MODESTUM
OPEN ACCESS

EURASIA Journal of Mathematics, Science and Technology Education, 2019, 15(4), em1689
ISSN:1305-8223 (online)
Research Paper https://doi.org/10.29333/ejmste/103071

The Impact of a STEM Inquiry Game Learning Scenario on Computational Thinking and Computer Self-confidence

Sarantos Psycharis [1*], Evangelia Kotzampasaki [2]

[1] Professor School of Pedagogical and Technological Education, ASPETE, Athens, GREECE
[2] School of Pedagogical and Technological Education, ASPETE, Athens, GREECE

Received 30 May 2018 ▪ Revised 26 December 2018 ▪ Accepted 8 January 2019

ABSTRACT

Computational thinking is an ability which is considered to be essential for the process of problem solving in every science. The current empirical research aims to study the impact of a STEM content Inquiry based scenario using computational tools and educational games, regarding computational thinking (CT) and confidence for "computers use" of 115 students of Greek public schools of the 5th-6th grade. For the needs of this research, a didactic scenario was developed and implemented, using computational tools, such as the Arduino microcontroller, RGB Led's while a computational model was designed and implemented. The assessment of computational thinking improvement and confidence for computers use was conducted with the use of questionnaires that were administered before and after the intervention. The findings indicate a positive influence of the intervention on the dimensions of computational thinking in the experimental group. The findings can be applied to educational settings that integrate STEM in the teaching sequence in order to enhance students' confidence with computational experiments.

Keywords: computational thinking, STEM, game learning, Arduino, computational pedagogy, self-confidence

INTRODUCTION

There is an increased interest for the way Computational Thinking (CT) should be implemented in the teaching approach, especially for students without previous use of computers; specifically, which tools should be used and what would constitute a proper pedagogical structure in order for the learning process to be effective (e.g. Guzdial, 2008). This raises also a number of questions, including how to integrate computational thinking into the curriculum (e.g. Barr et al, 2011; Sentance & Csizmadia, 2015; Voogt et al., 2015), how CT can be embedded in a pedagogical framework integrated with the STEM epistemology (Psycharis, 2018), which didactic model should be appropriate for its inclusion, and what computational tools and computational methods are appropriate for the school education settings (Bower et al., 2017; Psycharis & Kotzampasaki, 2017).

Computational Thinking (CT)

Computational Thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the fundamental concepts of computer science, while it is also considered as a universal skill and attitude that complements thinking in mathematics and engineering with a focus on designing systems that facilitate the solution of complex problems humans face (Wing, 2006, 2008). Researchers argue that there is no clear-cut definition of computational thinking (Bower et al, 2017; Hu, 2011). While computational thinking draws upon concepts that are fundamental to computing and computer science (Denning, 2007), it also includes practices, such as problem representation, abstraction, pattern recognition (machine learning) decomposition, simulation, verification, and prediction (Bower et al, 2017, Psycharis, 2018; Sengupta et al., 2013). These practices are related to the development of models (as representations of the physical world), the scientific reasoning, as well as to Science

✉ spsycharis@gmail.com **(*Correspondence)** ✉ evikotza@gmail.com

Contribution of this paper to the literature

- It proposes a frame of reference of a pedagogical framework that utilizes CT in primary school education, even for students without previous computer experience, by implementing specific computational tools.
- It provides quantitative data for the effectiveness of the implementation of a "STEM inquiry game learning" scenario on students' computational thinking skills for every CT dimension and for every different difficulty level, as well as for the impact in students' self-confidence in computer use.
- It provides a systematic evaluation method in order to determine whether or not a pedagogical framework is successful in encouraging the development/improvement of CT skills.

and Mathematics teaching. According to Weintrop et al. (2016), Science and Mathematics are becoming computational endeavors highlighting the need of use computational methods. Next Generation Science Standards (NGSS, 2013) also suggest that CT "is a core scientific practice and due to the increased presence of computation in mathematics and scientific contexts, a new urgency has come to the challenge of defining computational thinking and providing a theoretical grounding for what form it should take in Science and Mathematics".

Bundy (2007) declared that the ability to think computationally is pervasive thus has consequential impact on every kind of thought and to all disciplines. National Research Council refer to concepts from Computer Science (NRC, 2010), while many researchers place emphasis on the fact that "Despite the obvious relevance of CT to computer science, scholars argue that CT needs to be taught in disciplines outside of computer science beginning in kindergarten" (Barr & Stephenson 2011; Kotsopoulos et al. 2017, pp. 2; Yadav et al. 2011). According to Einhorn (2012), computational thinking favors a variety of skills, such as logical reasoning, critical thinking, creativity, algorithmic thinking, modelling and simulations, and assists learners become engaged in the scientific methodology. Educational technology also considers CT as an essential skill for the 21st-century skills (Einhorn, 2012). Selby and Woolard (2014) propose the following "dimensions" of CT as the more fundamental ones.

Abstraction (AB): Although researchers have accepted abstraction as a central concept in computational thinking, they disagree on the meaning of it (Cetin & Dubinsyb, 2017). Piaget introduced the concept of reflective abstraction to describe children's construction of abstract logico-mathematical structures (Beth & Piaget, 1966) and he distinguished three types of abstraction: empirical, pseudo-empirical, and reflective abstraction. According to (Cetin & Dubinsyb, 2017), reflective abstraction can be used as a tool in the study of computational thinking. They suggested that "The most common meaning of abstraction of a concept in computer science and mathematics, is extraction, that is, the idea of considering common features of several examples and building a structure or category which has all of these features". They also address another component of abstraction, which is the decontextualization, as a way of thinking about a concept in dependently of any context is what makes abstraction difficult (Gravemeijer & Doorman, 1999). Within a next section of the current study, referring to the Computational Science Education (CSE), we will connect the abstraction with the models of simulations. Wing (2008) connected abstraction with automation arguingthat the mechanization of abstraction layers and the relationships between them leads to abstraction, she defined that computing is the "automation of our abstractions".

Algorithm (AL): The term algorithm is interpreted as a step-by-step procedure for accomplishing tasks, not just in computer science, but in other disciplines (Selby & Woolard, 2014). In literature, algorithms are connected to different levels of abstraction, and we believe that the most abstract level connected to the algorithm, is its relation to the "problem" that we have to solve.

Algorithms provide solutions to problems and they have the following properties:

1. They consist of a step-by –step set of instructions
2. An algorithm is a finite process, i.e. it finishes at some point

Decomposition (DE): Selby and Woolard (2014, pp7) stated that according to NRC (2011), "the creation of solutions requires breaking problems down into chunks of particular functionality and sequencing the chunks

Generalization (GE): It is considered as the ability to expand from a specific to a broader applicability also related to pattern recognition. It should for example use an exponential decrease law in an R-C circuit and understand that the same mathematical function also applies to the decay rate of the nuclei. "The ability to recognize parts of solutions that have been used in previous situations or that might be used in future situations is included by Kolodner in a definition of computational thinking" (Selby & Woolard, 2014, pp. 12; NRC, 2011).

Evaluation (EV): Computational Thinking includes evaluation the ability to evaluate processes, in terms of efficiency and resource utilization, and the ability to recognize and evaluate outcomes" (L'Heureux et al. (2012); Selby & Woolard, 2014, pp12)

These dimensions seem to be more widely accepted (Dorling, 2017) and there also used for the evaluation of CT in the International computational thinking competition "Bebras" (http://www.bebras.uk/). This completion is

held in more than 50 countries, and it includes questions that are differentiated according to their level of difficulty (A, B, C) and to the age group of students (six different groups of ages are included). Dagiene & Stupuriene (2016) argued for the added value of this completion in which for each question there is a clear correspondence for which dimensions of CT are involved.

Engineering Education Epistemology (EEE)

There is a strong link between Computational thinking dimensions and Engineering Education "According to Shirey (2017), the discipline of engineering can be divided into engineering content and engineering design. Engineering content arises from the intersection of science, mathematics, and encompasses a collection of tools, which engineers can use to design solutions to specific problems based on criteria and constraints. Rugarcia et al. (2000) described engineering education as the development of engineering knowledge (facts and concepts), skills (design, computation, and analysis), and attitudes (values, concerns and preferences). Berland et al. (2013) consider that engineering in high schools can influence students' deep learning and teach students the engineering design process (Psycharis, 2018, pp.51).

Katehi et al. (2009) state that "perhaps the most important for engineering is design, the basic engineering approach to solving problems and when students are engaged in the design process, they can integrate various skills and types of thinking—analytical and synthetic thinking and detailed understanding". They also state that the engineering design process is "(1) highly iterative; (2) open to the idea that a problem may have many possible solutions; (3) provides a meaningful context for learning scientific, mathematical, and technological concepts; and (4) provides stimulus to systems thinking, modeling, and analysis while engineering design is a potentially useful pedagogical strategy". We comment on the emphasis given in this report in the modelling concept/process, which is a CT skill and the types of thinking that are closely related to the dimensions included in CT (Psycharis, 2018, pp.51). Moreover, according to NRC (2012, a, b) CT is closely connected to Engineering Education Epistemology (EEE).

