Information organization via computational thinking: case study in a primary school classroom*

Simone A. C. Cavalheiro[1], Ana M. Pernas[1], Marilton S. de Aguiar[1], Luciana Foss[1], André Du Bois[1], Renata H. S. Reiser[1], Clause de Brum Piana[1], Yuri Weisshahn[1]

[1]Technology Development Center - Federal University of Pelotas (UFPEL), 1, Gomes Carneiro St - 96.010-610 - Pelotas, RS, Brazil

Abstract. *Technological innovations have changed the profile of professionals required by economy sectors demanding skills related to problem solving and logical reasoning. Concurrently with this reality, Computational Thinking emerges as a methodology for solving problems, which is able to develop many of these skills. This article reports the experience in applying the activities referred as Binary Numbers, Color by Numbers, Sorting Algorithms and Battleships, which aims at developing some of necessary skills for the practice of Computational Thinking by fourth-graders. The article resumes the activities, and evaluates the obtained results.*

1. Introduction

Computational Thinking (CT) is the idea of using basic concepts of Computer Science (CS) to solve problems, develop systems and to understand human behavior. In a general way, abilities once developed and used to design programs are applied in a more general context, solving general problems from different domains of knowledge [Bundy 2007, Denning 2009]. Wing argues that CT is in fact human thinking instead of machine/mechanized thinking. It consists of conceptualized thinking instead of programmed thinking, i.e., it is a way of thinking instead of a product. More than a specialty of computer scientists [Yinnan and Chaosheng 2012], CT can be perceived as a human capability. In this context, CT is thought as a basic human ability such as reading, writing and solving basic arithmetic problems. CT requires a high level of abstract thinking. As it is a new way to organize thought and solve problems, and as humans are more intelligent and imaginative than computational devices, CT helps us to solve problems once unsolvable before the computing era.

Primary education is an important stage to begin developing CT as children are starting to develop logical and deductive reasoning. Companies like Google and Microsoft are promoting CT in different levels of the educational system. For example, Google is proposing a set of activities to develop CT in primary and secondary education in the USA [Google 2013]. In Brazil, there are also many projects in this direction [França and Amaral 2013, de Francca and Tedesco 2014, Ramos and da Silva Teixeira 2015, Rodrigues et al. 2015, Rodriguez et al. 2015]. Although there are some advances in the area, the research is still in its early stages and far from making CT a regular subject in basic education.

This paper describes our experiences with a pilot project for introducing some basic concepts of CT in Brazilian public schools. More precisely, we discuss the application

*The authors gratefully acknowledge financial support received from PROEXT - MEC/SESu.

of activities based on fundamental concepts of CT as proposed on [CSTA et al. 2010], mainly restricted to two classes and four activities: (i) data and image representation, referred as Binary Numbers (BN) and Colour by Numbers (CN); and (ii) sorting and searching algorithms, Battleships (BS) and Sorting Algorithms (SA). From the results obtained we can conclude that these activities promoted different abilities such as to correlate, count, apply, memorize and sort. The students' achievement assessment was obtained by analysis of pre- and post-tests. Case studies and results or our research are included to illustrate how schools and academia can be integrated to plan for instructional improvement in basic education, especially in the early school years.

The paper is organized as follows. Section 2 presents a brief theoretical framework of CT. In Section 3 the methodological aspects for developing the selected activities is described. Section 4 reports the case study: the application of the activities in two classes of elementary school. Finally, Section 5 the main conclusions and further work are presented.

2. Computational Thinking

The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) have collaborated to develop an operational definition of CT for K-12 education. CT includes (but is not limited to) the following specific skills [ISTE 2013] and [Deng et al. 2009]: (i) reformulating an apparently difficult problem into another one we know the solution by reduction, simulation, or transformation; (ii) thinking recursively and using parallel processing as approaches for data and code interpretation; (iii) using abstraction and decomposition for solving a large complex task or designing a large system via piecewise analysis; (iv) choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable; (v) thinking in terms of prevention, protection, and recovery from worst-case scenarios using redundancy, damage containment, and error correction concepts; (vi) using heuristic reasoning to discover a solution – planning, learning, and scheduling in the presence of uncertainty; (vii) using massive amounts of data to speed up solution and computation processes.

ISTE and CSTA supported by National Science Foundation (NSF), in collaboration with leaders from higher education, industry, and K-12 education created an important reference in CT called *Computational Thinking in K-12 Education Leadership Toolkit* [CSTA et al. 2010]. This toolkit provides a theoretical foundation for the process of developing CT and resources for creating a systemic change. In this toolkit, nine fundamental concepts of CS are presented in a progression chart for CT development: data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, simulation, and parallelization.

