/things/61WiXXrWL6m | https://www.tinkercad.com/things/hukm3v1JgK4 |
| https://www.tinkercad.com/things/eSMDW1FwtLp | https://www.tinkercad.com/things/0mtjh07qkJg |
| https://www.tinkercad.com/things/739Z2Ag5HU1 | https://www.tinkercad.com/things/edlzAmhWxsZ |
| https://www.tinkercad.com/things/lFOxuVI5Mtk | https://www.tinkercad.com/things/48SdqVDkuwj |
| https://www.tinkercad.com/things/g7mOiplCzBN | https://www.tinkercad.com/things/9uC2KRp7NRn |
| https://www.tinkercad.com/things/7OkfsGyaF9B | https://www.tinkercad.com/things/6x6jVDZSEOz |
| https://www.tinkercad.com/things/5edDpJjlqMH | https://www.tinkercad.com/things/eBZZpmRAWhO |
| https://www.tinkercad.com/things/fTWfuBY9y5Y | https://www.tinkercad.com/things/4W2wj3w01kY |

# Chapter 18

# Folk Culture and Education:

## The Role of Information Technology and Information and Communication Technologies in the Production of Digital Educational Materials

**Alexandros G. Kapaniaris**
*Democritus University of Thrace, Greece*

## ABSTRACT

*The subject matter of the chapter is the result of a doctoral thesis conducted in the Department of Preschool Education and Educational Design of the University of the Aegean in Greece. The purpose of this chapter, derived from a corresponding thesis on the issue of folk culture and education, is to explore the role of information technology and information and communication technologies (ICT) in the production of digital educational material. In essence, the work comes to contribute to the scientific debate on whether technology can enhance the relationship between folk culture and education through interactive-multimedia and online technologies. Simultaneously, this project also aspires to contribute to the configuration of the instruction of folk culture through enriched teaching interventions by analog and digital means.*

## INTRODUCTION

This paper presents the implementation of a research applied, for the first time in Greece, as far as we know in the field of folk culture and education, the experimental research with intervention groups (IGs) and control groups (CGs) on a fairly large sample of students (eight Elementary School classes) attempting to compare the effectiveness of digital enriched teaching to the analog enriched teaching. Thus, useful conclusions will be exported about whether multiple media (analog and digital) along with appropriate teaching design, can create enriched teaching situations, in which students will be able to search the context for learning experiences by combining authentic situations and learning within groups.

DOI: 10.4018/978-1-7998-4576-8.ch018

It is a research method, which mainly uses the quantitative but also the qualitative method (observation and open discussion). Regarding the sample selection criteria, Fourth Grade with two separate classes were preferred, while location was also another criterion, schools in urban, semi-urban and rural areas were preferred.

Information and Communication Technologies (ICTs), with multimedia and web applications considered as the spearhead, have years developed a new interactive way of communication (Kalogiannakis & Papadakis, 2007). New digital forms of everyday expression on the Internet are daily being shaped by online behaviors, where women and men can easily use text, video and audio in their informal on-line chat, creating thus new forms of language (Kalogiannakis & Papadakis, 2008; Papadakis, Tousia, & Polychronaki, 2018). Thus, from texting (producing and sending written messages), we were eventually led to viding (making video with the use of montage) (Howard 2016: 50). In recent years communication has evolved and conversations are taking place with the use of avatars in 3D computer-generated environments in which one can be immersed (e.g. second life) (Lanier, Minsky, Fisher & Druin, 1989). Simultaneously, new forms of communication with augmented reality applications are initiated (Vidakis et al., 2019). Such applications integrate the real world, as perceived by the user or through his / her senses, information and material, mainly visual and audio, created by a computer unit (Mustakas, Paliokas, Tsakiris & Tsovaras 2015: 215).

One of the reasons that would facilitate the selection of ebooks in the future, especially in the educational field, is to avoid problems regarding the form and often the incomplete content of a printed text. The disadvantages of a conventional book are: a) rough layout, b) a variety of terms that require the recourse to various dictionaries or glossaries for photographs, drawings and paintings which are located in other pages, c) recourse to references at the end of chapters, d) recourse to references on other books and e) recourse to references of internet links that cannot be directly accessed. The enriched eBooks come to eliminate all the above with a variety of tools that increase the functionality and interactivity of reading. Indicatively some of the characteristics of enriched eBooks are associated respectively with e-book devices:

- reading and automatic turning of pages
- touch or mouse selection of photos that are simultaneously magnified
- transfer to a specific page
- content's table view
- multimedia content (video, video from You tube, sound, recorded speech, animation, illustration)
- internal and external links
- full access to embedded Internet browser to navigate on sites related to the book and the work of the student, simultaneously with reading
- drafting short essays using a rich text editor that can be displayed next to the open eBook
- appearance of enriched material without the student having to move from the page that he/she reads (Shadowbox),
- use of bookmarks, attachments, markings, and generally many interventions that students can make up on the printed books
- search by keywords
- change font or font size
- change page color, text and luminance
- horizontal and vertical display

- activation or deactivation of alignment and hyphenation (Gasouka, Kapaniaris, Arvanitidou, Foulidi & Raptou, 2013:4177-4178)

Besides, we must not forget that the world of technology essentially involves devices and machines penetrating into everyday life. Nevertheless, the concept also includes social changes, such as increased population mobility and bureaucratic administration and networking (Bausinger, 2009: 5).

Nowadays technology, having prevailed as a total event, extends to all human activities, both practical and instrumental, as well as the most personal and spiritual (Kalogiannakis et al., 2009). The internet as a classic case of technology spreading to potential or virtual1 worlds, which tend to gradually replace the real-world's primacy, has brought about cultural upheavals and shifts (Skarpelos, 1999: 22.61).

Consequently, the internet, as a technology carrier of a new emerging digital culture, will affect both the space of popular culture2 and education over time. The new digital culture formed in the Internet and the importance of teaching folk culture in education leads to the utilization of all digital media in the field of folklore. According to Gasouka & Foulidi (2012) relationship between folklore, Internet and related digital media is of great interest. Specifically, the folkloric dimension of «Web 2.0» and related digital media helps in regeneration of folk culture (habits, beliefs, myths, tales, practices, attitudes, norms, etc.) (Kapaniaris, Gasouka, Zisiadis, Papadimitriou & Kalogirou, 2013a: 107)

Attempting to define the Internet space as a medium for the production of digital artifacts,3 learning objects 4or more general as a space for the production and creation of information 5and applications in the field of culture and education, one would conclude that the Internet is a subset of Information and Communication Technologies (ICT) (Orfanakis, Papadakis, Kalogiannakis, Ampartzaki, & Vassilakis, 2016). The Internet, therefore, based on information technologies and related information organization structures, is used for all kinds of communication and activities, while being a useful tool in the field of education and popular culture.

Thus, when referring to ICTs and the internet, it has already been made clear that ICT is a broader set that includes the internet. Of course, a huge volume of ICT work and activities is devoted to the internet and its applications. It is reasonable therefore, for an extensive chapter to be devoted to exploring and understanding the role of the internet, in relation to content in general, popular culture and education.

Xanthippis Foulidis' innovative doctoral thesis on "New horizons in Greek Folklore Studies: The internet as a field of production, dissemination and educational exploitation of Folk Culture" conducted at the University of the Aegean - Department of Preschool Sciences and Educational Design, clearly defines the relationship of popular culture with the Internet and in particular the science of folklore.

According to Foulidis (2015: xii), the relationship between folklore and the internet is not a competitive one as "… folklore is neither shrunk by technology nor threatened by the internet, but is instead enriched and constantly renewed, creating a new field called digital folklore". Moreover, regarding the subject of digital folklore Foulidis (2015: 51) argues that "… the subject of digital folklore studies is the variety of online folk groups, electronic folklore groups, e-groups and communities, their characteristics, their reason, their habits, their motifs urging them to participate in these communities, their laws, their beliefs."

Then, our guides in exploring the role of the internet in relation to popular culture, will be the two facets of the internet. First of all, the first facet is the 'information provider' and the ability to use applications, which gives the user a more passive role as a 'consumer' and on the other hand, there is the author - producer and creator of information and web-based applications, which allows the user a more energetic and constructive role.

At the same time, following the path of discovery and delineation of the Internet as a permanent and contemporary phenomenon in the field of education and popular culture, we will be critical towards the view it presents it as a 'treasure of information' (Dreyfus, 2003: 11).

Certainly, the relationship between folklore and the internet is of great interest. This is also confirmed by Gasoukas & Foulidis (2012: 104), who state that: "… popular culture is regenerated through the internet in some way and its habits, convictions, beliefs, myths, narratives, practices, behaviors, rules, etc., find a way out on the web, in an online language and practice interaction. "

Dundes6' (2005: 406) view in Gasoukas & Fulidis (2012) is also indicative: "… if folklore continues to be alive and growing in the modern world, it is partly due to its increasing transmission through email and the Internet".

# ENRICHED TEACHING INTERVENTIONS

## Introduction

The educational exploitation of ICTs remains an on ongoing challenge for education throughout the years. Simultaneously, an ongoing question for the educational community is the effectiveness of new technologies as auxiliary to teaching strategies developed in educational scenarios. This reflection even extends when comparing an enriched digital teaching to an analog enriched teaching. Thus, questions are raised, such as whether a teaching approach can be equally effective and pleasant for students without the use of ICT tools, in relation to a teaching intervention using other analog enriched techniques.

Learning objects, when used as part of a digital enrichment process, are called digital enrichment learning resources. Digital enrichment resources when combined with the proper pedagogic/teaching plan form the basic assets of the digital enrichment of electronic books. Therefore, it is important to analyse the learning object concept and also to define a learning object taxonomy (Kapaniaris et al, 2013a:105)

In cases where both techniques (enriched digital techniques and analog enriched teaching) lead to the approximately same result, it may legitimately be wondered whether a student will gain more using either one of these techniques. In addition to the obvious learning benefit which is first and foremost measured in the educational process, there are a number of issues to consider:

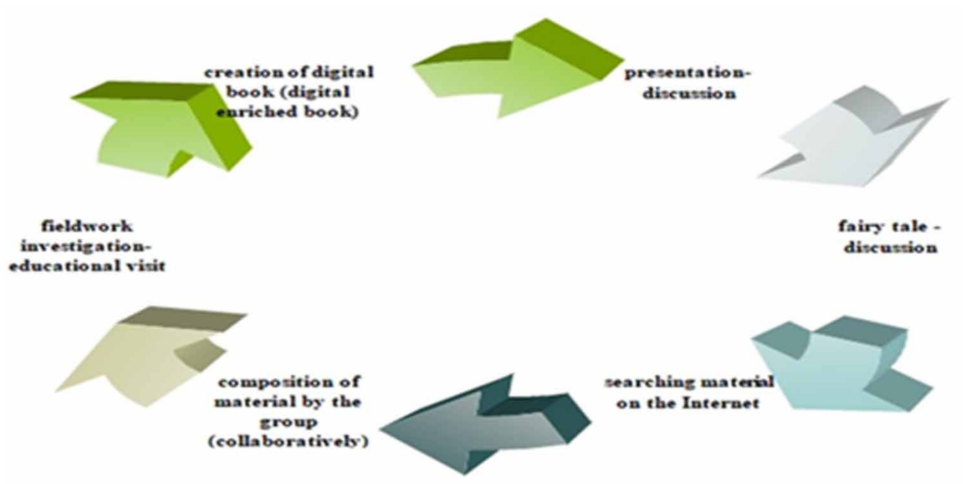
a) The preparation of the learning material (learning resources) which should be created by teachers. Respectively this material is related to the speed, efficiency, scalability and reuse of the material,
b) The diversity and multiple representations of the learning material that can be offered to students from multiple perspectives so as to avoid saturation and fatigue.
c) The link between modern day representations and learning practices in an era dominated by image, video and speed,
d) The time and cost needed for representations, visits, observations of phenomena and situations that can be done either digitally or analogously.
e) The interactivity of teaching in relation to the means used for teaching (digital - analog) and,
f) The learning theory on which the teaching intervention (constructive, behaviorism) is based and how it is successfully achieved through activities.

Digital reading can be exploited in folklore education through simple and complex learning object development. Learning object development provides the raw material for the digital enrichment process of e-books on folk culture in HTML or any other format (Kapaniaris, Gasouka, Zisiadis, Papadimitriou & Kalogirou, 2013b: 317)

## Digital Enriched Teaching: Designing a Teaching Proposal

A digital enriched teaching in the field of folk material could use, in addition to tools and strategies, tools of analog enriched teaching (authentic learning frameworks, i.e. educational visit and fieldwork research), and purely digital media such as searching information on the internet, wi-book writing tools, and collaborative learning tools (wiki).

*Figure 1. The cycle of creating an enriched digital teaching using an educational scenario*




This teaching proposal, dealing with digital enriched teaching (using ICT), includes the following interventions: a) presentation - discussion, b) folk tale - discussion, c) search for material on the internet, d) fieldwork research - educational visit, e) group composition via wiki (collaboratively), f) digital book creation (digitally enriched book, containing images photos, text, videos, hyperlinks, software tool painting etc.). The ultimate goal is to teach the learning object through activities, research, visits, investigations, material compositions, collaborative text production online and the final deliverable is the creation of artifacts (digital enriched e-book).

The selection of the above means was made in order for the students to: a) develop critical thinking through engaging in activities and producing written text (language), b) develop synthetic skills through the construction of the enriched digital book, and be able to handle multimodal text, c) create the final digital deliverable material (text summaries, photo selection, links, videos), which was initially uploaded to their group's wiki and then further refined so as to produce their final enriched digital book, d) reconstruct ideas and images for concepts through the cognitive conflict that can result from different ways of representing concepts using exploratory material search tools (articles, tributes, wikipedia, online journals, digital museum material) or through chat via e-mail as well as within the classroom (group - plenary), e) working collaboratively, remotely and face-to-face, through the process of material selection and production.

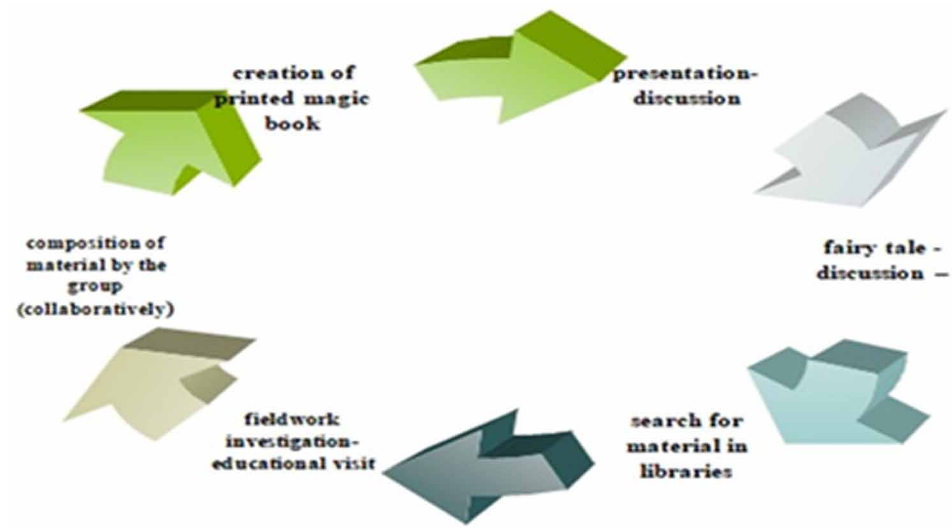
Moreover, iBooks offer the opportunity to comprehend complex concepts bringing together mobile technologies with folk culture areas and concepts. It is quite impressive how students can use (multi) touch technology enhancing the museum experience with learning through gaming and social media. Children interaction with archived material through publications-reading applications enhances the dialectical process not only in the classroom but also in cultural sites by combining digital gaming with raw material (Kapaniaris et al, 2013b:320).

## Analog Enriched Teaching: Designing a Teaching Proposal

An analog enriched teaching can use authentic learning frameworks (educational visit and fieldwork research), a folk tale with magical elements, searching for materials in libraries and creating analog artifacts (collage, painting, handcrafts, etc.).

This teaching proposal includes the following interventions: a) presentation - discussion, b) folk tale - discussion, c) search for printed material in the libraries, d) fieldwork research - educational visit, e) group composition via wiki (collaboratively), f) magical book creation (printed book enriched with pictures, photos, text, painting, stickers, colors, etc.).

*Figure 2. The cycle of creating an enriched analog teaching using an educational scenario*




The selection of the above means was made in order for the students to: a) develop critical thinking through engaging in activities and producing written text (language), b) develop synthetic skills through the construction of the enriches magical book, as and be able to handle multimodal text, c) to be activated and engaged in a group learning process with discussion, reflection and exchange of views and ideas, to complete both a collaborative modular text and the creation of an enriched "magic book", d) reconstruct ideas and images for concepts through the cognitive conflict that can result from different ways of representing concepts using exploratory material search tools (book magazines folk tales) or through discussion within the classroom (group - plenary), e) working collaboratively, through the process of material selection and production (enriched printed book - “magical book”).

### Comparison of the two Different Teaching Approaches

Examining the two different teaching approaches, that is, digital enriched teaching and analog enriched teaching, we can argue on three axes: (a) the cognitive coverage of the subject, (b) the ease or difficulty in implementing them, and c) the stimulation of the students’ interest.

Regarding the cognitive coverage of the subject to be taught (learning object) through digital enriched teaching, it is envisaged that students will: a) develop critical thinking by engaging in activities, b) develop synthetic skills while constructing the enriched digital book, c) to be activated and engaged in a group learning process with discussion, reflection and exchange of views and ideas, to complete both a collaborative text and the creation of an enriched book, d) reconstruct ideas and images, due to the cognitive conflict that can arise from the different way of representing concepts, using exploratory searches on the Internet and through the classroom discussion as well (group - plenary), e) work collaboratively through the process of selecting and producing material (collaborative text, enriched book).

As mentioned above, analog enriched teaching will achieve cognitive coverage and will be as important an effort as digital enriched teaching. This assumption will eventually be confirmed or reversed by the presentation of the results emerged from the present research.

In addition to whether the two teaching interventions have the same or different effect on the subject’s cognitive coverage and whether the learning process has ultimately been strengthened, there are also student-related issues that need to be taken into account. These issues are related to:

- what the students like,
- the diversity and multiple representations of learning material,
- the audiovisual culture of today’s students of our generation (digital generation),
- the interactivity of the teaching and the interest it generates,
- the time, the cost and the reusable digital training material.

Thus, based on the above analysis, it was initially estimated that digitally enriched teaching might have had a lead in relation to the interest the students would show due to the use of new technologies. However, this assumption was yet to be confirmed with the results of the research presented in a following chapter.

## LEARNING OBJECTIVES & DIGITAL ENHANCEMENT IN THE FIELD OF POPULAR CULTURE

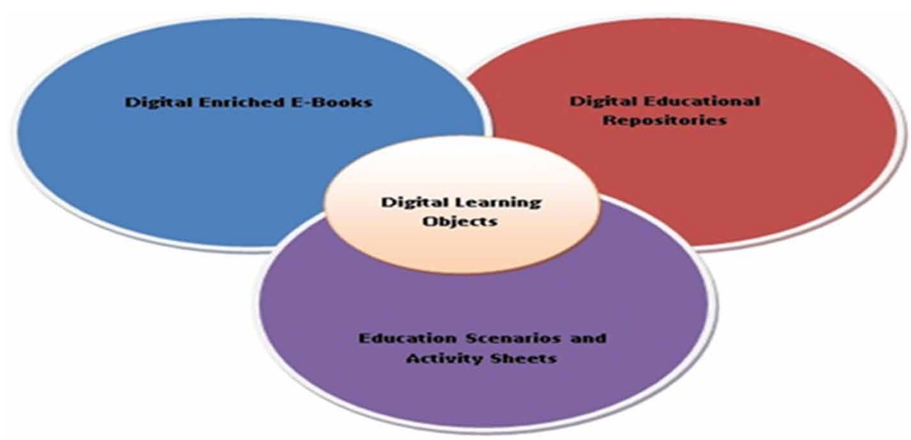
### The Learning Objects in the Field of Popular Culture

A key element of digital teaching enrichment is learning objects (simple and complex). Based on digital learning objects, learning experiences can be designed, embedded in an educational scenario. Folk culture in all its forms, as well as all folklore as a whole, can constitute the ‘core material’, that is, content that can often be transformed into a learning experience when integrated into an educational scenario and corresponding activity sheets. Also, material from a potential student folklore survey (folklore) in the context of a research project may provide evidence that on their own (fragmented) might not have the same educational value as they would have if presented as integrated learning objects. The collection, distribution and editing of photographs, sounds, folk songs, narratives, fairy tales, riddles, folk paintings, folk art can be possible with appropriate digitization techniques to “bring to life” a story, a legend, a memory, an event and be used as a basic material for the creation of simple and complex learning objects. (Kapaniaris & Gassoukas 2016).

It should be underlined that archival material and popular culture collections, wherever they come from (public bodies, individuals), are also an inexhaustible source for the creation of digital educational material, if modified, for use in teaching interventions. The role of information technology and ICT is proving crucial in the new digital information society. Especially after the digitization of many Folklore Archives in Greece, from the relevant “information society program”, it is possible to create educational learning objects for both students (simple learning objects) and teachers (complex learning objects) (Kapaniaris 2013b).

The role of the internet is also important, especially web 2.0 applications, because rare material can be transformed into multiple learning aids. We must not forget that the internet is a vast pool of folklore material, practically serving as a bridge between the popular culture, education and popular culture (museums, exhibition centers, private collections, etc.) (Kapaniaris 2013b).

*Figure 3. Uses of digital learning objects (Kapaniaris 2013b).*




In conclusion, digital learning objects (simple and complex) can be used to produce digitally enriched books, be uploaded to digital educational repositories, used as stand-alone educational resources, or integrated into activities through educational scenarios. However, in any case, the teaching of popular culture in education can be greatly enhanced by learning experiences based on digital learning objects. Additionally, the link between on-site folklore research and research work in Primary Education in the context of the Flexible Zone or cultural programs (school activities) can be upgraded. Nevertheless, we must not forget that technology alone is not enough to create neither an authentic learning framework nor an integrated learning experience. First and foremost, appropriate pedagogical design based on a corresponding learning theory incorporating ICT, as well as the development of appropriate teaching material must be implemented (Kapaniaris 2013b).

### The Exploitation of the Enriched Digital Books in the Teaching of Folk Culture

Enriched digital books can offer learning experiences through simple (learning asset) and complex learning resources as long as, they are integrated into a comprehensive pedagogical design based on the lesson scenario and corresponding activity sheets. In fact, enriched books are the means to create learning experiences for students and perhaps this is the innovation they can offer to education (Kapaniaris, 2013a). It should be underlined moreover, that the learning experience is not just the production of an activity using a computer system, with the aim of making the activity effective, easily digestible and flexible, giving the learner satisfaction. It has to be useful in the lives of students as well, and that is the true essence of the process (Thomas et al. 2012).

Thus, utilizing enriched books and reading equipment in education is not a simple process. It requires punctual preparation and planning in advance on behalf of the teacher and the school community. In more detail, it requires careful consideration of the factors that determine the content development process, the readability, the exploration provided by digital resources, and the overall interactivity that digital enriched books can provide in the educational process. (Kapaniaris, Gasoukas 2016). Besides, the use of enriched textbooks and correspondingly new modern multimedia gadgets, which often comes as a train (trend), is not always a reason for the teacher to include them in the educational process (Kapaniaris 2013a).

## THE RESEARCH

### The Idea of the Research

The impetus for the implementation of the research emerged from the question such as whether combining interactive digital enrichment technologies, with the addition of other tools and teaching strategies (visits and research on real sources), could be proven more effective than analog enriched teaching in the field of popular culture.

Thus, he result was the idea to explore whether enriched digital book writing tools are really a vehicle for teaching popular culture, considering that combining technology with other learning tools and strategies leads to an active, experiential approach to teaching.

Our long experience combined with various indications, have led us to the assessment that the practical results of such an approach would confirm the importance of the digital enrichment model, because: a) the field of popular culture is appropriate for such teaching interventions, since it comprises images, sounds, narration, etc. or access to authentic sources and original material, b) new multimedia internet technologies offer many possibilities for experimentation, collaboration and integration of other digital tools and resources of the internet, c) students seek both diversity and digital technologies, as they are quite familiar with them from young ages (digital natives), d) the possibility for digital sharing of digital artifacts creates another positive framework and e) the possibility of extending and restructuring digital artifacts at different times is offered. However, we must note from the beginning, that the results of the research we present below, partially disprove this view, or expectation as others would suggest, providing an interesting perspective drawn from a fairly large sample of Magnesia students.

The research highlights the field of teaching of popular culture through the use of ICT, which to this day has remained relatively unexplored. It focuses on digitally enriched teaching interventions compared to teaching interventions which do not implement ICT, yet they are enriched with other analog tools and strategies. with other analog tools and strategies. Given the fact that up to now ICT researches that have been conducted were merely focused on the intervention group (with ICT) and the control group (without ICT), the result was being significantly preconceived and several times, converging towards the intervention group (with ICT), the present study avoided this methodological framework by comparing enriched teaching interventions with or without ICT. It is worth mentioning, however, that in the meager, perhaps anyway, field of popular culture teaching, we do not encounter surveys on enriched digital teachings compared to corresponding enriched analog teachings. It is extremely difficult, in general, to find research on folk culture and ICT, within integrated teaching interventions with an educational scenarios and corresponding activity sheets.

Consequently, the research tried to shed light upon an area that has not been sufficiently studied in Greek and international literature, with the exception of some related researches, which focus on teaching interventions with or without the use of ICT, mainly in literary courses (e.g. history, literature, modern Greek etc.). These researches were presented in the theoretical context of our study. Thus, this research is, to a certain extent, an innovation and contributes both to the promotion of the science of folklore and to the link between popular culture and education and ICT more generally.

## Purpose of the Research

The purpose of this research is to understand and interpret whether the use of ICTs (wiki, internet, enriched digital books, etc.) can increase the efficiency and attractiveness of popular culture-based teaching interventions without ICT tools, in relation to other enriched analogue processes and tools.

## Aim of the Research

The specific aims of the research were to investigate whether students after the intervention: a) understood the concept of popular culture, b) what forms of art and entertainment are associated with popular culture, c) how they perceive the concept of tradition today, d) how they respond to teamwork with multiple means and stimuli, e) how they respond to the production of a complex artifact (analog or digital) using authentic sources, f) how they build new concepts by developing a different attitude within the group, h) how they work with multiple media connecting the past with the present, (digital enriched teaching) i) how they can be assisted by using a specific combination of ICT tools (digital enriched teaching).

## Target Group of the Research

The target group of the research was students of Primary School of Magnesia (Greece), where the research was conducted, with experimental and comparative groups. At the same time, the teachers of the classes involved, the principals as well as the teachers of the specialties that collaborated in the implementation of the teaching interventions (Informatics and Visual Arts).

## Research Questions of Research

The research questions were the basis for the design of the teaching interventions, which were then evaluated by methods of: a) quantitative research (pre-test / post-test), evaluation sheet, b) qualitative research (calendar - observation discussion with teachers, principals).

The main research question was whether the combination of interactive digital enrichment technologies, with the addition of other tools and teaching strategies (visits and research in real sources) in teaching popular culture modules in primary education, could be proven more effective than analog learning which also comprises other tools and teaching strategies.

The individual research questions and answers that emerged from the research clarify and answer the above main research question. The secondary research questions were:

- To what extent did students understood the concept of popular culture, in teaching interventions in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what cultural forms of entertainment and art do the students relate the concept of popular culture, in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent do students think that popular culture still exists today as an important expression of society, in teaching interventions in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent can students understand the meaning of tradition, in teaching interventions in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent can students understand the concept of teamwork, and in particular teamwork with many means, materials and tools, in teaching interventions in teaching interventions in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent can students work on producing an artifact (e.g. a fairy tale or a presentation or a story) in collaboration with their team members through discussion, reflection, exchange of views and ideas in teaching interventions, in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent can students discover new knowledge as a group, developing a different attitude and building concepts in teaching interventions in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent do students wish to work in multiple tools linking the past to the present, in teaching interventions in which digital enrichment tools have been used, in relation to enriched analog teaching interventions?
- To what extent the use of ICTs can enhance the students' interest ("I like") in folk culture topics (nautical tradition), in a digital enriched teaching intervention, in relation to enriched analog teaching interventions?
- To what extent the use of ICTs can enhance the learning skills ('what helped me learn') of students in the field of folk culture (nautical tradition) in a digital enriched teaching intervention, in relation to analog enriched teaching?

- To what extent can students be assisted by the use of ICT tools, (digital enriched teaching), to compose integrated digital history (scenario) on popular culture issues (nautical tradition), in relation to enriched analog teaching?
- To what extent can students be assisted by using a specific combination of ICT tools, (digital enriched teaching), to compose an integrated digital story (scenario) on folk culture (nautical tradition), in relation to a specific combination of analog enriched analog teaching tools?

## Research Structure: Methodology

Regarding the methodology, an experimental research was followed, with experimental and comparative groups. The aim was to evaluate, through a specific pre-control / post-control experimental design (Robson, 2010: 150), the effectiveness of a set of teaching interventions (project) related to the teaching of popular culture, using ICT. (digital enriched interventions), compared to another set of popular culture teaching interventions without the use of ICT (analog enriched interventions). The sample selected consisted of two classes of the same grade, (classes E1, E2) from four selected Primary Schools.

In particular, throughout the teaching process, both in the intervention and the control group, the researcher was present as an observer, which had been made clear to the students from the beginning and when the teacher assessed that help was needed in the use of ICTs, the observer became an assistant. In addition to teaching interventions in the classroom, the researcher was also present at the educational visit, which took place at the boatyard Cartapanis and the Museum - Christopoulos House.

In all teaching interventions and activities, the researcher observed and kept a detailed diary, as a field observation tool. The teaching interventions took place on different days and hours and therefore it was possible to record them in detail, thereby enhancing the research with quality tools.

For the convenience of the researcher, the principal of the school unit, and the teachers of both groups (intervention group, control group), there was a recorded planning, including an intervention schedule (day, time, class), as well as teacher preparation instructions from previous days. For each group (intervention or control group), a different table with corresponding instructions was composed, for the teacher who applied the intervention.

Pre and post the teaching interventions discussions were held, with the teacher of the class of the schools where the interventions were implemented, with the principal mainly on organizational issues, and with the IT teacher and the visual arts teacher on the cross-curricular part of the interventions. First of all, for reasons of order and ethics after the approval of the relevant permission by the Institute of Educational Policy, the competent licensing body for educational and pedagogical research, and prior to the commencement of teaching interventions, the school counselor to whom the school belongs was requested a relevant permission. The counselor was informed in advance of the educational scenario for all the groups (intervention group, control group) and the methodology of the intervention was also discussed. Then, with the school counselor's assent, a meeting and discussion was held with the head of the school unit and he was introduced in detail in the process of the teaching instructions.

At a subsequent time, a discussion was separately held with the teachers of the classrooms where the teaching interventions would take place. The first meeting was with the intervention group (IG) teachers and the second meeting with the control group (CG) teacher.

The issues discussed were: a) the content and objectives of the teaching intervention, b) familiarizing students with ICTs outside of school, c) how to work with their students (teamwork etc.), d) subject selection issues, e) technical issues and difficulties, f) the scheduling of teaching interventions, g) group creation issues, h) the form of students derivable.

The teachers of both classes were informed that, in the context of interdisciplinarity, they would work with the Visual Art teacher and the IT teacher, in order to participate respectively in the projects of the other two subject areas (Informatics, Visual Arts).

### Educational Scenario and Printed Educational Material

Two educational scenarios were created with the corresponding activities. One deployed digital enriched teaching and the other analog enriched teaching, while both scenarios explored the same subject matter, implementing several common actions (educational visit, writing, storytelling etc.). Further down, part of the educational scenario implemented with the use of ICT (digital enriched teaching), will be presented.

## THE EDUCATIONAL SCENARIO USING ICT (THE USE OF DIGITAL ENRICHED TEACHING)

### Title of the Scenario

*"Approaching the Nautical Tradition of My Country: The Case of Magnesia."*

### Brief Description of the Scenario

The educational scenario is pertained to 4th grade of Elementary School and will be implemented in three three-hour teaching interventions and a two-hour preliminary intervention, a project on folk culture (nautical tradition of my country). The scenario will be greatly facilitated by the use of ICTs (creation of enriched books) and will be implemented in the context of flexible zones (flexible projects), within the timetable following the principles of the interdisciplinary approach.

The educational scenario discusses modules on concepts: nautical tradition, folk culture, past, present and future of shipping, the environment in relation to shipping and tradition, folk art, local history perspectives, mythology and folk songs of the sea. In particular, the instructor approaches the issue of Magnesia's nautical tradition in an experiential way, using constructive digital learning frameworks, through the guided project methodology that includes several teaching interventions, workshops and visits. The ultimate goal of the design of the teaching interventions is for the concepts of the above subject, related to popular culture, nautical tradition and local history, to be taught.

### Estimated Duration

The 11-hour teaching intervention consists of three three-hour tutorials as well as a two-hour preliminary intervention.

### Aims and Objective

The students are expected to:

### Regarding the Subject

- To discover and understand the notion of continuation of folk culture and nautical tradition,
- To investigate and record alternatively the impact and connection of nautical tradition with individual themes (local history, social life and environment, art, nautical tradition in Greece, Greece: from mythical to modern times),
- To compose a complete story (scenario) with audio-visual material on a theme of nautical tradition, combining past and present.

### Regarding the Use of New Technologies

- To use collaborative learning tools (wiki) based on a specific subject area,
- To familiarize learners with ICT tools. - Computer literacy (online enriched book creating software i-w-book),

### Regarding the Learning Process

- To develop critical thinking through engaging in activities related to traditional and modern popular culture,
- To develop synthetic skills through the production of the enriched book,
- To be motivated and engaged in a group learning process through discussion, reflection and exchange of views and ideas, so as to complete both a collaborative text and be involved in the process of creating an enriched book on nautical tradition,
- To reconstruct ideas and images related to the concepts of "folk culture", "nautical tradition" and "marine occupation", through the cognitive conflict that may arise from the different way of representing nautical tradition (past - present - future - evolution) using exploratory internet searches and classroom discussion (group - plenary),
- To work collaboratively through the process of selecting and producing material (collaborative text, enriched book).

## Cognitive Areas Involved

Informatics (computer literacy), Visual arts (art forms, history of art - artists), Language - Literature (comprehension of oral texts and writing production, fairy tale), History (local history), Environmental study (sea).

## Correlation with the Curriculum

Always according the Curriculum for the "New School", Field: Culture - Art Activities for compulsory education: "students should not owe respect for their cultural heritage, nor should they bear a debt to memorize information regarding the past, in order to secure their national identity. On the contrary, tomorrow's society will multiple advantages if its citizens can creatively converse with the past, derive personal emotions through it, and enrich their present with valuable elements that each individual has chosen to spread from the past, building on through the mental staff of the "museum" a conscious attitude of love and care for what was conquered and tested over time." (P.I. 2011a: 8)

### Students' Alternative Perceptions (Representations, Ideas, etc.)

Alternative perceptions, that is, the representations and ideas that students have about folk culture, and therefore about nautical tradition, are related to the usefulness (why?) of this object (motivation to learn) and whether it practically (how?) follows their daily practices (learning benefit). The continuation of the past with the present thus, is fundamental in the approach of our teaching intervention and should not be transmuted into popular culture lessons aimed to mere memorization; the ultimate aim is, through an experiential approach, the discovery of the other (past) which is the base of contemporary culture.