STEM Epistemology

There are two approaches for STEM education integration: the content integration and the context integration. Content integration (Moore, 2008) "focuses on the merging of the content fields into a single curricular activity or unit to highlight "big ideas" from multiple content areas". the operation of a transistor in order to illustrate the power and possibilities of teaching a fully integrated STEM context could be a possible illustration of highlighting a big idea. The concept of the transistor contains concepts form all the STEM cognitive areas, while hands-on artefacts about the construction of speakers could allow/enable teachers and students explore the variables that impact the current and voltage amplifiers. The big idea is the "amplification of the current" and all the STEM cognitive areas are included in this idea. Physics concepts, such as electric field and voltage are combined with numerical analysis from mathematics (in order to find by iteration the values of the current in the diode p-n) and with technology (selection of semi- conduction materials). Finally engineering uses all of them to design artefacts that utilize the transistor in industry.

In this big idea, engineering epistemology is implemented through the engineering design and the by developing a model (variables and selection of variables). Learners should simulate the model (simulation acts as a working model) and they can compare the results with known data. For example, using the LabView software (http://www.ni.com/en-us/shop/labview.html), learners can create the design and the model; they collect the data and make the refinements to adjust the variables of the model in order to be in agreement with the data form a physical experiment. The whole process is a typical engineering design method, in alignment with the STEM content approach. According tothis approach, the activity about the design and use of the transistor, needs a series of lectures to be implemented in the classroom and faces a problem of real life. The "problem" of the transistor is faced by an integrated approach in a holistic way and not in separate issues (i.e. first discussing issues form physics, next move to mathematics etc). This process can be implemented either by using the computational experiment (see next section) via physical computing (e.g. Arduino construction, LabView design), or without the implementation of computers, i.e. unplugged computing (student can create a transistor from simple materials).

Within the context STEM integration approach, the focus lays on the content of one discipline and concepts from other disciplines are used to make the content more relevant. For instance, a mathematics teacher might choose a unit from probability about Bayes theorem and then he can ask students to analyses samples from a biochemistry lab in order to examine the probability for diseases using conditional probabilities. In another example, the teacher teaches algorithms and then asks engineering students to visit different networks and register the response time in a network with different number of nodes.

It is considered that any epistemological approach for STEM should be connected to the approaches of Mode-2 (Nowotny, 2003) and "Nicolescuian" methodological approach (Nicolescu, 2004).

STEM epistemology is related to Mode-2 system as it faces problems that emerge from different disciplines and loose organizational structures, flat hierarchies, and open-ended chains of command

STEM integration even shares some issues alongside with the Nicolescuian methodological approach (Nicolescu, 2004). Realities of Nicolescuian methodological approach can appear in personal epistemology when students create their own model. Complexity, according to Nicolescuian methodological approach, could also be related to STEM content epistemology. According to (Nicolescu, 2004), complexity "is a modern form of the ancient principle of universal interdependence, in that everything is dependent on everything else, everything is connected, and nothing is separate". This definition of complexity, alongside current research definitions efforts of complexity, raises awareness about issues, such as emerging behavior or connection of scales that could be related to STEM content, since STEM faces complex problems.

Issues, such as the relationship between the interdependence of the constituents of a complex system, the structure of a complex system which spans several scales can only be confronted through the STEM contact approach.

Based on the analysis above, we consider that STEM epistemology should follow the Mode-2 Transdisciplinarity as it faces problems that emerge not only form one cognitive area and the "Nicolescuian" methodological approach, while it utilizes the main issues of the engineering education epistemology.

Physical Computing

Physical computing is considered as a linkage between the computers to the physical world (Martinez & Stager, 2013) combining digital elements with real situations, by creating an interface conversation between the physical world and the virtual world of the computer (Schulz & Pinkwart, 2015).

In the context of computer science education, most of the research focus on programming (e.g. Qiu et al., 2013, Psycharis at al., 2017). Physical computing can be implemented in computer science in two ways: either to teach concepts of computer science using physical computing, or to use physical computing selectively as an entry point to different topic areas of computer science (Przybylla & Romeike, 2014). Physical computing "takes the computational concepts "out of the screen" and into the real world so that the student can interact with them by changing the model" (Rubio et al., 2013, pp. 1). Our argument is that physical computing is strongly connected to the dimensions of CT, namely: abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization. Using a STEM epistemology as a starting point, we implemented physical computing using the Arduino platform combined with the S4A software (http://s4a.cat/).

According to Zieris, Gerstberger and Müller (2015), it is difficult to engage students in CT skills, when the computational tools act as "black box". For this reason, many researchers, support the view of using Arduino instead of "closed" tools. The use of Arduino is also supported in research, which also stress its connection with the STEM epistemology, especially for primary school students. According to Cheng et al. (2016), Arduino can be the bridge between theory and practice in the educational settings. Programming in Scratch is considered as a serious game with a strong impact on students' enhancement of CT skills and problem solving (Kazimoglou et al., 2012).

The Computational Science Education

Computational Science (C.S.), in general, has its origins in Monte Carlo modeling and algorithms, such as Lanczos algorithm, and has been applied for solving complex problems in Physics (Landau et al., 2008; Psycharis, 2018).

According to a number of authors (Yasar, 2004; Yasar & Landau, 2003), Computational Science (C.S.) overlaps with many other knowledge areas, so an educational program in (C.S.) naturally draws strength from all of them. Nevertheless, in addition to overlapping with computer science, math, and science and engineering application areas, (C.S.) has developed its own core knowledge area.

Juszczak (2015) states that (C.S.), in both natural and social sciences, "is different than the usage of computers to analyze complex systems and data sets. Computational Science is a non-empirical science. Data that is gathered in computational science is the result of simulations and virtual experiments. The key distinction between a true "Computational Science" and a science that uses computation is in the nature of evidence: traditional science and science experimentation that use computation to assist in the analytic and experimental process have, as their threshold of truth, empirical evidence. Computational Science, on the other hand, conducts experiments that are only virtually true and attempts to use data about the real world in order to conduct real experiments in a virtual universe".

Using Computational Science, "we accept that we can conduct experiments and iterations that could never be conducted in the real world with results equivalent to the classical experiments. For this reason, cognitive areas like

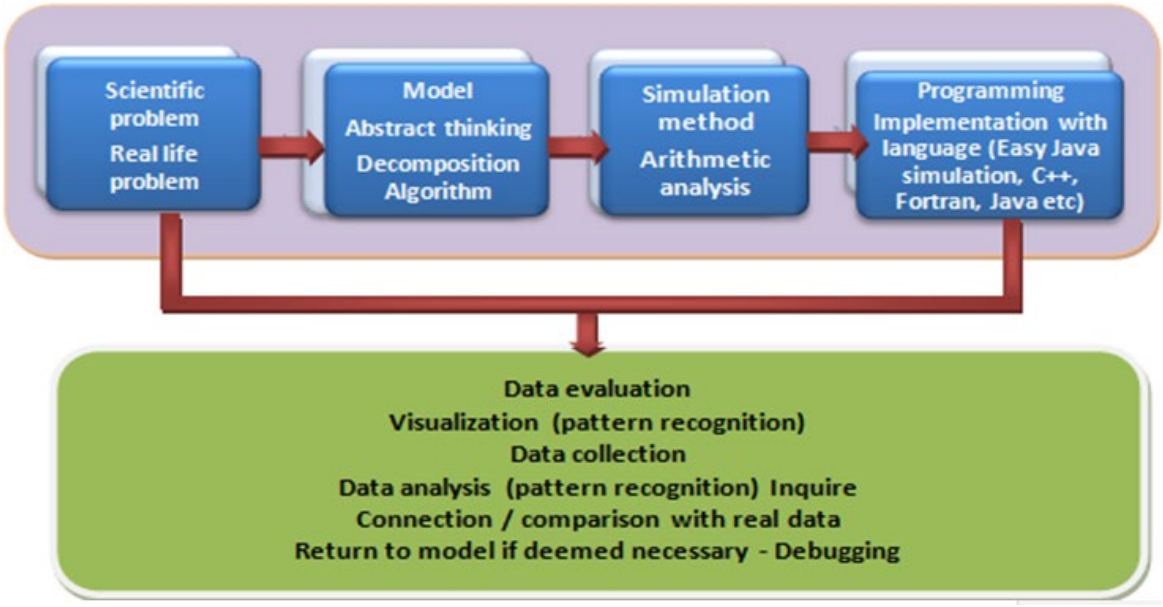

Figure 1. The Computational Science Experiment (CSE experiment)

computational psychology; computational biology, computational astrophysics, computational chemistry, and computational sociology have developed over the past several decades" (Psycharis, 2018, pp.62).