Programs and projects focusing on designing activities for the development of CT in distinct levels of education may be cited [Caetano and Oliveira 2015, Google 2013, Carnegie Mellon , Barr and Stephenson 2011, Bell et al. 2010, França and Amaral 2011, Lee et al. 2011, Kafura and Tatar 2011, Repenning et al. 2010, Hambrusch et al. 2009]. In Brazil, it is possible to cite different projects aimed at introducing CT in basic education. Just to give some examples, The *Game Design Brasil*, hosted at PUC-RJ, aims to develop a Brazilian technology to stimulate teaching and learning of comput-

1. Fill the doted space with the element that completes correctly each sequence: ○ ● ○ ● .....   1 2 4 8 .....
2. Observe how we can use simbols to write the words below: apple= ♠ △ △ ◇ ♡ panel= △ ♠ ♣ ♡ ◇ paper= △ ♠ △ ♡ √ Using the same simbols, write the word: appear = ...
3. From the seven cards ordered on the table, write down the sequence of cards turned face up by the student to find: a. The card with the highest value: b. The card with the number 2: c. The card with the number 8:
4. From the four shuffled cards on the table, put them in order counting how many pairs of cards were turned face up, at a time, by the student: ....

Figure 1. Pre and post-test

ing in K-12 education [PUC-Rio 2014]; the *Computação Desplugada* (in English Computer Science Unplugged), which aims to propagate the foundations of CS without the use of computer to students in public schools [Scaico et al. 2012]; and, our project, *Explorando o Pensamento Computacional para a Qualificação do Ensino Fundamental* which aims to create and apply activities that develop the essential concepts of CT [UFPel 2014, Andrade et al. 2013].

3. Methodology and Development of the Activities

The proposed activities were developed based on the material available in the book Computer Science Unplugged [Bell et al. 2010], created by the Universities of Carnegie Mellon and Canterbury with the support of Google enterprise.

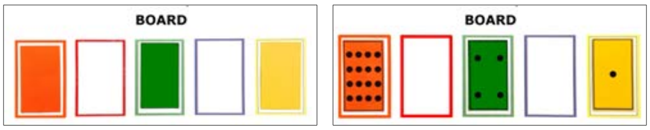
For this work, we apply four of these activities, named: (i) Binary Numbers (BN); (ii) Colour by Numbers (CN); (iii) Battleships (BS); and (iv) Sorting Algorithms (SA). A general description of these activities is presented in [Bell et al. 2010], with suggestions of materials and discussions to be raised with children during practice. Some extensions were proposed as adaptations for the target group, fourth grade students (age range from 9 to 13 years) of Basic Education, and were aimed at helping with the development of abilities such as represent, count, correlate, classify and order. The methodology for each one of the four activities includes their description and development in the classroom. Activities were applied weekly, in one hour and thirty minutes, being organized as tasks, each one with concepts and exercises. In order to measure the increase in student learning related to the CT abilities, a test, presented in Figure 1, was developed to be applied individually before (pre-test) and after (post-test) the activities. Each question of Figure 1 was first explained to the student, leaving him/her free to answer. In question 3, seven ordered playing cards are put on the table, asking the students to turn the cards one by one, to answer each question. In question 4, four cards were randomly put on the table. Students were instructed to order the cards turning pairs of cards.

First the pre-test was applied and then, the development of the activities started (details in next subsections). After them, the same test was reapplied.

3.1. Activity 1 - Binary Numbers

Summary: The objective of this activity is to communicate using binary codes, working with the conversion of binary codes to decimal numbers, further their conversion to letters and the reverse conversion of letters into numbers and binary codes. For that, students are invited to participate in a secret club, where the binary codes are used for communication. The activity is composed of three tasks, each one with concepts, exercises and evaluation.

Educational Material: the activity is presented using boards and each board contains five colored cards (with 1, 2, 4, 8 and 16 black dots on one side). An enlarged version of the material was made for demonstration in class and a smaller version was made for each student (Figure 2). Also, a table was designed associating decimal numbers (1 to 26) to letters (A to Z).



(a) Binary 10101 (b) Decimal 21

Figure 2. Student' Material

Methodology: In order to convert from binary codes to decimal, we initially present the binary codes, with some examples. Then, each student gets five coloured cards and a board, knowing that each card can only be placed in the board space of the same color. Next, the teacher explains how we can represent a binary code on the board: when the code digit was 0 they should not place the card on the board; when the digit was 1 they should place the corresponding card. The next step is to find out what is the secret number represented by the code. For this, the cards must be flipped and the black dots counted. They use the table described in the Educational Material to convert the secret number into letters. In the reverse conversion - letters to binary code - the students receive two pages where, in the first, they write his/her name and favorite fruit. For each letter used, they fill the respective number (using the table) and convert to binary code. After that, the second page containing only the binary code is delivery randomly to another student, who must decipher the message. Concluding, a method of converting decimal numbers to binary code by successive divisions was also practiced with the students.