Problems that may arise from students' passive attitude towards popular culture may be overcome, if first of all, folklore and sterile reproduction of a stereotypical image of the past are exceeded. Thus, the subject expected to be taught through the use of ICTs (nautical tradition in past and present) must be taught through a realistic approach with interpretations of perceptions, social relationships or even habits.

The relationship between daily amateur - artistic creation with the contemporary one can be seen alongside traditional folk culture. This was helped by the fact that in the area of maritime tradition there were traditional boats, fishermen and sailors continuing the tradition of teaching. A great contribution towards this direction was the fact that, in the field of nautical tradition during the prosecution of the teaching implementation, there were still present traditional shipwrights, fishermen and sea men, who continued the tradition.

## CONCLUSION – SUGGESTIONS

Regarding the basic research question, to what extent students understood the meaning of popular culture before and after the two different teaching interventions, we concluded from the quantitative research data that the control group (CG) had slightly higher rating than the intervention group (IG). Consequently, towards the concept of 'popular culture', the control group (GC) responds better after the teaching interventions than the intervention group (IG) does. The fact attests that the control group without the use of technology, understood the subject quite well and the students were able to relate the concept of popular culture to its content. However, it should not be forgotten that both groups show improvement after the teaching interventions, which confirms the effectiveness of the teaching interventions respectively. It should also be noted, that both groups started with a fairly high rate level of understanding regarding the subject.

Based on qualitative research (observation and diary), we can identify the causes of this divergence between the two groups. One possible explanation could be that the intervention group completed the questionnaires rather impetuously and may not have sufficiently understood the functioning of the Likert scale. Another issue recorded regarding the control group, is that the classroom teacher was able to devote more time to the dialogue and discussion when completing the questionnaires, and more generally during all teaching interventions, due to the time saved from non-preparation.

In relation to the qualitive research, it should be added that after the teaching interventions, both groups showed a positive change with respect to the subject. Students also responded positively to both teaching interventions respectively, highlighting the variety of tools used in conjunction with the educational visit.

Moreover, another important issue emerged from the fieldwork-observation, is the children's willingness to engage in dialogue and the effort to connect the knowledge they have gained with their daily lives and practices. There was also a significant tendency for students to relate subjects from all lessons to the new subject, as well as to relate their personal experiences and knowledge to it.

Another important issue that emerged, especially regarding the groups using ICT tools, was the initial enthusiasm for the advent of computers and the internet in the classroom environment, which was linked to the logic of "I like to learn in this way". This enthusiasm, nevertheless, was tempered in the face of "what I have to learn in order to finally deliver a digital artifact following given specifications".

Specifically, the selection and management of text at the composition level within groups using ICT, caused stress to children, which forced the teacher to facilitate and suggest directions. Accordingly, the students of the groups where ICT tools were not used, had less stress probably, due to the fact that the shaping of texts could be also continued at home, depending on the extent they wished.

There is no statistically significant dependence on the gender of the student. However, a statistically significant dependency arises from the teacher's gender, according to the assessment of "I would like more lessons to be delivered in this way". Examining the averages rates per gender, it seems that the differentiation is due to the comparatively more positive evaluation of the analog teaching proposal by the C.G. students who had a female teacher. Consequently, students in these classes state, to a higher degree, in relation to their classmates in a male teacher's classroom, that they would like the teaching intervention to be applied to other subjects as well

## Which Teaching Intervention Should the Teacher Choose and why?

After completing the two teaching interventions in the intervention group (IG) and the control group (CG), the question arises as to which teaching intervention the teacher should choose and why.

As proven from the results, both teaching interventions showed approximately the same positive change during the pre-post test and the other intervention assessment sheets, fact that primarily attests that both enrichment teachings had approximately the same effectiveness. However, issues arise concerning teaching interventions (digital or analog enriched) with respect to the students and the teachers.

In particular, enriched digital teaching has positive effects on students, such as: a) children's interest in learning is stimulated, b) is better suited for modules with rich digital material, because searching for sources is easier and faster, c) the possibility of producing digital material, which can be shared online, is offered, thereby contributing to the extraversion of the classroom and the school, d) the possibility of extending and continuing digital artifact formatting assignment over time, is also offered, e) children are offered the possibility to work collaboratively also outside classroom (use of wiki, search engines, etc.).

Regarding the teacher, it is essential that he is familiarized with the philosophy of using ICT tools prior to the implementation of an enriched digital teaching, which practically means that the teacher must have already been trained in the use of ICT. (Second level of training). Furthermore, at least for the first time, it is required for the teachers to devote a significant amount of their time to formulate digital and printed material for teaching interventions. Also, the implementation of digital enriched teaching requires in advance familiarization of students with software and applications.

Another important issue is to ensure digital class equipment and time to settle computers and other tools (projector, wi-fi router, etc.), which can generate teaching noise as well as the need for extra preparation time and overall more time to implement the teaching intervention.

In the case of the analog enriched teaching, preparation on the behalf of the teacher and the technical arrangements needed, are less. However, analog teaching intervention also requires detailed educational material design and careful handling of group interactivity issues. Especially in the case of artifact production, respectively, as in digital enriched teaching, the teacher should adequately prepare the students giving them the precise instructions for the course of work and the operation of the group.

In conclusion, it should be underlined that the final choice between one of the two (digital - analog) teaching interventions is consisted of many issues, such as: the level of familiarity of the class and the teacher with the ICT, the teaching time available, the particularities of each module, the diversity of its teaching material and, finally, the degree of difficulty and understanding.

What is certain, however, is that enriched teaching (digital or analog) seems to have multiple learning benefits and is more likely to stimulate student interest.

Following the above and summarizing, we note that: the intervention group started at a lower level regarding the subject than the control group, which is later confirmed by the answers to most of the questions.

In more detail, it is observed from the position of the average values of the variables that the control group has a higher average initial cognitive level than the intervention group, but does not differ significantly statistically.

The changes noted in both groups, following the different teaching interventions, show approximately the same upward change. That is, approximately the same significant progress has been noted in student attitudes and perceptions. The possible explanation is that the two teachings may differ in respect to ICT tools being used in the intervention group, whereas in the control group they were not used, however they also present basic similarities: a) common authentic learning resources (educational visit), b) empirical tools (fieldwork research), c) multiple tools and tools (digital-analog), d) similar thematic approach (visual arts), e) motivation for artifact creation (digital enriched book, magic analog book). Furthermore, the different cognitive level and way of working, at the level of daily lesson / teaching practice by each classroom teacher, affects to a certain extend the above results. We should also not forget that, had we compared a digital enriched teaching with a teacher-centered teaching, we would end up in the comfortable dominance of teaching with the use of ICTs.

In this case, however, a different reality is being recorded where the key to teaching (digital - analog) is teamwork, the variety of tools and resources, the authentic learning framework and the motivation for art production (play and learn).

Certainly, the image, no matter how we choose to treat it, in combination with the text (multimedia, in the case of digital enriched teaching, or analog in the case of analog enriched teaching) offers significant opportunities for learning.

After the end of the teaching interventions in both groups, implementing two different teachings (digitally and analog enriched), using qualitative and quantitative tools with the corresponding interpretation techniques, it arises that:

- They, above all, prefer authentic learning, research and experience contexts (educational visit to Christopoulos Museum – Tarsana Kartapani, fieldwork research, film presentation on nautical tradition),
- They are seeking a combination of tools and diversity in every form of teaching both in the case of digitally enriched teaching (wiki, enriched book writing software, web search, etc.) and in the

case of analog teaching (searching for material from books, creating magic book, educational visit and field research, folk tale and discussion).
- They like to be involved with technology (e.g., enriched book writing), but they learn through a combination of tools and especially, authentic situations,
- They prefer the subjects to be taught in an interdisciplinary way (visual arts, computer science, literature, language) and they expect the activities to be connected,
- They are already familiarized with image and less with text and are therefore helped by the use of multimedia ICT environments, such as enriched book writing software,
- In the context of both different enriched teachings, they are equally taught and create artifacts when the above are combined with research and educational visits.

Students in the control group (analog enriched teaching) understood to a greater extent the concept of folk culture after teaching interventions compared to the intervention group (digitally enriched teaching), possibly due to the greater investment in time on behalf of their teacher for discussion and commentary, as well as because of the background of the particular group (note that the control group started at a higher level in relation to the cognitive level than the intervention group and this is reflected throughout research).

Students in the control group (analog enriched teaching) linked to a greater extent (a remarkable change in all sub-questions) the cultural forms of entertainment (today and yesterday), to the sense of popular culture, compared to the intervention group (digitally enriched).), possibly due to the greater investment in time on behalf of their for discussion and commentary, as well as the cognitive background of the group's cognitive background.

The members of both groups, after the completion of the teaching interventions, agreed equally and to a large extent that popular culture still exists today as an important expression of society.

The members of both groups (intervention - control) understood equally the content and importance of nautical tradition, and this shows that with both enriched teachings we can probably achieve similar results.

Students of the intervention group (digitally enriched teaching) through the use of ICT are significantly reinforced and show an interest in folk culture (nautical tradition), while similarly, the control group (analog enriched teaching) are equally reinforced, demonstrating eventually that diversity of media enrichment is a key factor in successful teaching.

Whereas students state that they primarily and greatly enjoy the use of ICTs in teaching, the same does not apply to learning, that is, what they eventually learn.

Students can be greatly helped through the use of ICT tools to compose integrated digital history (scenario) on folk culture (nautical tradition), but they perform just as well when they are offered a corresponding variety and quality of tools.

Students did not encounter any problem learning ICT tools and this shows their great familiarity with technologies, but also their ease with integrating ICT tools. in a wider field of objects.

## Suggestions

Consequently, based on both the quantitative and qualitative findings of the research, targeted proposals are being made regarding the design and implementation of enriched teaching interventions using digital or analog means.

1. Regarding the teachers it occurs that a generalized training is immediately required: a) on ICT usage, b) on the development of digital learning objects (simple and complex) and digital enrichment c) in ICT tools. (web 2.0), d) on issues concerning the development of popular culture programs in the Flexible Zone, e) issues related to the design of wider free-zone programs (free projects), f) issues on the exploitation of educational visits and the organization of field research by pupils.
2. Regarding the school's infrastructures, it is evident that at classroom and laboratory level in many schools there are no modern IT laboratories, internet access points, wi-fi and video projectors. Some schools also have an urgent need to obtain an event venue or a multipurpose use room and an organized library, where various research projects or innovative programs could be implemented.
3. At the level of curriculum design, and in particular of popular culture programs, it is necessary to: a) plan ahead at the beginning of each school year, b) taking into account the popular culture curriculum references, c) taking into account the Flexible Zone Guidelines and innovative curricula, d) combine multiple tools and techniques, taking into account what means can best enhance learning objectives each time, e) provision of the use of ICTs in the teaching process should be ensured, if the classroom has probationary practiced the use of ICT tools. and IT, either in the context of IT or in particular in an innovative program, f) the teaching plan should be recorded (scenario and activity sheets) presenting in detail the course of one or more teaching interventions.
4. Regarding the popular culture curriculum, it still currently relevant and the material created for its teaching seeks to be exploited. This material was created under the scientific supervision of Professor M.G. Meraklis and his then scientific associate Dr. of Folklore, Mr. Rea Kakabouras (now Assistant Professor of Folklore of NTUA)[7] and resulted to the production of a guide book entitled "Interdisciplinary Training Material for the Flexible Zone - Clothing - Folk Culture[8]". This guide includes, as it is well known, lesson plans, folklore texts, instructions, and more. Moreover, the teaching models of the appendix "Attempts to create educational material on folk culture for kindergarten and first grade of Primary School" of Gasoukas & Foulidis (2015)[9] can also be used.
5. Fifteen years later, it is useful to develop a new Curriculum in the context of Flexible Zone or the Area of Free Activities (free projects), taking into account the current trends in the field of popular culture teaching, good practices, instructional and models, as well as related digital material developed in the context of various public actions. This is due to the fact that the teaching of folk culture in Primary Education does not have an organized teaching framework, accompanied by an official Curriculum, textbooks and an individual lesson in the time schedule of school. However, modules or sub-topics related to popular culture are found in school textbooks scattered across various subjects in the form of texts and pictures. In particular, Literature and Language textbooks contain several topics specific to popular culture, as well as lesser references to other subjects such as Environmental Studies, Music, Religious Education, Visual, Environmental Education, Physical Education or even Mathematics.

For all of the above-mentioned topics, which are scattered throughout all elementary school lessons, additional auxiliary educational scenarios, with or without ICTs, need to be developed. (digital or analog teaching), taking into account new concepts and practices in the field of teaching.Guides can also be developed to exploit texts that refer to popular culture, such as: a) guide on educational visits, b) guide to utilizing simple and complex digital learning objects, c) the research project implementation guide.

6. The digital material of enriched books and public repositories of learning objects (e.g. Photo-Tree) needs to be systematized and exploited in educational scenarios or suggested activities in the various sections of popular culture, found in Primary School books.
7. The following platforms and digital frameworks should be used to develop teaching interventions on popular culture issues: a) the method of Pedagogical Folklore and Learning by Design, b) "Aesop", the integrated tool for the Development, Design, Writing, Evaluation and Presentation of Digital Interactive Teaching Scenarios, c) the model for development of educational scenarios through the online application MITIDA, Production, Reconstruction and Distribution of Educational Scripts service.
8. There is an urgent need to develop curricula (enriched teachings) for popular culture in the context of the Flexible Zone or in the area of free activities (free projects).

In order to implement all of the above suggestions and ideas, a special evaluation team of existing programs and curricula and guides for popular culture in Primary Education should be set up:

a) Autonomous curriculum for popular culture in the context of the Flexible Zone or the Area of Free Activities (free projects) for Preschool and Primary Education.

## REFERENCES

Araka, H., Koutras, N., & Makridou, E. (2014). Access to Information: Evolution and the Digital Divide. In Information History - From Papyrus to Electronic Document. Athens: Law Library.

Bausinger, H. (2009). *The popular culture in the world of technology*. Athens: Patakis.

Curriculum for "New School", Field: Culture - Art Activities for Compulsory Education, Athens. (2011a). Institute of Education - Ministry of Lifelong Learning and Religious Education. Available at: http://ebooks.edu.gr/

Dreyfus, L. H. (2003). The Internet. Athens: Kritiki.

Dundes, A. (2005). Folkloristics in the Twenty-First Century (AFS Invited Presidential Plenary Address, 2004). *Journal of American Folklore*, *118*(470), 385–408. doi:10.1353/jaf.2005.0044

Foulidis, X. (2015). *New horizons in Greek Folklore Studies: The Internet as a field of production, dissemination and educational exploitation of Folk Culture* (Doctoral Thesis). University of the Aegean, Department of Preschool Education and Educational Design.

Gasouka, M., Kapaniaris, A., Arvanitidou, Z., Foulidi, X., & Raptou, E. (2013). Folk culture and digital enriched books: The case study of thw digital enriched toll "E-Anagnosis". *Proceedings of INTED2013 Conference,* 4176-4184.

Gasoukas, M., & Foulidis, X. (2012). *New Horizons of Folklore Studies - Conceptual Framework, Research, Gender, Internet, School*. Athens: Sideris Publications.

Howard, G. R. (2016). The New Regularity of Digital Web Hybridization. In E. Avdikos (Ed.), *Internet Cultures*. Athens: Pedio.

Kalawsky, R. (1994). *The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments*. Addison Wesley Publishing Company.

Kalogiannakis, M., & Papadakis, S. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In *Proceedings of the 8th International Conference On Computer Based Learning in Science* (pp. 265-276). Academic Press.

Kalogiannakis, M., & Papadakis, S. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. *Training, 6*, 2.

Kalogiannakis, M., Vassilakis, K., Alafodimos, C., Papadakis, S., Papachristos, D., & Zafeiri, E. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece. *International Journal of Advanced Corporate Learning*, *2*(4), 15–20. doi:10.3991/ijac.v2i4.981

Kapaniaris, A. (2013a). From e-books to enriched e-books: Tools for digital enrichment in education. In Interdisciplinary Conference "History of Information: Papyrus Reference Points in Electronic Document. Athens: Department of Archives and Libraries of the Ionian University.

Kapaniaris, A. (2013b). Cultural Archives and Folklore Collections: The Role of Informatics and Information Technologies in the Formulation of Digital Educational Materials. In *Proceedings of the 7th Congress of Postgraduate Students and Doctoral Candidates*. Athens: Department of Philology, National and Kapodistrian University of Athens.

Kapaniaris, A., Gasouka, M., Zisiadis, D., Papadimitriou, E., & Kalogirou, E. (2013a). Learning Object Design and Development in Folklore Education Using Web 2.0 Tools. Mediterranean Journal of Social Sciences, 4(11), 104-110.

Kapaniaris, A., Gasouka, M., Zisiadis, D., Papadimitriou, E., & Kalogirou, E. (2013b). Digital Books Taxonomy: From Text E-Books to Digitally Enriched E-Books in Folklore Education Using the iPad. in Mediterranean Journal of Social Sciences, 4(11), 316-322.

Kapaniaris, A., & Gasoukas, M. (2016). Folk Culture and Enriched E-Books: Designing Learning Resources for Internet-Based Education. In E. Avdikos (Ed.), *Internet Cultures*. Athens: Pedio.

Lanier, J., Minsky, M., Fisher, S., & Druin, A. (1989). Virtual Environments and Interactivity: Windows To The Future. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques.* ACM.

Matsagouras, H. (2010). *The Innovation of Research Works in High School*. Athens: OEDB.

Meraklis, M., & Kakabouras-Tilis, P. (2001). Intersectional Material for Flexible Zone, 1st-2nd Grades of Elementary School, Aesthetic Education, Teaching Greek Sign Language, Clothing - Folk Culture. Athens: Institute of Education - Ministry of Education & Religious Affairs.

Moustakas, K., Paliokas, I., Tsakiris, A., & Tzovaras, D. (2015). *Graphic and Virtual Reality*. Athens: Greek Academic Electronic Textbooks and Tutorials - Hellenic Academic Libraries Association.

Mpilalis, M. (2015). *The Past on the Internet, Image, Technology and Historical Culture in Modern Greece (1994-2005)*. Athens: HISTOREIN - National Documentation Center.

Orfanakis, V., Papadakis, S., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital student conference platform implementation: The case study of the “research project” course. *The Open Education Journal*, *12*(2), 5–23.

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *International Journal of Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Robson, C. (2010). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers.* Αθήνα: Gutenberg.

Skarpelos, G. (1999). *Terra Virtualis, The Construction of Cyberspace*. Athens: Nefeli.

Thomas, N., Carroll, F., Kop, R., & Stocking, S. (2012). iBook learning experience: the challenge of teaching computer architecture to first year university students. In *WORLDCOMP’12- The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing*. Las Vegas, NV: WORLDCOMP.

Vidakis, N., Barianos, A. K., Trampas, A. M., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (Vol. 1, pp. 36-43). 10.5220/0007810800360043

## ADDITIONAL READING

Kalogiannakis, M., & Papadakis, St. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-service Teachers to Promote STEM Activities in Preschool Classrooms. In St. Papadakis and M. Kalogiannakis (Eds), Mobile Learning Applications in Early Childhood Education, USA-PA: IGI Global (under publication). doi:10.4018/978-1-7998-1486-3.ch005

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, St. (2019). Evaluating the efficiency of two programming environments in shaping novices’ attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *Int. J. Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, St. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-primary School Age, In St. Papadakis and M. Kalogiannakis (Eds), Mobile Learning Applications in Early Childhood Education, USA-PA: IGI Global (under publication).

Papadakis, S., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. *HSci2010 7th International Conference Hands-on Science Bridging the Science and Society Gap*.

Papadakis, St., & Kalogiannakis, M. (2018). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed), Early Childhood Education, 1-19, i-first chapter, UK-London: IntechOpen Limited.

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231–246. doi:10.1504/IJTEL.2019.100478

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Technology Enhanced Learning*, *10*(3), 235–250.

Papadakis, St., & Kalogiannakis, M. (2020). A Research Synthesis of the Educational Value of Self-proclaimed Mobile Educational Applications for Young Age Children. In St. Papadakis & M. Kalogiannakis (Eds.), *Mobile Learning Applications in Early Childhood Education*. IGI Global. doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, St. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

## KEY TERMS AND DEFINITIONS

**Digital Artifact:** The term refers to any form of digital creation, digital art composition, digital representation with the help of any web tool or other software.

**Enriched Analog Teaching:** The term analog teaching refers to a teaching process that students using various means, materials and techniques which are analog rather than digital (collage, comic, painting, drawing, diagrams, etc., can create artifacts through analog learning deliverables resources (photography, interviewing, creative writing, fairy tale, archive material) within an authentic learning environment where students themselves create and become producers of educational material.

**Enriched Digital Teaching:** The term enriched digital teaching refers to a teaching process which encourages and supports the use of ICT and more particularly, online writing environments for the production of artifacts with enriched interactive learning resources. Thus, students within a genuine learning framework that they themselves create become producers of educational material.

**ICT (Information and Communication Technology):** A term that covers all technical means used to handle information and aid communication, including software and hardware.

**STEM:** The term STEM (science, technology, engineering, and mathematics) is an acronym used by those relevant to the educational method concerning the fields of science, technology, engineering, and mathematics.

## ENDNOTES

[1] According to Kalawsky, "… a virtual environment is a synthetic sensory experience, transmitting, naturally and abstractly, information to the person experiencing it. It is born out of a computer system through the presentation to human sensory systems, a human-computer interface that approaches various properties of the real world and which is in the form of three-dimensional visual environments consisting of objects and phenomena." Kalawsky, R. (1994) The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments, Addison Wesley Publishing Company.

[2] Mitsos Bilalis (2015), in relation to the developments triggered by technology into the field of culture, states: ... I believe that one must keep distance from techno-deterministic considerations, which presuppose that the emergence of new technology triggers - almost legally - chaining developments in the field of culture", see p. Bilalis, M. (2015). The Past on the Internet, Image, Technology and Historical Culture in Modern Greece (1994-2005). Athens: HISTORY - National Documentation Center.

[3] The above definition was based on the concept of the term artifact, which was defined in the context of the subject research project, offered by the 1st Lyceum under the supervision of Elias Matsagouras. According to Matsangouras"… an artifact expresses the central idea of research work and can take the form of construction, artistic composition, two-dimensional or three-dimensional representation, collage, poster, video, film, activities for expressing a central idea or highlighting an issue Matsagouras, E. (2010). *The Innovation of Research Project in High School*..Athens: OEDB, p.23.

[4] According to the definition of L.T.S.C. (Learning Technology Standards Committee) Learning objects are defined as entities, whether digital or not, that can be reused by various teaching systems and educational environments. LTSC. (2000). Learning technology standards committee Available at: http://ltsc.ieee.org/ (last accessed July 20, 2014).

[5] Davis and Olson (1985, p. 200) state that information is data that have been processed in a way that makes sense to the recipient in: Araka, H.,,Koutras, N., Makridou, E. (2014). Access to Information: Evolution and the Digital Divide *In Information History - From Papyrus to Electronic Document* (ed. Kanellopoulos - Boti, M.). Athens: Law Library.

[6] Dundes, A. (2005). Folkloristics in the Twenty-First Century (AFS Invited Presidential Plenary Address, 2004), in *Journal of American Folklore* 118, pp. 385-408.

[7] Pedagogical Institute, Flexible Zone, available at: http://www.pi-schools.gr/programs/EuZin/.

[8] Meraklis, M., Kakabouras- Tilis, P. (2001). Intersectional Material for Flexible Zone, 1st-2nd Grades of Elementary School, Aesthetic Education, Teaching Greek Sign Language, Clothing - Folk Culture. Athens: Institute of Education - Ministry of Education & Religious Affairs. pp. 55-99.

[9] Gasoukas, M. & Foulidis, X. (2012). New horizons of folklore - Conceptual framework, research, gender, internet, school. Athens: Sideris Publications.

# Chapter 19
# Folk Culture and Enriched Digital Teaching:
## Designing Educational Scenarios With the Use of ICT

**Alexandros G. Kapaniaris**
*Democritus University of Thrace, Greece*

## ABSTRACT

*The ongoing developments in the field of information technology and, in particular, information and communication technologies (ICT) combined with the new digital culture that is rapidly emerging on the internet create a new perspective in the teaching of popular culture in primary education. More specifically, the teaching of popular culture can be greatly enhanced by learning experiences based on digital learning objects. Moreover, with the use of IT tools and enriched teaching can enhance the relationship between local folklore research and research work in primary education in the context of flexible zone or cultural programs (school activities). The internet at large and online technologies, Web 2.0, can transfuse special dynamic entities to popular culture evidence.*

## INTRODUCTION

Folk culture never constituted an independent discipline in the history of primary education yet, in most disciplines, correlated topics are taught to students. Specifically, in literature and language school textbooks there are separate units devoted to popular culture as well as, lesser references (texts and images) to other subjects. This content is often accompanied by activities. In addition to literature and language, references to popular culture are also found in religious education, natural sciences, environmental education, art, gymnastics and even mathematics. Consequently, popular culture may not be an autonomous subject but in many cases it occupies quite a large part of the material either by specific themes (literature) or by references to all lessons (thematic approach) or through school activities (representations of popular events, shadow theater, dramatization, traditional dances, etc.) (Kakampouras 2010: 216).

DOI: 10.4018/978-1-7998-4576-8.ch019

The necessity of instructing and familiarizing children around the works of folk culture, combined with the formal culture provided by the whole school curriculum, can only be sustained if the traditional folk culture is linked to parts of children's daily life (school culture). Thus, when teachers try to communicate with their students they weave a web connecting everyday life (school culture) with the folk culture that is an integrated part of their daily life (local or global), so that positive results can be achieved for the learning process (Tassin 2010 at: Diamanou 2013).

## Folklore and Popular Culture

Looking back in time, it is prominent that the instruction of folk culture in the elementary school was closely related to the science of folklore and the folkloric science in general. The contribution of folklore as an autonomous discipline in the study of Greek culture was consolidated by the emergence of a section of society that identified with the rural class, at least for the period from the 18th century until its vigorous transformation after World War II, in its midst. 20th century. After the war, a transformation occurred, leading to changes in the rural class combined with the increase in the number of urban populations. Consequently, these changes inflamed social and urban content to folklore (Dundes 1965, 1980 in: Diamanou 2013).

The teachers' relationship with folklore also dates back to the second half of the 19th century. At this time schoolteachers, scholars, headmasters, elementary school inspectors, teachers and curators are engaged in a struggle to collect and preserve folklore material. This endeavor is also supported by a series of circulars from institutional bodies such as the Ministry of Ecclesiastical Affairs and Public Education as well as, related proclamation and competitions on behalf of associations and scientific companies. It is no coincidence that, at this time, with the assistance of Nikolaos Politis, student folklore collections are being produced in schools and universities. The aforementioned endeavor resulted in the publication of folklore material in the Symmetika in the Bulletin of the Hellenic Folklore Society as well as in the Folklore Archive of the Academy of Athens (Kakabouras 2010: 217).

According to Bawman, Avdikos and Kakabouras folklore has always been up to this day one of the areas where activities can be developed and function parallel to or be integrated into education (school curriculum). One of the most important benefits that folklore has to offer to the classroom, is the awareness of the modern identity of the modern man and the encounter with the "other" in the context of the students' correspondence with each other, or wider, the correspondence between students, parents and teachers (Bawman 2006, Avdikos 1999. Kakabouras 1999. 2005 at: Diamanou 2013).

# THE CONTINUATION OF THE PAST (TRADITIONAL CULTURE) WITH THE PRESENT (MODERN POPULAR CULTURE)

The pinnacle of traditional culture is linked to the prevalence of community life during the Turkish occupation, where crafts and commerce flourished while folk art is also at its prime. With the establishing of the independent Greek state in 1832, communities, as a key core of administrative organization, are gradually weakened and Greece is progressively entering industrial development. With the advent of World War II, we move on to another era of modern popular culture (Meraklis, 2001: 62).

An important factor that should be examined and greatly contributes to the continuity of the past with the present is the acceptance of the fact that, the teaching of popular culture must also be linked to the local tradition of each region. In this way, students will be able to build a more complete understanding of local culture and history. A striking example of enhancing students' cultural self-awareness is knowing that formal history has left its mark (e.g. in the "monuments of their place") and whether the daily and customary lives of their ancestors remain functional even in several instances or not (Kakabouras 2001: 87).

The continuation of the past with the present, as the key to the approach of teaching interventions to popular culture, should not be transmuted into popular culture lessons aimed to mere memorization; instead the ultimate aim through such an experiential approach is the discovery of the other (past) which is eventually the continuum of a contemporary culture.

Moreover, the teaching of folk culture should be integrated into modern intercultural education that takes into account the importance of particular culture, in understanding individual and cultural identity (Kakabouras, 2010: 220).

## HOW CAN POPULAR CULTURE BE LINKED TO MODERN POPULAR CULTURE?

One of the ways to connect popular culture with modern popular culture can be achieved by understanding the phenomenon of the complex identity of modern man resting on the substrate of popular culture. Thus, references among all peoples share a link in common cultural elements (folk narratives, rituals, folk art) that can function fruitfully in corresponding intercultural encounters and educational processes (Diamanou 2013: xx).

Linking popular culture with modern school presupposes the transcendence of folklore and the sterile reproduction of a stereotypical image of the past. Today, popular culture must be taught through a realistic approach with interpretations of perceptions, social relationships or even habits. This is possible, and can be achieved to some extent, through the design of long-term school curricula that will use recording and research tools (traditions, fairy tales, doctrines, testimonies, narratives related to the daily life of the immediate past). If the above research method is complemented by data analysis with tools that any age group can handle, it will eventually lead to creative learning of parts of social history and the production of conscious collective memory (Kirdis xx). The idea of using research tools in popular culture is also reinforced by Diamantidis, Frosis and Patel (2004: 85-94) who state that fieldwork research, interviewing, etc., are among the key tools that can be used in school as research methods of folklore.

The connection of popular culture with modern culture and monuments that are present in the daily lives of students can also be achieved by cultivating interest in popular culture and local history. Thus, the curriculum proposes programs that link folklore to local and oral history. In this way, students have an incentive to learn information about buildings, constrictions and monuments they encounter daily (cultural environment) but have never been given the opportunity to construct a more circumstantial view on them.

Another method of teaching folk culture and connecting it to the present day, giving this way a functional direction to what the student is learning, is the comparison that should be sought through folk culture material in order to designate timeless concepts and values. Through examples and critical discussion at a plenary or group level, students, in collaboration with the teacher, can search for timeless concepts, ideas and values. These concepts, ideas and values may be related to quality yesterday and today (e.g. food culture), endurance, hygiene and aesthetics yesterday and today (e.g. wood culture), aesthetics and diversity yesterday and today (e.g. weaving culture), the concept of handmade and original as opposed to imitation, substitute and mass production and culture (e.g. reproduction of works of art).

Moreover, the daily amateur - artistic creation of today's modern popular culture can be examined alongside traditional folk culture. Moreover, modern day occupations and interactions (market, marches and demonstrations, online matchmaking centers) as a modern expression of today's popular culture, can be associated with yesterday (traditional matchmakers, bazaars, revolts). Even fairy tale, myth and folk song as authentic aspects of folk creation are still prevalent to this day in many areas of Greece (e.g. Crete with mantinades).

The popular creation in times of crisis with the corresponding production of anecdotes is transferred and modified from "email to email" as "mouth to mouth". This modern production of anecdotes1 or poems with the help of new technologies is dispersed in incredible time and forms a new peculiar form of folk creation. Their potential gleaning and teaching connect education with a practice that students themselves exert in their daily lives. Consequently, if the word "folk culture" corresponds to anonymous art creations that are put to popular consumption, then we have to admit that initially, the anecdotes and riddles constitute the most vivid forms of folk creation today rather the lyrics (Dermitzakis 2006).

Accordingly, in the field of folk painting, folk painters depict scenes of the daily life of yesterday and today. Typical examples in the Magnesia region is Theophilos Hadzimichail 2(folk painter) where his traversal and works in the region (Milies, Makrinitsa, Anakasia, Ano Volos) represent the folk culture of the past and the folk painter of Tarnasas Nikos Christopoulos 3(Pefkakia) representing contemporary years. The continuity of folk painting is intense, and examples exist in many areas of Greece. Engagement with such issues could possibly offer to education elements linking yesterday with today, popular culture with local and national history.

## RESEARCH PROJECT METHOD AND POPULAR CULTURE

As mentioned above according to Kakabouras (2010: 218) the projekt method is suggested by Dr. Benekos, Professor of Folklore, in various applications of folk culture in primary school. According to Benekos (2006: 197), the creative involvement with Greek tradition in the form of a project, enables multilevel connections and multilevel correlations between linguistic, artistic, craft and music lessons. The aim of the project is for the children to construct their own theme by producing a project that has its own seal transferring tradition to the present. The project allows students to take initiative by activating their innate tendency for action. Thus, the appropriate environment is created for traditions, myths, fairy tales, folk songs to be understood alongside other elements that children themselves draw from popular culture, giving life to all these.

Benekos (2006: 198) in his same work entitled "Cultural Tradition and Education" extends the project method by applying the globular lesson analysis (Projektunterricht). According to this approach, students are particularly interested in research that extends over a longer period of time. In this way, they seek each topic flexibly and overcome the difficulties that arise, creatively by developing individual and group solidarity. The interdisciplinary approach to the method of globular lesson analysis (Projektunterricht) is extensively presented in Benekos' work, (2008: 149-153) entitled "Perspectives of Popular Culture". Indicatively, regarding the inter-disciplinary proposal on folk song, the project proposed to facilitate and give ideas for further implementation, consists of the following thematic sections: a) history, b) environment - nature, c) religious education, d) social issues (beauty of women and men), e) migration,

f) professions (pastoral, agricultural, nautical) and g) intercultural education. This interdisciplinary approach is often illustrated by a block diagram, as a guide for its use in program implementation.

## THE METHOD OF TRAINING SCENARIOS USING ICT (B-LEVEL TRAINING IN GREECE)

According to the former Institute of Education of Greece (today's Institute of Educational Policy), the term educational scenario refers to that "complete teaching model, based on one or more learning theories and frames through an organized structure the specific subject area, which will be taught together along with psychoeducational theories and teaching methodology that will be applied "(PI, 2006). More specifically, it is a complete learning framework, characterized by:

- focused subject,
- specific educational / teaching goals,
- building upon specific pedagogical principles or theories,
- its evolution through a series of activities in which students and teachers have well-defined roles, and
- utilizing specific educational tools / software (Tzimas 2009).