Computational Science in Education (C.S.E.) can be an effective methodology to support learners to solve a STEM problem using models of simulations. In this process there are included diverse tasks, such as: formulating the problem in a way suitable for simulations using models ,choosing an efficient computational algorithm, running the simulations and collecting numerical data, analyzing the data obtained, finding patterns in order to generalize the method to other problems and extracting the solution of the problem in a form that can lead to the creation of artifacts. All the above mentioned components of the (C.S.E.) methodology related to the dimensions of CT establish a clear connection between (C.S.E.) and CT.

C.S.E. focuses on a real life problem and follows a scientific problem–solving paradigm with a sequence of steps: a. Problem (from science/real world); b. Modelling (Mathematical relations between selected variables-decomposition of the problem); c. Simulation Method (time dependence of the state variables, discrete, continuous or stochastic processes, selection of proper interfaces); d. Development of the algorithm based on numerical analysis methods; e. Implementation of the algorithm (using Java, Scratch, Python, Arduino, raspberry pi etc); and f. Assessment and Visualization through exploration of the results and comparison with real data received from real life phenomena. C.S.E shares many commonalities with CT and may serve as the background platform to implement applications that include the dimensions of CT (Psycharis, 2018). In **Figure 1** we present the methodology known as Computational Experiment (C.S.E. experiment).

The Computational Pedagogy Model

The term Computational Pedagogy was introduced by Yasar et al. (2016) as an extension of Technological Pedagogical Content Knowledge (TPACK) and was called Computational Pedagogical Content Knowledge (CPACK). Yasar (2013, pp. 10) states that "Computational modeling and simulations provide us with a deductive pedagogical approach by enabling us to introduce a topic from a simplistic framework and then move deeper into details after learners gain a level of interest to help them endure the hardships and frustration of deeper learning. Computational pedagogy puts the learner at the center of a constructivist experience that utilizes both bottom-up (abstraction) and top-down approaches to teaching".

The process of abstraction is inductive processes by which we sort out/organize details and connect the dots to arrive at more general patterns and conclusions. Abstraction is also connected to pattern recognition and is included in the CSE experiment methodology.

Yasar et al (2016, pp. 1) propose a model (CMST) in which "computational modeling and simulation technology (CMST) is used to improve technological pedagogical content knowledge (TPACK) of teachers". According to Yasar et al. (2016), "when mathematics, computing, and sciences are integrated, "their integration gives birth not only to a new content domain of computational science, as witnessed by degree programs in the past two decades but also a particular computational pedagogy. This multi- faceted interdisciplinary knowledge domain has been called

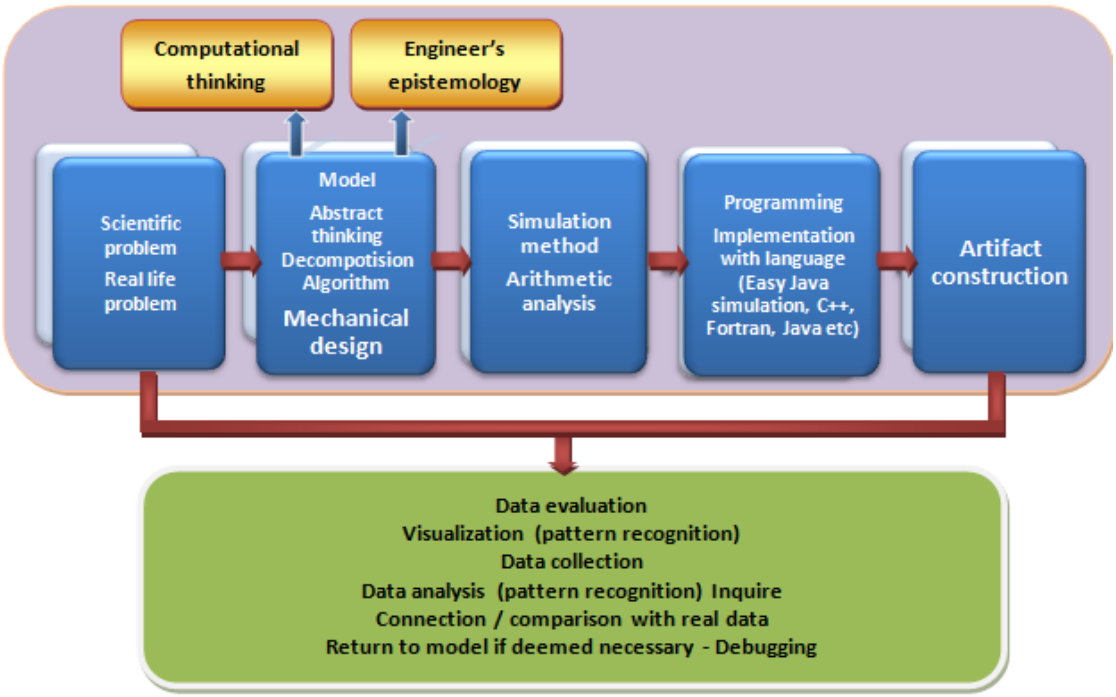

Figure 2. The Computational Science Experiment (CSE experiment) with engineering design and CT

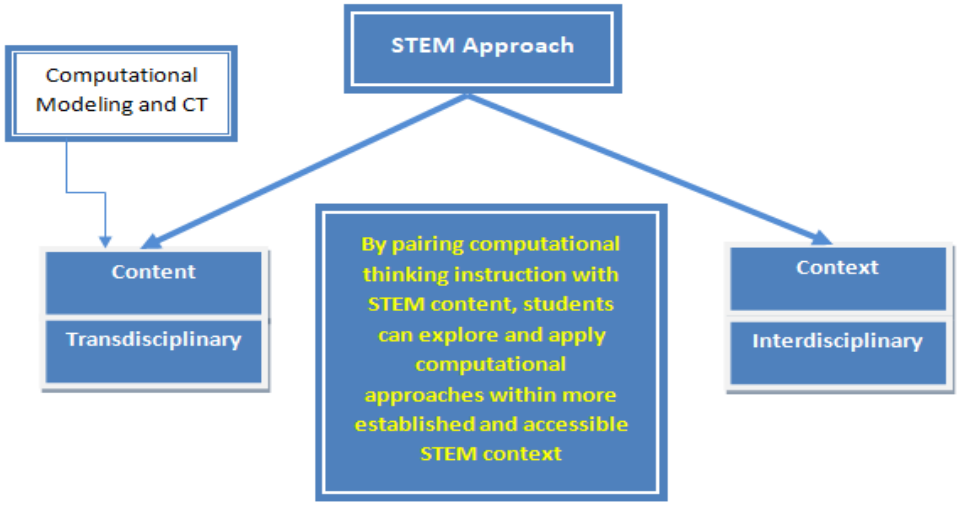

Figure 3. The "Computational STEM Pedagogy"

Computational Pedagogical Content Knowledge (CPACK) domain framework". Psycharis (2018) adopted the model of Yasar et al. (2016) with some slight modifications and to engineering design practices were added to the so called computational experiment spaces. This lead to the modification of **Figure 1** leading to **Figure 2**, which includes CT and engineering epistemology.

Psycharis (2015, 2016), Psycharis and Kotzampasaki (2017), and Psycharis et al. (2017) discussed the spaces of the computational experiment and proposed inquiry based activities at each space. To use the model and simulation in the inductive process of teaching, we need proper environments that favor the use of mathematics and algorithms, so the computational experiment will be "equivalent" to the physical experiment.

In our model we integrate the inquiry based teaching and learning approach, the C.S.E. spaces (C.S.E. experiment), C.S.E. and Engineering Education Epistemology (EEE). Using a unified label we call our model of teaching "Computational STEM Pedagogy" (**Figure 3**) (Psycharis, 2018).

Game Based Learning- STEM and Computational Thinking

Brown, Collins and Duguid (1989) developed the theory of situated cognition or situated learning and produced a proposal for a model of instruction that has implications for classroom practice. They stated that a situated understanding of a word or concept supports a deeper comprehension and the ability to use this knowledge in ways that are customizable to several situations of use. Collins (1988) defines situated learning as: 'the notion of learning knowledge and skills in contexts that reflect the way the knowledge will be useful in real life'.

A fundamental issue of the situated learning model is the notion of the apprentice observing the 'community of practice'. According to Lave and Wenger (1991), participation in a culture of practice can, in the first instance, be observation from the boundary or "legitimate peripheral" participation (Herrington & Oliver, 2000). According to Lo et al. (2008, pp. 51), "Digital game-based situated learning integrates concepts of situated learning into digital game -based learning so that the learning process can take place unintentionally".