3.2. Activity 2 - Colour by Numbers

Summary: The objective of this activity is to show students how computer recognizes images, working with the process of coding and decoding pixels that form images on the screen. For this, black and white images are presented to the students, simulating a computer screen. First, from a drawing, the students learn to determine the code that represents the arrangement of pixels on the screen. So, they do the reverse process, determining the drawing from the arrangements of pixels, given by a code. This activity is developed in two tasks, considering learning concepts, practice exercises and evaluation.

Educational Material: the activity is presented using polystyrene boards measuring 48cm wide by 48cm high. In each board, a grid is drawn with a specific number of squares, depending on the number of pixels to be simulated on the screen. In the presentation of the images on the screen, we used small black squares built on cardboard, which were fixed to the board, forming the drawings used to explain the activity.

Methodology: using the polystyrene boards we first present a simple drawing, such as a letter of the alphabet, to show how each line of the image can be represented by a sequence of numbers: the first representing the number of white pixels (or squares without black cardboard) on the line from the first small square on the grid; the second representing the number of black pixels; and so on until the end of the first line. This first example is explained from the first to the last line. Then, the students receive some exercises to do in class and go through an evaluation. In the second task, the reverse is shown: from a code, with the number of white and black squares on each line, we derive an image. After the explanation, they do some exercises and an evaluation.

3.3. Activity III - Battleships

Summary: the aim of this activity is to introduce through a battleships game three different search methods - linear search, binary search and search by hashing - often employed to find information in (large) collections of data. The activity is split into three tasks one for each search method.

Educational Material: the necessary materials for the development of the activity are: (i) game boards, as illustrated in Figure 3, in large format for the teacher, posted on Styrofoam, and in A4 sheets for students; (ii) squares made of black cardboard with dimensions 6 cm × 6 cm; (iii) pins to secure the squares in the Styrofoam; (iv) beans.

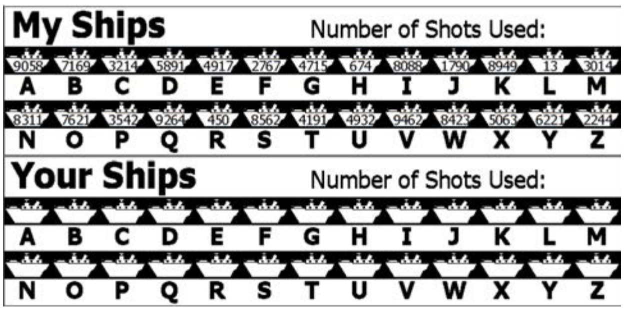


Figure 3. Battleships board [Scaico et al. 2012]

Methodology: the battleships game is a guessing game for two players. The goal of the game is to sink a ship chosen by the partner. Each player must tell his/her partner the boat number (of their ships) that the opponent must overthrow. The players take turns to guess where the partner's ship is. One player says the letter name of a ship and the partner tells him the number of the ship at that letter. The numbers of shots taking to locate the partner's ship is the score of the game. The player who hits fewer shots wins. In the first task the boats are randomly numbered, and the players execute a linear search. They go through along the positions, one by one. In the second task ship numbers are sorted, hence they learn that the best strategy to win is to use binary search. That is, the player must shoot in the middle of the ships list and see whether the number is greater than or less than the value he is looking for. This determines if the ship is in the first or second half of the board. This strategy should be repeated until the end of the game. In the third task, the ships are organized in ten columns (0 to 9) and it is possible to find out which column the ship is in simply by adding all the digits of the ship's number and computing the sum modulo 10 (the modulo operation finds the remainder after division of the sum by 10). Each task is presented in a different meeting. At the beginning of each lesson the teacher presents the game playing with a volunteer, using the larger board posted in front of the classroom. The black cards are used to mark where the shots have been taken, allowing the reuse of the board. With the same purpose the students can use beans. Then, some game rounds must be performed between the students and the teacher must collect and discuss the scores. The idea is to follow up a discussion, questioning what would be the minimum and maximum scores possible, if there is a strategy which leads to a lower score and what the advantages of each search method are. The binary search is usually faster than the linear search, but the linear search doesn't require the ships to be sorted into order. The search by hashing is usually faster than the other two, but it is possible, by chance, for it to be very slow. In the worst case, if all the ships end up in the same column, it is just as slow as the first strategy.