The educational scenario with the use of ICT implemented in teacher training for ICT utilization and implementation in the teaching process (level B) 4has a specific structure and is a teaching proposal that leverages Information and Communication Technologies (ICT) however, is not consisted solely of ICT lectures.

The development of an educational scenario involves seven (7) phases: a) Identification of the subject: title, class, cognitive areas involved, prerequisites, b) Detection of students' prior knowledge and representations: what the students know, possible difficulties in contemplating about the subject, c) Setting objectives of the scenario: on the subject (s), on the learning process and on the use of ICT, d) Teaching material: the required logistics, books, maps, software, constructions, worksheets, e) Creating scenario activities: organizing teaching based on appropriate classroom scenario implementation activities (teaching approaches and strategies, exploiting the added value of ICT in the learning process, creating corresponding worksheets, etc.) f) Assessment: student and scenario and possible extensions of the scenario, g) Teachers' remarks and instructions, bibliography, references, sources (RACTI 2013).

## THE IMPLEMENTATION OF ICT IN THE TEACHING PROCESS

Up to these days, the exploitation of ICT in the teaching process, in the majority of cases, has been treated as a panacea or an end in itself, resulting to the emergence of new technologies as a teacher rather than a 'partner' along the learning process. Consequently, at this point it should be elucidated that students do not learn from ICT itself, but ICT can support productive thinking and concept building. So, if students learn through technology and the ICT become students' spiritual collaborator, we can have an innovative, more creative approach to the learning process (Jonassen, Howland, Marra & Crismond, 2011).

Attempting to identify the role of technology to a more "meaningful learning5" we will eventually be lead to some suggestions and assumptions such as:

- ICT are not just software and multimedia but, they can faction as digital learning frameworks where students can develop plans and activities by approaching learning creatively, along with the aid of cognitive learning and critical thinking strategies,
- as ICT any environment or specific set of activities can be identified, providing that engages students in active, constructive, conscious, authentic and collaborative learning,
- ICTs can effectively support meaningful learning when fulfilling a specific learning need. Students interact with these digital environments in a way that allows them to have control over these tools, giving substance to their learning process.
- ICT should assist students, while as intellectual tools they should help them build more meaningful personal interpretations and representations of the world (Jonassen et al., 2011).

## Implementation of ICT in the Teaching Process: The Case of Popular Culture

The educational exploitation of ICTs remains an on ongoing challenge for education throughout the years. Simultaneously, an ongoing question for the educational community is the effectiveness of new technologies as auxiliary to teaching strategies developed in educational scenarios. This reflection even extends when comparing an enriched digital teaching to an analog enriched teaching. Thus, questions are raised, such as whether a teaching approach can be equally effective and pleasant for students without the use of ICT tools, in relation to a teaching intervention using other analog enriched techniques.

In cases where both techniques (enriched digital techniques and analog enriched teaching) lead to the approximately same result, it may legitimately be wondered whether a student will gain more using one of these techniques. In addition to the obvious learning benefit which is first and foremost measured in the educational process, there are a number of issues to consider:

a) The preparation of the learning material (learning resources) which should be created by teachers. Respectively this material is related to the speed, efficiency, scalability and reuse of the material,
b) The diversity and multiple representations of the learning material that can be offered to students from multiple perspectives so as to avoid saturation and fatigue.
c) The link between modern day representations and learning practices to an era dominated by image, video and speed,
d) The time and cost needed for representations, visits, observations of phenomena and situations that can be done either digitally or analogously.
e) The interactivity of teaching in relation to the means to be used for teaching (digital - analog) and,
f) The learning theory on which the teaching intervention (constructive, behaviorism) is based and how it is successfully achieved through activities.

## Learning Theories and ICT: Constructive Learning Environments

The learning theory that supports the construction of learning through teaching interventions using ICT is the theory of constructivism. According to this notion, the individual creates his or her own representations of knowledge based on their previous experiences. Consequently, children's pre-existing ideas of things and the world surrounding them have a dominant role in their learning. The fundamental idea that rules the theory of constructivism is the acceptance that there in not a single right version of right knowledge (Solomonides, 2006).

However, according to many researchers, as Solomonides points out, the most common features of constructive learning environments are:

- The exploitation of authentic environments rather than pre-constructed educational sequences, in real contexts, that make sense to children (authentic environments).
- The representation of the physical complexity of the real world by avoiding the usual oversimplification of knowledge, using realistic approaches rather than academic learning contexts. (complex environment).
- Learners are directed to assume a more active role, they are active agents of their own learning, that is, to manipulate tools and objects and give thought their actions (active / manipulative environment.)
- Focus on building, rather than on reproducing, empowering the knowledge-building process that depends on both context and content. In addition, provides opportunities and tools for children to incorporate new ideas into pre-existing knowledge for interpreting multiple facets of the world and constructing substance and meaning of notions (constructive environment.)
- Strengthening the process of communication, dialogue, collaboration between children, as well as, supporting collaborative knowledge building, by trying to find solutions through the connection of school to society and the world, so that learners understand that there are many ways to interpret the world and various solutions to various problems of life (conversational / collaborative environment).
- Implementing learning activities within learning environments that foster the construction of meaning and facilitate conceptual interconnection between their cognitive (contextual environments).
- Creating a design focused on clear pedagogical goals and supporting learners along the way in shaping their learning goals (intentional environment).
- Strengthening the process of reflection on what they are doing, the decisions they make, in order learners themselves to gain internal control of their learning and use the process of evaluation as a self-analysis and progress tool (reflective environment.)
- Finally, teachers are helpers, companions and analysts of strategies used to solve problems. Simultaneously, every mistake is utilized and converted to a mechanism for deeper understanding of the subject. (teacher as coach) (Solomonides, 2006; Kanoukas, & Anderson, 1999; Vlachos, 2004).

## ENRICHED DIGITAL TEACHING

In general, a digitally enriched teaching could implement, besides its common tools, strategies and tools of analog enriched teaching (authentic learning frameworks, such as educational visits and fieldwork research) and pure digital means such as the creation of artifacts with writing tools for enriched e-books or the creation of collaborative presentations and texts.

The ultimate goal is to teach the learning object through activities, research, visits, fieldwork investigations, material compositions, collaborative text production online and the final deliverable is the creation of artifacts (digital enriched e-book).

## The Selection of Digital Tools

The choice of web 2.0 technologies, and the particular selection of wiki, was not accidental. This technology was primarily chosen since part of our educational scenario or our teaching proposal, involves activities of producing a collaborative material that must be produced remotely, outside teaching hours (homework). This collaborative text along with other accompanying material (photos, videos, hyperlinks) will constitute as a part of the core material for the enriched digital book. The use of wiki in our teaching entails t in advance learners' familiarization with this technology so that they can easily handle the tool.

Moreover, choosing the wiki as an ICT tool supports meaningful learning since as a social tool it a) enhances cooperation between team members, b) provides space for discussion, disagreement and consensus building among team members regarding their final deliverables, c) supports knowledge building (representation of ideas, understanding and beliefs of students) d) produces an organized multimedia knowledge base. Regarding the search for information material (text, images, links, videos) through the search engines, ICT respectively provide an information vehicle for knowledge exploration, learning support by building access to necessary information and the ability to compare perspectives, beliefs and opinions (Jonassen et al., 2011).

The use of digital enriched book creating software with the "w-i-book" tool besides being easy to use, since it is appropriately built for the corresponding age groups (primary education),the use of ICTs, also offer students the support of thoughtful learning which functions as conceptual partner aiming to:

a) articulate and visualize what they already know
b) reflect on what they have learned and how they have learned by selecting and synthesizing the information (material assembled in the wiki) and cognitive material (acquired knowledge),
c) support their internal negotiations giving a purpose to their meaning,
d) structure personal representations of concepts,
e) support their diligent thinking,
f) support the logic "learning by doing" (Jonassen et al., 2011).

## How can Enriched Digital Books Be Used in the Teaching of Popular Culture

Enriched digital books can provide a learning experience framework through simple (Learning Asset) and complex learning objectives (Learning Resource), provided these are integrated into a comprehensive pedagogical design based on the lesson scenario and corresponding activity sheets.

In fact, enriched books are the means to create learning experiences for students and perhaps this is the innovation they can offer to education (Kapaniaris, 2013).

It should be underlined moreover, that the learning experience is not just the production of an activity using a computer system, with the aim of making the activity effective, easily digestible and flexible, giving the learner satisfaction. It has to be useful in the lives of students as well, and that is the true essence of the process (Thomas, Carroll, Kop, & Stocking, 2012).

## SELECTING THE CONTENT OF TEACHING INTERVENTIONS AND ITS CONNECTION TO ICT

The content, the learning object in other words, chosen for the aforementioned teaching interventions to be applied (digitally enriched teaching), is the folk culture and in particular the theme "Approaching the work of folk painters yesterday and today: The case of Theofilos"

The aim of the design of the teaching intervention was for students to:

a) discover and understand the concept of the continuation of folk culture and folk art (folk painters),
b) research and record alternatively the impact and connection between folk painting and individual themes (local history, social life and environment, art, folk painting in Greece, Greece: from mythical to contemporary years),
c) compile a complete story (scenario) for a theme of folk painting combining past with present.

Specifically, the reasons for choosing this particular teaching object and the specific thematic area, in combination with the teaching intervention of digitally enriched teaching were:

a) folk culture by its nature comprises a wealth of data (texts, images, videos, music) that can be represented by students and teachers in multiple and creative ways,
b) folk culture includes "hot" and experiential material (educational visits / fieldwork-research and digitization) that can be utilized in multiple ways and used descriptively in related activities,
c) research interest, in an area where teaching interventions in the form of educational scenarios designed on the basis of digitally enriched teaching, have not been implemented;
d) the material of folk culture folk painting and art is a subject that can be interpreted to a large extent (exploiting photos - works of art) without cost and time constraints with modern digital media and Web 2.0 applications (wiki, enriched digital e-book),
e) The specific theme "The Case of the Folk Painter Theofilos" offers children opportunities to work on intersectionality (language, visual arts, computer science, local history) through the use of new technologies.

## EDUCATIONAL SCENARIO "APPROACHING THE WORK OF FOLK PAINTERS YESTERDAY AND TODAY: THE CASE OF THEOFILOS"

**Scenario Title**

"Approaching the Work of Folk Painters Yesterday and Today: The Case of Theofilos".

**Scenario Creator**

Alexandros G. Kapaniaris

## BRIEF DESCRIPTION OF THE SCENARIO

This educational scenario is applied to the 4th grade of the elementary school and will be implemented in three three-hour teaching interventions and one two-hour preliminary intervention, a project on folk culture (folk artists yesterday and today). The scenario will be greatly facilitated by the use of ICTs (creation of enriched books) and will be implemented in the context of flexible zones (flexible projects), within the timetable following the principles of the interdisciplinary approach.

## ESTIMATED DURATION

The 11-hour teaching intervention consists of three three-hour tutorials as well as a two-hour preliminary intervention.

## AIMS AND OBJECTIVES

### Regarding the Subject

- To discover and understand the notion of continuation of folk culture and folk art (folk painters),
- To investigate and record alternatively the impact and connection of folk painting with individual themes (local history, social life and environment, art, folk painting in Greece, Greece: from mythical to modern times),
- To compose a complete story (scenario) with audio-visual material on a theme of folk painting, combining past and present.

### Regarding the Use of New Technologies

- To use collaborative learning tools (wiki) based on a specific subject area,
- To familiarize learners with ICT tools. - Computer literacy (online enriched book creating software i-w-book),

### Regarding the Learning Process

- To develop critical thinking through engaging in activities related to traditional and modern popular culture
- To develop synthetic skills through the production of the enriched book,
- To be motivated and engaged in a group learning process through discussion, reflection and exchange of views and ideas, so as to complete both a collaborative text and be involved in the process of creating an enriched book on folk painting,
- To reconstruct ideas and images related to the concepts of “folk culture”, “folk painting” and “folk painter”, through the cognitive conflict that may arise from the different way of representing folk art (past - present - future - evolution) using exploratory internet searches and classroom discussion (group - plenary),
- To work collaboratively through the process of selecting and producing material (collaborative text, enriched book).

## COGNITIVE AREAS INVOLVED

Informatics (computer literacy), Visual arts (art forms, history of art - artists), Language - Literature (comprehension of oral texts and writing production, fairy tale), History (local history).

## LEARNERS' PREREQUISITE KNOWLEDGE

Students in the language and literature classes have already practiced written production and have been exposed to popular culture texts. Therefore, there is no prerequisite knowledge that students should have already acquired in relation to the subject. Additionally, students are not expected to possess any special skills in using computers since they have already become accustomed to using the painting program, the word processor and the browser from previous classes. Regarding the use of the w-i-book software, a relative learning activity will be executed, while the use of the wiki as a collaborative tool has been taught in previous lessons as part of the literature lesson by the classroom teacher.

## CORRELATION WITH THE CURRICULUM

It is compatible with the Analytical Curriculum and the CTCF and especially in the Pilot School Units implementing the New Curriculum developed in the context of the New School (21st Century School) - New Curriculum and in particular in the section "Culture - Art" and in the field "Culture - Art Activities for compulsory education ". In this program there is also provision for interventions and educational activities that encourage the dissemination of the concepts of Culture within the foreseen subjects of the Curriculum. Namely, the section "dealing with the past" refers to the exact need for a fruitful connection to what we call cultural heritage and how it is linked to popular culture.

Always in line with the Curriculum for the "New School", Field: Culture - Art Activities for compulsory education: "*students should not owe respect for their cultural heritage, nor should they bear a debt to memorize information regarding the past, in order to secure their national identity. On the contrary, tomorrow's society will multiple advantages if its citizens can creatively converse with the past, derive personal emotions through it, and enrich their present with valuable elements that each individual has chosen to spread from the past, building on through the mental staff of the "museum" a conscious attitude of love and care for what was conquered and tested over time.*"

## TEACHING MATERIAL AND LOGISTICS REQUIRED FOR THE SCENARIO

In order for this scenario to be implemented, a portable computer lab with internet access will be needed and an office suit management application installed as well. Moreover, a separate educational material folder will be created for both the teacher and the student. Part of the educational package "Following the footsteps of Theophilos" will also be used.

## EPISTEMOLOGICAL APPROACH AND CONCEPTUAL ANALYSIS

The introduction of popular culture into education, or more specifically, the need for popular culture teaching, reinforces children's expectations and needs since many facets of popular culture: a) contribute to the child's acquisition of cultural experiences, b) strengthen or enhance the child's imagination and meaning, c) act as a vehicle for understanding, as far as possible, cultural diversity, ethnicity and social diversity, accepting and responding to it critically, d) develop the child's cognitive, linguistic, social, emotional, kinetic, mental and aesthetic development, and (e) enhancing the confidence of children involved in storytelling or dramatization (Gasoukas & Foulidis 2012: 133-134).

Another key parameter that overlaps the teaching of folk culture and this particular teaching unit (folk artists) is whether the teacher can perceive that cultural phenomena do not work out of history and consequently be able to explain to students, the inevitable distinction between modern Greek traditional and modern popular culture. Thus, unless both the connection and the continuity of the past (traditional culture) with the present (modern popular culture) is achieved, we will not be able to build sufficiently our scenarios and their corresponding activities (Meraklis 2001: 62).

An important factor that should be examined and greatly contributes to the continuity of the past with the present is the acceptance of the fact that, the teaching of popular culture must also be linked to the local tradition of each region. In this way, students will be able to build a more complete understanding of local culture and history. A striking example of enhancing students' cultural self-awareness is knowing that formal history has left its mark (Kakabouras 2001: 87). A relevant example in our teaching intervention could be the buildings and monuments of our place (eg Theophilos' traversal in Magnesia and his artistic imprint on Magnesia buildings such as the Kontos house in Anakasia and Velentzas bakery)

## STUDENTS' ALTERNATIVE PERCEPTIONS (REPRESENTATIONS, IDEAS, ETC.)

Alternative perceptions, that is, the representations and ideas that students have about folk culture, and therefore about folk art, are related to the usefulness (why?) of this object (motivation to learn) and whether it practically (how?) follows their daily practices (learning benefit). The continuation of the past with the present thus, is fundamental in the approach of our teaching intervention and should not be transmuted into popular culture lessons aimed to mere memorization; the ultimate aim through an experiential approach, the discovery of the other (past) which is the continuum of a contemporary culture.

Problems that may arise from students' passive attitude towards popular culture may be overcome, if first of all, folklore and sterile reproduction of a stereotypical image of the past are exceeded. Thus, the subject expected to be taught through the use of ICTs (folk painting and in paste and present) must be taught through a realistic approach with interpretations of perceptions, social relationships or even habits.

Therefore, the daily amateur - artistic creation of today's modern popular culture can be seen alongside traditional folk culture. Accordingly, in the field of folk painting, folk painters depict scenes of the daily life of yesterday and today. Typical examples in the Magnesia region is Theophilos Hadzimichail (folk painter) where his traversal and works in the region (Milies, Makrinitsa, Anakasia, Ano Volos) represent the folk culture of the past and the folk painter of Tarnasas Nikos Christopoulos (Pefkakia) representing contemporary years. An example of such a link to present art could be the wall painting (city graffiti). In conclusion, engagement with such issues could possibly offer to education elements linking yesterday with today, popular culture to local and national history.

## CLASS ORGANIZATION

The lesson will be conducted in the classroom, with desks divided in groups of three, while the portable computer lab will be used when needed. Students will work in groups of three on each computer. Students, after a brief introduction to the software environment following the relevant familiarization required, will be invited to use w-i-book software and other digital resources based on corresponding activity sheets.

## TEACHING APPROACHES AND LEARNING STRATEGIES / THEORIES

The teaching approaches and strategies that will be followed will be illustrated separately for each two-hour teaching intervention, followed by a comprehensive view of the link between teaching interventions and learning theories. It should also be noted that between the two-hour teaching intervention one week will interfere, so that the teaching can be implemented using the logic of a research project. An important point to pay particular attention to and will possibly ensure the successful outcome of the overall teaching intervention, is the collaboration of the IT and Visual Instructor with the teacher and students of the teaching intervention in the interdisciplinary approach. In particular, students throughout the IT course should be encouraged to process the resulting digital material as well as search for information material on the Internet (video editing and posting on YouTube of the educational visit, photo editing, wiki editing). Similarly, students participating in the visual art lesson can engage in folk painting modules, with emphasis on folk painters and contemporary folk art respectively.

The teaching intervention will also be linked to an educational visit to Theofilos Museum - Kontos House in Anakasia (after the completion of the second two-hour teaching intervention) with the aim of researching, recording and capturing information about the subject of the group. This material will then be used so as to complete the subject, in the classroom's wiki and in the creation of the enriched wi-book as well.

Overall, the teaching approaches and strategies to be used are: a) presentation, b) storytelling, c) comprehension of oral texts and writing, d) the use of the collaborative Web 2.0 (wiki) technologies, e) discussion, f) fieldwork research (interviews, recording, photographing) at Theofilos Museum (educational visit), g) the use of enriched book writing tools (use of ICT - online educational software, h) the online search of work-related sources and their processing, and i) the use of visual arts techniques. The learning theory, that underpins this educational intervention as a whole, is related to the theory of constructivism. According to this notion, the individual creates his/ her own representations of knowledge based on their experiences.

## SCENARIO ACTIVITIES DESCRIPTION

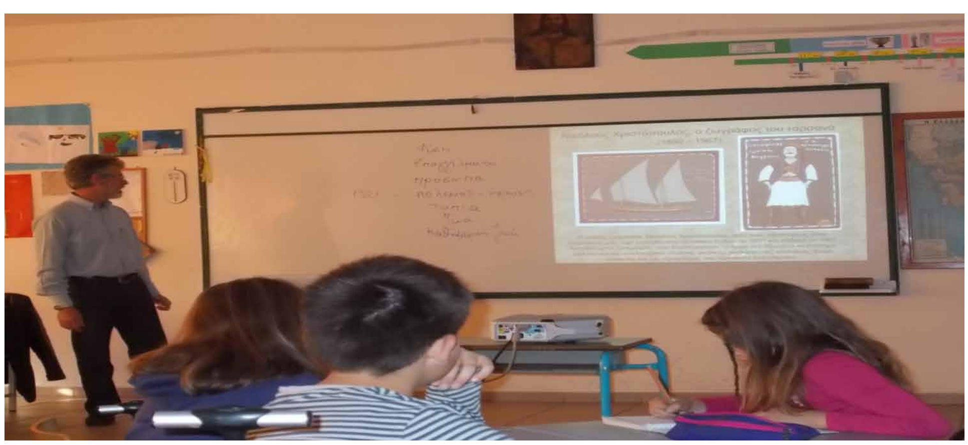
### Preparatory Actions (2hours)

The 2 classes of the 4th grade (intervention group D1 and control group D2) completed the pre-test under the instructions of the classroom teachers and then watched extracts of the film by Lakis Papastathis "THEOFILOS". The children had the opportunity to watch selected passages of the film decorated with pictures of Theophilos' works. The studets had the opportunity to watch selected extracts of the film sprinkled with pictures of Theophilos' works. The ultimate purpose of this preliminary teaching was to introduce students to the subject through a visualized way that would stimulate their interest while informing them of the value of folk art.

## 1st Teaching Intervention (3hours)

During the first teaching intervention, a 10-minute presentation will be given by the class teacher on folk art using the power point software and a video projector in the classroom. The presentation on folk art will focus on folk painting and on the following subtopics: a) local history, b) social life and environment, c) art, d) folk painting in Greece, e) Greece: from mythical to the contemporary years. In the material, there will be references to folk painters: a) Theophilos Hadjimichael - Folk painting, b) Pagonis – Folk religious painting, c) Panagiotis Painter - Folk painting echoing gunpowder and braveρy, d) Nikos Christopoulos today - folk painting that reaches contemporary years. Then, for fifteen minutes, an interactive discussion will follow and a dialogue will be developed on the basis that cultural phenomena (folk painting and naïve painters) do not survive outside history, and will consequently be able to explain to students the inevitable distinction between traditional Greek and modern Greek culture.

*Figure 1. The teacher based on the presentation, records the subject units*



Afterwards, the teacher will narrate the story "THEOFILOS HATZIMICHAIL" by Irene Kamaratos – Gialoussis. Guides accompanying the students through the adventurous journey through time will be George and Mary, two peer heroes who watch what happens in past times and are impressed to see how all of this can be applied in their own time, in the present, in the daily lives of their school and share their story with others around them. During the next teaching hour, the teacher asks the students to form groups of three / four and decide on which subject they will work based on the work of the popular painter Theophilos (groups are formed by lot or otherwise through any other way taking into account that groups should be heterogeneous in terms of gender and performance). After the formation of the groups, auxiliary corresponding tabs will be distributed with indicative interdisciplinary subjects from the educational package "Following the footsteps of Theophilos". A discussion will follow along with the completion of an activity sheet (psychological and cognitive preparation).

Depending on the groups and themes that will arise depending on the students decisions, they will develop their own material (texts, photos, narratives, fairy tales, interviews, contemporary art forms today) in the wiki classroom (Web 2.0 collaborative tool) until their next teaching meeting (homework research and writing assignment). This material will be re-used multiple times in other activities, functioning as students' reference and research material in bibliography (books) and sources (internet).

*Figure 2. Painting folk motifs*

## 2nd Teaching Intervention (3hours)

Given the formation of six groups and their respective themes, students at a plenary level, summarize the material they have produced through the wiki. The main activity sheet is then distributed and based on the subject of each group, the students complete the sub- activities using the portable computer lab and the corresponding w-i-book software. The activity sheet has two goals: a)to familiarize students with free online wi-book software, b) the initial compilation of material from the collaborative wiki tool so as students to begin designing the story in the environment of the enriched wi-book writing tool (creation of first pages). The final deliverable, that is, the enriched book of each group and the class as a whole (along the way, the material will be consolidated) will be enriched with material from the fieldwork research conducted during the students' visit to Theofilos Museum - Kontos House.

## 3rd Teaching Intervention (3hours)

During the third teaching intervention, in the first part an activity sheet (subject matter consolidation) will be distributed in order to: a) the material derived from the wiki and the research material from the fieldwork investigation (educational visit to Theofilos Museum),to be rearranged and associated with new material from other contemporary forms of folk art (current folk artists). In this way students will form their views, but they will also be able to understand the continuity of popular culture with modern popular culture, b) complete and publish the digital enriched book using the free w-i-book software (http://software.i-nous.org/w-i-book/). The students then fill out an assessment sheet and discuss the benefits of teaching intervention (reflection).

*Figure 3. The teacher in the role of coordinator in the environment of the portable computer lab*

## WORKSHEETS

### Psychological and Cognitive Preparation Activity Sheet

Through the psychological and cognitive preparation activity sheet we will try to stimulate the students, inform about the purpose and objectives of the lesson, first, assess the existing knowledge of folk art and finally, identify the cognitive difficulties and representations of the learners.

### Activity Sheet (Exploration & Writing Production) [Teamwork Homework]

The activity sheet, exploration and production of written discourse in a collaborative learning environment (wiki), aims to enable children work remotely (homework) in an activity that involves exploration of print and digital resources, collaboration and learning that combines game, the previous stimuli they had from classroom instruction and support for thematic units. Through this activity, children will produce the very first material (texts, images, videos) that will form the basis material for building the enriched book, an integrated attempt in other words, to imprint their thematic area. Simultaneously, the same activity sheet also functions as an evaluation index of all the groups and their members.

## ADDITIONAL INFORMATION: ASSESSMENT OF STUDENTS AND SCENARIO

### Students Assessment

Assessment of students will result from the assessment sheet given at the end of the third (two-hour) teaching intervention. There is also provision for homework (creating collaborative text on the wiki) as well as activities to be implemented in collaboration with visual arts and IT teachers.

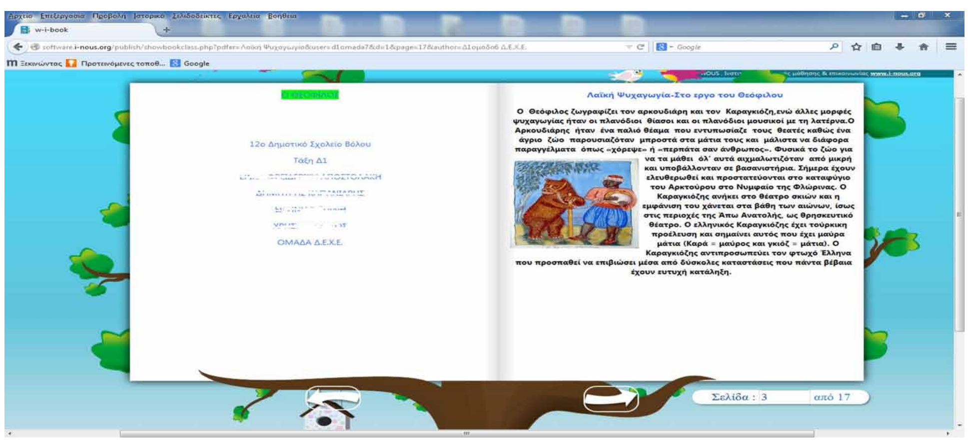
## Scenario Evaluation

The scenario will be evaluated from classroom fieldwork research (visitor-assistant), video recording of teaching intervention and diary keeping.

## Scenario Extensions and Interconnection with Other Concepts

This teaching intervention may be the beginning of other sub-activities that extend the original one using similar methods and activities as traditional professions, from the perspective of popular artists yesterday and today. Moreover, after the implementation of this specific teaching intervention, students' concerns will be recorded as a result of a specific activity for a further expanding (eg Pelion folk architecture).

*Figure 4. Digital enriched book*




## CONCLUSIONS FROM THE IMPLEMENTATION OF THE TEACHING SCENARIO

The selection of the particular subject used for the teaching intervention on folk culture and the theme "Approaching the work of folk painters yesterday and today: The case of Theophilos" was not accidental; it was due to a number of concerns, questions and interests that pre-existed on the researcher's side.

Specifically, the reasons for selecting this subject when designing the teaching approach, were related to the age of the students (elementary) and the expected response they might show to such teaching approaches, to the particularity and interest of their particular medium itself, that is the technology used (enriched book writing tool), in a relatively poor research area such as that of folk culture and ICT, in experiential material that can be imprinted and researched through such teaching interventions (actual works of art at Theofilos Museum – Kontos House), the domination of the image in the subject under study at an age where the visual dominates the word (image, color and design) and finally, the nature of the themes offered for teamwork and research (image and text composition) in a familiar environment in the children's world (child painting - folk painting).

It should be underlined that the designed and implemented teaching interventions followed the design of an interdisciplinary project and required several teaching hours to be implemented.

Throughout the teaching process the researcher was present holding the role of observer, which had been explained to the students from the beginning, and when the teacher assessed that the assistance of ICT was needed, the observer became an assistant. In addition to the classroom interventions, the researcher was also present during the educational visit to the Theofilos Museum - Kontos House. In all teaching interventions and activities, the researcher observed and kept a detailed diary as a fieldwork observation tool. The teaching interventions took place on different days and hours and therefore it was possible to record them in detail.

Pre and post the teaching interventions discussions were held, with the teacher of the class, with the principal mainly on organizational issues, and with the IT teacher and the visual arts teacher on the cross-curricular part of the interventions.

First of all, for reasons of order and ethics, prior to the beginning of the teaching intervention, the school counselor to whom the school belongs was requested a relevant permission. The counselor was informed in advance of the educational scenario for all the groups and the methodology of the intervention was also discussed. Furthermore, with the school counselor's assent, a meeting and discussion was held with the head of the school unit and he was introduced in detail in the process of the educational scenario.

Subsequently, there was a separate discussion with the classroom teacher where the teaching interventions were to be held. The issues on which the discussion focused were: a) the content and objectives of the teaching intervention, b) the students' familiarization with ICT outside the school lesson, c) the best way to work with students (cooperative, etc.), d) subject selection issues, e) technical issues and difficulties, f) scheduling of teaching interventions, g) group creation issues, h) the form of student assignments.

The classroom teacher was also informed that in the context of interdisciplinary, collaboration would be required with the Art teacher and the IT teacher so as to have two other subjects (Informatics, Visual Art) explored in the project.

The discussion with the teachers and the exchange of views continued after the end of the teaching interventions. The conclusions of the teachers involved were focused on: a) the duration of the teaching interventions, that they would like to be even longer than they initially suggested and b) in the formation activity sheets (more informal language).

After the end of the interventions, the conclusions were analyzed and formulated using the qualitative tools (researcher diary, activity sheets, notes, interviews of the teachers and the principal) which show that:

- Students, above all, prefer authentic learning, research and experience contexts (educational visit to Theophilos Museum - Kontos House, fieldwork research, film presentation on Theophilos),
- Students are seeking for a combination of tools and diversity in every form of teaching (wiki, enriched book writing software, web search, etc.),
- Students like to be involved with technology (e.g., enriched book writing), but they learn through a combination of tools and especially, authentic situations,
- Students prefer the subjects to be taught in an interdisciplinary way (visual arts, computer science, literature, language) and they expect the activities to be connected,
- Students are already familiarized with image and less with text and are therefore helped by the use of multimedia ICT environments, such as enriched book writing software,
- Students understood the concept of folk painter, such as Theofilos,
- Students understood the concept of folk painting and its continuation that expands to the present,
- Students through the use of ICT, are significantly enhanced and show interest in folk culture (folk painters),

- Students state that they primarily and largely enjoy the use of ICTs in teaching, but the same does not apply in terms of learning, that is, what I am finally learning,
- Students can be supported to a great extent by the use of ICT tools, so as to compose a complete digital story (scenario) on folk culture (folk artists),
- Students did not encounter any problem learning along ICT tools. This is an indicative example of the students' great familiarity with technology and more generally, of their willingness to integrate with ICT tools in a wider field of objects.

## REFERENCES

Audikos, E. (1999). *Once upon a time….but….it can also be done now - Education as a space for storytellers configuration*. Ellinika Grammata.

Benekos, D. (2006). *Cultural Tradition and Education, Educational Approach - Educational Application*. TYPOTHITO- George Dardanos.

Benekos, D. (2008). *Aspects of Folklore, Folklore Analyses, Educational Applications, Text Research*. Gutenberg.

Bowman, P. (2006). Standing at the Crossroads of Folklore and Education. *Journal of American Folklore*, *119*(471), 66–79. doi:10.1353/jaf.2006.0002

Damianou, K., Frosis, L., & Pateli, O. Cultural Diversity in the Classroom: Students' Needs and Expectations. In D. Sakkas, & A. Psaltis (Eds.), *Workshop Proceedings on" Cultural Diversity in School: The Role of the Teacher* (pp. 85-94). Alexandroupolis.

Dermitzakis, C. (2006). Contemporary Forms of Popular Culture. In *Proceedings of the 1st Conference on Popular Culture & Education.* Volos: Research Institute of Culture & Education. Available at: http://www.eipe.gr/index.php/eisigiseis-2

Diamanou, D. (2013). The pedagogical dimensions of greek Folklore and the instruction of Folklore in Greek education. In *Bulletin of Greek Folklore* (Vol. 42). Hellenic Folklore Society.

Gasoukas, M., & Xanthippi, F. (2012). *Contemporary Horizons of Folklore Studies, Conceptual Framework, Research, Gender, Internet, School*. I. Sideris.

Jonassen, D., Howland, J., Marra, R., & Crismond, D. (2011). *Meaningful learning with Technology* (V. Mitropoulou, Trans.). Methexis.

Kakampoura, P. (2010). Teaching Folk Culture in Primary Education. In *Proceedings of the 1st Panhellenic Congress of Education Sciences*. Athens: Smyrniotakis.

Kanoukas, H., & Anderson, T. (1999). Using constructivism in technology-mediated learning: Constructing order out of the chaos in the literature. *Radical Pedagogy*, *1*(1). http:// radicalpedagogy.icaap.org/content/ issue1_2/02kanuka1_2.html

Kapaniaris, A. (2013). From electronic to enriched e-book: digital enrichment tools in education. Proceedings of Interdisciplinary Conference "History Information: references from papyrus to electronic document." Athens: Department of Archives and Libraries of the Ionian University.

Meraklis, M. (2001). *Folk Culture Curriculum (introductory note) in: Flexible Zone Intersectional Material, 1st-2nd Grade of Elementary School, Aesthetic Education, Teaching Greek Sign Language, Clothing - Folk Culture*. Pedagogical Institute.

Solomonides, X. (2006). *New trends in educational technology. Constructivism and modern learning environments*. Metaixmio.

Tassin M. (2010). Pour une "pédagogie culturelle". *Cahiers pédagogiques, 479*, 29-30.

Thomas, N., Carroll, F., Kop, R., & Stocking, S. (2012). iBook learning experience: the challenge of teaching computer architecture to first year university students. In *Proceedings WORLDCOMP'12- The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing*. Las Vegas, NV: WORLDCOMP.

Tzimas, B. (2009). Educational scenarios with the support of ICT. In *Proceedings of the 1st Educational Conference "Integration and use of ICTs" in the Educational Process.* ETPE, University of Thessaly.

Vlachos, I. (2004). *Training in Natural Sciences. The proposal of constructivism*. Grigoris.