Research stress the aspect that that technology is difficult to learn in an abstract way and practical activities are needed in order to apply scientific concepts (Berta et al 2017). STEM topics "typically involve facts and concepts that could be effectively implemented and/or shown through smart objects according to the Internet-of-Things paradigm. Such objects, called "iBlocks," could be manipulated by young learners to study various types of phenomena/artifacts and compose new aggregations" (Berta et al., 2017).

In the current research we built an environment consisting of physical objects enhanced with computing operations, and communicating capabilities in order to engage students in multimodal, in alignment with the current research on the relation between STEM and serious games.

Game based learning is attractive for students, causing a kind of commitment and is considered as a proper teaching and learning approach for students' engagement in CT skills (Fogli et al., 2017).

STEM and Self-confidence

Self-confidence is considered as a self-awareness of capability (Weinberg, 2009), and a basic component of the inquiry teaching and learning process (Greenwald, 2010). Self-confidence is related to logical reasoning and argumentation in inquiry based teaching and learning approach, and this leads to a first indication of its relation to the dimensions of CT (Carin, Bass, & Contant, 2005). Other studies suggest that self-confidence is a proper predictor of achievement and is related to both cognitive and self-belief measures (Stankov et al., 2012)

Some other studies also suggest that increasing the self-confidence of students in the use of Computers, will enhance their capacity to be engaged in the solution of complex phenomena (Aesaert & vanBraak, 2014). According to Cretchley (2007), the high level of self-confidence had a strong impact on students' motives for learning in computer-based learning environments.

METHODOLOGY

Research Questions

This article proposes a pedagogical frame for the design of "STEM inquiry game learning" scenarios for primary school students, without previous computer experience and it adopts the Computational Pedagogy approach, as it was previously described.

This study aimed to investigate the self-confidence of primary school students in computer use and their performance in CT skills, and research was carried using a quantitative survey.

The sample of the study consisted of 115 children aged 11 and 12 years old, selected using the sampling method. It is presentedhe main research questions that we have investigated. .

What is the impact of an inquiry based-STEM Content epistemology-Computational Pedagogy based didactic scenario, on students'

1. Improvement in CT dimensions, when computational tools like (Easy Java Simulations-EJS-, Arduino, Scratch for Arduino-S4A) are used in the teaching approach?
2. Self-confidence in the computers' use?

To answer the above research questions, a correlational, pretest–posttest research design was implemented through the use of questionnaires. Each variable was measured at the beginning and at the end of the intervention. Data were collected from the questionnaires given to students before and after the intervention. Intervention was lasted for seven weeks and the whole research lasted 4 months.

Participants

In this study, we report on findings (115 students) from three K5 classes (58 students) and three K6 classes (57 students) from two public schools in Attica, Greece, during the Academic year 2017-18. 62 (54%) students were male and 53 (46%) female.

Questionnaires

CT questionnaire

To evaluate students' "performance" in CT, the Bebras questionnaire was used (UK Bebras; Dagiene & Stupuriene, 2016). Sixteen (16) questions were used from the UK Bebras competition of 2016, and one question was selected from the 2015 competition, since it was relevant to the RGB color model. Taking into account that students had no previous experience in CT questions, we selected questions from the Senior level with difficulty level A-B, and questions for the Intermediate level with difficulty level A-B-C. After discussions of the researchers with the teachers and their suggestions about the appropriateness of the questions, we ended to 12 questions, selected according to: a) teachers' views for the appropriateness of the questions, in relevance to students' capacity, b) the decision to include four (4) questions for every level of difficulty and c) to include the five (5) dimensions of CT as discussed in the introduction.

The Questionnaire was adapted into Greek language by English teachers with Authors' cooperation. Pilot research was implemented by providing the questionnaire to two students out of the sample and their remarks were taken into account mainly in changing the wording of some questions.

Self-confidence in "computer use"

The students' self-confidence in "Computer use" was based on the instrument developed by Fogarty et al. (2001). Questionnaire has Cronbach alpha internal consistency reliability .92. All items employed a Likert-style response format, with options ranging from 1 (Strongly agree), 2 (Agree), 3 (Neutral), 4 (Disagree), to 5 (Strongly disagree). Questionnaire used a mixture of positively worded items (4 questions) and negatively worded items (8 questions). Negative orientation was selected thus reflecting the primary concern of most educators, which is the possible handicapping effect of negative attitudes towards computers.

Factor analysis and Cronbach alpha were carried out in order to be adjusted to Greek culture and the age of the students. Students' responses follow the normal distribution, (N=115). To codify the answers in SPSS 23.0, we reversed the meaning of the 4 "positive oriented" and the reliability alpha was a=0.770 (for the pre-test) and a=0.795 (for the post-test).

Factor analysis (Extraction Method: Principal Component Analysis) was carried out and three factors (eigenvalues) were produced. The most significant of them contained nine (9) questions with close contributions (Q.2=0,523· Q.3=0,645· Q4=0,611· Q5=0,715· Q6=0,604· Q7=0,591· Q10=0.602· Q11=0,658·Q12=0,513), and the other two considered as no significant. An index of self- confidence was attributed to each student, before and after the intervention.

The Intervention

We designed a STEM content inquiry game learning scenario where teaching is implemented through a drama educational game and students are engaged in a rescue game. The different phases of the scenario follow, as well as a short description of the computational tools used, are presented. An alternative and more detailed description of the game is presented in Psycharis & Kotzampasaki (2017).

Phase 1: Motivation phase. The problem of how to design a rescue process of a yellow box is presented, as well as the restrictions and the rules of the game (for example students are informed that to rescue the box form glass building, they cannot bring with them any digital detector). All students decided that they should be separated in two sub-teams and the rules were known and agreed to all of them before the game. Sub-team B should emit different colors-light signals that will be used in order to guide sub-team A of the glass building and inform them about the "coding" (**Figure 4**). (CT dimensions involved: AB-DE-EV).

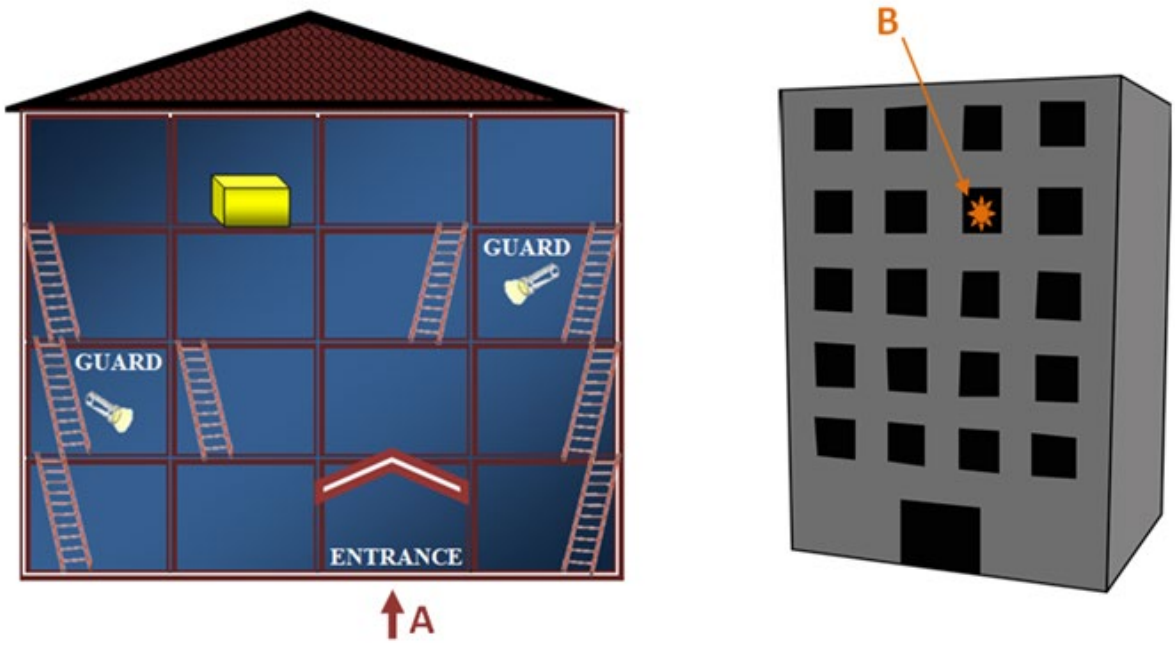

Figure 4. Buildings used in the "rescue scenario"

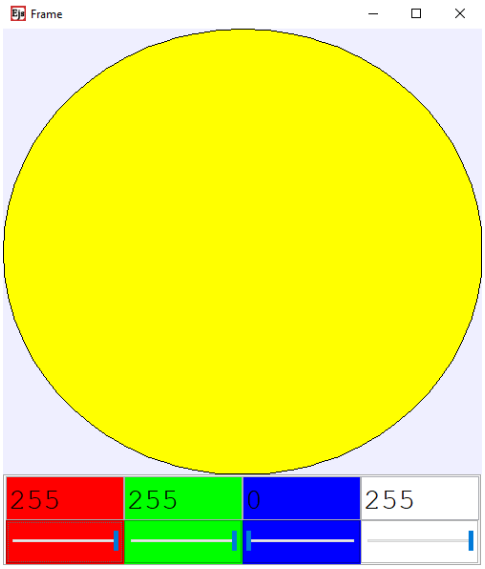

Figure 5. Creating colors in EJS according to RGB color model

Phase 2: The rescue design-Simulation of the model. Students of both teams "create colors" using the "RGB" developed in EJS software, and they also agree for the representation code of the color. For example, the yellow light corresponds to "move to the right". The yellow colour is created in EJS software by setting the prices Red=255, Green=255, Blue=0. (CT dimensions involved: AB-EV-AL).