## ADDITIONAL READING

Kalogiannakis, M., & Papadakis, S. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In C. Constantinou, Z. Zacharias & M. Papaevripidou, *Proceedings of the 8th International Conference On Computer Based Learning in Science* (pp. 265-276).

Kalogiannakis, M., & Papadakis, St. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy.*EE'08: Proceedings of the 5th WSEAS/IASME international conference on Engineering education July 2008 Pages 305–311*

Kalogiannakis, M., Vassilakis, K., Alafodimos, C., Papadakis, S., Papachristos, D., & Zafeiri, E. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece. *International Journal of Advanced Corporate Learning*, *2*(4), 15–20. doi:10.3991/ijac.v2i4.981

Papadakis, St. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *Int. J. Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, St. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-primary School Age, In St. Papadakis and M. Kalogiannakis (Eds), Mobile Learning Applications in Early Childhood Education, USA-PA: IGI Global (under publication).

Papadakis, St., & Kalogiannakis, M. (2018). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed), Early Childhood Education, 1-19, i-first chapter, UK-London: IntechOpen Limited.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating the effectiveness of a game-based learning approach in modifying students behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Technology Enhanced Learning*, *10*(3), 235–250.

Papadakis, St. & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *Int. J. Technology Enhanced Learning, 10*, 1-2, 44–72.

## KEY TERMS AND DEFINITIONS

**Analog Teaching:** The term analog teaching refers to a teaching process that students using various means, materials and techniques which are analog rather than digital (collage, comic, painting, drawing, diagrams, etc., can create artifacts through analog learning deliverables resources (photography, interviewing, creative writing, fairy tale, archive material) within an authentic learning environment where students themselves create and become producers of educational material.

**Educational Scenario:** An educational scenario is the complete teaching model, based on one or more theories of learning that in an organized structure frames the specific subject that will be taught along with the psychoeducational theories and teaching methodology that will be applied.

**Enriched Books:** The enriched digital books of the digital school are open source digital textbooks (html) enriched with digital interactive educational materials. The digital material is associated with the section of the book that refers to or is related to. Interactive textbook pages include "active" icons or hyperlinks that refer to digital learning objects, such as simulations, research, experiments, images, educational games, 3D maps, crosswords, and more.

**Enriched Digital Teaching:** The term enriched digital teaching refers to a teaching process which encourages and supports the use of ICT and more particularly, online writing environments for the production of artifacts with enriched interactive learning resources. Thus, students within a genuine learning framework that they themselves create become producers of educational material.

**Learning Object:** A learning object is an autonomous and independent unit of learning material, linked to one or more learning outcomes, which has been initially developed in such a way, that it can be reused in different educational contexts.

## ENDNOTES

1 "- Boss, when will I get some leave??? – Ahahahahahaha nice one!!! Do you know the other joke where you get paid???", They called me from Hellas Online. I told them, "Thanks, I steal Conn-X internet from my neighbour, I am not interested in changing company" ...

2 Theophilos Hatzimihail or Theofilos Kefalas or Kefalas, known simply as Theophilos, was a major folk painter of modern Greek art. Source: wikipedia

[3] Nikolaos Christopoulos, the painter of Tarsanas: List of works in his exhibition at Pefkakia Volos, Album with 191 works by the popular folk painter from Volos of the sea and Tarsanaa N. Christopoulos (2008), Volos: Municipality of Volos.

[4] Official website of the project, http://b-epipedo2.cti.gr/el-GR/

[5] According to Jonassen et al., 2011 "In order for meaningful learning to take place, the work that students pursue, should include active, constructive, purposeful / conscious, authentic and collaborative activities".

# Compilation of References


A Bit of CS4FN. (n.d.). *A Bit of CS4FN*. Retrieved from https://abitofcs4fn.org/

Abdulhamid, M., & Mutheke, M. (2018). Design of Digital Control System for Line Following Robot. *Technological Engineering*, *15*(2), 48–52. doi:10.1515/teen-2018-0018

Abelson, H. (2010). *Is App Inventor based on Scratch?* [Online forum comment]. Retrieved November 25, 2013, from https://groups.google.com/forum/#!topic/App Inventor/LvhZz8NZZ5g

Abelson, H. (2012). From computational thinking to computational values. In *Special Interest Group on Computer Science Education* (pp. 239–240). SIGCSE.

ACARA. (2015). *Australian Curriculum, Assessment and Reporting Authority*. Retrieved from https://www.acara.edu.au/

Agalianos, A., Noss, R., & Whitty, G. (2001). Logo in mainstream schools: The struggle over the soul of an educational innovation. *British Journal of Sociology of Education*, *22*(4), 479–500. doi:10.1080/01425690120094449

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835. doi:10.1093/comjnl/bxs074

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education*, *33*(2), 131–152. doi:10.1016/S0360-1315(99)00029-9

Ainsworth, S., & Van Labeke, N. (2004). Multiple forms of dynamic representation. *Learning and Instruction*, *14*(3), 241–255. doi:10.1016/j.learninstruc.2004.06.002

Akman, B., & Özgül, S. G. (2015). Role of play in teaching science in the early childhood years. In *Research in early childhood science education* (pp. 237–258). Dordrecht: Springer. doi:10.1007/978-94-017-9505-0_11

Akram, B., Min, W., Wiebe, E., Mott, B., Boyer, K. E., & Lester, J. (2019). Assessing Middle School Students' Computational Thinking Through Programming Trajectory Analysis. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1269-1269). ACM. 10.1145/3287324.3293798

Alafodimos, C., Kalogiannakis, M., Papadakis, St., & Papachristos, D. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece, In D. Guralnick (Ed.), *Proceedings of the International Conference on E-Learning in the Workplace (ICELW-09)*. New York: Kaleidoscope Learning.

Alafodimos, C., Kalogiannakis, M., Papadakis, St., & Papachristos, D. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece. In D. Guralnick (Ed.), *Proceedings of the International Conference on E-Learning in the Workplace (ICELW-09)*. New York: Kaleidoscope Learning.

Ala-Mutka, K., & Jarvinen, H.-M. (2004). Assessment process for programming assignments. In *Proceedings of IEEE International Conference on Advanced Learning Technologies* (pp. 181-185). Washington, DC: IEEE. 10.1109/ICALT.2004.1357399

Alimisis, D. (2014). Educational Robotics in Teacher Education: an Innovative Tool for Promoting Quality Education. In L. Daniel, I. Lūka, L. Rutka, & I. Žogla (Eds.), *Teacher of the 21st Century: Quality Education for Quality Teaching* (pp. 14–27). Newcastle, UK: Cambridge Scholars Publishing. Retrieved from https://books.google.gr/books?hl=el&lr=&id=x3ExBwAAQBAJ&oi=fnd&pg=PA14&dq=%5B7%5D%09Alimisis,+D.+(2014).+Educational+Robotics+in+Teacher+Education:+an+Innovative+Tool+for+Promoting+Quality+Educatio.,+In+L.+Daniela,+I.+Lūka,+L.+Rutka+and+I.+Žogla+(Ed.)+"Te

Alimisis, D., & Loukatos, D. (2018, June). STEM education post-graduate students' training in the eCraft2Learn ecosystem. In *Proceedings of the 2nd International Conference on Innovating STEM Education*, *Athen, Greece* (pp. 22-24). Academic Press.

Alimisis, D. (2009). *Teacher education on robotics-enhanced constructivist pedagogical methods*. Athens: School of Pedagogical and Technological Education. Retrieved from http://www.terecop.eu/en/Products1.html

Alimisis, D. (2013). Educational robotics : Open questions and new challenges. *Themes in Science & Technology Education*, *6*(1), 63–71.

Alimisis, D., Alimisi, R., Loukatos, D., & Zoulias, E. (2019). Introducing Maker Movement in Educational Robotics: Beyond Prefabricated Robots and "Black Boxes". In L. Daniela (Ed.), *Smart Learning with Educational Robotics*. Cham: Springer. doi:10.1007/978-3-030-19913-5_4

Alimisis, D., Moro, M., & Menegatti, E. (2016). *Educational Robotics in the Makers Era. Advances in Intelligent Systems and Computing* (Vol. 560). Springer; doi:10.1007/978-3-319-55553-9

Alsancak Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Turkish Journal of Social Research/Turkiye Sosyal Arastirmalar Dergisi, 23*(2).

Alves, N. da C., Gresse von Wangenheim, C., & Hauck, J. C. R. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, *18*(1), 17–39. doi:10.15388/infedu.2019.02

Alves, N. da C., Gresse von Wangenheim, C., Hauck, J. C. R., & Borgatto, A. F. (2020). A Large-scale Evaluation of a Rubric for the Automatic Assessment of Algorithms and Programming Concepts. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 556–562). Portland, OR: ACM. 10.1145/3328778.3366840

Alves, N. da C., Gresse von Wangenheim, C., Hauck, J. C. R., Borgatto, A. F., & de Andrade, D. F. (2020). An Item Response Theory Analysis of the Sequencing of Algorithms & Programming Concepts. *Proceedings of the International Conference on Computational Thinking Education*.

Amariei, C. (2015). *Arduino Development Cookbook*. Packt Publishing Ltd.

American Association for the Advancement of Science (AAAS). (2001). *American Association for the Advancement of Science: Project 2061*. Washington, DC: AAAS.

Anderson, C. (2012). *Makers: The New Industrial Revolution*. Crown Publishing.

Anderson, C. (2012). *Makers: The new Industrial Revolution*. New York: Crown.

Angeli, C. (2019). Computational thinking education: Issues and challenges. *Computers in Human Behavior*. doi:10.1016/j.chb.2019.106185

Angeli, C., & Jaipal-Jamani, K. (2018). Preparing Pre-service Teachers to Promote Computational Thinking in School Classrooms. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines* (pp. 127–150). Springer. doi:10.1007/978-3-319-93566-9_7

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, 105954. doi:10.1016/j.chb.2019.03.018

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47–57.

App Inventor. (2019). *MIT App Inventor*. Retrieved from https://appinventor.mit.edu/

Araka, H., Koutras, N., & Makridou, E. (2014). Access to Information: Evolution and the Digital Divide. In Information History - From Papyrus to Electronic Document. Athens: Law Library.

Araujo, A. L. S. O., Andrade, W. L., Guerrero, D. D. S., & Melo, M. R. A. (2019). How Many Abilities Can We Measure in Computational Thinking?:A Study on Bebras Challenge. In *Proceedings of 50th ACM Technical Symposium on CS Education* (pp. 545-551). New York: ACM.

Ardoin, N. M., Bowers, A. W., Roth, N. W., & Holthuis, N. (2018). Environmental education and K-12 student outcomes: A review and analysis of research. *The Journal of Environmental Education*, *49*(1), 1–17. doi:10.1080/00958964.2017.1366155

Arduino, I. D. E. (2020). *Arduino web editor*. Retrieved from https://www.arduino.cc

Aronson, E. (1978). *The jigsaw classroom*. Retrieved January 2, 2020, from http://psycnet.apa.org/psycinfo/1980-51351-000

Astrachan, O., Hambrusch, S., Peckham, J., & Settle, A. (2009). The present and future of computational thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education* (SIGCSE '09). ACM. https://doi.acm.org/10.1145/1508865.1509053

Atmatzidou, S., & Demetriadis, S. (2014). How to Support Students ' Computational Thinking Skills in Educational Robotics Activities. *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, 43–50.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75B*, 661–670. doi:10.1016/j.robot.2015.10.008

Atmatzidou, S., Demetriadis, S., & Nika, P. (2018). How Does the Degree of Guidance Support Students' Metacognitive and Problem Solving Skills in Educational Robotics? *Journal of Science Education and Technology*, *27*(1), 70–85. doi:10.100710956-017-9709-x

Atmatzidou, S., Markelis, I., & Demetriadis, S. (2008). The use of LEGO Mindstorms in elementary and secondary education : game as a way of triggering learning. *Workshop Proceedings of International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 22–30.

Audikos, E. (1999). *Once upon a time….but….it can also be done now - Education as a space for storytellers configuration*. Ellinika Grammata.

Austin, H., & Sax, L. (1996). Developing scientific talent in undergraduate women. In C. Davis, A. Ginorio, C. Hollenhead, B. Lazarus, P. Rayman, & ... (Eds.), *The equity equation: Fostering the advancement of women in the sciences, mathematics, and engineering*. San Francisco, CA: Jossey-Bass.

Australian Mathematics Trust. (n.d.). *Computational and Algorithmic Thinking (CAT)*. Retrieved from https://goo.gl/eVCFis

Autodesk. (2019). *Tinkercad*. Retrieved from https://www.tinkercad.com/

Ayman, R., Sharaf, N., Ahmed, G., & Abdennadher, S. (2018). MiniColon; Teaching Kids Computational Thinking Using an Interactive Serious Game. In. Lecture Notes in Computer Science: Vol. 11243. *Serious Games. JCSG 2018*. Springer; doi:10.1007/978-3-030-02762-9_9

Bakhsh, S. A. (2016). Using Games as a Tool in Teaching Vocabulary to Young Learners. *English Language Teaching*, *9*(7), 120–128. doi:10.5539/elt.v9n7p120

Balanskat, A., Engelhardt, K., & Licht, A. H. (2018). Strategies to include computational thinking in school curricula in Norway and Sweden. *European Schoolnet's Study Visit. European Schoolnet.* retrieved by http://www.eun.org/documents/411753/817341/Computational_thinking_report_2018pdf/4d3d6fa0-dedd-4b62-a201-a26bf4dfd3a0

Balanskat, A., Engelhardt, K., & Licht, A. H. (2018). *Strategies to include computational thinking in school curricula in Norway and Sweden*. Retrieved from http://www.eun.org/documents/411753/817341/Computational_thinking_report_2018.pdf/4d3d6fa0-dedd-4b62-a201-a26bf4dfd3a0

Barbiery, S., Vrgova, R., & Bliznakovski, J. (2013). *Overcoming of the ethnic segregation: how public schools to be integrated in Macedonia and Bosnia and Herzegovina, Why not*. Skopje: Sarajevo and Institute for Democracy Societas Civilis.

Barbopoulos, G. (2015). *Programming tools and syllabuses for preschool and school children*. School of Technology, Department of Informatics Engineering, Antirrio.

Barefoot Computing. (n.d.). *Barefoot. Unlock the future of your classroom*. Retrieved from https://www.barefootcomputing.org/

Barkley, E. F., Major, C. H., & Cross, K. P. (2014). *Collaborative Learning Techniques: A Handbook for College Faculty*. John Wiley & Sons, Inc. Retrieved from https://www.wiley.com/en-us/Collaborative+Learning+Techniques%3A+A+Handbook+for+College+Faculty%2C+2nd+Edition-p-9781118761557

Barnes, T., Richter, H., Chaffin, A., Godwin, A., Powell, E., Ralph, T., ... Jordan, H. (2007). Game2learn: a study of games as tools for learning introductory programming concepts. In *Proceedings of the ACM Special Interest Group on Computer Science Education.*. New York, NY: ACM.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning and Leading with Technology*, *38*(6), 20–23.

Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *Learning and Leading with Technology*, *38*(6), 20–23.

Barrows, H. S. (1996). Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*, *1996*(68), 3–12. doi:10.1002/tl.37219966804

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. doi:10.1145/1929887.1929905

Barshay, J. (2014). *Getting Boys – and Girls – Interested in Computer Coding*. http://educationbythenumbers.org/content/girls-computer-coding_1691/31

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010). Using scalable game design to teach computer science from middle school to graduate school. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, 224–228. 10.1145/1822090.1822154

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric approach. In J. J. Marciniak (Ed.), *Encyclopedia of Software Engineering* (pp. 528–532). New York: Wiley.

Battelle for Kids. (2019). Framework for 21st Century Learning. *Partnership For 21st Century Learning*. Retrieved from https://www.battelleforkids.org/networks/p21/frameworks-resources

Bausinger, H. (2009). *The popular culture in the world of technology*. Athens: Patakis.

BBC. (n.d.). *Introduction to computational thinking*. Retrieved from https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1

Becker, T. (2010). The Character of Successful Trainings with Serious Games. *International Journal of Emerging Technologies In Learning, 5*(SI3). Retrieved April 17, 2012, from http://online-journals.org/i-jet/article/view/1498

Becker, K., & Park, K. (2011). Integrative approaches among science, technology, engineering, and mathematics (STEM) subjects on students' learning: A meta-analysis. *Journal of STEM Education*, *12*(5), 23–37.

Beecham, S., Hall, T., Britton, C., Cottee, M., & Rainer, A. (2005). Using an expert panel to validate a requirements process improvement model. *Journal of Systems and Software*, *76*(3), 251–275. doi:10.1016/j.jss.2004.06.004

Bell, T., & Lodi, M. (2019). *Constructing computational thinking without using computers.* Retrieved March 8, 2020, from https://hal.inria.fr/hal-02378761/document

Bell, T., & Roberts, J. (2016). Computational thinking is more about humans than computers. *I SET, 1*, 3-7. Doi:10.18296et.0030

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008). CS without computers: new outreach methods from old tricks. In *Proceedings of 21st Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ)*. Auckland, New Zealand: Academic Press.

Bell, T., Witten, I. H., & Fellows, M. (1998). *CS Unplugged . . .off-line activities and games for all ages*. Retrieved from https://classic.csunplugged.org/wp-content/uploads/2015/01/unplugged-book-v1.pdf

Bell, T., Witten, I. H., & Fellows, M. (2015). *CS Unplugged An enrichment and extension programme for primary-aged students*. Retrieved from https://classic.csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf

Bell, J., & Bell, T. (2018). Integrating computational thinking with a music education context. *Informatics in Education*, *17*(2), 151–166. doi:10.15388/infedu.2018.09

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). CS Unplugged: School students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

Bell, T., Curzon, P., Cutts, Q., Dagiene, V., & Haberman, B. (2011). Overcoming obstacles to CS education by using non-programming outreach programmes. In I. Kalas & R. T. Mittermeir (Eds.), *ISSEP 2011, LNCS 7013* (pp. 71–81). Berlin: Springer. doi:10.1007/978-3-642-24722-4_7

Bell, T., & Lodi, M. (2019). Constructing Computational Thinking Without Using Computers. *Constructivist Foundations*, *14*(3), 342–351.

Bell, T., & Vahrenhold, J. (2018). CS Unplugged – How is it used, and does it work? In H. J. Bockenhauer, D. Komm, & W. Unger (Eds.), *Adventures between lower bounds and higher altitudes: Essays dedicated to Juraj Hromkovic on the occasion of his 60th birthday*. New York: Springer.

Belur, J., Tompson, L., Thornton, A., & Simon, M. (2018). Interrater reliability in systematic review methodology: Exploring variation in coder decision-making. *Sociological Methods & Research*.

Benekos, D. (2006). *Cultural Tradition and Education, Educational Approach - Educational Application*. TYPOTHITO-George Dardanos.

Benekos, D. (2008). *Aspects of Folklore, Folklore Analyses, Educational Applications, Text Research*. Gutenberg.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978–988. doi:10.1016/j.compedu.2011.10.006

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin Inroads*, *39*(2), 32–36. doi:10.1145/1272848.1272879

Bennett, V. E., Koh, K. H., & Repenning, A. (2013). Computing creativity: Divergence in computational thinking. In *Proceedings of the 44th ACM technical symposium on computer science education* (pp. 359-364). Denver, CO: ACM. 10.1145/2445196.2445302

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging Primary Programming and Mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, *3*(2), 115–138. doi:10.100740751-017-0028-x

Berninger, V. (2015). *Interdisciplinary frameworks for schools: Best professional practices for serving the needs of all students*. Washington, DC: American Psychological Association. doi:10.1037/14437-000

Berry, R. Q., Bull, G., Browning, C., Thomas, C. D., Starkweather, G., & Aylor, J. (2010). Use of digital fabrication to incorporate engineering design principles in elementary mathematics education. *Contemporary Issues in Technology & Teacher Education*, *10*(2), 167–172.

Bers, M. U. (2014). Tangible Kindergarten. *The Go-To Guide for Engineering Curricula PreK-5: Choosing and Using the Best Instructional Materials for Your Students, 133*, 145.

Bers, M. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York: Teacher's College Press.

Bers, M. U. (2008). *Blocks to Robots Learning with Technology in the Early Childhood Classroom*. New York, NY: Teachers College Press.

Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, *12*(2), 1–20.

Bers, M. U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, *12*(2), n2.

Bers, M. U. (2017). The Seymour test. *International Journal of Child-Computer Interaction*, *14*(100), 10–14. doi:10.1016/j.ijcci.2017.06.004

Bers, M. U. (2018a). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 8. doi:10.20897/ejsteme/3868

Bers, M. U. (2018b). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 2094-2102). IEEE. 10.1109/EDUCON.2018.8363498

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. doi:10.1016/j.compedu.2013.10.020

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130–145. doi:10.1016/j.compedu.2019.04.013

Bian, L., Leslie, S. J., & Cimpian, A. (2017). Gender stereotypes about intellectual ability emerge early and influence children's interests. *Science*, *355*(6323), 389–391. doi:10.1126cience.aah6524 PMID:28126816

Bikos, K., Stamovlasis, D., & Tzifopoulos, M. (2018). Dimensions of digital divide and relationships with social factors: A study of Greek pre-service teachers. *Themes in eLearning, 11*(1), 23-34.

Blackley, S., & Howell, J. (2015). A STEM narrative: 15 years in the making. *The Australian Journal of Teacher Education*, *40*(7), 102–112. doi:10.14221/ajte.2015v40n7.8

Blanchard, S., Freiman, V., & Lirrete-Pitre, N. (2010). Strategies used by elementary schoolchildren solving robotics-based complex tasks: Innovative potential of technology. *Procedia: Social and Behavioral Sciences*, *2*(2), 2851–2857. doi:10.1016/j.sbspro.2010.03.427

Bleumers, L., All, A., Mariën, I., Schurmans, D., Van Looy, J., Jacobs, A., ... de Grove, F. (2012). *State of play of digital games for empowerment and inclusion: A review of the literature and empirical cases.* European Commission.

Blickenstaff, J. C. (2005). Women and science careers: Leaky pipeline or gender filter? *Gender and Education*, *17*(4), 369–386. doi:10.1080/09540250500145072

Blikstein, P. (2013). Digital fabrication and 'making' in education: The democratization of invention. In FabLabs: Of Machines, Makers and Inventors, 4, 1-21.

Blikstein, P. (2018). *Pre-College Computer Science Education: a Survey of the Field*. Google LLC. Retrieved from https://goo.gl/gmS1Vm

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. Retrieved from https://goo.gl/dmWYfU

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education –Implications for policy and practice*. EUR 28295EN; European Commission. Retrieved from https://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint Research Centre (Seville site).

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). *Developing computational thinking in compulsory education.* European Commission, JRC Science for Policy Report.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice. Technical report, European Union Scientific and Technical Research Reports. EUR 28295 EN*. Publications Office of the European Union.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education –Implications for policy and practice. *EUR*, *28295EN*. Advance online publication. doi:10.2791/792158

Boechler, P., Artym, C., Dejong, E., Carbonaro, M., & Stroulia, E. (2014). Computational thinking, code complexity, and prior experience in a videogame-building assignment. In *2014 IEEE 14th International Conference on Advanced Learning Technologies* (pp. 396-398). IEEE. 10.1109/ICALT.2014.118

Bonderud, D. (2019). *How K–12 Classrooms Can Benefit from Robotics*. Retrieved from https://edtechmagazine.com/k12/article/2019/09/how-k-12-classrooms-can-benefit-robotics-perfcon

Borko, H., Whitcomb, J., & Liston, D. (2008). An Education President for the 21st Century. *Journal of Teacher Education*, *59*(3), 207–211. doi:10.1177/0022487108317803

Bowman, P. (2006). Standing at the Crossroads of Folklore and Education. *Journal of American Folklore*, *119*(471), 66–79. doi:10.1353/jaf.2006.0002

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 65-72). 10.1145/3137065.3137069

Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York: Springer. doi:10.1007/978-0-387-09506-6

Brandt, A. M., & Colton, M. B. (2008). *Toys in the classroom: LEGO MindStorms as an educational haptics platform. In 2008 symposium on haptic interfaces for virtual environment and teleoperator systems* (pp. 389–395). IEEE. doi:10.1109/HAPTICS.2008.4479982

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)* (vol. 1, pp. 1-25). Academic Press.

Brennan, K., & Resnick, M. (2012, April). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the annual meeting of the American Educational Research Association, Vancouver, British Columbia, Canada. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Brennan, K., Chung, M., Martin, W., Cervantes, F., Tally, B., & Resnick, M. (2019). *Supporting computational practices*. Harvard Graduate School of Education.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking* (pp. 1–25). Vancouver, BC, Canada: Annual American Educational Research Association Meeting. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association*.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the Annual Meeting of the American Educational Research Association*.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1-7), 107-117.

Briones-Garcia, S. J. G. (2019). *Diverse Learners in the Classroom: Students with Special Needs enrolled in Science, Technology, Engineering, and Mathematics (STEM) Texas Public Classrooms* (Dissertation). University of North Texas.

Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages: A way to learn programming principles. *Education and Information Technologies*, *2*(1), 65–83. doi:10.1023/A:1018636507883

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, *87*(4), 834–860. doi:10.3102/0034654317710096

Bull, G., & Garofalo, J. (2009). Personal fabrication systems: From bits to atoms. *Learning and Leading with Technology*, *36*(7), 10–12.

Bullock, D. (2004). Moving from theory to practice: An examination of the factors that preservice teachers encounter as they attempt to gain experience teaching with technology during field placement experiences. *Journal of Technology and Teacher Education*, *12*(2), 211–237.

Butler, Y. G. (2015). The use of computer games as foreign language learning tasks for digital natives. *System*, *54*, 91–102. doi:10.1016/j.system.2014.10.010

Büyüköztürk, Ş. (2005). *Handbook of data analysis for social sciences*. Ankara: PegemA.

Bybee, R. (2010). Advancing STEM Education: A 2020 Vision. *Technology and Engineering Teacher*, *70*(1), 30–35.

Caballero-Gonzalez, Y.-A., Muñoz-Repiso, A. G.-V., & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics. In *TEEM'19 Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 19–23). 10.1145/3362789.3362874

Caeli, E. N., & Yadav, A. (2019). *Unplugged approaches to computational thinking: A historical perspective.* Springer. doi:10.100711528-019-00410-5

Caldwell, H., & Smith, N. (2017). Teaching Computing Unplugged in Primary Schools. London: Learning Matters (SAGE Publications Ltd).

Çalik, M., Ebenezer, J., Özsevgeç, T., Küçük, Z., & Artun, H. (2015). Improving Science Student Teachers' Self-perceptions of Fluency with Innovative Technologies and Scientific Inquiry Abilities. *Journal of Science Education and Technology*, *24*(4), 448–460. doi:10.100710956-014-9529-1

Çalik, M., Özsevgeç, T., Ebenezer, J., Artun, H., & Küçük, Z. (2014). Effects of "Environmental Chemistry" Elective Course Via Technology-Embedded Scientific Inquiry Model on Some Variables. *Journal of Science Education and Technology*, *23*(3), 412–430. doi:10.100710956-013-9473-5

Callan, N. (2013). *ESL Jigsaws*. Retrieved from https://www.esljigsaws.com/assets/lessons/samplelesson.pdf

Campbell, C., Speldewinde, C., Howitt, C., & MacDonald, A. (2018). STEM practice in the early years. Creative Education Journal, 9(1), 11-25.

CAS. (2015). *Computing at School*. Retrieved from https://www.computingatschool.org.uk/

Castledine, A., & Chalmers, C. (2011). LEGO Robotics: An authentic problem solving tool? *Design and Technology Education: An International Journal, 16*(3), 19–27. Retrieved from https://ojs.lboro.ac.uk/DATE/article/view/1661/1562

Catlin, D., & Woollard, J. (2014). Educational Robots and Computational Thinking. Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, 144–151. 10.1017/CBO9781107415324.004

Çetinkaya, H. N. (2019). *Bilişim teknolojileri ve yazılım dersindeki etkinliklerin bilgi işlemsel düşünme ve bazı değişkenler açısından incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi. İnönü Üniversitesi, Eğitim Bilimleri Enstitüsü.

Chadha, K. (2014). *Improving the Usability of App Inventor through Conversion between Blocks and Text* (Honors thesis). Wellesley College.

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, *17*, 93–100. doi:10.1016/j.ijcci.2018.06.005

Chang, K. E., Wu, L. J., Weng, S. E., & Sung, Y. T. (2012). Embedding game-based problem-solving phase into problem-posing system for mathematics learning. *Computers & Education*, *58*(2), 775–786. doi:10.1016/j.compedu.2011.10.002

Chatzopoulos, A., Papoutsidakis, M., Kalogiannakis, M., & Psycharis, S. (2019). Action Research Implementation in Developing an Open Source and Low Cost Robotic Platform for STEM Education. *International Journal of Computers and Applications*, *178*(24), 33–46. doi:10.5120/ijca2019919039

Chaudhary, V., Agrawal, V., & Sureka, A. (2016). *An experimental study on the learning outcome of teaching elementary level children using Lego mindstorms EV3 robotics education kit.* arXiv preprint arXiv:1610.09610

Chen, P., & Huang, R. (2017). Design thinking in app inventor game design and development: a case study. In *Advanced Learning Technologies (ICALT), 2017 IEEE 17th International Conference on*, (pp.139–141). IEEE. 10.1109/ICALT.2017.161

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162–175. doi:10.1016/j.compedu.2017.03.001

Cheng, G. (2019). Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. *Computers in Human Behavior*, *92*, 361–372. doi:10.1016/j.chb.2018.11.043

Chevalier, M., Riedo, F., & Mondada, F. (2016). Pedagogical Uses of Thymio II: How Do Teachers Perceive Educational Robots in Formal Education? *IEEE Robotics & Automation Magazine*, *23*(2), 16–23. doi:10.1109/MRA.2016.2535080

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019). Educational Robotics in Primary School: Measuring the Development of Computational Thinking Skills with the Bebras Tasks. *Informatics (MDPI)*, *6*(43), 1–12. doi:10.3390/informatics6040043

Chiu, C. F. (2014). Use of problem-solving approach to teach scratch programming for adult novice programmers. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, (pp.710–711). ACM. 10.1145/2538862.2544284

Çiftci, S., & Bildiren, A. (2019). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer Science Education*, *§§§*, 1–19.

Cinganotto, L. (2017). *When content and learning join up, great things can happen*. Retrieved from https://www.schooleducationgateway.eu/en/pub/viewpoints/experts/when-content-and-learning-join.htm

Classic C. S. Unplugged. (n.d.). *Classic CS Unplugged*. Retrieved from https://classic.csunplugged.org/

Clements, D. H., & Sarama, J. (2016). Math, science, and technology in the early grades. *The Future of Children*, *26*(2), 75–94. doi:10.1353/foc.2016.0013

Clogg, C. C. (1995). Latent class models. In G. Arminger, C. C. Clogg, & M. E. Sobel (Eds.), *Handbook of Statistical Modeling for the Social and Behavioral Sciences* (pp. 311–359). New York, NY: Plenum. doi:10.1007/978-1-4899-1292-3_6

CODE. (2015). *Hour of Code*. Retrieved from https://hourofcode.com

Code.org. (n.d.). *Code.org About us*. Retrieved from https://code.org/about

CODECOMBAT. (2016). *Code Combat*. Retrieved from https://codecombat.com

CodeWeek. (n.d.). *Code Week*. Retrieved from https://codeweek.eu/about

Cohen, J. (1998). *Statistical Power Analysis for the Behavioral Sciences*. New York: Routledge Academic.

Cohen, L., Manion, L., & Morrison, K. (2002). *Research methods in education*. Routledge. doi:10.4324/9780203224342

Cohen, L., Manion, L., & Morrison, K. (2018). *Research Methods in Education* (8th ed.). London, UK: Routledge.

Colorado School of Mines. (n.d.). *CS Unplugged at Mines for Middle Schools*. Retrieved from http://csunplugged.mines.edu/

Combéfis, S., Beresnevičius, G., & Dagienė, V. (2016). Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, *10*(1), 39–60. doi:10.15388/ioi.2016.03

Computational Fairy Tales. (n.d.). *Computational Fairy Tales*. Retrieved from http://computationaltales.blogspot.com/

Computational Thinking with Scratch. (2012). *Computational thinking*. Retrieved from https://goo.gl/mxzTnp

Computer Science for Fun. (n.d.). *CS4FN: the fun side of CS*. Retrieved from http://www.cs4fn.org/

Computer Science Teachers Association, & International Society for Technology in Education. (2011). *Computational Thinking: Leadership Toolkit* (1st ed.) Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf

Computer Science Teachers Association. (2016). *K-12 Computer Science Framework*. Retrieved from https://k12cs.org/

Computer Science Teachers Association. (2017). *CSTA K-12 Computer Science Standards*. Revised 2017. Retrieved from http://www.csteachers.org/standards

Computing at School. (2013). *Computing in the national curriculum. A guide for primary teachers*. Bedford: Newnorth Print.

Computing at School. (2014). *Computing in the national curriculum. A guide for secondary teachers*. Bedford: Newnorth Print.

Computing at School. (2015). *Computational Thinking. A guide for Teachers*. Retrieved from https://community.computingatschool.org.uk/files/8550/original.pdf

Computing at School. (n.d.). *Computing at school*. Retrieved from https://www.computingatschool.org.uk/

Conde, M. Á., Fernandez-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellan-Olivera, V., & García-Peñalvo, F. J. (2017). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. In *Proceedings of International Conference Technological Ecosystems for Enhancing Multiculturality (TEEM 2017)*. 10.1145/3144826.3145355

Connolly, T., Stansfield, M., & Hainey, T. (2009). Towards the development of a games-based learning evaluation framework. In *Games-based learning advancements for multi-sensory human computer interfaces: Techniques and effective practices* (pp. 251–273). IGI Global. doi:10.4018/978-1-60566-360-9.ch015

Constitution of Greece, Resolution of the Z Revision Chamber of the Greeks, April 6th, 2001

Cooper, S., Dann, W., & Pausch, R. (2000). Developing algorithmic thinking with Alice. In The proceedings of ISECON (Vol. 17, pp. 506-539). Academic Press.

Corder, G. W., & Foreman, D. I. (2014). Nonparametric Statistics: A Step-by-Step Approach. *Wiley. Cura Tutorial.* Retrieved by https://all3dp.com/1/cura-tutorial-software-slicer-cura-3d/

Corder, G. W., & Foreman, D. I. (2014). *Nonparametric Statistics: A Step-by-Step Approach*. Wiley.