Phase 3: Development of artefact in Arduino and algorithm in S4A: Students construct the "circuit" using the Arduino platform and the RGB-LED. They create the algorithm in S4A, so they can handle the different colors of the LED, according to the code created at Phase 2 (CT dimensions involved: GE-EV-DE-AL).

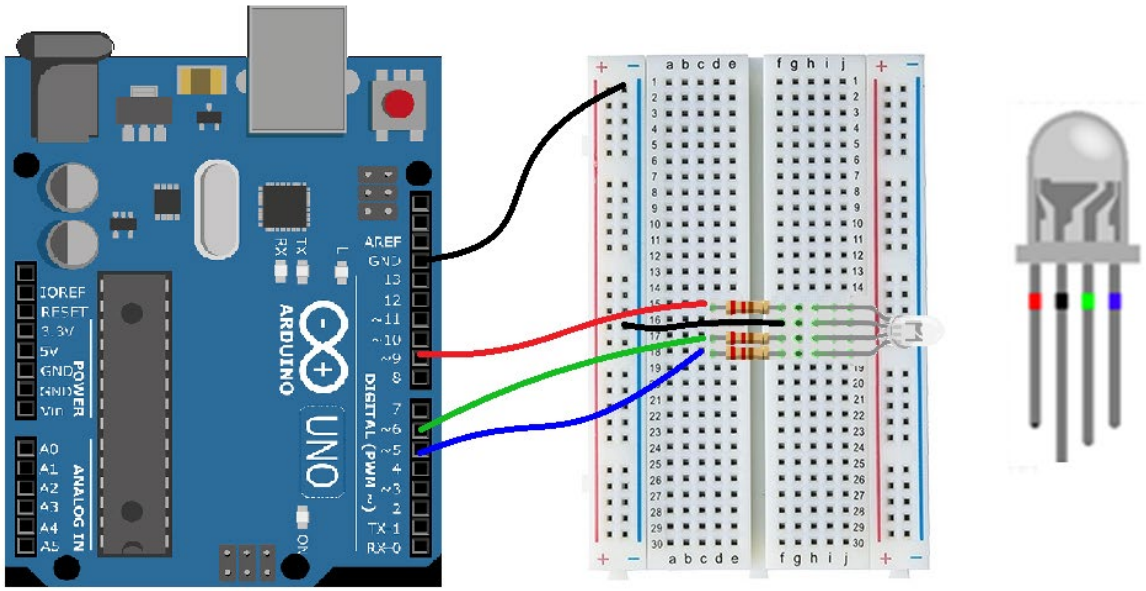

Figure 6. Arduino circuit and RGB led

Phase 4: The rescue Algorithm: Students are provided with the positions of the guards for every phase. In the digital tableau, students move the guards and they register the commands for the movement in every direction and they make correspondences between the directions and the colors. Process is registered in a work-sheet that will be used in Phase 5. (AL-AB-EV-GE).

Phase 5: Coding in S4A: Students' answers for the Phase 4 are transferred as code in S4A and the RGB LED signals different colors depending on the commands for the directions of motion. (AL-GE).

Phase 6: Connection with Technology and Biology: Connection of the RGB LED with the operation of the pixel using a simulation in PixelViewer. In Addition, using a video, students realized that the operation of the eye resembles that of the operation of the pixel, as they both use the RGB color model (GE).

During the intervention, teachers acted as critical friends. They also attended the intervention and answered the questions given to students.

DATA COLLECTION AND ANALYSIS OF DATA

Computational Thinking Analysis

Computational thinking tests results

In **Table 1** the responses before and after the intervention are presented. A χ^2 statistical test was used in SPSS, but the requirements for χ^2 were not fulfilled (Expected frequencies was not at least 5 for the majority of the cells in 3x3 tables), and thus χ^2 was rejected. To apply Mc Nemar test for independence, we merged the responses "no answer" and "wrong answer". Then, the variables were re-coded to make paired nominal data. The conditions of the test are met because of the prices that both variables can get are two (Correct answer; Yes or No) in a 2x2 contingency table. Mc Nemar proportion test was applied to each question to find out if the answers are independent before and after the intervention.

Table 1. Answers in Computational Thinking Test

Computational Thinking Test		No answer	Correct	False
		N %	N %	N %
Q1:Beaver Code (A) AL, DE, GE	**Pre**	23.70%	53.50%	22.80%
Students connect symbols with words, according to the symbols already provided form a Table	**Post**	6.10%	87.80%	6.10%
	McNemar: p_value=0.000			
Q2: Blossom (B) EV, GE	**Pre**	40.40%	16.70%	43.00%
Students find the colors of flowers and put them in the correct order, by using the trial and error method and the logical reasoning.	**Post**	18.30%	58.30%	23.50%
	McNemar: p_value=0.000			
Q3: Theater 2015 (B) AB, DE, GE	**Pre**	42.10%	3.50%	54.40%
.Students find colors according to the mixing code of RGB	**Post**	40.00%	20.90%	39.10%
	McNemar: p_value=0.000			
Q4:Party Banner (A) AB, EV,GE	**Pre**	51.80%	7.90%	40.40%
Students track the missing rectangles form a color belt and they recognize the pattern	**Post**	40.00%	20.90%	39.10%
	McNemar: p_value=0.001			
Q5: Party Guests (A) AL,DE	**Pre**	27.20%	14%	58.80%
Students determine the correct order in a telephone communication with five friends, according to specific restrictions and rules.	**Post**	23.50%	36.50%	40%
	McNemar: p_value=0.000			
Q6:ConcurrentDirections (B) AL, DE	**Pre**	42.10%	26.30%	31.60%
Students select the correct path for simultaneous directions of three robots, with specific outcomes at the end of the path followed.	**Post**	27%	34.80%	38.30%
	McNemar: p_value=0.099			
Q7: Primary Health Care (B) EV, AB	**Pre**	19.30%	2.60%	78.10%
.Students select from a map three points I order to propose the construction of a hospital, so all people could arrive at the hospital with one movement	**Post**	13.00%	31.30%	55.70%
	McNemar: p_value=0.000			
Q8:RobotExit (A) AL	**Pre**	40.40%	13.20%	46.50%
Students design an algorithm so robots can find the exit. They design the arrows to represent the motion and they repeat the commands(loop control)	**Post**	9.60%	63.50%	27.00%
	McNemar: p_value=0.000			
Q9: Rafting (C) EV, AL	**Pre**	49.10%	5.30%	45.60%
Students find two wrong signals by studying a specific graph.	**Post**	43.50%	28.70%	27.80%
	McNemar: p_value=0.000			
Q10: Secret Messages (C) EV, GE	**Pre**	51.80%	25.40%	22.80%
Students are provided with an example about a cipher and they have to decipher another message,	**Post**	57.40%	28.70%	13.90%
	McNemar: p_value=0.585			
Q11: Cave game (C) AL	**Pre**	49.10%	7.00%	43.90%
Students are provided with a map with caves, where a present is hidden. A player asks for the position of the present and the other player responds either that he is correct or provides the direction that he should follow. Students are asked to find the optimal number of trials.	**Post**	60.00%	7.00%	33.00%
	McNemar: p_value=1			
Q12: Segwey (C) AB, AL	**Pre**	50.90%	8.80%	40.40%
Students are provided with a graph with buttons in a game and they are asked to find the final direction.	**Post**	66.10%	10.40%	23.50%
	McNemar: p_value=0.774			

Mc Nemar N=114, binomial distribution (A)= Easy, (B)= Medium, (C)= Difficult
AL = Algorithmical Thinking, AB=Abstraction, DE=Decomposition, GE= Generalization, EV=Evaluation
http://www.bebras.uk/uploads/2/1/8/6/21861082/uk-bebras-2016-answers.pdf

According to the results from **Table 1**, an increase in the percentage of the correct answers is easily observed in eleven (11) out of the twelve (12) questions. For eight (8) questions, (1,2,3,4,5,7,8,9), using the Mc Nemar Test, we reject the Ho hypothesis (p_value<0.05), and we conclude that a greater percentage of students are expected to respond correctly to answers after the intervention.