Cormen, T. H. (2009). *Introduction to algorithms*. MIT Press. cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Couse, L. J., & Chen, D. W. (2010). A tablet computer for young children? Exploring its viability for early childhood education. *Journal of Research on Technology in Education*, *43*(1), 75–96. doi:10.1080/15391523.2010.10782562

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*(3), 297–334. doi:10.1007/BF02310555

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational Thinking: a Guide for Teachers. *Computing at Schools*. Retrieved from https://community.computingatschool.org.uk/resources/2324/single

Cura Tutorial. (2019). https://all3dp.com/1/cura-tutorial-software-slicer-cura-3d/

Curriculum for "New School", Field: Culture - Art Activities for Compulsory Education, Athens. (2011a). Institute of Education - Ministry of Lifelong Learning and Religious Education. Available at: http://ebooks.edu.gr/

CurriculumS. (n.d.). https://www.maketolearn.org/inventions-kits/motor/activity-1-2/

Curzon, P. (2014). *Computing without computers. A gentle introduction to computer programming, data structures and algorithms*. Retrieved from https://teachinglondoncomputing.org/resources/inspiring-computing-stories/computing-withoutcomputers/

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). *Developing computational thinking in the classroom: a framework*. Retrieved from https://goo.gl/28ScgY

Curzon, P., & McOwan, P. W. (2017). *The Power of Computational Thinking*. London: World Scientific Publishing Europe Ltd. doi:10.1142/q0054

Curzon, P., McOwan, P., Cutts, Q., & Bell, T. (2009). Enthusing & Inspiring with Reusable Kinaesthetic Activities. *ACM SIGCSE Bulletin*, *41*(3), 94–98. doi:10.1145/1595496.1562911

Cussó-Calabuig, R., Farran, X. C., & Bosch-Capblanch, X. (2018). Effects of intensive use of computers in secondary school on gender differences in attitudes towards ICT: A systematic review. *Education and Information Technologies*, *23*(5), 2111–2139. doi:10.100710639-018-9706-6

Cutcher-Gershenfeld, J., Gershenfeld, A., & Gershenfeld, N. (2018). *Digital fabrication and the future of work, High Tech Futures*. Retrieved from http://cba.mit.edu/docs/papers/19.01.POW.pdf

Czerkawski, B. (2013). Instructional Design for Computational Thinking. In R. McBride & M. Searson (Eds.), *Proceedings of SITE 2013--Society for Information Technology & Teacher Education International Conference* (pp. 10-17). New Orleans, LA: Association for the Advancement of Computing in Education (AACE). Retrieved from https://www.learntechlib.org/primary/p/48062/

Dagiene, V., Stupuriene, G., & Vinikiene, L. (2016). Promoting inclusive informatics education through the Bebras challenge to all K-12 students. Proceedings of CompSys Tech '16. doi:10.1145/2983468.2983517

Dagienė, V., Pelikis, E., & Stupurienė, G. (2015). Introducing Computational Thinking through a Contest on Informatics: Problem-solving and Gender Issues. *Informacijos Mokslai*. *Information Sciences*, 73.

Dagiene, V., & Stupuriene, G. (2016). Bebras – a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in Education*, *15*(1), 25–44. doi:10.15388/infedu.2016.02

Damianou, K., Frosis, L., & Pateli, O. Cultural Diversity in the Classroom: Students' Needs and Expectations. In D. Sakkas, & A. Psaltis (Eds.), *Workshop Proceedings on" Cultural Diversity in School: The Role of the Teacher* (pp. 85-94). Alexandroupolis.

Daniela, L., & Lytras, M. D. (2018). *Educational Robotics for Inclusive Education*. Technology, Knowledge and Learning; doi:10.100710758-018-9397-5

Dasgupta, C., Magana, A. J., & Vieira, C. (2019). Investigating the affordances of a CAD enabled learning environment for promoting integrated STEM learning. *Computers & Education*, *129*, 122–142. doi:10.1016/j.compedu.2018.10.014

De Coito, I. (2014). Focusing on science, technology, engineering, and mathematics (STEM) in the 21st century. *Ontario Professional Surveyor, 57*(1), 34-36. Retrieved from https://www.krcmar.ca/resource-articles/2014_Winter_Focusing%20on%20STEM_0.pdf

de Freitas, S. (2006). *Learning in immersive worlds: A review of game-based learning*. Retrieved January 5, 2020, from https://researchrepository.murdoch.edu.au/id/eprint/35774/1/gamingreport_v3.pdf

de Freitas, S., & Jarvis, S. (2006). A Framework for Developing Serious Games to meet Learner Needs. *Interservice/Industry Training*, *Simulation and Education Conference*.

DeJarnette, N. (2012). America's children: Providing early exposure to STEM (science, technology, engineering and math) initiatives. *Education*, *133*(1), 77–84.

Dekhan, S., Xu, X., & Tsoi, M. Y. (2013). Mobile app development to increase student engagement and problem-solving skills. *Journal of Information Systems Education*, *24*(4), 299.

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832. doi:10.1016/j.compedu.2020.103832

Demetriou, K., & Nikiforidou, Z. (2019). The relational space of educational technology: Early childhood students' views. *Global Studies of Childhood*, *9*(4), 290–305. doi:10.1177/2043610619881458

Denzin, N. K., & Lincoln, Y. S. (2005). The Sage Handbook of Qualitative Research (3rd ed.). London, UK: SAGE Publications Ltd.

Department for Education. (2014). *The national curriculum in England. Framework document*. Retrieved from www.gov.uk/dfe/nationalcurriculum

DePryck, K. (2016). From computational thinking to coding and back. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 27-29). New York: ACM. 10.1145/3012430.3012492

Dermitzakis, C. (2006). Contemporary Forms of Popular Culture. In *Proceedings of the 1st Conference on Popular Culture & Education.* Volos: Research Institute of Culture & Education. Available at: http://www.eipe.gr/index.php/eisigiseis-2

Derrick, B., & White, P. (2017). Comparing Two Samples from an Individual Likert Question. *International Journal of Mathematics and Statistics*, *18*(3), 1–13.

DeVellis, R. F. (2003). *Scale development: theory and applications*. Thousand Oaks: SAGE Publications.

Dewey, J. (1986, September). Experience and education. In The Educational Forum (Vol. 50, No. 3, pp. 241-252). Taylor & Francis Group. doi:10.1080/00131728609335764

Dewey, R. A. (2018). *Psychology: An Introduction.* Retrieved from http://www.intropsych.com/ch01_ psychology_and_science/independent_dependent_and_subject_variables.html

Diamanou, D. (2013). The pedagogical dimensions of greek Folklore and the instruction of Folklore in Greek education. In *Bulletin of Greek Folklore* (Vol. 42). Hellenic Folklore Society.

Dillon, J., Rickinson, M., Teamey, K., Morris, M., Choi, M. Y., Sanders, D., & Benefield, P. (2006). The value of outdoor learning: Evidence from research in the UK and elsewhere. *The School Science Review*, *87*(320), 107–113.

DIMACS. (n.d.). *The Value of Computational Thinking across Grade Levels 9-12 (VCTAL)*. Retrieved from http://dimacs.rutgers.edu/archive/VCTAL/

Dlab, M. H., Hoić-Božić, N., Anđelić, M., & Botički, I. (2019). Digital games and tools for development of computational thinking in primary school. In *Proceedings of the International Conference on Management, Economics & Social Science-ICMESS* (pp. 1-7). Academic Press.

Dougherty, C. (2001). *The Raft of Odysseus: The Ethnographic Imagination of Homer's Odyssey*. Oxford University Press.

Dresel, M., Schober, B., & Ziegler, A. (2007). Golem und Pygmalion. Scheitert die Chancengleichheit von Mädchen im mathematisch-naturwissenschaftlich-technischen Bereich am geschlechtsstereotypen Denken der Eltern? In Erwartungen in Himmelblau und Rosarot. Effekte, Determinanten und Konsequenzen von Geschlechterdifferenzen in der Schule. Weinheim: Juventa.

Dreyfus, L. H. (2003). The Internet. Athens: Kritiki.

Dryden-Peterson, S. (2016). Refugee education in countries of first asylum: Breaking open the black box of pre-resettlement experiences. *Theory and Research in Education*, *14*(2), 131–148. doi:10.1177/1477878515622703

Dufoyer, J.-P. (1988). *Informatique, éducation et psychologie de l'enfant*. Paris: PUF.

Dundes, A. (2005). Folkloristics in the Twenty-First Century (AFS Invited Presidential Plenary Address, 2004). *Journal of American Folklore*, *118*(470), 385–408. doi:10.1353/jaf.2005.0044

Durak, H. Y. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology*. *Knowledge and Learning*, *25*(1), 179–195. doi:10.100710758-018-9391-y

Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational Thinking, Programming Self-Efficacy, Problem Solving and Experiences in the Programming Process Conducted with Robotic Activities. *Contemporary Educational Technology*, *10*(2), 173–197. doi:10.30935/cet.554493

Duschl, R., Schweingruber, H., & Shouse, A. (2007). *Taking Science to School: Learning and Teaching Science in Grades K-8*. Washington, DC: National Academies Press.

Economou, B. (2011). *The Scratch Programming Language*. Retrieved from https://economu.wordpress.com/h-language-programming-scratch /

Education, L. (2017). Education WeDo Toolbox, (v1), 31.

Education, L. (2019). *LEGO Education WeDo 2.0*. Retrieved January 4, 2020, from https://education.lego.com/en-us/elementary/intro/wedo2

Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2010* (pp. 4006-4014). Chesapeake, VA: AACE.

Eguchi, A. (2014a). Educational Robotics for promoting 21st century skills. *Journal of Automation. Mobile Robotics & Intelligent Systems*, *8*(1), 1–23. doi:10.14313/JAMRIS

Eguchi, A. (2014b). Learning experience through robocupjunior: Promoting engineering and computational thinking skills thourgh robotics competition. *ASEE Annual Conference and Exposition, Conference Proceedings*.

Eguchi, A. (2015). RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. *Robotics and Autonomous Systems*, *75*, 692–699. doi:10.1016/j.robot.2015.05.013

Eguíluz, A., Garaizar, P., & Guenaga, M. (2018). An Evaluation of Open Digital Gaming Platforms for Developing Computational Thinking Skills. In *Simulation and Gaming*. InTech. doi:10.5772/intechopen.71339

Eguíluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodríguez, C. (2017). Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis. *IEEE Transactions on Emerging Topics in Computing*, 1–1. doi:10.1109/TETC.2017.2768550

Ejiwale, J. A. (2013). Barriers to Successful Implementation of STEM Education. *Journal of Education and Learning*, *7*(2), 63–74. doi:10.1007/978-3-319-24436-5_20

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. *Computers in the Schools*, *33*(3), 169–186. doi:10.1080/07380569.2016.1216251

Engeser, S., Limbert, N., & Kehr, H. (2008). *Abschlussbericht zur Untersuchung Studienwahl Informatik*. Retrieved from: http://www.psy.wi.tum.de/Docs/Studienwahl_Informatik-Abschlussbericht.pdf

English, L., & King, D. (2015). STEM learning through engineering design: Fourth-grade students' investigations in aerospace. *International Journal of STEM Education*, *2*(1), 14. doi:10.118640594-015-0027-7

Eordanidis, S., Gee, E., & Carmichael, G. (2017). The effectiveness of pairing analog and digital games to teach computer science principles to female youth. *Journal of Computing Sciences in Colleges*, *32*(3), 12–19.

Ershov, A. P. (1981). *Programming, the second literacy*. Academic Press.

Ertl, B., Luttenberger, S., & Paechter, M. (2017). The Impact of Gender Stereotypes on the Self-Concept of Female Students in STEM Subjects with an Under-Representation of Females. *Frontiers in Psychology*, *8*, 703. doi:10.3389/fpsyg.2017.00703 PMID:28567022

Ertmer, P. A. (2005). Teacher pedagogical beliefs: The final frontier in our quest for technology integration? *Educational Technology Research and Development*, *53*(4), 25–39. doi:10.1007/BF02504683

Eshach, H., & Fried, M. N. (2005). Should science be taught in early childhood? *Journal of Science Education and Technology*, *14*(3), 315–336. doi:10.100710956-005-7198-9

Esper, S., Foster, S. P., Griswold, W. G., Herrera, C., & Snyder, W. (2014). *CodeSpells: bridging educational language features with industry-standard languages*. Koli Calling. doi:10.1145/2674683.2674684

Euclid, & Joyce, D. E. (1996). *Euclid's elements*. Clark University, Department of Mathematics and Computer Science.

European Commission. (2015). *She Figures 2015*. Retrieved from: https://ec.europa.eu/research/swafs/pdf/pub_gender_equality/she_figures_2015-final.pdf

European Commission. (2016). *The Computational Thinking Study*. Retrieved from https://goo.gl/y5TL2K

European Commission. (2018). *Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions on the Digital Education Action Plan*. Retrieved from https://ec.europa.eu

European Schoolnet. (2014). *Computing our future – Priorities, school curricula and initiatives across Europe*. Retrieved from https://goo.gl/WAjaCs

European Schoolnet. (2019). *Survey of Schools: ICT in education. Final report*. European Commission, DG Communications Networks, Content & Technology.

Eurostat (2019a). *Persons employed as ICT specialists and total employment, EU-28, 2006-2016 ICT*. Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php/File:Persons_employed_as_ICT_specialists_and_total_employment,_EU-28,_2006-2016_ICT17.png

Eurostat. (2019b). *ICT specialists - statistics on hard-to-fill vacancies in enterprises*. Retrieved from https://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_specialists_-_statistics_on_hard-to-fill_vacancies_in_enterprises

Eurostat. (2019c). *ICT specialists are predominantly male*. Retrieved from https://ec.europa.eu/eurostat/en/web/products-eurostat-news/-/DDN-20190513-1

Everett, L., Imbrie, P., & Morgan, J. (2000). Integrated curricula: Purpose and design. *Journal of Engineering Education*, *89*(2), 167–175. doi:10.1002/j.2168-9830.2000.tb00511.x

Faber, H., Wierdsma, M., Doornbos, R., van der Ven, J., & de Vette, K. (2017). Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Educational Network*, *12*, 13–24.

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning*, *32*(6), 576–593. doi:10.1111/jcal.12155

Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. (2011). Teaching CS Unplugged in the High School (with Limited Success). In *Proceedings of 16th Annual Conference on Innovation and Technology in CS Education (ITiCSE '11)*. 10.1145/1999747.1999817

Federici, S. (2011). A minimal, extensible, drag-and-drop implementation of the C programming language. In *Proceedings of the 2011 conference on Information technology education (SIGITE '11)*. New York: ACM. 10.1145/2047594.2047646

Ferrini-Mundy, J., Khargonekar, P., & Kurose, J. K. (2016). *Dear colleague letter*. Washington, DC: U.S. National Science Foundation. Retrieved from https://www.nsf.gov/pubs/2015/nsf15086.jsp

Fesakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97. doi:10.1016/j.compedu.2012.11.016

Fessakis, G., Komis, V., Mavroudi, E., & Prantsoudi, S. (2018). Exploring the scope and the conceptualization of Computational Thinking at the K-12 classroom level curriculum. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines: Foundations and Research Highlights*. Springer. doi:10.1007/978-3-319-93566-9_10

Feurzeig, W., Papert, S. A., & Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, *19*(5), 487–501. doi:10.1080/10494820903520040

Field, A. P. (2009). *Discovering statistics using SPSS* (3rd ed.). London: Sage.

Fincher, S., Cooper, S., Kölling, M., & Maloney, J. (2010). Comparing alice, greenfoot & scratch. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp.192–193). 10.1145/1734263.1734327

Fitzgerald, S., Hanks, B., Lister, R., McCauley, R., & Murphy, L. (2013). What are we thinking when we grade programs? In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 471-476). Denver, CO: ACM. 10.1145/2445196.2445339

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: Support for early childhood learning through computer programming. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 1-10). 10.1145/2485760.2485785

Flores, I. M. (2015). Developing Preservice Teachers' Self-Efficacy through Field-Based Science Teaching Practice with Elementary Students. *Research in Higher Education*, *§§§*, 27.

Fontichiaro, K. (2016). Help! My principal says I need to start a makerspace in my elementary library. *Teacher Librarian*, *44*(1), 49–51.

Ford, J. L. (2008). *Scratch Programming for Teens*. Boston, MA: Course Technology Press.

Foulidis, X. (2015). *New horizons in Greek Folklore Studies: The Internet as a field of production, dissemination and educational exploitation of Folk Culture* (Doctoral Thesis). University of the Aegean, Department of Preschool Education and Educational Design.

Foundation, F. A. B. (2019). *Scaling a Community of Practice for Education in STEM through Digital Fabrication Reflection and Playbook*. Academic Press.

Frazer, A., Recio, A., Gilbert, L., & Wills, G. (2014). Profiling the educational value of computer games. *Interaction Design & Architecture (s) Journal-IxD&A, 19*, 1–19.

Freudenthal, H. (1983). *Didactical phenomenology of mathematical structures*. Dordrecht: Reidel.

Fronza, I., El Ioini, N., & Corral, L. (2017). Leveraging Robot Programming to Foster Computational Thinking. doi:10.5220/0006310101090116

Full Scale. (2019). *Talent Shortage of Software Developers*. Retrieved from https://fullscale.io/talent-shortage-software-developers/

Fundamentals Unplugged, C. S. (n.d.). *Code.org CS Fundamentals Unplugged*. Retrieved from https://code.org/curriculum/unplugged

Funk, C., & Parker, K. (2018). Women and Men Often at Odds over Workplace Equit. Pew Research Center.

Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. In *Lecture Notes in Computer Science 4226* (pp. 159–168). Springer; doi:10.1007/11915355_15

Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. *Proceedings of the 2010 Constructionist Approaches to Creative Learning, Thinking and Education: Lessons for the 21st Century (Constructionism 2010)*, 1-10.

Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 155-164). Springer. 10.1007/978-3-642-24722-4_14

Gal, L., Hershkovitz, A., Eguíluz, A., Guenaga, M., & Garaizar, P. (2017). *Suggesting a Log-Based Creativity Measurement for Online Programming Learning Environment*. doi:10.1145/3051457.3054003

Gallagher, A. C. (2012). Jigsaw puzzles with pieces of unknown orientation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 382-389). IEEE. 10.1109/CVPR.2012.6247699

Gallagher, T. (2010). *Key Issues in Coexistence and Education*. Coexistence International at Brandeis University.

Garcia, P. G. F., & De la Rosa, F. (2016). RoBlock–Web App for Programming Learning. International. *Journal of Emerging Technologies in Learning*, *11*(12), 45–53. doi:10.3991/ijet.v11i12.6004

García-Peñalvo, F. J. (2017, October). Computational thinking issues. In *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality* (p. 1). ACM.

García-Peñalvo, F. J., & Cruz-Benito, J. (2016). Computational thinking in pre-university education. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*. 10.1145/3012430.3012490

García-Peñalvo, F. J., & Mendes, A. J. (2017). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, *80*, 407–411. doi:10.1016/j.chb.2017.12.005

Gardner, H. (2011). *Frames of mind: The theory of multiple intelligences*. Hachette UK.

Gasouka, M., Kapaniaris, A., Arvanitidou, Z., Foulidi, X., & Raptou, E. (2013). Folk culture and digital enriched books: The case study of thw digital enriched toll "E-Anagnosis". *Proceedings of INTED2013 Conference,* 4176-4184.

Gasoukas, M., & Foulidis, X. (2012). *New Horizons of Folklore Studies - Conceptual Framework, Research, Gender, Internet, School*. Athens: Sideris Publications.

Gasoukas, M., & Xanthippi, F. (2012). *Contemporary Horizons of Folklore Studies, Conceptual Framework, Research, Gender, Internet, School*. I. Sideris.

Gershenfeld, N. (2008). *Fab: the coming revolution on your desktop--from personal computers to personal fabrication*. Basic Books.

Gerstein, J. (2019). *Learning in the Making: How to Plan, Execute, and Assess Powerful Makerspace Lessons*. ASCD.

Giannakoulas, A., & Xinogalos, S. (2018). *A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. In Education and Information Technologies*. Springer. doi:10.100710639-018-9702-x

Gomes, T. C. S., Falcão, T. P., & Tedesco, P. C. D. A. R. (2018). Exploring an approach based on digital games for teaching programming concepts to young children. *International Journal of Child-Computer Interaction*, *16*, 77–84. doi:10.1016/j.ijcci.2017.12.005

Gonida, E., Karabenick, S., Stamovlasis, D., Metallidou, P., & Greece, C. T. Y. (2018). Help seeking as a self-regulated learning strategy and achievement goals: The case of academically talented adolescents. *High Ability Studies*, *30*(1-2), 147–166. doi:10.1080/13598139.2018.1535244

Gonzalez, H. B., & Kuenzi, J. J. (2012). Science, technology, engineering, and mathematics (STEM): A Primer. *Congressional Research Service*, 1–15. Retrieved from https://www.ccc.edu/departments/Documents/STEM_labor.pdf

González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., & Infante-Moro, A. (2019). Computational Thinking and Down Syndrome: An Exploratory Study Using the KIBO Robot. *Informatics (MDPI)*, *6*(2), 25–45. doi:10.3390/informatics6020025

Google for Education. (n.d.). Retrieved from https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-materials

Government Gazette (2014). *Curriculum for Informatics courses in High School*. Greek Ministry of Education, FEK 932/14-4-2014.

Govindarajan, M. (2019). *Families That Code Together Learn Together: Exploring Family-Oriented Programming in Early Childhood with ScratchJr and KIBO Robotics* (Doctoral dissertation). Tufts University.

Goyal, S., Chopra, S., & Mohanan, D. (2017, October). CodeFruits: Teaching Computational Thinking Skills Through Hand Gestures. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play* (pp. 291-298). ACM. https://dl.acm.org/citation.cfm?id=3131335

Gray, J., Abelson, H., Wolber, D., & Friend, M. (2012). Teaching CS principles with app inventor. In *Proceedings of the 50th Annual Southeast Regional Conference* (pp.405–406). ACM. 10.1145/2184512.2184628

Greenberg, R. I., & Reed, D. F. (2018). Using magic in computing education and outreach. *Proceedings of 2018 IEEE Frontiers in Education Conference*. 10.1109/FIE.2018.8658626

Greenfoot. (n.d.). Retrieved from https://greenfoot.org/download

Gresse von Wangenheim, C., Hauck, J. C. R., Demetrio, M. F., Pelle, R., Alves, N. da C., Barbosa, H., & Azevedo, L. F. (2018). CodeMaster – automatic assessment and grading of app inventor and Snap! programs. *Informatics in Education*, *17*(1), 117–150. doi:10.15388/infedu.2018.08

Gribbons, B., & Herman, J. (1996). True and quasi-experimental designs. *Practical Assessment, Research & Evaluation*, *5*(1), 14.

Grover, S. (2011). Robotics and engineering for middle and high school students to develop computational thinking. *Annual Meeting of the American Educational Research Association,* (650), 1–15. Retrieved from http://web.stanford.edu/~shuchig/docs/AERA2011-Shuchi Grover-Robotics and Engineering for Middle and High School Students to Develop Computational Thinking.pdf

Grover, S. (2015). "Systems of Assessments" for deeper learning of computational thinking in K-12. In *Proceedings of the 2015 annual meeting of the American educational research association* (pp. 15-20). Academic Press.

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School, 19*.

Grover, S., & Basu, S. (2017). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 267–272. 10.1145/3017680.3017723

Grover, S., Basu, S., & Schank, P. (2018). What We Can Learn About Student Learning From Open-Ended Programming Projects in Middle School Computer Science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 999-1004). Baltimore, MD. ACM. 10.1145/3159450.3159522

Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 57–62). Uppsala, Sweden. ACM.

Grover, S., Lundh, P., & Jackiw, N. (2019). Non-Programming Activities for Engagement with Foundational Concepts in Introductory Programming. In *Proceedings of 50th ACM Technical Symposium on CS Education* (pp. 1136–1142). New York: ACM. 10.1145/3287324.3287468

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. doi:10.3102/0013189X12463051

Grubbs, M. (2013). Robotics Intrigue middle school students and build STEM skills. *Technology and Engineering Teacher*, *72*(6), 12–16.

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği: Geçerlik ve Güvenirlik Çalışması. *Turkish Journal of Computer and Mathematics Education*, *10*(1), 1–29.

Guzdial, M. (2018). *Confusion over the Forms of Programming Problems: Mathematics/Physics versus CS.* Retrieved from https://computinged.wordpress.com/2018/02/26/confusion-over-the-formsof-programming-problems-mathematics-physics-versus-cs/

Guzdial, M. (2004). Programming environments for novices. In S. Fincher & M. Petre (Eds.), *Computer Science Education Research* (pp. 127–154). Abingdon, UK: Taylor & Francis.

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, *8*(6), 1–165. doi:10.2200/S00684ED1V01Y201511HCI033

Halverson, E. R., & Sheridan, K. (2014). The maker movement in education. *Harvard Educational Review*, *84*(4), 495–504. doi:10.17763/haer.84.4.34j1g68140382063

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *SIGCSE Bulletin*, *41*(1), 183–187. doi:10.1145/1539024.1508931

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2019). An Emerging Technology Report on Computational Toys in Early Childhood. *Technology, Knowledge and Learning*, 1-12.

Hao, S. (2016). *Effects of Faded Scaffolding in Computer-Based Instruction on Learners' Performance, Cognitive Load, and Test Anxiety*. Retrieved from http://purl.flvc.org/fsu/fd/FSU_2016SP_Hao_fsu_0071E_12940

Harris, J., Mishra, P., & Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education*, *41*(4), 393–416. doi:10.1080/15391523.2009.10782536

Harvey, B., & Monig, J. (2010). Bringing 'No Ceiling' to Scratch: Can One Language Serve Kids and Computer Scientists? Proceedings of Constructionism 2010.

Heath, O. (2016). *5 reasons why DIY is good for you*. Retrieved from https://www.housebeautiful.com/uk/renovate/diy/tips/a1131/why-diy-is-good-for-your-wellbeing/

Heilig, S. (2015). K-12 STEM Education. *The Institute for the Promotion of Teaching Science and Technology, 1*(2), 112.

Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Proceedings of IEEE Frontiers in Education Conference (FIE) 2016*. Erie: IEEE. 10.1109/FIE.2016.7757410

Herdzina, J. (2019). *Young children: What's code got to do with it? Guest blog by Sarah Pila*. Retrieved from http://teccenter.erikson.edu/what-were-reading/pila_coding/

Hermans, F., & Aivaloglou, E. (2017). To Scratch or not to Scratch? In *Proceedings of 12th Workshop on Primary and Secondary Computing Education* (pp. 49-56). Nijmegen, Netherlands: Academic Press.

Hicks, A., Catete, V., & Barnes, T. (2014). Part of the game: Changing level creation to identify and filter low-quality user generated levels. In *Proceedings of the international conference on the foundations of digital games* (pp. 242-245). ACM.

Higgins, J. M. (2013). The Future of Jobs. *World Future Review*, *5*(, 11–23. doi:10.1177/1946756712473437

Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting education in "educational" apps: Lessons from the science of learning. *Psychological Science in the Public Interest*, *16*(1), 3–34. doi:10.1177/1529100615569721 PMID:25985468

Hoachlander & Yanofsky. (2011, March). Making STEM Real (2011). *Educational Leadership*, *68*(6), 60–65.

Holt, L. (2011). *Creating Digital Stories with Scratch to Promote Computational Thinking*. Paper presented at the Society for Information Technology & Teacher Education International Conference.

Hour of Code. (n.d.). *Hour of Code activities from Code.org*. Retrieved from https://code.org/hourofcode/overview

Howard, G. R. (2016). The New Regularity of Digital Web Hybridization. In E. Avdikos (Ed.), *Internet Cultures*. Athens: Pedio.

Hromkovič, J., Kohn, T., Komm, D., & Serafini, G. (2016). Examples of algorithmic thinking in programming education. *Olympiads in Informatics*, *10*(1-2), 111–124. doi:10.15388/ioi.2016.08

Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. doi:10.1016/j.compedu.2018.07.004

Hughes, J., & Donnelly, C. (2006). *Contact as a mechanism for promoting better relations in integrated schools in Northern Ireland and bilingual/bi-national schools in Israel*. Paper presented at IACM 18TH Annual Conference.

Hughes, J., Gadanidis, G., & Yiu, C. (2017). Digital Making in Elementary Mathematics Education. *Digital Experiences in Mathematics Education*, *3*(2), 139–153. doi:10.100740751-016-0020-x

Huh, J. H., Kim, H. B., & Seo, K. (2016). A design of smart-based education gamification platform using mobile devices. *Advanced Science and Technology Letters*, *127*, 85–90. doi:10.14257/astl.2016.127.18

Hussain, A. J., Fergus, P., Al-jumeily, D., Pich, A., & Hind, J. (2015). Teaching Primary School Children the Concept of Computer Programming. *2015 International Conference on Developments of E-Systems Engineering*, 180–184. 10.1109/DeSE.2015.57

Imberman, S. P., Sturm, D., & Azhar, M. Q. (2014). Computational thinking: Expanding the toolkit. *Journal of Computing Sciences in Colleges*, *29*(6), 39–46.

Instefjord, E. J., & Munthe, E. (2017). Educating digitally competent teachers: A study of integration of professional digital competence in teacher education. *Teaching and Teacher Education*, *67*, 37–45. doi:10.1016/j.tate.2017.05.016

Instructables, Yours for the making. (2020). Retrieved from https://www.instructables.com/

Ioannidou, A., Bennett, V., Repenning, A., Koh, H., & Basawapatna, A. (2011). Computational Thinking Patterns. *Annual Meeting of the American Educational Research Association, 2*. Retrieved from http://www.agentsheets.com

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544. doi:10.100710639-018-9729-z

Ioannou, M., & Bratitsis, T. (2016). Utilizing Sphero for a speed related STEM activity in Kindergarten. *Hellenic Conference on Innovating STEM Education.*

ISTE & CSTA. (2011). *Operational definition of Computational Thinking in K-12 education*. Retrieved from https://goo.gl/AlSSp

IWG. (2010). *Computational Thinking for Youth*. Newton, MA: Education Development Center, Inc.

Jagušt, T., Sovic Krzic, A., Gledec, G., Grgic, M., & Bojic, I. (2018). Exploring Different Unplugged Game-like Activities for Teaching Computational Thinking. In *Proceedings of 2018 Frontiers in Education Conference (FIE)*. IEEE. 10.1109/FIE.2018.8659077

James, I., & Novak, I. J. (2019). *Re-educating the Educators: Collaborative 3D Printing Education. In Interdisciplinary and International Perspectives on 3D Printing in Education* (pp. 28–49). IGI Global.

Jamieson, P. (2011). Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat? In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Jiang, S., & Wong, G. K. W. (2017). Assessing primary school students' intrinsic motivation of computational thinking. In *Proceedings of 2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE).* IEEE.

Johnson, L., Smith, R., Willis, H., Levine, A., & Haywood, K. (2011). *The Horizon Report* (2011 Edition). Austin, TX: The New Media Consortium.

Jonassen, D., Howland, J., Marra, R., & Crismond, D. (2011). *Meaningful learning with Technology* (V. Mitropoulou, Trans.). Methexis.

Jordaan, D. B. (2018). Board games in the computer science class to improve students' knowledge of the Python programming language. In *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)* (pp.1–5). IEEE. 10.1109/ICONIC.2018.8601207

Juhadi, J., Nugraha, S. B., & Banowati, E. (2017). Disaster Education Model for Early Childhood. *Sumatra Journal of Disaster*. *Geography and Geography Education*, *1*(2), 89–93.

K–12 CS Framework. (2016). *K–12 CS Framework*. Retrieved from http://www.k12cs.org

Kafai, Y. (2016). From computational thinking to computational participation in K–12 education. *Communications of the ACM*, *59*(8), 26–27. doi:10.1145/2955114

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture*, *1*(1), 36–40. doi:10.1177/1555412005281767

Kafai, Y. B., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, *95*(1), 61–65. doi:10.1177/003172171309500111

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, *50*(4), 313–334. doi:10.1080/00461520.2015.1124022 PMID:27019536

Kafai, Y. B., Fields, D. A., & Searle, K. A. (2011, November). Everyday creativity in novice e-textile designs. In *Proceedings of the 8th ACM conference on Creativity and cognition* (pp. 353-354). 10.1145/2069618.2069692

Kafai, Y. B., Searle, K., Kaplan, E., Fields, D., Lee, E., & Lui, D. (2013, March). Cupcake cushions, scooby doo shirts, and soft boomboxes: e-textiles in high school to promote computational concepts, practices, and perceptions. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 311-316). 10.1145/2445196.2445291

Kakampoura, P. (2010). Teaching Folk Culture in Primary Education. In *Proceedings of the 1st Panhellenic Congress of Education Sciences*. Athens: Smyrniotakis.

Kalawsky, R. (1994). *The Science of Virtual Reality and Virtual Environments: A Technical, Scientific and Engineering Reference on Virtual Environments*. Addison Wesley Publishing Company.

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic J. Modern Computing*, *4*(3), 583–596.

Kalogiannakis, M., & Papadakis, S. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In *Proceedings of the 8th International Conference On Computer Based Learning in Science* (pp. 265-276). Academic Press.

Kalogiannakis, M., & Papadakis, S. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. *Training, 6*, 2.

Kalogiannakis, M., & Papadakis, St. (2007). The dual form of further education of educators in ICT: technological and pedagogical training. In C. Constantinou, Z. Zacharias & M. Papaevripidou (Eds.), *Proceedings of the 8th International Conference On Computer Based Learning in Science*, (pp. 265-276). Academic Press.

Kalogiannakis, M., & Papadakis, St. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. In *Proceeding of the 5th WSEAS/IASME International Conference on Engineering Education* (pp. 305-311). Academic Press.

Kalogiannakis, M., Rekoumi, C., Antipa, E., & Poullou, V. (2010). Preschool education and geology within the scope of environmental education: the case of a teaching intervention at kindergarten. In *Proceedings of the 3rd World Conference on Science and Technology Education (ICASE 2010), Innovation in Science and Technology Education: Research, Policy, Practice, Tartu-Estonia* (*Vol. 28*, pp. 159-163). Academic Press.

Kalogiannakis, M. (2008). From Learning to Use ICT to Use ICT for Learning: Technological Capabilities and Pedagogical Principles. In R. Kobayashi (Ed.), *New Educational Technology* (pp. 13–42). New York: Nova Publishers.

Kalogiannakis, M., & Kanaki, K. (2020). Introducing Computational Thinking Unplugged in Early Childhood Education Within the Context of Physical and Natural Science Courses: A Pilot Study in Greece. In J. Keengwe & P. Wachira (Eds.), *Handbook of Research on Integrating Computer Science and Computational Thinking in K-12 Education* (pp. 164–190). Hershey, PA: IGI Global. doi:10.4018/978-1-7998-1479-5.ch010

Kalogiannakis, M., & Papadakis, S. (2020). The Use of Developmentally Mobile Applications for Preparing Pre-Service Teachers to Promote STEM Activities in Preschool Classrooms. In *Mobile Learning Applications in Early Childhood Education* (pp. 82–100). IGI Global. doi:10.4018/978-1-7998-1486-3.ch005

Kalogiannakis, M., & Papadakis, St. (2008). Hybrid learning for women and socially sensitive groups for the promotion of digital literacy. *Proceeding of the 5th WSEAS/IASME International Conference on Engineering Education*, 305-311.

Kalogiannakis, M., Vassilakis, K., Alafodimos, C., Papadakis, S., Papachristos, D., & Zafeiri, E. (2009). Adult Education and Lifelong Learning. The case of GSAE (General Secretary for Adult Education) in Greece. *International Journal of Advanced Corporate Learning*, *2*(4), 15–20. doi:10.3991/ijac.v2i4.981

Kalovrektis, K. (2015). Getcoding for everyone - The first steps in the world of the Code. Academic Press.

Kanaki, K., & Kalogianakis, M. (2019). Assessing Computational Thinking Skills at First Stages of Schooling. In *Proceedings of the 2019 3rd International Conference on Education and E-Learning* (pp. 135-139). New York: ACM. doi:10.1145/3371647.3371651

Kanaki, K., & Kalogiannakis, M. (2018). Introducing fundamental object-oriented programming concepts in preschool education within the context of physical science courses. *Education and Information Technologies*, *23*(6), 2673–2698. doi:10.100710639-018-9736-0

Kanoukas, H., & Anderson, T. (1999). Using constructivism in technology-mediated learning: Constructing order out of the chaos in the literature. *Radical Pedagogy*, *1*(1). http:// radicalpedagogy.icaap.org/content/ issue1_2/02kanuka1_2.html

Kantrowitz, A. (2019). *Maker Ed in the Art Studio: Preparing Pre-Service Artist/Educators to Integrate Maker Education into P-12 Art Studio/Classrooms*. Retrieved from: https://nyc2019.fablearn.org/wp-content/uploads/2019/04/FL2019_paper_109.pdf

Kapaniaris, A. (2013). From electronic to enriched e-book: digital enrichment tools in education. Proceedings of Interdisciplinary Conference "History Information: references from papyrus to electronic document." Athens: Department of Archives and Libraries of the Ionian University.

Kapaniaris, A. (2013a). From e-books to enriched e-books: Tools for digital enrichment in education. In Interdisciplinary Conference "History of Information: Papyrus Reference Points in Electronic Document. Athens: Department of Archives and Libraries of the Ionian University.

Kapaniaris, A. (2013b). Cultural Archives and Folklore Collections: The Role of Informatics and Information Technologies in the Formulation of Digital Educational Materials. In *Proceedings of the 7th Congress of Postgraduate Students and Doctoral Candidates*. Athens: Department of Philology, National and Kapodistrian University of Athens.

Kapaniaris, A., Gasouka, M., Zisiadis, D., Papadimitriou, E., & Kalogirou, E. (2013a). Learning Object Design and Development in Folklore Education Using Web 2.0 Tools. Mediterranean Journal of Social Sciences, 4(11), 104-110.

Kapaniaris, A., Gasouka, M., Zisiadis, D., Papadimitriou, E., & Kalogirou, E. (2013b). Digital Books Taxonomy: From Text E-Books to Digitally Enriched E-Books in Folklore Education Using the iPad. in Mediterranean Journal of Social Sciences, 4(11), 316-322.

Kapaniaris, A., & Gasoukas, M. (2016). Folk Culture and Enriched E-Books: Designing Learning Resources for Internet-Based Education. In E. Avdikos (Ed.), *Internet Cultures*. Athens: Pedio.

Karagöz, Z. (2016). *The robotel Türkiye project*. Retrieved February 27, 2017, from http://enablingthefuture.org/2016/12/05/enablingturkey/

Karakus, M., Uludag, S., Guler, E., Turner, S. W., & Ugur, A. (2012, June). Teaching computing and programming fundamentals via App Inventor for Android. In *2012 International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1-8). IEEE. 10.1109/ITHET.2012.6246020

Karasar, N. (2005). *Scientific research method*. Ankara: Nobel Publishing.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255. doi:10.100710643-012-0554-5

Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergar-ten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, *21*(4), 371–391.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2011). Understanding computational thinking before programming: Developing guidelines for the design of games to learn introductory programming through game-play. *International Journal of Game-Based Learning*, *1*(3), 30–52. doi:10.4018/ijgbl.2011070103

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia: Social and Behavioral Sciences*, *47*, 1991–1999. doi:10.1016/j.sbspro.2012.06.938

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522–531. doi:10.1016/j.procs.2012.04.056

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, *37*(2), 83–137. doi:10.1145/1089733.1089734

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, *50*(7), 58–64. doi:10.1145/1272516.1272540

Kelleher, C., Pausch, R., Pausch, R., & Kiesler, S. (2007, April). Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). ACM. 10.1145/1240624.1240844

Kent State University Libraries. (2017). *SPSS tutorials: Independent samples t test.* Retrieved from https://libguides.library.kent.edu/SPSS/IndependentTTest

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and Higher Education*, *8*(1), 13–24. doi:10.1016/j.iheduc.2004.12.001

Kindon, S., Pain, R., & Kesby, M. (2007). *Participatory Action Research Approaches and Methods*. London: Routledge.

Kinzie, M. B., & Joseph, D. R. (2008). Gender differences in game activity preferences of middle school children: Implications for educational game design. *Educational Technology Research and Development*, *56*(5-6), 643–663. doi:10.100711423-007-9076-z

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews* (Vol. 4). CRC Press. doi:10.1201/b19467

Kleinschmager, S., & Hanenberg, S. (2011). How to Rate Programming Skills in Programming Experiments? A Preliminary, Exploratory, Study Based on University Marks, Pretests, and Self-Estimation. In *Proc. ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools* (pp. 15–24). Portland, OR: ACM.

Knuth, D. (1968). *The art of computer programming 1: Fundamental algorithms 2: Seminumerical algorithms 3: Sorting and searching*. Addison-Wesley.

Kodotchigova, M. A. (2002). Role Play in Teaching Culture: Six Quick Steps for Classroom Implementation. *The Internet TESL Journal, 8*(7). Retrieved from http://iteslj.org/Techniques/Kodotchigova-RolePlay.html

Koh, K. H., Basawapatna, A., Nickerson, H., & Repenning, A. (2014). Real Time Assessment of Computational Thinking. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 49-52). Melbourne, Australia. IEEE.

Koning, J. I., Faber, H. H., & Wierdsma, M. D. M. (2017). Introducing computational thinking to 5 and 6 year old students in Dutch primary schools; an educational design research study. In *Proceedings of 17th Koli Calling International Conference on Computing Education Research* (pp. 189–190). New York: ACM.

Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, *82*, 162–178. doi:10.1016/j.compedu.2014.11.020

Kordaki, M., & Kakavas, P. (2017). Digital Storytelling as an effective framework for the development of computational thinking skills. *EDULEARN2017*, 3-5.

Korkmaz, Ö. (2018). The effect of scratch-and lego mindstorms Ev3-Based programming activities on academic achievement, problem-solving skills and logical-mathematical thinking skills of students. MOJES. *Malaysian Online Journal of Educational Sciences*, *4*(3), 73–88.

Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., & Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, *34*(2), 68–87.

Korpi, J. (2014). *Teaching programming to children through games* (Master of Science Thesis). Tampere University of Technology.

Kozhevnikov, M. (2007). Cognitive styles in the context of modern psychology: Toward an integrated framework of cognitive style. *Psychological Bulletin*, *133*(3), 464–481. doi:10.1037/0033-2909.133.3.464 PMID:17469987

Krul, K. Y. (2012). *Teaching Control Structures Using App Inventor*. Master thesis. Retrieved from https://dspace.library.uu.nl/bitstream/handle/1874/254527/researchpaper-kevinkrul-25-6.pdf?sequence=1

Kubica, J. (2013). Best practices of spell design. A computational fairy tale. San Bernardino, CA: Academic Press.

Kubica, J. (2012). *Computational fairy tales*. Monee.

Kubica, J. (2016). *The CS detective. An algorithmic tale of crime, conspiracy, and computation*. San Francisco: No Starch Press.

Kukul, V. (2018). *The effect of different structured processes on students' computational thinking skills, self-efficacy levels and programming achievements in programming teaching* (Unpublished PhD Thesis). Gazi University, Turkey

Kunduracıoğlu, İ. (2018). Examining the interface of lego Mindstorms ev3 robot programming. *Journal of Educational Technology and Online Learning*, *1*(1), 28–46. doi:10.31681/jetol.372826

Lamagna, E. A. (2015). Algorithmic thinking unplugged. *Journal of Computing Sciences in Colleges*, *30*(6), 45–52.

Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in Colleges*, *24*(3), 118–124.

Lamprou, A., & Repenning, A. (n.d.). Computational Thinking ≠ Programming. *SI Magazine*. Retrieved from https://magazine.swissinformatics.org/en/computational-thinking-%E2%89%A0-programming/

Lanier, J., Minsky, M., Fisher, S., & Druin, A. (1989). Virtual Environments and Interactivity: Windows To The Future. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques.* ACM.

Laporte, L., & Zaman, B. (2016). Informing Content-driven Design of Computer Programming Games: a Problems Analysis and a Game Review. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (p. 10). ACM. 10.1145/2971485.2971499

Larman, C., & Basili, V. (2003). Iterative and incremental developments: A brief history. *IEEE Computer*, *36*(6), 47–56. doi:10.1109/MC.2003.1204375

Lavidas, K., Komis, V., & Gialamas, V. (2012). Spreadsheets as cognitive tools: A study of the impact of spreadsheets on problem solving of math story problems. *Education and Information Technologies*, *18*(1), 113–129. doi:10.100710639-011-9174-8

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. doi:10.1145/1929887.1929902

Lee, J. (2019). Coding in early childhood. *Contemporary Issues in Early Childhood*, 1463949119846541.

Lee, J., & Junoh, J. (2019). Implementing Unplugged Coding Activities in Early Childhood Classrooms. *Early Childhood Education Journal*, *47*(6), 709–716. doi:10.100710643-019-00967-z

Lee, V. R., & Vincent, H. (2019). An expansively-framed unplugged weaving sequence intended to bear computational fruit of the loom. In P. Blikstein, & N. Holbert (Eds.), *Proceedings of FabLearn 2019*. New York: ACM. 10.1145/3311890.3311907

Lego. (2016a). *LEGO Education WeDo 2. 0 - Computational Thinking - Teacher's Guide*. LEGO Group. Retrieved from https://le-www-live-s.legocdn.com/sc/media/files/user-guides/wedo-2/computational-thinking-teacher-guides/computationalthinkingteacherguide-en-gb-v1-69cd677d61d21b2a44999328c40e7122.pdf

Lego. (2019). *LEGO Education – LEGO History.* Retrieved December 30, 2019, from https://www.lego.com/en-us/lego-history/lego-education-604484c3b3de4314a678ef280d04d216

Lego, E. (2016b). *Teacher ' s Guide LEGO Education WeDo 2. 0 Computational Thinking*. The LEGO Group.

Leon, J. A., Abad Jiménez, W. E., Barraza Ramírez, J. L., Frontado Aldana, S. A., Puma Alvites, A. T. A., & Solórzano Pinzás, R. L. (2019). Work in progress: Llup'ix. In *Proceedings of 2019 IEEE World Conference on Engineering Education (EDUNINE)*. IEEE. 10.1109/EDUNINE.2019.8875816

Lester, J. C., Spires, H. A., Nietfeld, J. L., Minogue, J., Mott, B. W., & Lobene, E. V. (2014). Designing game-based learning environments for elementary science education: A narrative-centered learning perspective. *Information Sciences*, *264*, 4–18. doi:10.1016/j.ins.2013.09.005

Lieberman, D., Bates, C., & So, J. (2009a). Young children's learning with digital media. *Computers in the Schools*, *26*(4), 271–283. doi:10.1080/07380560903360194

Lieberman, D., Fisk, C. M., & Biely, E. (2009b). Digital games for young children ages three to six: From research to design. *Computers in the Schools*, *26*(4), 299–313. doi:10.1080/07380560903360178

Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology*, *50*(4), 1979–1995. doi:10.1111/bjet.12685

Liu, A., Newsom, J., Schunn, C., & Shoop, R. (2013). Students Learn Programming Faster through Robotic Simulation. *Tech Directions*, *72*(8), 16–19.

Liu, J., Lin, C.-H., Potter, P., Philip, E., Zebulun, H., Barnett, D.-B., & Singleton, M. (2013). Going mobile with app inventor for android: a one-week computing workshop for K-12 teachers. *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13)*, 433-438. 10.1145/2445196.2445324

Liukas, L. (2015). *Hello Ruby: Adventures in coding*. New York: Feiwel & Friends.

Liukas, L. (2017). *Hello Ruby: Journey inside the computer*. New York: Feiwel & Friends.

Liukas, L. (2018). *Hello Ruby: Expedition to the internet*. New York: Feiwel & Friends.

Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, *27*(1), 1–29. doi:10.1080/08993408.2017.1308651

Livingstone, S., & Locatelli, E. (2012). Ethical dilemmas in qualitative research with youth on/offline. *International Journal of Learning and Media*, *4*(2), 67–75. doi:10.1162/IJLM_a_00096

Lochhead, J., & Whimbey, A. (1987). Teaching analytical reasoning through thinking aloud pair problem solving. *New Directions for Teaching and Learning,* (30), 73-92. Retrieved from https://eric.ed.gov/?id=EJ355732

Locke, T., Alcorn, N., & O'Neill, J. (2013). Ethical issues in collaborative action research. *Educational Action Research*, *21*(1), 107–123. doi:10.1080/09650792.2013.763448

Lockwood, J., & Mooney, A. (2017). *Computational Thinking in Education: Where does it fit? A systematic literary review.* ArXiv Preprint ArXiv:1703.07659

Lockwood, J., & Mooney, A. (2017). *Computational thinking in education: Where does it fit? A systematic literary review*. Retrieved from https://arxiv.org/abs/1703.07659v3

Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2015). Is coding the way to go? In A. Brodnik & J. Vahrenhold (Eds.), ISSEP 2015, LNCS 9378 (pp. 165–174). Academic Press.

Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2017). Nothing to fear but fear itself: Introducing recursion in lower secondary schools. In *Proceedings of 2017 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*. IEEE. 10.1109/LaTiCE.2017.23

Lood, F. (2017). *Assistive technology*. Retrieved 28, 2017), from https://www.frankieflood.com/studio.html

Lotfi, E., & Mohammed, B. (2018). Teaching Object Oriented Programming Concepts Through a Mobile Serious Game. *Proceedings of the 3rd International Conference on Smart City Applications*. 10.1145/3286606.3286851

Loukides, M. (2010). *App Inventor and the culture wars*. Retrieved November 26, 2019, from http://radar.oreilly.com/2010/07/culture-wars.html

Loyd, B. H., & Gressard, C. (1984). Reliability and factoral validity of computer attitude scale. *Educational and Psychological Measurement*, *44*(2), 501–505. doi:10.1177/0013164484442033

Lucas, B. (2016). A Five-Dimensional Model of Creativity and its Assessment in Schools. *Applied Measurement in Education*, *29*(4), 278–290. doi:10.1080/08957347.2016.1209206

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. doi:10.1016/j.chb.2014.09.012

Mader, A., & Dertien, E. (2016). Tinkering as a Method for Academic Teaching. *International Conference on Engineering and Product Design Education*. Aalborg University.

Maiorana, F., Giordano, D., & Morelli, R. (2015). Quizly: A live coding assessment platform for App Inventor. *Proceeding of IEEE Blocks and Beyond Workshop*, 25-30. 10.1109/BLOCKS.2015.7368995

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014a). Designing educational games for computer programming: A holistic framework. *Electronic Journal of e-Learning*, *12*(3), 281–298.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014c). Integrating learning analytics in an educational MMORPG for computer programming. In *Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies*. Athens, Greece: IEEE Computer Society Press.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Educational games for teaching computer programming. In *Research on e-Learning and ICT in Education* (pp. 87–98). Springer.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014b). Educational games for teaching computer programming. In *Research on e-learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 87–98). Springer.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2017). CMX: The effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, *10*(2), 219–235. doi:10.1109/TLT.2016.2556666

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *Transaction on Computing Education, 10*(4), 16:1–16:15.

Manabe, H., Kanemune, S., Namiki, M., & Nakano, Y. (2011). CS Unplugged assisted by digital materials for handicapped people at schools. In I. Kalas & R. T. Mittermeir (Eds.), *ISSEP 2011, LNCS 7013* (pp. 82–93). Berlin: Springer. doi:10.1007/978-3-642-24722-4_8

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the working group reports of the 2014 on innovation and technology in computer science education conference* (pp. 1-29). New York: ACM. doi:10.1145/2713609.2713610

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational Thinking and scratch at distance. *Computers in Human Behavior*, *80*, 470–477. doi:10.1016/j.chb.2017.09.025

Marr, B. (2016). Why Everyone Must Get Ready For The 4th Industrial Revolution. *Forbes*. Retrieved from https://www.forbes.com/sites/bernardmarr/2016/04/05/why-everyone-must-get-ready-for-4th-industrial-revolution/print/

Marr, B. (2017). *Kroger: How This U.S. Retail Giant Is Using AI And Robots To Prepare For The 4th Industrial Revolution.* Bernardmarr.Com.

Marr, B. (2018). What is Industry 4.0? Here's A Super Easy Explanation For Anyone. *Forbes*, 4–7. Retrieved from https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/#5b3ba6539788

Marsh, H. W., & Scalas, L. F. (2011). Self-concept in learning: reciprocal effects model between academic self-concept and academic achievement. In S. Järvela (Ed.), *Social and Emotional Aspects of Learning* (pp. 191–197). Amsterdam: Elsevier.

Marsh, J. A. (2016). The digital literacy skills and competences of children of pre-school age. *Media Education: Studi, Ricerche*. *Buone Practice*, *7*(2), 197–214.

Martin, L. (2015). The promise of the maker movement for education. *Journal of Pre-College Engineering Education Research, 5*(1), 4.

Martín-Ramos, P., da Silva, M. M. L., Lopes, M. J., & Silva, M. R. (2016, November). Student2student: arduino project-based learning. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 79-84).

Massachusetts Institute of Technology. (2013). *What is App Inventor*? Retrieved December 01, 2013, from http://AppInventor.mit.edu/explore/content/what-app-inventor.html

Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology*, *160*, 92–106. doi:10.1016/j.jecp.2017.03.013 PMID:28433822

Matsagouras, H. (2010). *The Innovation of Research Works in High School*. Athens: OEDB.

Mattar, J. (2018). Constructivism and connectivism in education technology: Active, situated, authentic, experiential, and anchored learning. *RIED. Revista Iberoamericana de Educación a Distancia*, *21*(2), 201–217. doi:10.5944/ried.21.2.20055

Mavrommatis, G. (2004). *The children of Calcantza. Education, poverty and social exclusion in a Muslim community of Thrace*. Athens: Metaixmio.

Mavrovounioti, V., Chatzopoulos, A., Papoutsidakis, M., & Piromalis, D. (2018). Implementation of an 2-wheel Educational Platform for STEM Applications. *Journal of Multidisciplinary Engineering Science and Technology*, *5*(10), 8944–8948.

McKean, R. (2017). Heritage Perth and the mixed reality makeover. *Education Technology Solutions*, *78*, 44–46.

McManis, L. D., & Gunnewig, S. B. (2012). Finding the education in educational technology with early learners. *YC Young Children*, *67*(3), 14.

Melkonian, S., Chatzopoulos, A., Papoutsidakis, M., & Piromalis, D. (2018). Remote Control via Android for a Small Vehicle ' s 2-Wheels Balancing. *Journal of Multidisciplinary Engineering Science and Technology*, *5*(10), 8964–8967.

Meraklis, M., & Kakabouras-Tilis, P. (2001). Intersectional Material for Flexible Zone, 1st-2nd Grades of Elementary School, Aesthetic Education, Teaching Greek Sign Language, Clothing - Folk Culture. Athens: Institute of Education - Ministry of Education & Religious Affairs.

Meraklis, M. (2001). *Folk Culture Curriculum (introductory note) in: Flexible Zone Intersectional Material, 1st-2nd Grade of Elementary School, Aesthetic Education, Teaching Greek Sign Language, Clothing - Folk Culture*. Pedagogical Institute.

Miljanovic, M. A., & Bradbury, J. S. (2018). A Review of Serious Games for Programming. In Lecture Notes in Computer Science: Vol. 11243. *Serious Games. JCSG 2018*. Cham: Springer. doi:10.1007/978-3-030-02762-9_21

Miller, B., Major, J. C., Kirn, A., Feil-Seifer, D., Anderson, M., & Jurkiewicz, M. (2018). Unplugged Robotics to Increase K-12 Students' Engineering Interest and Attitudes. In *Proceedings of 2018 IEEE Frontiers in Education Conference (FIE)*. IEEE. 10.1109/FIE.2018.8658959

Miller, L. D., Soh, L. K., Chiriacescu, V., Ingraham, E., Shell, D. F., & Hazley, M. P. (2014, March). Integrating computational and creative thinking to improve learning and performance in CS1. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 475-480). 10.1145/2538862.2538940

Mishra, J., Anguera, J. A., & Gazzaley, A. (2016). Video games for neuro-cognitive optimization. *Neuron*, *90*(2), 214–218. doi:10.1016/j.neuron.2016.04.010 PMID:27100194

MIT App Inventor. (2019). *MIT App Inventor*. Retrieved from http://appinventor.mit.edu/

Montola, M., Nummenmaa, T., Lucero, A., Boberg, M., & Korhonen, H. (2009). Applying game achievement systems to enhance user experience in a photo sharing service. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era* (pp. 94-97). ACM. 10.1145/1621841.1621859

Moore, T., Stohlmann, M., Wang, H., Tank, K., Glancy, A., & Roehrig, G. (2014). Implementation and integration of engineering in K-12 STEM education. In S. Purzer, J. Strobel, & M. Cardella (Eds.), *Engineering in Pre-College Settings: Synthesizing Research, Policy, and Practices* (pp. 35–60). West Lafayette: Purdue University Press. doi:10.2307/j.ctt6wq7bh.7

Moreno-León, J., & Robles, G. (2015). Dr. Scratch: A web tool to automatically evaluate Scratch projects. In *Proceedings of the 10th Workshop in Primary and Secondary Computing Education* (pp. 132–133). London, UK: ACM. 10.1145/2818314.2818338

Moreno-León, J., Román-González, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. *2018 IEEE Global Engineering Education Conference*, 1684-1689. 10.1109/EDUCON.2018.8363437

Morrison, J., & Bartlett, R. V. (2009). STEM as a curriculum: An experiential approach. *Education Week*, *28*(23), 28–31.

Moskal, B. M., & Leydens, J. A. (2000). Scoring Rubric Development: Validity and Reliability. *Practical Assessment, Research & Evaluation*, *7*(10).

Moustakas, K., Paliokas, I., Tsakiris, A., & Tzovaras, D. (2015). *Graphic and Virtual Reality*. Athens: Greek Academic Electronic Textbooks and Tutorials - Hellenic Academic Libraries Association.

Mpilalis, M. (2015). *The Past on the Internet, Image, Technology and Historical Culture in Modern Greece (1994-2005)*. Athens: HISTOREIN - National Documentation Center.

Munari, B. (1977). *Disegnare um albero*. Zanichelli.

Municipality, A. V. (2019a). *Educational Robotics Event for Elementary-High School-High School Students*. Retrieved January 2, 2020, from https://agiavarvara.gr/event/ekdilosi-ekpaideytikis-rompotikis-gia-mathites-dimotikoy-gymnasioy-lykeioy/

Municipality, A. V. (2019b). *Giannis Ritsos Municipal Theater*. Retrieved January 2, 2020, from https://agiavarvara.gr/building/dimotiko-kinimatotheatro-giannis-ritsos/

Muñoz-Repiso, A. G. V., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar*, *27*(59), 63–72. doi:10.3916/C59-2019-06

Murcia, K., Campbell, C., & Aranda, G. (2018). Trends in Early Childhood Education Practice and Professional Learning with Digital Technologies. *Pedagogika*, *68*(3), 249–264. doi:10.14712/23362189.2018.858

Murcia, K., & Pelliccione, L. (2017). Early childhood iSTEM: Playing as learning with digital technology toys. *Digitising Early Childhood International Conference*.

Myrsiades, K. (1987). *Approaches to teaching Homer's Iliad and Odyssey*. Modern Language Association of America.

NAEYC. (2012). *Position statement of the National Association for the Education of Young Children and the Fred Rogers Center for Early Learning and Children's Media at Saint Vincent College*. Retrieved from https:// www.naeyc.org/resources/topics/technology-and-media

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners. *The Asia-Pacific Education Researcher*, *28*(5), 1–11. doi:10.100740299-019-00438-4

Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM*, *62*(2), 32–35. doi:10.1145/3231587

National Association for the Education of Young Children, & Fred Rogers Center for Early Learning and Children's Media at Saint Vincent College. (2012). *Technology and interactive media as tools in early childhood programs serving children birth through age 8*. Retrieved from https://www.naeyc.org/sites/default/files/globally-shared/downloads/PDFs/resources/topics/PS_technology_WEB.pdf

National Research Council (NRC). (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.

National Research Council. (2006). *To recruit and advance: Women students and faculty in science and engineering*. Washington, DC: National Academies Press.

National Research Council. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. Washington, DC: National Academies Press.

NCWIT. (2018). *CS-in-a-Box: Unplug Your Curriculum (2018 Update)*. Retrieved from https://www.ncwit.org/resources/computer-science-box-unplug-your-curriculum-2018-update

Neeman, H. (2019). Retrieved January 20, 2020, from https://www.dezeen.com/2018/08/10/p-rouette-3d-printed-ballet-shoe-hadar-neeman-graduate-design/

Nemorin, S., & Selwyn, N. (2016). Making the best of it? Exploring the realities of 3D printing in school. *Research Papers in Education*, *32*, 1–18. doi:10.1080/02671522.2016.1225802

NewZoo. (2020). *Global Mobile Market Report*. Retrieved from https://newzoo.com/products/reports/global-mobile-market-report/

Ng, O.-L., & Chan, T. (2019). Learning as Making: Using 3D computer-aided design to enhance the learning of shape and space in STEM-integrated ways. *British Journal of Educational Technology*, *50*(1), 294–308. doi:10.1111/bjet.12643

NGSS Lead States. (2013). *Next Generation Science Standards: For states, by states*. Washington, DC: The National Academies Press; doi:10.17226/18290

Nishida, T., Idosaka, Y., Hofuku, Y., Kanemune, S., & Kuno, Y. (2008). New methodology of information education with "CS Unplugged". In R. T. Mittermeier & M. M. Syslo (Eds.), *ISSEP 2008, LNCS 5090* (pp. 241–252).

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS Unplugged Design Pattern. In *Proceedings of 40th SIGCSE Technical Symposium on CS Education, SIGCSE 2009* (pp. 231-235). New York: ACM.

Nistor, A. (2019). *Bringing Research into the Classroom – The Citizen Science approach in schools.* Scientix Observatory Report.

Nosek, B. A., Smyth, F. L., Sriram, N., Lindner, N. M., Devos, T., Ayala, A., ... Greenwald, A. G. (2009). National differences in gender-science stereotypes predict national sex differences in science and math achievement. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(26), 10593–10597. doi:10.1073/pnas.0809921106 PMID:19549876

NRC. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

O'Brian, M. (2020). Retrieved March 3, 2020, from https://www.definedstem.com/blog/what-is-project-based-learning/

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performances. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interactions in cooperative groups: the theoretical anatomy of group learning* (pp. 120–141). Cambridge: Cambridge University Press.

Oberhuber, S., Kothe, T., Schneegass, S., & Alt, F. (2017). Augmented Games: Exploring Design Opportunities in AR Settings With Children. In *Proceedings of the 2017 Conference on Interaction Design and Children* (pp. 371-377). ACM. 10.1145/3078072.3079734

Oblinger, D. G. (2010). The next generation of educational engagement. *Journal of Interactive Media in Education*, *2004*(1), 10. doi:10.5334/2004-8-oblinger

OECD. (2015). Excellence and Equity in Education. *PISA 2015 Results, 1*. Available at: https://www.oecd.org/education/pisa-2015-results-volume-i-9789264266490-en.htm

OECD. (2015). *Excellence and Equity in Education*. Retrieved from https://www.oecd.org/education/pisa-2015-results-volume-i-9789264266490-en.htm

OECD. (2016). *PISA 2015: PISA Results in Focus*. Paris: OECD Publishing.

OECD. (2018). *Excellence and Equity in Education*. Retrieved from https://www.oecd-ilibrary.org/education/pisa-2018-results-volume-i_5f07c754-en

Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, *44*(4), 443–460. doi:10.1007/BF02296207

Oluk, A., & Korkmaz, Ö. (2016). Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables. *Online Submission*, *8*(11), 1–7.

Orfanakis, V., & Papadakis, St. (2014). A new programming environment for teaching programming. A first acquaintance with enchanting. *Proceedings of the 2nd international virtual Scientific Conference - Scieconf 2014, EDIS,* 268–273.

Orfanakis, V., Papadakis, S., Kalogiannakis, M., Ampartzaki, M., & Vassilakis, K. (2016). Digital student conference platform implementation: The case study of the "research project" course. *The Open Education Journal*, *12*(2), 5–23.

Orfanakis, V., & Papadakis, St. (2016). Teaching basic programming concepts to novice programmers in Secondary Education using Twitter, Python, Ardruino and a coffee machine. In *Hellenic Conference on Innovating STEM Education (HISTEM)*. University of Athens.

Orfanakis, V., & Papadakis, St. (2016). Teaching basic programming concepts to novice programmers in Secondary Education using Twitter, Python, Arduino and a coffee machine. In *Proceedings of the Hellenic Conference on Innovating STEM Education (HISTEM)*. University of Athens.

Organization for Economic Cooperation and Development (OECD). (2013). *OECD Skills Outlook 2013*. Paris: OECD Publishing. doi:10.1787/9789264204256-

Ospennikova, E., Ershov, M., & Iljin, I. (2015). Educational Robotics as an Inovative Educational Technology. *Procedia: Social and Behavioral Sciences*, *214*(June), 18–26. doi:10.1016/j.sbspro.2015.11.588

Ota, G., Morimoto, Y., & Kato, H. (2016). Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 238-239). Cambridge, UK: IEEE. 10.1109/VLHCC.2016.7739695

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia: Social and Behavioral Sciences*, *191*, 1479–1482. doi:10.1016/j.sbspro.2015.04.224

Paikin, G., & Tal, A. (2015). Solving multiple square jigsaw puzzles with missing pieces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4832-4839). 10.1109/CVPR.2015.7299116

Palaiologou, I. (2016). Children under five and digital technologies: Implications for early years pedagogy. *European Early Childhood Education Research Journal*, *24*(1), 5–24. doi:10.1080/1350293X.2014.929876

Panagiotakopoulos, C. T., Sarris, M. E., & Koleza, E. G. (2013). Playing with numbers: development issues and evaluation results of a computer game for primary school students. In T. Sobh & K. Elleithy (Eds.), *Emerging trends in computing, informatics, systems sciences, and engineering* (Vol. 151, pp. 263–275). New York: Springer. doi:10.1007/978-1-4614-3558-7_22

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In *Proceedings of the 2014 workshop on interaction design in educational environments* (pp. 1-7). New York: ACM.

Papadakis, St., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In M. Kalogiannakis, D. Stavrou & P.G. Michaelidis (Eds.), *Proceedings of the 7th International Conference on Hands-on Science*, (pp. 235-240). Academic Press.

Papadakis, St., & Kalogiannakis, M. (2010). eTwinning in the early childhood as starting line of innovative practices for the didactic of natural sciences. In M. Kalogiannakis, D. Stavrou & P.G. Michaelidis (Eds.), *Proceedings of the 7th International Conference on Hands-on Science, Bridging the Science and Society Gap*, (pp. 235-240). Academic Press.

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In Fardoun, H.M. & Gallud, J.A. (Ed.), *Proceedings of the 2014 Workshop on Interaction Design in Educational Environments*, (pp.1–7). ACM.

Papadakis, S. (2016). Creativity and innovation in European education. 10 years eTwinning. Past, present and the future. *International Journal of Technology Enhanced Learning*, *8*(3/4), 279–296. doi:10.1504/IJTEL.2016.082315

Papadakis, S. (2018b). The use of computer games in classroom environment. *International Journal of Teaching and Case Studies*, *9*(1), 1. doi:10.1504/IJTCS.2018.090191

Papadakis, S. (2018c). Evaluating pre-service teachers' acceptance of mobile devices with regards to their age and gender: A case study in Greece. *International Journal of Mobile Learning and Organisation*, *12*(4), 336–352. doi:10.1504/IJMLO.2018.095130

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *Int. J. Teaching and Case Studies*, *10*(1), 31–52.

Papadakis, S. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*, *10*(1), 31–52. doi:10.1504/IJTCS.2019.096871

Papadakis, S. (2020). Apps to Promote Computational Thinking Concepts and Coding Skills in Children of Preschool and Pre-Primary School Age. In *Mobile Learning Applications in Early Childhood Education* (pp. 101–121). IGI Global. doi:10.4018/978-1-7998-1486-3.ch006

Papadakis, S., & Kalogiannakis, M. (2017). Mobile educational applications for children: What educators and parents need to know. *International Journal of Mobile Learning and Organisation*, *11*(3), 256–277. doi:10.1504/IJMLO.2017.085338

Papadakis, S., & Kalogiannakis, M. (2018). Using gamification for supporting an introductory programming course. The case of classcraft in a secondary education classroom. *Lecture Notes of the Institute for Computer Sciences. Social-Informatics and Telecommunications Engineering, LNICST*, *229*, 366–375. doi:10.1007/978-3-319-76908-0_35

Papadakis, S., & Kalogiannakis, M. (2020). A Research Synthesis of the Real Value of Self-Proclaimed Mobile Educational Applications for Young Children. In *Mobile Learning Applications in Early Childhood Education* (pp. 1–19). IGI Global. doi:10.4018/978-1-7998-1486-3.ch001

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2014). Novice programming environments. Scratch & app inventor: a first comparison. In *Proceedings of the 2014 workshop on interaction design in educational environments* (pp. 1-7). 10.1145/2643604.2643613

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2016). Using Scratch and App Inventor for teaching introductory programming in Secondary Education. A case study. *International Journal of Technology Enhanced Learning*, *1*(1), 1. doi:10.1504/IJTEL.2016.10001503

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. In *Early Childhood Development: Concepts, Methodologies, Tools, and Applications* (pp. 797–819). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-7507-8.ch039

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Comparing tablets and PCs in teaching mathematics: An attempt to improve mathematics competence in early childhood education. *Preschool & Primary Education*, *4*(2), 241–253. doi:10.12681/ppej.8779

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2018). The effectiveness of computer and tablet assisted intervention in early childhood students' understanding of numbers. An empirical study conducted in Greece. *Education and Information Technologies*, *23*(5), 1849–1871. doi:10.100710639-018-9693-7

Papadakis, S., & Orfanakis, V. (2016). The combined use of Lego Mindstorms NXT and App Inventor for teaching novice programmers. In *International Conference EduRobotics 2016* (pp. 193-204). Springer.