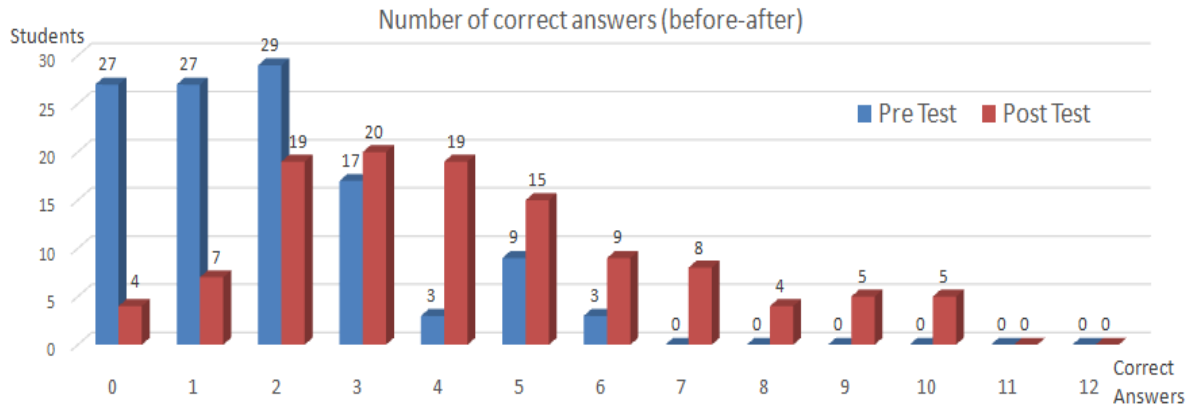

Figure 7. Number of correct answers before and after the intervention

Table 2. Calculate score by difficulty level according to the Bebras competition standards

Difficulty level	False points	No answer points	Wright points	Questions	Start points	Max score
A	0	0	6	1-4-5-8	0	24
B	-2	0	9	2-3-7-8	8	44
C	-4	0	12	9-10-11-12	16	64
		Total test			24	**132**

Table 3. Difficulty Level analysis

Difficulty Level	M.O. Pre Test	M.O. Post Test	percentage increase	Pearson p_value (N=115)
A	5,27	12,52	**237%**	r=0.499, p_value=0.000
B	8,28	17,94	**217%**	r=0.430, p_value=0.000
C	15,48	21,04	135%	r=0.148, p_value=0.116

The mean value of the number of correct answers was increased from M=1,83 (Std=1.595) (before the intervention), to M=4,29 (after the intervention) with standard deviation Std=1,975. We also observed –using a paired sample t-test that the mean values of the correct answers has a statistically significant difference (p_value=0.000).

Grades were also calculated according to the standard Bebras competition method (calculated for the 12 answers). The level of difficulty was counted by attributing negative and positive scale according to the description provided in **Table 2**, so the lower grade was set to zero (**Table 2**).

The mean value calculated in alignment to the standard Bebras method, changed from (M=29,07, Std=14,96) to (M=51,50, Std=23,89) , p=0.000 and we conclude that the intervention had a positive impact on students' CT skills.

Computational thinking results for the difficulty levels

The grades of every student were calculated in alignment with the difficulty level as it is predefined in Bebras completion. Results presented in **Table 3**, show that the intervention had statistical significant impact for the difficulty levels A, B.

Computational thinking results per CT dimension

In order to measure the "performance" of every student and for every dimension of CT, we added the grades for each student, taking into account the level of difficulty according to the Bebras standards as described in **Tables 1** and **3**.

From the results presented in **Table 4**, we notice that there is a statistically significant increase in CT performance for every CT dimension.

Table 4. Statistical analysis per CT dimensions – Paired Sample T-test

.	Pearson r	p_value	St. Deviation	St.Error Mean	MeanPre	Mean Post	Mean Increase
Algorithm	0.323	0,000	12.010	1.120	2.16	13.43	**622%**
Decomposition	0.511	0,000	6.015	0.561	3.23	8.56	**265%**
Abstraction	0.330	0,000	9.377	0.874	0.87	5.03	**578%**
Evaluation	0.476	0,000	9.603	0.895	6.41	14.18	**221%**
Generalization	0.536	0,000	14.217	1.326	2.45	15.31	**624%**

Table 5. Descriptive analysis of computer confidence questions

		1	2	3	4	5
1. I have more trouble learning how to use a computer than I do learning other things. (Reversed)	Pre	7.0%	14.8%	27.0%	28.7%	21.7%
	Post	0.0%	5.2%	27.0%	32.2%	35.7%
2. When I have difficulties using a computer I don't know I can handle them. (Reversed)	Pre	11.3%	15.7%	24.3%	33.0%	15.7%
	Post	0.0%	4.3%	12.2%	37.4%	46.1%
3. I am not what I would call a computer person.	Pre	20.0%	20.9%	24.3%	18.3%	16.5%
	Post	3.5%	6.1%	23.5%	32.2%	34.8%
4. It takes me much longer to understand how to use computers than the average person.	Pre	12.2%	15.7%	16.5%	27.8%	27.8%
	Post	0.0%	1.7%	1.7%	16.5%	80.0%
5. I have never felt myself able to learn how to use computers.	Pre	2.6%	10.4%	7.8%	27.0%	51.3%
	Post	1.7%	1.7%	7.8%	12.2%	76.5%
6. I don't enjoy trying new things on a computer. (Reversed)	Pre	4.3%	3.5%	8.7%	27.0%	56.5%
	Post	0.0%	1.7%	1.7%	16.5%	80.0%
7. I find having to use computers frightening.	Pre	2.6%	6.1%	16.5%	20.0%	54.8%
	Post	0.9%	0.9%	3.5%	18.3%	76.5%
8. I don't find many aspects of using computers interesting and challenging. (Reversed)	Pre	3.5%	3.5%	16.5%	28.7%	47.8%
	Post	0.0%	0.9%	11.3%	30.4%	57.4%
9. I don't understand how some people can seem to enjoy spending so much time using computers	Pre	20.9%	14.8%	17.4%	15.7%	31.3%
	Post	10.4%	10.4%	20.0%	21.7%	37.4%
10. I have never been very excited about using computers.	Pre	9.6%	13.0%	14.8%	23.5%	39.1%
	Post	0.9%	430.0%	5.2%	29.6%	60.0%
11. I find using computers confusing.	Pre	7.0%	8.7%	14.8%	41.7%	27.8%
	Post	0.9%	6.1%	16.5%	33.0%	43.5%
12. I'm nervous that I'm not good enough with computers to be able to use them to learn mathematics or science	Pre	6.1%	7.0%	25.2%	27.0%	34.8%
	Post	2.6%	1.7%	10.4%	27.0%	57.4%

1 = Strongly agree; 2 = Agree; 3 = Neutral; 4 = Disagree; 5 = Strongly disagree, N=115

Table 6. Descriptive measures for student's computer confidence indicator

	Computer confidence indicator	
	Pre Test	**Post Test**
Mean	3.7144	4.3685
Median	3.7800	4.5600
Std. Deviation	.71590	.52989
Std. ErrorMean	06676	.04941
Minimum	1.78	2.56
Maximum	5.00	5.00

The percentage increase is very impressive, especially for the "Generalization Dimension" (624%), the "Algorithmic Dimension" (622%), and the "Abstraction Dimension" (578%).

Self-confidence Results in Computer Use

From **Table 5**, it is evident that students have fewer problems in computers use and they feel that they can handle out problems related to their use. They also feel that they have the capacity to learn using computers without time as a prohibitive factor. They also enjoy trying new things on a computer and they consider that computers are an essential tool for learning Science and Mathematics. In **Table 5**, results for the confidence for computers' use are presented.

Applying-test for the mean values, the results are presented in **Table 6**. We mention here that the "index of self-confidence" for each student was calculated according to the factors analysis described before. From the data, it is evident that there is a statistically significant increase (p=0.000).

Table 7. Researches Comparison

TestMax=100	Current research N=114			Djambong & Freiman N=10		
	PreTest	Post Test	**Improvement**	PreTest	Post Test	**Improvement**
A	22	52	**30**	40	50	**10**
B	19	41	**22**	25	29	**4**
C	24	33	**9**	27	32	**5**
TOTAL	22	39	**17**	30	33	**3**

DISCUSSION

The research was guided by the research literature that suggests that we should find ways to implement CT in primary school education, especially for students without previous use of computers (Angeli et al, 2016; Guzdial, 2008). For this reason we proposed open computational tools (Arduino, S4A) and a didactic model that uses the Inquiry based teaching and learning model, embedded with a game based learning approach and issues of STEM content epistemology, such as the development of the color code, the design of the artefact and the connection with problems of real life.

Results analysis allows us to support that the presented above intervention resulted in a significant improvement on students' CT skills, with statistically significant results in: a) the number of correct answers b) the CT score c) score in A and B level of difficulty d) score in every investigated CT dimension / skill (scores calculated in alignment with Bebras standards). The more significant result is an increase in students' CT "grades", especially in the "Generalization", "Algorithmic Thinking" and "The abstraction" dimensions.

Table 7 presents the compartment data of the results of this research about CT with those of Djambong and Freiman (2016) for the same age group. They also used selected questions from the Bebras competition before and after five (5) weeks of intervention, using the Lego EV3 Mindstorms. In the following results, all grades are calculated with maximum score of 100 for the three level of difficulty. A comparison of each CT dimension could not be conducted due to the lack of relevant data by Djamong and Freiman (2016).

The results indicate that students who participated in the current research started with lower grades but they had greater grades after the intervention. One possible explanation for the difference in results can be the longer period of intervention, which is also recognized in the research of Djambong & Freiman (2016). Another possible explanation could be related to the didactic model adopted and the use of the specific computational tools, which are based on the Computational Science Education, through the implement of model construction and the development of simulation using open software, such as the Arduino platform. This can be justified by the view expressed by Zieris et al. (2015), who clearly stated that CT skills cannot be developed through the ICT tools that operate as a "black box", and they recommend the use of Arduino instead of the Lego Mindstorms. The Arduino is shown to be a very effective tool for students' ability to develop CT skills like algorithmic design, which is in accordance to the review of literature (e.g. Cheng et al, 2016). Based on the current results, it could be concluded, that Arduino with S4A is a proper computational tool for K5-K6 students, who haven't acquired previous experience in the use of computers.

This article also supports the findings of Goh et al. (2013), for the added value of simulations in the teaching sequence of primary school students. In our research, students did not use the simulation as a "craft", but they developed the code according to the RGB color model instead. Furthermore, the results verified those of Fogli et al (2017) regarding the effectiveness of a serious game in learning, as students feel committed to reaching a target, therefore learning seems to become more appealing.

The initial level of students' self-confidence in computers' use, shows a significant increase after the intervention. Apparently, this result could be attributed to various educational choices that have been made but clearly seem to be in agreement with the conclusions of Kalelioglu and Gülbahar (2014), that optical programming contributes to students' self-confidence and their capacity to solve problems.

Further research is under progress regarding the relation of CT improvement and self-confidence. While a lot of research focuses on the definition of CT, there is little research regarding its impact on students' self-confidence for computers' use for solving complex problems (Aesaert, & vanBraak, 2014).

Research limitations include the fact that the students' sample was taken from the capital of Greece, as well as the fact that students had no previous experience in computer's use. The total duration of the intervention was four months which could be considered adequate but an extend of our research interval in order to investigate among others whether students could handle more difficult questions would be rather interesting.

In this article we attempted to contribute to the research for the effectiveness of the implementation of an Inquiry based sequence that implements CT in primary school education when specific computational tools and the Computational Science Education methodology are applied.

An additional contribution of this paper is related to the systematic evaluation method that is presented due to the fact it provides the ability of counting the main dimensions of Computational Thinking. Taking under account the different dimensions, which are necessary to solve the problems of the Bebras competition, and combining the differential level of questions, we managed to export quantitative data for each dimension of CT. Through this methodology and by comparing data before and after the intervention we were able to measure the improvement of students that is related to CT dimensions. Furthermore, due to the above the impact of different interventions can be evaluated. In addition, the adaptation process of the questionnaires into the greek language was effectively conducted thus it can be used for future research. Finally, our research could contribute to the education policy in Greece for the development of new curricula for the primary school education, as CT is considered as a very essential skill for the 21st century.

A set of rigorous attempts are currently being designed to provide a systematic evaluation of the pedagogical framework and the game as well. These experiments will provide adequate data so to determine whether our pedagogical framework is successful or not in encouraging the development of CT skills and whether the game helps students to develop both of aspects analyzed separately and in combination, in order to accurately determine the impact and its possible benefits regarding our approach. The statistical data generated will also be made available to the research community, so to provide a further knowledge on the use of physical computing to CT skills.

CONCLUSION

To summarize, results analysis allows us to support that the implementation of a STEM content Inquiry based scenario using computational tools (Arduino, S4A, simulations) and educational games according to Computational Science Education methodology· has positive influence on the dimensions of Computational Thinking and self-confidence for "computers use" in 5K-6K students even without previous knowledge in the use of computers.

The additional contribution of the research is that it provides a systematic evaluation method with quantitative data in order to determine whether a pedagogical framework is successful or not in encouraging the improvement for every CT dimension and for every different level of difficulty.

REFERENCES

Aesaert, K., & vanBraak, J. (2014). Exploring factors related to primary school pupils' ICT self-efficacy: A multilevel approach. *Computers in Human Behavior*, *41*, 327-341. https://doi.org/10.1016/j.chb.2014.10.006

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.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & 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. https://doi.org/10.1145/1929887.1929905

Berland, L. K., Martin, T. H., Ko, P., Peacock, S. B., Rudolph, J. J. & Golubski, C. (2013). Student learning in challenge-based engineering curricula. Journal of Pre-College Engineering Education Research *(JPEER),* 3(1), 5. https://doi.org/10.7771/2157-9288.1080

Berta R., Bellotti F., van der Spek E., Winkler T. (2017) A Tangible Serious Game Approach to Science, Technology, Engineering, and Mathematics (STEM) Education. In: Nakatsu R., Rauterberg M., Ciancarini P. (eds) *Handbook of Digital Games and Entertainment Technologies*. Springer, Singapore. https://doi.org/10.1007/978-981-4560-50-4_32

Beth, E. W., & Piaget, J. (1966). *Mathematical epistemology and psychology*. Dordrecht, The Netherlands: Reidel.

Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the Computational Thinking Pedagogical Capabilities of School Teachers. *Australian Journal of Teacher Education*, *42*(3). https://doi.org/10.14221/ajte.2017v42n3.4

Brown, J.S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32-42. https://doi.org/10.3102/0013189X018001032

Bundy, A. (2007). Computational Thinking is Pervasive. *Journal of Scientific and Practical Computing*, *1*(2), 67- 69.

Carin, A. A., Bass, J. E., & Contant, T. L. (2005). *Methods for teaching science as inquiry* (9th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, *47*, 70-80. https://doi.org/10.1016/j.jmathb.2017.06.004

Cheng, H., Hao, L., Luo, Z., &Wang, F. (2016). Establishing the Connection between Control Theory Education and Application: An Arduino Based Rapid Control Prototyping Approach. *International Journal of Learning and Teaching*, *2*(1). https://doi.org/10.18178/ijlt.2.1.67-72

Collins, A. (1988). Cognitive apprenticeship and instructional technology (Technical Report 6899): BBN Labs Inc., Cambridge, MA.

Cretchley, P. (2007). Does computer confidence relate to levels of achievement in ICT-enriched learning models?. *Education and Information Technologies*, *12*(1), 29-39. https://doi.org/10.1007/s10639-006-9004-6

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. https://doi.org/10.15388/infedu.2016.02

Denning, P. J. 2007. Computing is a natural science. *Commun. ACM*, 50, 13-18. https://doi.org/10.1145/1272516.1272529

Djambong, T., & Freiman, V. (2016). Task-Based Assessment of Students' Computational Thinking Skills Developed through Visual Programming or Tangible Coding Environments. *International Association for Development of the Information Society*.

Einhorn, S. (2012). Microworlds, computational thinking, and 21st century learning. LCSI White Paper. Retrieved from http://www.microworlds.com

Fogarty, G., Cretchley, P., Harman, C., Ellerton, N., & Konki, N. (2001). Validation of a questionnaire to measure mathematics confidence, computer confidence, and attitudes towards the use of technology for learning mathematics. *Mathematics Education Research Journal*, *13*(2), 154-160. https://doi.org/10.1007/BF03217104

Goh, K. S. A., Wee, L. K., Yip, K. W., Toh, P. Y. J., & Lye, S. Y. (2013). Addressing learning difficulties in Newtons 1st and 3rd Laws through problem based inquiry using Easy Java Simulation. *arXiv preprint arXiv:1303.0081*.