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programming environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning*, *10*(1-2), 44–72. doi:10.1504/IJTEL.2018.088333

Papadakis, S., Tousia, C., & Polychronaki, K. (2018). Women in computer science. The case study of the Computer Science Department of the University of Crete, Greece. *Int. J. Teaching and Case Studies*, *9*(2), 142–151. doi:10.1504/IJTCS.2018.090963

Papadakis, S., Zaranis, N., & Kalogiannakis, M. (2019). Parental involvement and attitudes towards young Greek children's mobile usage. *International Journal of Child-Computer Interaction*, *22*, 100144. doi:10.1016/j.ijcci.2019.100144

Papadakis, St. (2018a). Gender stereotypes in Greek computer science school textbooks. *International Journal of Teaching and Case Studies*, *9*(1), 48–71. doi:10.1504/IJTCS.2018.090196

Papadakis, St. (2018b). Is pair programming more effective than solo programming for secondary education novice programmers? A case study. *International Journal of Web-Based Learning and Teaching Technologies*, *13*(1), 1–16. doi:10.4018/IJWLTT.2018010101

Papadakis, St., & Kalogiannakis, M. (2017). *Using Gamification for Supporting an Introductory Programming Course. The Case of ClassCraft in a Secondary Education Classroom*. *2nd EAI International Conference on Design, Learning & Innovation*, Heraklion, Crete, Greece.

Papadakis, St., & Kalogiannakis, M. (2017). Using Gamification for Supporting an Introductory Programming Course. The Case of ClassCraft in a Secondary Education Classroom. In *Interactivity, Game Creation, Design, Learning, and Innovation* (pp. 366–375). Cham: Springer.

Papadakis, St., & Kalogiannakis, M. (2019). Evaluating a Course for Teaching Advanced Programming Concepts with Scratch to Preservice Kindergarten Teachers: A Case Study in Greece. In D. Farland-Smith (Ed.), *Early Childhood Education* (pp. 1–19). London, UK: IntechOpen Limited. doi:10.5772/intechopen.81714

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2016). Using scratch and app inventor for teaching introductory programming in secondary education. A case study. *International Journal of Technology Enhanced Learning*, *8*(3–4), 217–233. doi:10.1504/IJTEL.2016.082317

Papadakis, St., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. doi:10.4018/IJWLTT.2017100106

Papadakis, St., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in Preschool Education. A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187–202. doi:10.1504/IJMLO.2016.077867

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, *99*, 415–427. doi:10.1016/j.chb.2019.01.008

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2018). How do you feel about learning to code? Investigating the effect of children's attitudes towards coding using eye-tracking. *International Journal of Child-Computer Interaction*, *17*, 50–60. doi:10.1016/j.ijcci.2018.01.004

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2019). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*. doi:10.1016/j.chb.2019.03.003

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. Basic Books.

Papert, S. (1986). Constructionism: A new opportunity for elementary science education. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1990). *Children, computers and powerful ideas*. New York: Basic Books.

Papert, S. (1993). *Mindstorms: Children, Computers, and Powerful Ideas* (2nd ed.). New York: Basic Books.

Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, *1*(1), 95–123. doi:10.1007/BF00191473

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*(2), 1–11.

Paraskeva, F., Bouta, H., & Papagianni, A. (2008). Individual characteristics and computer self-efficacy in secondary education teachers to integrate technology in educational practice. *Computers & Education*, *50*(3), 1084–1091. doi:10.1016/j.compedu.2006.10.006

Pardamean, B., & Suparyanto, T. (2015). Improving problem-solving skills through Logo programming language. *Stanisław Juszczyk*, 52.

Pate, M., & Miller, G. (2011). Effects of Think–Aloud Pair Problem Solving on Secondary–Level Students' Performance in Career and Technical Education Courses. *Journal of Agricultural Education*, *52*(1), 120–131. doi:10.5032/jae.2011.01120

Patrick, H., & Mantzicopoulos, P. (2015). Young children's motivation for learning science. In *Research in early childhood science education* (pp. 7–34). Dordrecht: Springer. doi:10.1007/978-94-017-9505-0_2

Peijnenborgh, J. C., Hurks, P. P., Aldenkamp, A. P., van der Spek, E. D., Rauterberg, M. G., Vles, J. S., & Hendriksen, J. G. (2016). A study on the validity of a computer-based game to assess cognitive processes, reward mechanisms, and time perception in children aged 4-8 years. *JMIR Serious Games, 4*(2), e15. doi:10.2196/games.5997

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering* (pp. 68-77). BCS Learning & Development.

Phillips, P. (2009). *Computational thinking: A problem-solving tool for every classroom*. Retrieved from http://education.sdsc.edu/resources/CompThinking.pdf

Piaget, J., & Cook, M. (1952). *The origins of intelligence in children*. New York: International Universities Press. doi:10.1037/11494-000

Piteira, M., & Costa, C. (2012, June). Computer programming and novice programmers. In *Proceedings of the Workshop on Information Systems and Design of Communication* (pp. 51-53). ACM.

Pomeranz, D., Shemesh, M., & Ben-Shahar, O. (2011). A fully automated greedy square jigsaw puzzle solver. In CVPR 2011 (pp. 9-16). IEEE. doi:10.1109/CVPR.2011.5995331

Prensky, M. (2001). Fun, play and games: What makes games engaging. *Digital Game-Based Learning, 5*(1), 5-31.

Prensky, M. (2008). Students as designers and creators of educational computer games: Who else? *British Journal of Educational Technology*, *39*(6), 1004–1019. doi:10.1111/j.1467-8535.2008.00823_2.x

Psaridou, V. (2015). *Accessibility Planning Interventions for Visually Impaired Greek Legislation*. Democritus University of Thrace.

Psycharis, S., & Kotzampasaki, E. (2017). A Didactic Scenario for Implementation of Computational Thinking using Inquiry Game Learning. *Proceedings of the 2017 International Conference on Education and E-Learning*, 26–29. 10.1145/3160908.3160918

Psycharis, S., Kalovrektis, K., Sakellaridi, E., & Korres, K. (2017). Unfolding the Curriculum: Physical Computing, Computational Thinking and Computational Experiment in STEM's Transdisciplinary Approach. *European Journal of Engineering Research and Science*, 19–24.

Psycharis, S. (2018). STEAM in Education: A Literature review on the role of Computational Thinking, Engineering Epistemology and Computational Science. Computational STEAM Pedagogy (CSP). *Scientific Culture*, *4*(2), 51–72.

Psycharis, S. (2018). STEAM in educations: A literature review on the role of Computational Thinking, Engineering Epistemology and Computational Science. *Computational STEAM Pedagogy*, *4*(2), 51–72. doi:10.5281/zenodo.1214565

Psycharis, S., & Kotzampasaki, E. (2019). The Impact of a STEM Inquiry Game Learning Scenario on Computational Thinking and Computer Self-confidence. *Eurasia Journal of Mathematics, Science and Technology Education*, *15*(4). doi:10.29333/ejmste/103071

Pugnali, A., Sullivan, A., & Umashi Bers, M. (2017). The Impact of User Interface on Young Children's Computational Thinking. *Journal of Information Technology Education: Innovations in Practice*, *16*, 171–193. doi:10.28945/3768

Qian, M., & Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, *63*, 50–58. doi:10.1016/j.chb.2016.05.023

Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education*, *18*(1), 1–24. doi:10.1145/3077618

Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, *25*(5), 66–71.

Queen Mary University of London & King's College London. (n.d.). *Teaching London Computing*. Retrieved from https://teachinglondoncomputing.org/

Quinn, F., & Lyons, F. (2011). High school students' perceptions of school science and science careers: A critical look at a critical issue. *Science Education International*, *22*(4), 225–238.

Raabe, A., Rodrigues, A. J., Santana, A. M., Vieira, M. V., Rosário, T., & Carneiro, A. C. R. (2015). Brinquedos de Programar na Educação Infantil: Um estudo de Caso. Anais do Workshop de Informática na Escola, 21(1), 42.

Rainwater, B. (2016). *How the maker movement is revitalizing industry in American cities*. Retrieved from https://www.fastcompany.com/3057349/how-the-maker-movement-is-revitalizing-industry-in-american-cities

Ramirez-Rosales, S., Vazquez-Reyes, S., Villa-Cisneros, J. L., & De Leon-Sigg, M. (2016). A serious game to promote object oriented programming and software engineering basic concepts learning. *Proceedings - 2016 4th International Conference in Software Engineering Research and Innovation, CONISOFT*, 97–103. 10.1109/CONISOFT.2016.23

Reed, D. (2002). The use of ill-defined problems for developing problem-solving and empirical skills in CS1. *Journal of Computing Sciences in Colleges*, *18*(1), 121–133.

Resnick, M. (2006). Computer as paintbrush: Technology, play, and the creative society. Academic Press.

Resnick, M. (2012). Mother's Day, warrior cats, and digital fluency, stories for the scratch online community. Constructionism 2012 Educational Technology Lab, 52–58.

Resnick, M. (2013). Coding for all: interest-driven trajectories to computational fluency. Proposal to the National Science Foundation.

Resnick, M., Ocko, S., & Papert, S. (1988). LEGO, LOGO, AND DESIGN. *Children and Interactive Electronic Environments, 5*(4), 14–18. Retrieved from https://www.jstor.org/stable/41514692

Resnick, M. (2008). Sowing the Seeds for a More Creative Society. *Learning and Leading with Technology*, *35*(4), 18–22.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. B. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. doi:10.1145/1592761.1592779

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the 2005 conference on Interaction design and children* (pp. 117-122). 10.1145/1109540.1109556

Richter, G., Raban, D. R., & Rafaeli, S. (2015). Studying gamification: the effect of rewards and incentives on motivation. In *Gamification in education and business* (pp. 21–46). Cham: Springer. doi:10.1007/978-3-319-10208-5_2

Riley, D., & Hunt, K. A. (2014). *Computational thinking for the modern problem solver*. Chapman and Hall/CRC. doi:10.1201/b16688

Robbins, N. B., & Heiberger, R. M. (2011). Plotting Likert and other rating scales. In *Proceedings of the 2011 Joint Statistical Meeting* (pp. 1058-1066). American Statistical Association.

Robson, C. (2010). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers.* Αθήνα: Gutenberg.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using Student Performance to Assess CS Unplugged Activities in a Classroom Environment. In *Proceedings of 2016 ACM Conference on Innovation and Technology in CS Education ITiCSE '16* (pp. 95-100). New York: ACM. 10.1145/2899415.2899465

Roehrig, G. H., Moore, T. J., & Wang, H. (2012). *Is Adding the E Enough?: Investigating the Impact of K-12 Engineering Standards on the Implementation of STEM Integration*. School of Engineering Education Faculty Publications.

Rogers, S. (2014). *Level Up! The guide to great video game design*. John Wiley & Sons.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. doi:10.1016/j.chb.2016.08.047

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(42), 1–15. doi:10.118641239-017-0080-z

Ronsivalle, G. B., Boldi, A., Gusella, V., Inama, C., & Carta, S. (2018). *How to Implement Educational Robotics' Programs in Italian Schools: A Brief Guideline According to an Instructional Design Point of View*. Technology, Knowledge and Learning; doi:10.100710758-018-9389-5

Rose, S. P., Habgood, J., & Jay, T. (2018). Pirate plunder: game-based computational thinking using scratch blocks. In *Proceedings of the 12th European Conference on Games Based Learning*. Academic Conferences and Publishing International Limited, 556-564.

Rosicka, C. (2016). Translating STEM education research into practice. Australian Council for Educational Research.

Rotherham, A. J., & Willingham, D. T. (2010). 21st-Century" skills. *American Educator*, *17*(1), 17–20.

Roy, K., Rousse, W. C., & DeMeritt, D. B. (2012). Comparing the mobile novice programming environments: App Inventor for Android vs. GameSalad. In *Proceedings of the 2012 IEEE Frontiers in Education Conference (FIE '12)*. Washington, DC: IEEE Computer Society. 10.1109/FIE.2012.6462363

Rukavina, S., Zuvic-Butorac, M., Ledic, J., Milotic, B., & Jurdana-Sepic, R. (2012). Developing positive attitude towards science and mathematics through motivational classroom experiences. *Science Education International*, *23*(1), 6–19.

Ruzzenente, M., Koo, M., Nielsen, K., Grespan, L., & Fiorini, P. (2012). A review of robotics kits for tertiary education. In *Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum* (pp. 153-162). Academic Press.

S4A. (2020). *Scratch for Arduino*. Retrieved from https://scratch.mit.edu/

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129–141. doi:10.1016/j.compedu.2016.03.003

Saito, T., Gunji, Y., & Kumano, Y. (2015). The Problem about Technology in STEM Education: Some Findings from Action Research on the Professional Development & Integrated STEM Lessons in Informal Fields. *K-12 STEM Education, 1*(2), 85–100.

Samejima, F. A. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometric Monograph*, *34*(4, Pt.2), 17. doi:10.1007/BF03372160

Sanchez, E. (2011). *Key criteria for Game Design. A Framework. MEET Project*. European Commission.

Sanders, M. (2009). STEM, STEM education, STEMmania. *Technology Teacher*, *68*(4), 20–26.

Sanders, M. (2009). STEM, STEM Education, STEMmania. *Technology Teacher*, *68*(4), 20–27.

Sands, P., Yadav, A., & Good, J. (2018). Computational Thinking in K-12: In-service teacher perceptions of computational thinking. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines* (pp. 151–164). New York: Springer. doi:10.1007/978-3-319-93566-9_8

Sangiuliano, M. (2015). *Role models leading girls into careers in science*. Retrieved from https://www.schooleducation-gateway.eu/en/pub/viewpoints/experts/role_models_leading_girls_into.htm

Sapounidis, T., Demetriadis, S., & Stamelos, I. (2015). Evaluating children performance with graphical and tangible robot programming tools. *Personal and Ubiquitous Computing*, *19*(1), 225–237. doi:10.100700779-014-0774-3

Sapounidis, T., Papadopoulos, P., Demetriadis, S., & Stamovlasis, D. (2019). Tangible and Graphical Programming with Experienced Children: A Mixed Methods Analysis. International. *Journal of Child-Computer Interaction*, *19*, 67–78. doi:10.1016/j.ijcci.2018.12.001

Sapounidis, T., Stamovlasis, D., & Dimitriadis, S. (2018). Latent Class Modeling of Children's Preferences Profiles on Tangible and Graphical Robot Programming. *IEEE Transactions on Education*, *62*(2), 127–133. doi:10.1109/TE.2018.2876363

Sarıtepeci, M. (2017). *Analysis of computational thinking skill level in secondary education in terms of various variables*. Paper presented at the 5th International Instructional Technologies & Teacher Education Symposium–ITTES, İzmir, Turkey.

Sarıtepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational Research and Practice*, 490-501.

Saritepeci, M. (2020). Developing Computational Thinking Skills of High School Students: Design-Based Learning Activities and Programming Tasks. *The Asia-Pacific Education Researcher*, *29*(1), 35–54. doi:10.100740299-019-00480-2

Saunders, M. N. K., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students* (5th ed.). Harlow: Prentice Hall.

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66. doi:10.100740299-019-00478-w

Schauble, L., Glaser, R., Duschl, R. A., Schulze, S., & John, J. (1995). Students' Understanding of the Objectives and Procedures of Experimentation in the Science Classroom. *Journal of the Learning Sciences*, *4*(2), 131–166. doi:10.120715327809jls0402_1

Scheuerman, J. (2015). AdventureCode: Computational Thinking Through Games. In S. Carliner, C. Fulford & N. Ostashewski (Eds.), *Proceedings of EdMedia 2015--World Conference on Educational Media and Technology* (pp. 1832-1837). Montreal, Quebec, Canada: Association for the Advancement of Computing in Education (AACE). Retrieved December 10, 2019 from https://www.learntechlib.org/primary/p/151458/

Schleicher, A., Zimmer, K., Evans, J., & Clements, N. (2009). *PISA 2009 Assessment Framework: Key Competencies in Reading, Mathematics and Science.* OECD Publishing. doi:10.1787/9789264062658-en

Schneider, B., Judy, J., & Mazuca, C. (2012). Boosting STEM interest in high school. *Phi Delta Kappan*, *94*(1), 62–65. doi:10.1177/003172171209400112

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. *Handbook for Research on Mathematics Teaching and Learning*, 334–370.

Schon, S., Ebner, M., & Kumar, S. (2014). The Maker Movement Implications from modern fabrication, new digital gadgets, and hacking for creative learning and teaching. eLearning Papers, 86-100.

Schoretsanitou, P., & Vekiri, I. (2010). ICT Integration in Education: Factors for Predicting the Educational Crisis. In A. Tzimogiannis (Ed.), *Proceedings of the 7th Panhellenic Conference with International Participation "ICT in Education"*, (*vol. 2*, pp. 617-624). Korinthos.

Schuster, C., & Martiny, S. E. (2017). Not feeling good in STEM: Effects of stereotype activation and anticipated affect on women's career aspirations. *Sex Roles*, *76*(1-2), 40–55. doi:10.100711199-016-0665-3

Scientix Observatory Report. (2018). *STEM – Education policies in Europe*. Retrieved from http://www.scientix.eu/documents/10137/782005/Scientix_Texas-Instruments_STEM-policies-October-2018.pdf/d56db8e4-cef1-4480-a420-1107bae513d5

Scopigno, R., Cignoni, P., Pietroni, N., Callieri, M., & Dellepiane, M. (2017). Digital fabrication techniques for cultural heritage: A survey. *Computer Graphics Forum*, *36*(1), 6–21. doi:10.1111/cgf.12781

Scott, C. (2015). *MakerBot announces winners of 3D printed Assistive Technology Challenge*. Retrieved February 27, 2017, from https://3dprint.com/111643/makerbot-assist-tech-winners/

Scott, C. (2016). *A look at some small 3D printed devices that make a huge difference in the lives of the disabled*. Retrieved February 27, 2017, from https://3dprint.com/139584/3d-printed-assistive-technology/

Scratch. (2019). Retrieved from https://scratch.mit.edu/

Seiter, L., & Foreman, B. (2013). Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (pp. 59-66). La Jolla, CA: ACM. 10.1145/2493394.2493403

Selby, C. C., & Woollard, J. (2013). *Computational Thinking: The Developing Definition (Project report)*. Retrieved from https://eprints.soton.ac.uk/356481/

Sendurur, P. (2019). Investigation of pre-service CS teachers' CS-unplugged design practices. *Education and Information Technologies*, *24*(6), 3823–3840. doi:10.100710639-019-09964-6

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351–380. doi:10.100710639-012-9240-x

Senler, B., & Sungur, S. (2009). Parental influences on students' self-concept, task value beliefs, and achievement in science. *The Spanish Journal of Psychology*, *12*(1), 106–117. doi:10.1017/S1138741600001529 PMID:19476224

Sentence, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. doi:10.100710639-016-9482-0

Seow, P., Looi, C. K., How, M. L., Wadhwa, B., & Wu, L. K. (2019). Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore. In *Computational Thinking Education* (pp. 345–361). Singapore: Springer. doi:10.1007/978-981-13-6528-7_19

Servoz, M. (2018). *The future of work? Work of the future! On how artificial intelligence, robotics and automation are transforming jobs and the economy in Europe*. Retrieved from https://ec.europa.eu/epsc/sites/epsc/files/ai-report_online-version.pdf

Severance, C. (2015). Python for Everybody, University of Michigan, Coursera. Retrieved from https://www.coursera.org/specializations/python

SFIA Foundation. (2017). *The Skills Framework for the Information Age – SFIA*. Retrieved from https://www.sfia-online.org/en/reference-guide

Shaffer, D. W. (2006). *How computer games help children learn*. Macmillan. doi:10.1057/9780230601994

Shah, M., Foster, A., & Barany, A. (2017). *Facilitating learning as identity change through game-based learning. In Game-Based Learning: Theory, Strategies and Performance Outcomes (pp. 257-278)*. New York, NY: Nova Publishers.

Shell, D. F., Soh, L. K., Flanigan, A. E., Peteranetz, M. S., & Ingraham, E. (2017, March). Improving Students' Learning and Achievement in CS Classrooms through Computational Creativity Exercises that Integrate Computational and Creative Thinking. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 543-548). 10.1145/3017680.3017718

Sherman, M., Martin, F., Baldwin, L., & DeFilippo, J. (2014). *App Inventor Project Rubric – Computational Thinking through Mobile Computing.* Retrieved from https://nsfmobilect.files.wordpress.com/2014/09/mobile-ct-rubric-forapp-inventor-2014-09-01.pdf

Sherman, M., & Martin, F. (2015). The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, *30*(6), 53–59.

Shiakka, F. (2017). *Minority education and social exclusion. Teacher perceptions in minority schools in Xanthi*. Thessaloniki: Aristotelio University of Thessaloniki.

Shi, N., Cui, W., Zhang, P., & Sun, X. (2018). Evaluating the effectiveness roles of variables in the novice programmers learning. *Journal of Educational Computing Research*, *56*(2), 181–201. doi:10.1177/0735633117707312

Shoop, R., Flot, J., Friez, T., Schunn, C., & Witherspoon, E. (2016). Can computational thinking practices betaught in robotics classrooms? In *International Technology and Engineering Education Conference*. Carnegie Mellon Robotics Academy/University of Pittsburgh.

Shoukry, L., Sturm, C., & Galal-Edeen, G. H. (2015). Pre-MEGa: a proposed framework for the design and evaluation of preschoolers' mobile educational games. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering* (pp. 385–390). Cham: Springer. doi:10.1007/978-3-319-06773-5_52

Shute, V. J., & Ke, F. (2012). Games, learning, and assessment. In *Assessment in game-based learning* (pp. 43–58). Springer. doi:10.1007/978-1-4614-3546-4_4

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. doi:10.1016/j.edurev.2017.09.003

Siegle, D. (2009). Developing Student Programming and Problem-Solving Skills with Visual Basic. *Gifted Child Today, 32*(4), 24-29. https://doi.acm.org/10.1145/2157136.21572

Singh, K., Granville, M., & Kida, S. (2002). Mathematics and Science Achievement: Effects of Motivation, Interest, and Academic Engagement. *The Journal of Educational Research*, *95*(6), 323–332. doi:10.1080/00220670209596607

Singmaster, H. (2008). *Seven Skills Students Need for Their Future*. Retrieved from https://asiasociety.org/education/resources-schools/professional-learning/seven-skills-students-need-their-future

Skarpelos, G. (1999). *Terra Virtualis, The Construction of Cyberspace*. Athens: Nefeli.

Slavin, R. E. (1997). *Educational psychology: Theory and practice* (5th ed.). Boston: Allyn & Bacon.

Smith, S., Tillman, D., Mishra, P., Slykhuis, D., Alexander, C., Henriksen, D., & Goodman, A. (2014). Building multidisciplinary connections: Intersections of content, creativity, and digital fabrication technologies. In M. Searson & M. Ochoa (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014* (pp. 2506-2510). Chesapeake, VA: Association for the Advancement of Computing in Education (AACE).

Smith, A. (2010). *The influence of education on conflict and peace building. The influence of education on conflict and peace building, Background paper prepared for the Education for All Global Monitoring Report 2011 The Hidden Crisis: Armed conflict and education*. Paris: UNESCO.

Smith, C. P., & Neumann, M. D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, *21*(3), 185–188. doi:10.5951/teacchilmath.21.3.0185

Smithsonian. (n.d.). https://www.smithsonianmag.com/smithsonian-institution/ten-smithsonian-artifacts-you-can-3-d-print-180973649/

Snap, B. Y. O. B. (2019). *Snap! Build Your Own Blocks*. Retrieved from https://snap.berkeley.edu/

Solecki, I., Porto, J., & Alves, N. da C. (2020). Automated Assessment of the Visual Design of Android Apps Developed with App Inventor. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 51–57). Portland, OR: ACM. 10.1145/3328778.3366868

Solomonides, X. (2006). *New trends in educational technology. Constructivism and modern learning environments*. Metaixmio.

Stamovlasis, D., Papageorgiou, G., & Tsitsipis, G. (2013). The coherent versus fragmented knowledge hypotheses for the structure of matter: An investigation with a robust statistical methodology. *Chemistry Education Research and Practice*, *14*(4), 485–495. doi:10.1039/C3RP00042G

Stamovlasis, D., Papageorgiou, G., Tsitsipis, G., Tsikalas, T., & Vaiopoulou, J. (2018). Illustration of Step-Wise Latent Class Modelling with Covariates and Taxometric Analysis in Research Probing Children's Mental Models in Learning Sciences. *Frontiers in Psychology*, *9*, 532. doi:10.3389/fpsyg.2018.00532 PMID:29713300

Stansell, A. (2016). *Transmedia STEM Intervention Book in Middle School for Educational Change* (Dissertation). University of North Texas Libraries, UNT Digital Library. https://digital.library.unt.edu

Stegeman, M., Barendsen, E., & Smetsers, S. (2016). Designing a rubric for feedback on code quality in programming courses. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 160-164). Koli, Finland: ACM. 10.1145/2999541.2999555

Stein, C., & Nickerson, K. (2004). Botball robotics and gender differences in middle school teams. *Proceedings American society for engineering education annual conference*. http://dpm.kipr.org/papers/asee04-gender.pdf

Strawhacker, A., & Bers, M. U. (2018). Promoting Positive Technological Development in a Kindergarten Makerspace: A Qualitative Case Study. *European Journal of STEM Education*, *3*(3), 9. doi:10.20897/ejsteme/3869

Sullivan, A. A. (2016). *Breaking the STEM Stereotype: Investigating the use of robotics to change young children's gender stereotypes about technology and engineering* (Dissertation). Tufts University.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20. doi:10.100710798-015-9304-5

Sullivan, A., & Bers, M. U. (2019). Investigating the use of robotics to increase girls' interest in engineering during early elementary school. *International Journal of Technology and Design Education*, *29*(5), 1033–1051. doi:10.100710798-018-9483-y

Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, *48*(2), 105–128. doi:10.1080/15391523.2016.1146563

Sun, C. T., Ye, S. H., & Wang, Y. J. (2015). Effects of commercial video games on cognitive elaboration of physical concepts. *Computers & Education*, *88*, 169–181. doi:10.1016/j.compedu.2015.05.002

Sung, H. Y., & Hwang, G. J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & Education*, *63*, 43–51. doi:10.1016/j.compedu.2012.11.019

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education, *Technology*. *Knowledge and Learning*, *22*(3), 443–463. doi:10.100710758-017-9328-x

Susnea, I., & Vasiliu, G. (2016). A Fuzzy Logic Software Tool and a New Scale for the Assessment of Creativity. *International Journal of Computers, Communications & Control*, *11*(3), 441–449. doi:10.15837/ijccc.2016.3.2192

Swaid, S. I. (2015). Bringing Computational Thinking to STEM Education. *Procedia Manufacturing*, *3*, 3657–3662. doi:10.1016/j.promfg.2015.07.761

Syslo, M. M., & Kwiatkowska, A. B. (2014). Playing with computing at a children's university. In *Proceedings of Workshop in Primary and Secondary Computing Education WIPSCE '14*. New York: ACM. 10.1145/2670757.2670790

Tassin M. (2010). Pour une "pédagogie culturelle". *Cahiers pédagogiques, 479*, 29-30.

Taub, R., Ben-Ari, M., & Armoni, M. (2009). The effect of CS Unplugged on middle-school students' views of CS. In *Proceedings of the 2009 ACM Conference on Innovation and Technology in CS Education, ITiCSE '09* (pp. 99-103). New York: ACM. 10.1145/1562877.1562912

Taylor, K., & Baek, Y. (2019). Grouping matters in computational robotic activities. *Computers in Human Behavior*, *93*, 99–105. doi:10.1016/j.chb.2018.12.010

TeachThought Staff. (2015). *10 ways 3D printing can be used in education*. https://www.teachthought.com/technology/10-ways-3d-printing-can-be-used-in-education//

TeachThought Staff. (2015). *10 ways 3D printing can be used in education*. Retrieved from https://www.teachthought.com/technology/10-ways-3d-printing-can-be-used-in-education/

Tella, A. (2007). The Impact of Motivation on Student's Academic and Learning Outcomes in Mathematics among Secondary School Students in Nigeria. *Eurasia Journal of Mathematics, Science, &. Technology (Elmsford, N.Y.)*, *3*(2), 149–156.

Texley, J., & Ruud, R. M. (2018). *Teaching STEM literacy: a constructivist approach for ages 3 to 8* (1st ed.). St. Paul, MN: Redleaf Press.

The Royal Society. (2012). *Shut down or restart: The way forward for computing in UK schools*. Retrieved from https://goo.gl/7Qa7fY

Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2017). How do different cognitive styles affect learning programming? Insights from a game-based approach in Greek schools. *ACM Transactions on Computing Education*, *17*(1), 3. doi:10.1145/2940330

Thies, R., & Vahrenhold, J. (2013). On plugging “unplugged” into CS classes. In *Proceedings of the 2013 ACM Special Interest Group on Computer Science Education, SIGCSE ’13* (pp. 365-370). New York: ACM.

Thies, R., & Vahrenhold, J. (2016). Back to school: CS unplugged in the wild. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in CS Education, ITiCSE ’16* (pp. 118-123). New York: ACM.

Thomas, N., Carroll, F., Kop, R., & Stocking, S. (2012). iBook learning experience: the challenge of teaching computer architecture to first year university students. In *Proceedings WORLDCOMP’12- The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing*. Las Vegas, NV: WORLDCOMP.

Thomas, N., Carroll, F., Kop, R., & Stocking, S. (2012). iBook learning experience: the challenge of teaching computer architecture to first year university students. In *WORLDCOMP’12- The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing*. Las Vegas, NV: WORLDCOMP.

Thorell, B., Lindqvist, S., Bergman, S., Bohlin, G., & Klingberg, T. (2009). Training and transfer effects of executive functions in pre-school children. *Developmental Science*, *12*(1), 106–113. doi:10.1111/j.1467-7687.2008.00745.x PMID:19120418

Tillman, D., Kjellstrom, W., Smith, S., & Yoder, E. (2011). Digital fabrication scaffolds for developing preservice elementary teachers’ mathematics pedagogy. In M. Koehler & P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference* (892-897). Chesapeake, VA: AACE.

Tinkercad Tutorial. (2019). https://www.tinkercad.com

Todd, R. (2016). A conceptual framework for integrated STEM education. *International Journal of STEM Education*, *3*(1), 11. doi:10.118640594-016-0046-z

Togyer, J., & Wing, M. J. (2017). *Research notebook: Computational thinking–what and why?* Academic Press.

Torres-Torres, Y. D., Roman-Gonzalez, M., & Perez-Gonzalez, J. C. (2019). Implementation of Unplugged Teaching Activities to Foster Computational Thinking Skills in Primary School from a Gender Perspective. In *Proceedings of 7th International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 209-215). New York: ACM. 10.1145/3362789.3362813

Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013). Accelerating k-12 computational thinking using scaffolding, staging, and abstraction. SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education, 609–614. 10.1145/2445196.2445374

Trochim, W. M., & Donnelly, J. P. (2008). *Research methods knowledge base* (3rd ed.). Mason, OH: Atomic Dog Publishing.

Trundle, K. C. (2010). Teaching science during the early childhood years. *Best practices and research base*.

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. In *Proceedings of European Conference on Games Based Learning* (pp.687-695). Academic Press.

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, *5*(2), 25–44. doi:10.17083/ijsg.v5i2.248

Tseng, K.-H., Chang, C.-C., Lou, S.-J., & Chen, W.-P. (2011). Attitudes towards science, technology, engineering and mathematics (STEM) in a project-based learning (PjBL) environment. *International Journal of Technology and Design Education*, *23*(1), 87–102. doi:10.100710798-011-9160-x

Tsupros, N., Kohler, R., & Hallinen, J. (2009). *STEM Education in Southwestern Pennsylvania the missing components*. Retrieved from https://www.cmu.edu/gelfand/documents/stem-survey-report-cmu-iu1.pdf

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A Model Curriculum for K-12 CS: Final Report of the ACM K-12 Task Force Curriculum Committee*. New York: ACM.

Tucker, C. M., Porter, T., Reinke, W. M., Herman, K. C., Ivery, P., Mack, C. E., & Jackson, E. S. (2005). Promoting Teacher Efficacy for Working with Culturally Diverse Students. *Preventing School Failure*, *50*(1), 29–34. doi:10.3200/PSFL.50.1.29-34

Turbak, F., Mustafaraj, F., Svanberg, M., & Dawson, M. (2017). Identifying and analyzing original projects in an open-ended blocks programming environment. In *Proceedings of the 23rd International Conference on Distributed Multimedia Systems, Visual Languages and Sentient Systems* (pp. 110-118). Pittsburgh, PA: KSI Research Inc. and Knowledge Systems Institute Graduate School. 10.18293/DMS2017-021

Turbak, F., Sherman, M., Martin, F., Wolber, D., & Pokress, S. C. (2014a). Events-first programming in app inventor. *Consortium for Computing Sciences in Colleges Northeastern Region Conference*, 81–89.

Turbak, F., Sherman, M., Martin, F., Wolber, D., & Pokress, S. C. (2014). Events First Programming in App Inventor. *Journal of Computing Sciences in Colleges*, *29*(6), 81–89.

Turbak, F., Wolber, D., & Medlock-Walton, P. (2014b). The design of naming features in app inventor 2. *IEEE Symposium on Visual Languages and Human-Centric Computing (VLHCC 2014)*. 10.1109/VLHCC.2014.6883034

Turner, S., & Hill, G. (2007). Robots in problem - solving and programming. *8th Annual Conference of the Subject Centre for Information and Computer*, 82–85. 10.13140/2.1.2425.6800

TutorialT. (n.d.). Retrieved from https://www.tinkercad.com

Twigg, S., Blair, L., & Winter, E. (2019). Using children's literature to introduce computing principles and concepts in primary schools: work in progress. In *Proceedings of 14th Workshop in Primary and Secondary Computing Education, WIPSCE '19*. New York: ACM. 10.1145/3361721.3362116

Tyler-Wood, T., Cockerham, D., & Johnson, K. R. (2018). Implementing New Technology in a Middle School Curriculum: A Rural Perspective. *Smart Learning Environments*, *5*(22), 22. doi:10.118640561-018-0073-y

Tzimas, B. (2009). Educational scenarios with the support of ICT. In *Proceedings of the 1st Educational Conference "Integration and use of ICTs" in the Educational Process.* ETPE, University of Thessaly.

University of Canterbury. (n.d.). *CS Education Research Group. Computational Thinking and CS Unplugged*. Retrieved from https://csunplugged.org/en/computational-thinking/

University of Nebraska Omaha. (n.d.). *Computational Thinking Bins*. Retrieved from https://www.unomaha.edu/college-of-information-science-and-technology/computer-science-education/community-engagement/ct-bins.php

Unplugged, C. S. (n.d.). *CS Unplugged*. Retrieved from https://csunplugged.org/en/

Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch--a discussion. *ACM Transactions on Computing Education*, *10*(4), 17. doi:10.1145/1868358.1868364

Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 1–7.

Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. In *Frontiers in Education Conference (FIE)* (pp. 1–7). IEEE. 10.1109/FIE.2014.7044114

Vaidyanathan, S. (2016). *What's the Difference Between Coding and Computational Thinking?* Retrieved from https://www.edsurge.com/news/2016-08-06-what-s-the-difference-between-coding-and-computational-thinking

Vamvakopoulos, P. (2019). *Event for Educational Robotics, with great interest*. Retrieved January 2, 2020, from http://proodeutiki.gr/?p=85159

Van de Bergh, B., & Callebaut, G. (n.d.). *Scratch unplugged. Computerless programming with Scratch*. Retrieved from http://www.scratchunplugged.com/

van der Kooij, K., Hoogendoorn, E., Spijkerman, R., & Visch, V. (2015). Validation of games for behavioral change: Connecting the playful and serious. *International Journal of Serious Games*, *2*(3), 63–75. doi:10.17083/ijsg.v2i3.75

Vasquez, J. A., Comer, M., & Sneider, C. (2013). *STEM Lesson Essentials, Grades 3-8: Integrating Science, Technology, Engineering, and Mathematics*. Portsmouth, NH: Heinemann. Retrieved from https://www.amazon.com/STEM-Lesson-Essentials-Grades-3-8/dp/0325043582

Ventola, C. L. (2014). Medical applications for 3D printing current and projected uses. *P&T*, *39*(20), 704–711. PMID:25336867

Vidakis, N., Barianos, A.-K., Trampas, A.-M., Papadakis, St., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. In B. McLaren, R. Reilly, S. Zvacek, & J. Uhomoibhi (Eds), *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)* (vol. 1, pp. 36-43). 10.5220/0007810800360043

Vilniaus Universitetas. (n.d.). *Bebras - International Challenge on Informatics and Computational Thinking*. Retrieved from https://goo.gl/7cWDp2

Vitanza, A., Rossetti, P., Mondada, F., & Trianni, V. (2019). Robot swarms as an educational tool: The Thymio's way. *International Journal of Advanced Robotic Systems*, *16*(1), 1729881418825186. doi:10.1177/1729881418825186

Vlachos, I. (2004). *Training in Natural Sciences. The proposal of constructivism*. Grigoris.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. doi:10.100710639-015-9412-6

Vuorikari, R. (2015). *Becoming digitally competent: A task for the 21st-century citizen*. Retrieved from https://www.schooleducationgateway.eu/en/pub/viewpoints/experts/riina_vuorikari_-_becoming_dig.htm

Waite, J., Maton, K., Curzon, P., & Tuttiett, L. (2019). Unplugged computing and semantic waves: Analysing crazy characters. In *Proceedings of 1st UK & Ireland Computing Education Research Conference (UKICER)*. ACM. 10.1145/3351287.3351291

Walsh, C., & Campbell, C. (2018). Introducing Coding as a Literacy on Mobile Devices in the Early Years. In Mobile Technologies in Children's Language and Literacy: Innovative Pedagogy in Preschool and Primary Education (pp. 51-66). Emerald Publishing Limited. doi:10.1108/978-1-78714-879-620181004

Wang, J. (2017). Is the US education system ready for CS for all? *Communications of the ACM*, *60*(8), 26–28. doi:10.1145/3108928

Wassila, D., & Tahar, B. (2012). Using serious game to simplify algorithm learning. *International Conference on Education and E-Learning Innovations*, 1–5. 10.1109/ICEELI.2012.6360569

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (ITiCSE'14)*, (pp.39–44). ACM.

Watters, J. J., Diezmann, C. M., Grieshaber, S. J., & Davis, J. M. (2001). Enhancing science education for young children: A contemporary initiative. *Australasian Journal of Early Childhood*, *26*(2), 1–7. doi:10.1177/183693910102600202

Weintrop, D. (2015). Comparing text-based, blocks-based, and hybrid blocks/text programming tools. In *Proceedings of the Eleventh annual International Conference on International Computing Education Research*, (pp.283–284). 10.1145/2787622.2787752

Weintrop, D., Holbert, N. S., Horn, M., & Wilensky, U. (2016). Computational thinking in constructionist video games. *International Journal of Game-Based Learning*, *6*(1), 1–17. doi:10.4018/IJGBL.2016010101

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children*, (pp.199–208). 10.1145/2771839.2771860

Weintrop, D., & Wilensky, U. (2015). Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 101-110). Omaha, NE: ACM. 10.1145/2787622.2787721

Wiebe, E., London, J., Aksit, O., Mott, B. W., Boyer, K. E., & Lester, J. C. (2019). Development of a Lean Computational Thinking Abilities Assessment for Middle Grades Students. In *Proceedings of 50th ACM Technical Symposium on CS Education (SIGCSE '19)* (pp. 456-461). New York: ACM. 10.1145/3287324.3287390

Wiegand, J. (2004). Eclipse: A platform for integrating development tools. *IBM Systems Journal*, *43*(2), 371–383. doi:10.1147j.432.0371

Wikipedia. (2018). *Science, technology, engineering, and mathematics.* Retrieved March 6, 2018, from https://en.wikipedia.org/wiki/Science,_technology,_engineering,_and_mathematics

Wikipedia. (2019). *Agia Varvara Attikis.* Retrieved January 2, 2020, from https://el.wikipedia.org/wiki/Αγία_Βαρβάρα_Αττικής

Wilson, A., & Moffat, D. 2010. Evaluating scratch to introduce younger schoolchildren to programming. J. Lawrance, R. Bellamy (Eds.), *Proceedings of the 22nd annual workshop of the psychology of programming interest group – PPIG2010*. Universidad Carlos III de Madrid.

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M. (2010). *Running on empty: The failure to teach K-12 computer science in the digital age*. New York: ACM.

Wing, J. (2010). *Research Notebook: Computational Thinking - What and Why? The Link*. Pittsburgh, PA: Carnegie Mellon. Retrieved from http://www.cs.cmu.edu/ ~CompThink/resources/TheLinkWing.pdf

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

Wing, J. M. (2011). Research Notebook: Computational Thinking - What and Why? *The Link*. Retrieved from https://goo.gl/ifrqcO

Wing, J. M. (2006, March). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. doi:10.1145/1118178.1118215

Winslow, L. E. (1996). Programming pedagogy—A psychological overview. *ACM Sigcse Bulletin*, *28*(3), 17–22. doi:10.1145/234867.234872

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 55-60). ACM. 10.1145/2818314.2818340

Wolber, D. (2011). App Inventor and real-world motivation. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*, (pp.601–606). ACM. 10.1145/1953163.1953329

Wolfson, H., Schonberg, E., Kalvin, A., & Lamdan, Y. (1988). Solving jigsaw puzzles by computer. *Annals of Operations Research*, *12*(1), 51–64. doi:10.1007/BF02186360

Wolz, U., Leitner, H. H., Malan, D. J., & Maloney, J. (2009). *Starting with scratch in CS1. ACM SIGCSE Bulletin, 41, 223–227*.

Wong, G. K. W., & Cheung, H. Y. (2018). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 1–13.

World Economic Forum. (2017). *The Global Gender Gap Report. WEF*. Retrieved from: http://hdl.voced.edu.au/10707/349201

World Economic Forum. (2019). *Education 4.0*. Retrieved January 15, 2020, from https://www.weforum.org/projects/learning-4-0

World Economic Forum. (2020). *Schools of the Future Defining New Models of Education for the Fourth Industrial Revolution World Economic Forum*. Retrieved January 18, 2020, from https://www.weforum.org/reports/schools-of-the-future-defining-new-models-of-education-for-the-fourth-industrial-revolution

Wozney, L., Venkatesh, V., & Abrami, P. (2006). Implementing computer technologies: Teachers' perceptions and practices. *Journal of Technology and Teacher Education*, *14*(1), 173–207.

Xatzopoulos, A., Papoutsidakis, M., & Chamilothoris, G. (2013). Mobile Robotic Platforms as Educational Tools in Mechatronics Engineering. In *International Scientific Conference eRA – 8* (pp. 41–51). Pireaus. Retrieved from era.teipir.gr

Xenofontos, C. (2014). The cultural dimensions of prospective mathematics teachers' beliefs: Insights from Cyprus and England. *Preschool & Primary Education*, *2*, 3–16. doi:10.12681/ppej.85

Xinogalos, S., & Satratzemi, M. (2004). Introducing novices to programming: a review of teaching approaches and educational tools. In *Proceedings of the 2nd International Conference on Education and Information Systems, Technologies and Applications* (*Vol. 2*, pp. 60-65). Academic Press.

Xinogalos, S. (2013). Using Flowchart-based Programming Environments for Simplifying Programming and Software Engineering Processes. In *Proceedings of 4th IEEE EDUCON Conference* (pp. 1313–1322). Berlin, Germany: IEEE Press. 10.1109/EduCon.2013.6530276

Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education*, *47*(2), 148–171. doi:10.1016/j.compedu.2004.09.005

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, (2), 465–470. 10.1145/1953163.1953297

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. doi:10.100711528-016-0087-7

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), 1–16. doi:10.1145/2576872

Yasar, O., & Landau, R. H. (2003). Elements of computational science and engineering education. *SIAM Review*, *45*(4), 787–805. doi:10.1137/S0036144502408075

Yelland, N. (2005). The future is now: A review of the literature on the use of computers in early childhood education (1994-2004). *AACE Journal*, *13*, 201–232.

Yelland, N. (2011). Reconceptualising play and learning in the lives of young children. *Australasian Journal of Early Childhood*, *36*(2), 4–12. doi:10.1177/183693911103600202

Yildiz Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, *34*(6), 740–752. doi:10.1111/jcal.12281

Yildiz Durak, H., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191–202. doi:10.1016/j.compedu.2017.09.004

Yin, R. K. (2017). *Case Study Research: Design and Methods* (6th ed.). Thousand Oaks: SAGE Publications.

Yoo, S.-W., Yeum, Y.-C., Kim, Y., Cha, S.-E., Kim, J. H., Jang, H.-S., ... Lee, W.-G. (2006). Development of an Integrated Informatics Curriculum for K-12 in Korea. In R. T. Mittermeir (Ed.), *Proceedings of International Conference on Informatics in Secondary Schools (ISSEP) 2006. Lecture Notes in CS 4226*. Berlin: Springer. 10.1007/11915355_19

Young, J. L., Young, J. R., & Paufler, N. A. (2017). Out of School and into STEM: Supporting Girls of Color Through Culturally Relevant Enrichment. *Journal of Interdisciplinary Teacher Leadership*, *1*(2), 2–7.

Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, *48*(3), 789–801. doi:10.1111/bjet.12453

Yusoff, A., Crowder, R., Gilbert, L., & Wills, G. (2009). A Conceptual Framework for Serious Games. *The 9th IEEE International Conference on Advanced Learning Technologies*, 21-23. 10.1109/ICALT.2009.19

Zaharija, G., Mladenović, S., & Boljat, I. (2013). Introducing basic programming concepts to elementary school children. *Procedia: Social and Behavioral Sciences*, *106*, 1576–1584. doi:10.1016/j.sbspro.2013.12.178

Zahilah Mohamed Zaki, F., Wong, S. L., & Ridzwan Yaakub, M. (2019). A Review of Common Features inComputational Thinking Frameworks in K-12 Education. *IOP Conference Series. Materials Science and Engineering*, *551*, 012063. doi:10.1088/1757-899X/551/1/012063

Zapušek, M., & Rugelj, J. (2013). Learning programming with serious games. *EAI Endorsed Transactions on Game-Based Learning*, *1*(1), e6. doi:10.4108/trans.gbl.01-06.2013.e6

Zaranis, N., & Kalogiannakis, M. (2011b). Greek primary students' attitudes towards the use of ICT for teaching natural sciences, In M. F. Costa, B. V. Dorrío, & S. Divjak (Eds.), *Proceedings of the 8th International Conference on Hands-on Science* (pp. 50-55). University of Ljubljana.

Zaranis, N., Kalogiannakis, M., & Papadakis, S. (2013). Using Mobile Devices for Teaching Realistic Mathematics in Kindergarten Education. *Creative Education*, *4*(7A1), 1-10.

Zaranis, N. (2011). The influence of ICT on the numeracy achievement of Greek kindergarten children, In A. Moreira, M. J. Loureiro, A. Balula, F. Nogueira, L. Pombo, L. Pedro, & P. Almeida (Eds.), *Proceedings of the 61st International Council for Educational Media and the XIII International Symposium on Computers in Education (ICEM&SIIE'2011) Joint Conference* (pp. 390-399). University of Aveiro.

Zaranis, N., & Kalogiannakis, M. (2011a). The use of ICT in preschool education for science teaching with the Van Hiele theory, In M. F. Costa, B. V. Dorrío, & S. Divjak (Eds.), *Proceedings of the 8th International Conference on Hands-on Science* (pp. 21-27). University of Ljubljana.

Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. doi:10.1016/j.compedu.2019.103607

Zhao, Y. X., Su, M. C., Chou, Z. L., & Lee, J. (2007). A puzzle solver and its application in speech descrambling. In *WSEAS International Conference on Computer Engineering and Applications* (pp. 171-176). Academic Press.

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, *141*, 103633. doi:10.1016/j.compedu.2019.103633

Zhu, Z. T., Yu, M. H., & Riezebos, P. (2016). A research framework of smart education. *Smart Learning Environments, 3*(4), 1-17. doi:10.118640561-016-0026-2

Zygouris, N. C., Striftou, A., Dadaliaris, A. N., Stamoulis, G. I., Xenakis, A. C., & Vavougios, D. (2017). The use of LEGO mindstorms in elementary schools. *IEEE Global Engineering Education Conference, EDUCON*, 514–516. 10.1109/EDUCON.2017.7942895

# About the Contributors

**Michail Kalogiannakis** is an Associate Professor at the Department of Preschool Education in the University of Crete (Greece) and Associate Tutor at the School of Humanities at the Hellenic Open University. He graduated from the Physics Department of the University of Crete and continued his postgraduate studies at the University Paris 7-Denis Diderot (D.E.A. in Didactic of Physics), University Paris 5-René Descartes-Sorbonne (D.E.A. in Science Education) and received his PhD degree at the University Paris 5-René Descartes-Sorbonne (PhD in Science Education). His research interests include science education in early childhood, mobile learning, science teaching and learning, e-learning and the use of ICT in science education. He has published over 120 articles in international conferences and journals, and has served on the program committees of numerous international conferences. Michail Kalogiannakis has published in journals such as “Computers & Education” (Elsevier), “Education and Information Technologies” (EAIT) (Springer) and “Early Childhood Education Journal” (Springer). He is co-editor of the following IGI title: Papadakis, St., & Kalogiannakis, M. (Eds). (2020). Mobile Learning Applications in Early Childhood Education. USA-PA: IGI Global. https://www.igi-global.com/book/mobile-learning-applications-early-childhood/233147.

**Stamatios Papadakis** is post doc researcher in Educational Technology with emphasis on mobile learning at the Department of Preschool Education at the University of Crete, Greece . He has graduated from the Department of Informatics, Athens University of Economics and Business, Athens, Greece. In 2016, he completed a Ph.D. at the University of Crete, Department of Education in Rethymnon, Greece under the supervision by Prof. Zaranis Nikolaos. His scientific/ research interests include the study of mobile learning and especially the use of smart mobile devices and their accompanied mobile applications (apps) in the use of preschool and primary education, with a special interest at the development of Computational Thinking and students’ understanding of numbers. Furthermore, his research interests also include ICT in education, novice programming environments and teaching of programming in primary and secondary education (novice programmers). Languages: English (fluent), German.

* * *

**Panagiotis Angelopoulos** is an Informatics and Maths teacher in secondary education. He was seconded at the Hellenic Ministry of Education MoE for 16 years as Head of various departments related to the development of STEM education and the development of new and innovative technologies in education. He has authored books for secondary vocational education and has published more than 25 articles in national and international conference proceedings and magazines, on issues related to the utilization of new technologies and the development of innovation in primary and secondary education. He collaborates with higher education institutions in European programs, Erasmus+ and Horizon 2020 on projects related to the development of STEM aducation and innovative technologies and didactic approaches at schools. Member of the European SchoolNet Steering Committee for 4 years and, on behalf of the MoE, responsible at national level for educational European programs, as eTwinning and Scientix.

**Alexandros Balatsoukas** is an academic scholar of the University of Groningen. He has completed a Bachelors in Spanish Studies and a Master in Intercultural Education at the National and Kapodistriakan University of Athens. Moreover, he holds a Master in Linguistics and an Honorary Master in Leadership awarded from the University of Groningen.

**Adriano Ferreti Borgatto** is a Professor at the Federal University of Santa Catarina (UFSC). His research interest is statistical analysis in educational assessment, particularly in Item Response Theory (IRT) and missing data analysis. He is currently a consultant at INEP / MEC and director of the Brazilian Association for Educational Assessment (ABAVE). He teaches the postgraduate program in methods and management in assessment and the postgraduate program in Physical Education. Previously, he worked at the Estate University of Londrina (UEL). He received a Ph.D. degree in Experimental Statistics at the ESALQ/USP (Brazil) and a Ph.D. degree in Statistics at the UNESP (Brazil).

**Avraam Chatzopoulos** is a Lecturer of Applications at the Department of Industrial Design and Production Engineering of the University of Western Attica. He has extensive experience in adult education and worked for many years as a freelancer in a significant number of IT and automation projects. Since 2000 he has been involved in the development of educational robotics platforms, while in 2005 he was a member of the research team that received an educational robot's patent from the Industrial Property Organization.

**Nathalia da Cruz Alves** is a Ph.D. student in the Graduate Program in Computer Science (PPGCC) at the Federal University of Santa Catarina (UFSC) and a research student at the initiative Computing at Schools/INCoD/INE/UFSC. She received a bachelor's degree and a master's degree in Computer Science from the Federal University of Santa Catarina. Her main research interests are in the area of computing education and assessment.

**Aykut Durak** is a teacher at Ministry of National Education, Computing and Technology & Design. He received his undergraduate degree in Department of Computer Education and Instructional Technology from Hacettepe University, Faculty of Education in 2009. He received his master degree in School Administration and Educational Supervision (Non-Thesis) from Hacettepe University, Institute of Education Sciences in 2016. He received his master degree in Lifelong Learning and Adult Education from Bartin University, Institute of Education Sciences in 2019. His research interests consist of instructional technology, technology usage and adults, programming training in K12, use of technology in education, educational use of social media, internet threats, problematic technology use, augmented reality.

**Andreas Giannakoulas** is a PhD Candidate, since 2018, in the Department of Applied Informatics at the University of Macedonia (Thessaloniki, Greece), supervised by Stelios Xinogalos, associate professor at the Department of Applied Informatics, University of Macedonia, Greece. He received his

BSc in Informatics from the Aristotle University of Thessaloniki in 2002 and his MSc in Applied Informatics from the University of Macedonia (Thessaloniki, Greece) in 2017. He also works in education as Informatics teacher since 2004. His current research is focused on programming environments and games for young children.

**Jean Hauck** holds a PhD in Knowledge Engineering and a master's degree in Computer Science from Federal University of Santa Catarina (UFSC) and a degree in Computer Science from UNIVALI. He gained practical experience in the software industry leading projects for large companies and with that experience, he was a consultant in the areas of Software Modeling, Software Architecture and Software Process Improvement. He was a visiting researcher at the Regulated Software Research Centre - Ireland. He is currently a professor in the Department of Informatics and Statistics - INE / UFSC, vice coordinator of the Software Quality Group and member of the Computing at School Initiative.

**Kalliopi Kanaki** is a Ph.D. candidate in the department of Early Childhood Education at the University of Crete, Greece. She received a bachelor's degree in Physics from the University of Heraklion, Greece, in 1994 and a bachelor's degree in Computer Science from the University of Heraklion, Greece, in 1999. She also received a master's degree in Leadership and Management in Education from the Roma Tre University of Rome, Italy, in 2014 and a master's degree in Informatics and Multimedia from the Technological Educational Institute of Crete, Greece, in 2015. She has been working as a Computer Science teacher in primary and secondary schools in Heraklion, Greece, since 2001. She also worked as a software engineer for CMI/HTA (Center for Medical Informatics and Health Telematics Applications) at ICS/FORTH (Institute of Computer Science, Foundation of Research and Technology), in Heraklion, Greece (1999 - 2005).

**Alexander G. Kapaniaris** was born in Zagora and has two children. He works as an IT teacher in the Secondary Education and he is a Regional Coordinator of Educational Work of Computer Science of Thessaly. His basic degree is as an IT Engineer of the Technological Institute of Computer Engineering and Information of Athens and is a graduate of the department of "Studies in Greek Culture" of the Faculty of Humanities of the Greek Open University and a graduate of the School of pedagogical of PATES / S.E.L.E.T.E. of Thessaloniki. He also has MSc degrees on the topics of "Graphic Arts - Multimedia" from the Greek Open University and Modern Learning Environments & Production of Teaching Material of University of Thessaly (Department of Education). Since 2017, he has been a PhD of Folk Culture, Education and ICT, of the University of the Aegean (Department of Early Childhood Education and Educational Planning). Also he is a post doc researcher in the Department of History and Ethnology, Democritus University of Thrace. He has published and has participated in Greek and international conferences as a rapporteur and a judge and his research interests include the E/H educational technology, internet addiction, pedagogical applications, educational software, ICT, cultural informatics, digitizing historical archives, local history and folk culture. He has authored two books on the teaching use of ICT in education and the digital school. He has taught "Systems of Electronic Learning and Digital Means in the Education in the Department of Digital Systems in the University of Piraeus". He is currently teaching in the Greek Open University in postgraduate level the course "Digital Storytelling in education".

**George Lepouras** is a Professor at the Department of Informatics and Telecommunications, University of Peloponnese and Vice-Rector for Finance, Planning and Development at the same institution.

He holds a first degree in Mathematics from University of Athens, MSc in Information Technology from University of Strathclyde, Scotland and a PhD in Human Computer Interaction from University of Athens. Dr Lepouras is a senior member of ACM and has served as the chairman of the Greek ACM SIGCHI. His research interests include game-based profiling, personal and task information management, multilingual interfaces, web based interfaces, virtual reality and augmented reality applications as well as cultural technologies. He is an author and co-author of more than hundred papers, of which more than forty appear in international journals. He has participated, as researcher, technical manager or team coordinator, in many national and European research and technological development projects, among which are SmartGov, DELOS, Experimedia, CROSSCULT and FORESIGHT.

**Gaia Lombardi** is an Italian primary school teacher with over 25 years of experience in the pedagogical and educational field. After a training in humanities, she started working with children in the 0-3 age group, with an action research activity in kindergartens published in 2006 ("The adventure of growing together", Junior Edizioni). As a primary school teacher she specialized in teaching English as a second language and in the CLIL methodology for young learners. Since 2016 she has been interested in the introduction of coding and computational thinking in the school, within the community of teachers codemooc.net.

**Marina Molla**, Teacher at the 9th Primary School of Komotini & SCIENTIX Ambassador. Studied at the University of Ioannina in the Pedagogic Faculty for Elementary Education and at the Democritus University of Thrace in the Faculty of History & Ethnology. Master's Degree in "Local History – Interdisciplinary Approaches" at the Democritus University of Thrace in the Faculty of History & Ethnology In-service Training at "Didaskaleio Theodoros Kastanos" in Alexandroupolis.

**Antonios D. Niros** received his B.Sc. in Computer Science from the National & Kapodistrian University of Athens, Department of Informatics & Telecommunications in 1996, his M.Sc. in Digital Communication Systems, from the Department of Electrical and Electronics Engineering, Loughborough University (UK) in 1997, his Master's Diploma in "Educational Studies" from the School of Humanities of the Hellenic Open University in 2016 and his Ph.D. from the Department of Cultural Technology & Communication of the University of the Aegean in 2012. He has extensive experience (over two decades) in teaching positions in Secondary - Tertiary but also Adult Education. He has published in International Scientific Journals and Conference proceedings on issues of Artificial Intelligence, Artificial Neural Networks and Image Processing. He is reviewer in many International Journals and Conferences. He is WRO, ST3dM and Scientix Ambassador for the school years 2019-20.

**Adina Nistor** works as a Project Manager in the Science Education Department of European Schoolnet (EUN). Adina is responsible for managing Impact EdTech, a H2020 project focusing on digital learning solutions, as well as EUN's activities in the ARETE and BLOOM projects. Since she started working at EUN, her activities revolved around community building, piloting and evaluation. For two years, she has been coordinating the organization of Scientix Projects Networking Events, as well as the STEM education research activities published in the Scientix Observatory (http://www.scientix.eu/observatory).

**Dimitrios Papachristos** is a Teaching Assistant of the Department of Industrial Design and Production Engineering at the School of Engineering (University of West Attica). He received a B.Sc. in Automation from the TEI Piraeus and Phd degree in Shipping from University of Aegean, Greece. Dimitrios Papachristos has extensive experience in private sector and he has been involved in several national and european research projects. He has published research papers in journals, books and workshops and his research interests involve, the educational technology and software, neuroscience in maritime education, green shipping, organizational behavior, human-machine interaction and information systems.

**Michail Papoutsidakis** is an Associate Professor at the Industrial Design and Production Engineering Department of the University of West Attica, Greece. He is a PhD holder from Bristol Robotics Laboratory, UK, in the field of artificial intelligence and control methods for robotic systems. He also gained a PostDoc degree in the same research field in 2017, from University of Thessaly, Greece. Since 2005, Dr.Papoutsidakis has been teaching Robotics, Mechatronics and Automation Control Systems in a University level and has already published more than 100 research articles in international journals and conferences focused in the area of robotics applications and education.

**Panagiotis Politis** is Associate Professor of "ICT in Education and Didactics of Informatics" at the Department of Primary Education, University of Thessaly (Greece). He has previously worked as a visitor Senior Lecturer at the University of Athens, Department of Primary Education postgraduate level. He holds a BSc in Physics from the University of Patras (Greece); a MSc in Didactics of Informatics, from the University Paris 7 (France) and a PhD in ICT in Education from the University Paris 7 (France). He specialises in the area of ICT in Education, Educational Software Development and Didactics of Informatics. His research interests include ICT in education, modelling software design, pedagogical software evaluation and adults training. He has extensive expertise on computers multimedia sector and has published more than 100 papers and books on various subjects of computer science education. He has participated in IST, ESPRIT research and national research projects.

**Emmanouil Poulakis** is a PhD student at the Department of Primary Education, University of Thessaly (Greece), and his dissertation's subject is "Computational thinking development using unplugged activities". He holds a BSc degree in Informatics and Telecommunications from the University of Athens (Greece) and a M.Ed. in Gender and New Educational and Work Environments in Information Society from the University of Aegean (Greece). He is currently employed as a Computer Science teacher in secondary education in Greece and serves as Assistant Director at 2nd Vocational Training Institute of Heraclion, Crete (Greece). He has 15 years of expertise in teaching ICT and Computer Science in secondary and primary education, as well as in vocational training. His research interests include computational thinking, ICT in education and Didactics of Informatics.

**Vasiliki Psaridou**, Sub-Headmaster in the Minority School of Dokos, Leading Teacher in Coding. Studied at the Democritus University of Thrace in the Faculty of Pedagogic Faculty for Elementary Education and at the Aegean University, Department of Preschool Education and Teacher Education Design, Master's Degree entitled "Models Design and Development of Educational Units" with direction Management and Evaluation of its Training Units.

**Sarantos Psycharis** is Professor at the School of pedagogical and technological education-ASPETE (www.aspete.gr), and for many years he teaches courses relevant to Engineering and Science Education and ICT in Education(sarantospsycharis.weebly.com, http://education.aspete.gr/index.php/en/personnel/

acc/research-commitee/206-psycharis-sarantos.html) He holds a PhD from the University of Glasgow in Computational Physics and Msc in Information Technology, National University of Athens AND A Bachelor Degree in Physics from the National Kapodistrian University of Athens-Greece He was Chancellor of ASPETE. He has published more than 120 publications in STEM Education and he has been involved in many European and National Projects for “STEM Education”. From 2015-2018 he was Scientific director of the Msc program “STEM in Education” a validated Msc program. Professor Psycharis has developed STEM applications that can be used in school education as well as in Tertiary Education. He has been also involved in European projects as developer of learning platforms. Throughout his career, he has also served as an Advisor to the Greek Minister of Education on issues related to the integration of ICT in Education, and Teacher Education Policies. He was also the Coordinator, representing the Greek Government – for the European Network of Teacher Education Policies (ENTEP), and for two years President of the Network and Evaluator for European Commission. He has published mora e than 120 papers in peer review journals and conferences, related to STEM Education and Science Didactics. He has worked at ASPETE and Other Universities in Europe, for Committees responsible for the design and development of educational material/curriculum for TPACK and STEM Education for teachers’ training curricula and the curriculum for undergraduate and postgraduate students. He is the President of the Hellenic Society for STEM Education-http://e3stem.edu.gr/wordpress/?lang=en and member of the Editorial board of the Journal Hellenic Journal of STEM Education http://www.hellenicstem.com/index.php/journal.

**Mustafa Saritepeci** is assistant professor of computer education and instructional technology in the Ereğli Faculty of Education, Necmettin Erbakan University, in Turkey. He received his undergraduate degree in Department of Computer Education and Instructional Technology from Hacettepe University, Faculty of Education in 2009. He received his master degree and PhD degrees in Department of Computer Education and Instructional Technology from Gazi University in 2012 and 2016, respectively. His research interests include educational technology, technology-based behavioral addictions, ICT in teacher education, computational thinking and design-based learning, programming training in K12, use of technology in education, blended learning, e- learning.

**Efthalia Solomou** is a professor of Classics (Ancient, Modern Greek Literature and History, Latin and Phiosophy) in the Secondary and High School. She has completed the bachelor of Classis in the National and Kapodistriakan University of Athens. She is currently participating in the Master ¨Public Health¨ at the University of West Attica. Last but not least. she is a certified tutor of the Ministry of Education with specialty at ¨Capitalization and application of the Digital Technologies on the educational activity¨.

**Dimitrios Stamovlasis** is an Associate Professor of Research Methodology and Applied Statistics in the Department of Philosophy and Education. His research interests focus on methodological and epistemological issues concerning the theory development in psychology, education, behavioral and life sciences. He has published numerous papers in international journals and chapters in collective volumes, and he is serving as reviewer and member of editorial boards. Specific domains of research are neo-Piagetian theories, creativity, motivation, conceptual change, mental models, children’s mental representations, problem solving and conceptual understanding, group dynamics, computational thinking and robotics Special interest is the application of multivariate statistical modelling, network analysis, complexity theory, catastrophe theory, and nonlinear dynamics to social science research.

**Anastasios Theodoropoulos** is a postdoctoral researcher in the field of games and learning technologies at the Department of Informatics and Telecommunications of University of Peloponnese. He received his Ph.D. from the same department and his research focus was on alternative teaching methods for Computer Science Education. Dr. Anastasios Theodoropoulos has extensive experience in teaching within the Computer Science domain, in a wide variety of courses, to students aged from 6 years old to mature adults and in multicultural settings. He has co-authored several manuscripts published in peer-reviewed journals and conferences and has served/serves as editor on highly recognized journals. He also serves as a national Ambassador for the European Code Week initiative, while he is an active member of ACM, SIGCSE and CSTA organizations. He does research on Computer Science Education, Programming environments & techniques, Learning Technologies, CHI & Education and Game-Based Learning. He pursues research concepts such as user experience modeling and procedural content generation for the design of personalized interactive systems for learning programming.

**Timoleon Theofanellis** has worked in the field of education for 25 years as a secondary and adult school teacher, teacher educator, curriculum designer and currently he works as Fellow Assistant of ASPETE School of Pedagogical and Technological Education and at the 5th junior high school. He has studied Physics and Computer Science from B.Sc. to PhD and has 25 Publications in journals and 60 presentations in conferences (Greek and international). Has participation in 17 books on educational issues, educational web use, computer science. His current interests are in the field of educational technology especially on educational uses of web 2.0 tools, social learning tools and social learning.

**Ahmet Topçu** is a teacher at MNE. His research interests consist of technology usage and adults.

**Savvas Tsolakis** received his bachelor degree in Department of Electrical and Computer Engineering of University of Thrace in 2000 and his MS in Automation In Irrigation, In Agricultural Constructions And In The Mechanization Of Agriculture of Department of Agriculture Crop Production and Rural Environment Of University of Thessaly in 2016. He worked as an Electrical Engineer for two years. From 2004 he is a secondary school teacher on informatics. One of his interests is educational robotics, especially Arduino system and LEGO EV3. He is preparing school teams for robotic competitions. He is now studying Archaeology in the University of Thessaly in the Department of History, Archaeology and Social Anthropology.

**Christiane Gresse von Wangenheim** is a professor at the Department of Informatics and Statistics of the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, where she coordinates the Software Quality Group focusing on scientific research, development and transfer of software engineering models, methods and tools and software engineering education in order to support the improvement of software quality and productivity. She also coordinates the initiative Computing at Schools, which aims at bringing computing education to schools in Brazil. She received the Dipl.-Inform. and Dr. rer. nat. degrees in Computer Science from the Technical University of Kaiserslautern (Germany), and the Dr. Eng. degree in Production Engineering from the Federal University of Santa Catarina (Brazil). She is also PMP – Project Management Professional and MPS. BR Assessor and Implementor.

**Evaggelia Voulgari** received her bachelor degree in Informatics from the University of Piraeus in 1997 and her MS in Department of Electrical and Computer Engineering of the University of Thessaly in 2008. She worked as a senior software engineer, developing in house software, for more than six years. She specialized in Document Management, Data Warehouse and Customer Relationship Management systems. From 2003 she is a secondary school teacher on informatics. One of her interests is educational robotics and she is preparing teams for robotic competitions. She is now studying economics in the University of Thessaly in the Department of Economics.

**Stelios Xinogalos** received a PhD degree from the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece in 2002. He currently works as an associate professor at the Department of Applied Informatics, University of Macedonia, Greece. His research interests include Serious Games, Programming Environments and Techniques, Object-oriented Design and Programming, Didactics of Programming and Educational Programming Environments. He has published more than 90 peer-reviewed articles in international journals, conference proceedings and books.

**Hatice Yildiz Durak** is a researcher at Bartin University, Faculty of Education / Science, Department of Computer Technology and Information Systems. She received her undergraduate degree in Department of Computer Education and Instructional Technology from Hacettepe University in 2009. She received her master degree and PhD degrees in Department of Computer Education and Instructional Technology from Hacettepe and Gazi Universities in 2011 and 2016, respectively. Her research interests include, programming training in K12, use of technology in education, flipped learning, e-learning, teacher training, pre-service and in-service training, educational use of social media, digital divide, internet threats, problematic technology use, cyber bullying, cyberloafing, cyber-psychology, computational thinking.

**Konstantinos V. Zacharis** serves today as Principal of 5th Lyceum of Karditsa. He received his Diploma in "Computer Engineering and Informatics" from Polytechnic School, University of Patras in 1991, his Master's Diploma in "Educational Studies" from School of Humanitarian Studies, Greek Open University in 2003 and his PhD by the Business, Education and Law School, Staffordshire University, UK, in 2016. He has extensive experience (over 25 years) in administrative and teaching positions in Secondary - Tertiary but also Adult Education (he is a Certified Tutor at B-level in Ministry of Education). He is an author of books, school texts and articles on ICT in education, while he participates at conferences and coordinates various EU Erasmus+ projects. He publishes regularly at international research journals and is also a reviewer. His research interests span from ICT pedagogic applications to Educational Leadership. He speaks fluently English and German.

# Index

## L



## M



## N



## P



## R



## S



## T