Gravemeijer, K., & Doorman, M. (1999). Context problems in realistic mathematics education: A calculus course as an example. *Educational Studies in Mathematics, 39*(1), 111–129. https://doi.org/10.1023/A:1003749919816

Greenwald, J. M. (2010). Antecedents of core confidence latent construct: Direct and reciprocal links. *Dissertation Abstracts International Section A, 72,* 270.

Guzdial, M. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25–27, https://doi.org/10.1145/1378704.1378713

Herrington, J., & Oliver, R. (2000). An instructional design framework for authentic learning environments. *Educational technology research and development*, *48*(3), 23-48. https://doi.org/10.1007/BF02319856

Hu, C. (2011, June). Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 223-227). ACM. https://doi.org/10.1145/1999747.1999811

Juszczak, M. D. (2015). From Towards a Computational Pedagogy – Analysis of ABM Deployment in Pedagogical Instances. *International Journal of Pedagogy Innovation and New Technologies*, *2*(1), 2-13. https://doi.org/10.5604/23920092.1159113

Kalelioglu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education*, *13*(1), 33.

Katehi, L., Pearson G., & Feder M. (2009). Engineering in K-12 education: Understanding the status and improving the prospects. Washington, DC: National Academy of Engineering and National Research Council.

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. https://doi.org/10.1016/j.procs.2012.04.056

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*(2), 154- 171. https://doi.org/10.1007/s40751-017-0031-2

Landau, RH., Páez, J. & Bordeianu, C. (2008). A Survey of Computational Physics: Introductory Computational Science. Princeton and Oxford: Princeton University Press

Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511815355

L'Heureux, J., Boisvert, D., Cohen, R. & Sanghera, K. 2012. IT problem solving: an implementation of computational thinking in information technology. Proceedings of the *13th annual conference on Information technology education*. Calgary, Alberta, Canada: ACM. https://doi.org/10.1145/2380552.2380606

Lo, J. J., Ji, N. W., Syu, Y. H., You, W. J., Chen, Y. T. (2008) Developing a Digital Game-Based Situated Learning System for Ocean Ecology. In: Pan Z., Cheok A.D., Müller W., El Rhalibi A. (eds) Transactions on Edutainment I. Lecture Notes in Computer Science, vol 5080. Springer, Berlin, Heidelberg, pp 51-61. https://doi.org/10.1007/978-3-540-69744-2_5

Martinez, S. L., & Stager, G. (2013). Invent to learn. Retrieved on May 2018 from http://courseshare.com/pdfs/Education_3-0__Chula_Bonk.pdf

Moore, T. J. (2008). STEM integration: Crossing disciplinary borders to promote learning and engagement. Invited presentation to the faculty and graduate students of the UTeachEngineering, UTeachNatural Sciences, and STEM Education program area at University of Texas at Austin, December 15, 2008.

National Academy of Engineering (NAE) & National Research Council (NRC). (2014). STEM integration in K-12 education: status, prospects, and an agenda for research. The National Academies Press, Washington NGSS, 2013.

National Research Council (2010) Report of a Workshop on the Scope and Nature of Computational Thinking. Retrieved from http://www.nap.edu/catalog.php?record_id=12840

National Research Council (2011). National Research Council Report of a Workshop of Pedagogical Aspects of Computational Thinking. Retrieved on 10-20-2011 from http://www.nap.edu/catalog.php?record_id=13170

National Research Council. (2012a) A framework for K-12 science education: practices, crosscutting concepts, and core ideas. National Academies Press, Washington, DC

National Research Council. (2012b) Discipline-based education research: understanding and improving learning in undergraduate science and engineering. National Academies Press, Washington, DC.

Nicolescu, B. (2004). Gurdjieff's philosophy of nature. In J. Needleman & G. Baker (Eds.), Gurdjieff (pp. 37- 69). New York, NY: The Continuum International Publishing Group.

Nowotny, H. (2003). Democratising expertise and socially robust knowledge. *Science and public policy, 30*(3), 151-156. https://doi.org/10.3152/147154303781780461

Przybylla, M., & Romeike, R. (2014). Physical computing and its scope-towards a constructionist computer science curriculum with physical computing. *Informatics in Education*, *13*(2), 225. https://doi.org/10.15388/infedu.2014.05

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 (JOST), 25*(2), 316-326. https://doi.org/10.1007/s10956-015-9595-z

Psycharis, S. (2016). Inquiry Based- Computational Experiment, Acquisition of Threshold Concepts and Argumentation in Science and Mathematics Education. *Educational Technology & Society, 19*(3), 282-293.

Psycharis, S. (2018). STEAM In Education: Literature review on the role of computational thinking, engineering epistemology and computational science. Computational STEAM Pedagogy (CSP). *Scientific Culture, 4*(2), 51-72. https://doi.org/10.5281/zenodo.1214565

Psycharis, S., Kalovrektis, K., Sakelalridi, E., Korres, K., & Mastorodimos, D. (2017). Unfolding the Curriculum: Physical Computing, Computational Thinking and Computational Experiment in STEM's Transdisciplinary Approach. *European Journal of Engineering Research and Science (EJERS)*. https://doi.org/10.24018/ejers.2018.0

Psycharis, S., & Kotzampasaki, E. (2017). A didactic Scenario for Implementation of Computational Thinking using Inquiry Game Learning. In *Proceedings of the 2017 International Conference on Education and E-Learning*, Bangkok Thailand 2-4 November 2017 (pp. 26-29). ACM. https://doi.org/10.1145/3160908.3160918

Qiu, K., Buechley, L., Baafi, E., & Dubow, W. (2013, June). A curriculum for teaching computer science through computational textiles. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 20-27). ACM. https://doi.org/10.1145/2485760.2485787

Rubio, M. A., Hierro, C. M., & Pablo, A. P. D. M. (2013, July). Using arduino to enhance computer programming courses in science and engineering. In *Proceedings of EDULEARN13 conference* (pp. 5127-5133).

Rugarcia, A., Felder, R. M., Woods, D. R. & Stice, J. E. (2000). The future of engineering education: I. A vision for a new century. *Chemical Engineering Education, 34*, 16–25.

Schulz, S., & Pinkwart, N. (2015, November). Physical computing in stem education. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 134-135). ACM. https://doi.org/10.1145/2818314.2818327

Selby, C., & Woollard, J. (2014). Refining an understanding of computational thinking. *Author's Original*, 1-23.

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. https://doi.org/10.1007/s10639-012-9240-x

Sentance, S., & Csizmadia, A. (2015). Teachers' perspectives on successful strategies for teaching Computing in school. Paper presented at IFIP TCS, 2015. http://community.computingatschool.org.uk/files/6769/original.pdf

Shirey, K. (2017). Teacher Productive Resources for Engineering Design Integration in High School Physics Instruction (Fundamental). In Proceedings of the 2017 ASEE Annual Conference, Columbus, OH, June 2017. https://doi.org/10.18260/1-2--28908

Stankov, L., Lee, J., Luo, W., & Hogan, D. J. (2012). Confidence: A better predictor of academic achievement than self-efficacy, self-concept and anxiety?. *Learning and Individual Differences*, *22*(6), 747-758. https://doi.org/10.1016/j.lindif.2012.05.013

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. https://doi.org/10.1007/s10639-015-9412-6

Weinberg, B. A. (2009). A model of over-confidence. *Pacific Economic Review, 14,* 502–515. https://doi.org/10.1111/j.1468-0106.2009.00466.x

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., &Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technol-ogy*, *25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M .(2006). Computational thinking. *Communications of the ACM, 49*, 33-35. https://doi.org/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. https://doi.org/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 (TOCE)*, *14*(1), 5. https://doi.org/10.1145/2576872

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., &Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 465-470). ACM. https://doi.org/10.1145/1953163.1953297

Yasar O., Veronesi P., Maliekal J., Little L. J., Vattana S. E., & Yeter I. H. (2016). Computational Pedagogy: Fostering a New Method of Teaching. Presented at: ASEE Annual Conference and Exposition. Presented: June 2016. Project: SCOLLARCIT.

Yaşar, O. (2004). Computational math, science and technology: A new pedagogical approach to math and science education. In: Lagana, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds) ICCSA 2004. *LNCS, 3045*, 807-816. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24767-8_85

Yaşar, O. (2013). Teaching Science through Computation. *International Journal of Science, Technology and Society, 1*(1), 9-18. https://doi.org/10.11648/j.ijsts.20130101.12

Yaşar, O., & Landau, R (2003): Elements of CSE Education. *SIAM Review, 45*(4), 787–805. 72.

Zieris, H., Gerstberger, H., & Müller, W. (2015). Using Arduino-Based Experiments to Integrate Computer Science Education and Natural Science (Eds.). *KEYCIT 2014: key competencies in informatics and ICT* (Vol. 7). Universitätsverlag Potsdam.