3.4. Activity IV - Sorting Algorithms

Summary: this activity introduces two sorting algorithms, named as selection sort and quicksort. The purpose of this activity is to show that there are different strategies to sort a set of data and that the choice of the method to be applied directly impacts the speed at which you get the sorted data. According to the method chosen, it can take a long time to sort a large list into order, even on a fast computer.

Educational Material: the material used for the activity was: scales; kits of 8 bottles, all with different weights, in a box; kits of 30 squares made of black cardboard with dimensions 6 cm x 6 cm; board containing 8 horizontally aligned positions, one for each bottle; and rubber bands. We considered one material set for each trio of students.

Methodology: at the beginning of each meeting, before distributing the material to the students, the teacher presents the sorting algorithm for the whole class. Particularly, in the first task, before presenting the method, is reviewed the concept of comparison. The teacher chooses two bottles randomly and asks which is the lighter (or heavier). He weighs the two bottles and then points out that a comparison operation was done to find the answer. Following, the selection sort is introduced. Two bottles are removed from the box, a black square is picked up and the bottles are weighed. The lighter remains with the teacher and the heaviest is placed on the floor. Another bottle is removed from the box, a black square is picked up, and the bottle is compared with the one that the teacher has at hand. The teacher gets the lighter one and the other is placed on the floor. This process is repeated for each remaining bottle in the box. When the box is empty, the teacher has the lightest bottle in his hand. Then, he puts the lighter in the leftmost free position of the board, places the other bottles back into the box and repeats the process until it has all the bottles placed on the board. The black squares are used to count the total number of comparisons required to sort the whole kit of bottles. For each comparison made, a black square is picked up. After the teacher's presentation, students apply the method to sort subsets of 3 to 8 bottles, always recording the cost of ordering each subset. In the selection sort they note that the comparison cost for each bottle subset is the same for all groups of students. Before the final evaluation of the first class, the number of comparisons obtained by subset are discussed. The teacher concludes the presentation, describing that by using this ordering method, the number of comparisons required to sort

x bottles is given by $\sum_{i=1}^{x-1} i$. In the second task quicksort is introduced. A bottle of the kit is selected and chosen as pivot. The teacher puts a rubber band on the bottle to identify that it is the pivot. Then the bottles are removed from the box one by one and compared to the pivot. For each comparison a black card is picked up. If the bottle is lighter than the pivot, it is placed to the left of the pivot. If it is heavier than the pivot, it is placed to the right of the pivot. When the box is empty, the pivot has on its left hand side all the bottles that are lighter than it, and on its right all the heavier. With this information it is possible to put the pivot on the board in its correct position. If we have x lighter bottles than the pivot, its position on the board from left to right is $x + 1$. Then, the remaining bottles are replaced into the box and the process is repeated until all bottles are correctly positioned on the board. In such case, the students note that the comparison cost varies among groups, even considering the same number of bottles in the subset. So, a final discussion shall be conducted. The teacher guides a comparison between the two approaches and emphasizes that according to the method chosen, the ordering can be achieved in fewer steps (time).

4. Results about the pilot project

First, the profile of classroom practices is described. Then, the main results concerned with pre- and post-tests are presented.

4.1. Profile of classroom practices

Decisions taken for the implementation of this pilot project took into consideration meetings with our City Department of Education and Sports. The activities were applied in a primary public school, which is located near our university, making it easier not only the access of undergraduate students and professors' support to the practices in school classes, but also increasing the integration between school and academia communities.

Regular meetings were held with the management and coordination of the school, identifying the profiles of the groups and preparing preliminary work schedule. The study was conducted on two fourth year classes since in order to develop the chosen CT concepts, the children should know how to read, write and basic arithmetic. The survey addressed a sample of 46 students divided into a morning class (22 students) and an afternoon class (24 students). The age of the students ranged from 9 to 13 years old, but the vast majority (82%) was 9 or 10 years old. Regarding gender, 63% of students were male while 37% were female.

Regarding the schedule, the activities were held in the 2nd semester of 2013, organized as follows: (i) two meetings, one for the pre- and the other for the post-test; (ii) three meetings (one per week) for each activity and evaluations (a day for each task) . Moreover, the practice of all activities began in the second meeting, after applying the pre-test. The meetings were held with the presence of two or three Computer Science undergraduate students sharing the following responsibilities: (i) the oral presentation of the activity; (ii) the support with concrete material related to the development of activities (only when necessary); and (iii) the individual assistance (when required by students). In addition, two professors/researchers, together with the class teacher were always present. The formers supervised the undergraduate students and helped collecting data for the project; and the latter, helped with previous knowledge about behaviour and personal characteristics of each student.

4.2. Results about pre- and post-tests

The test included four questions (Q1 and Q2, worth 2 points each; Q3 and Q4, worth 3 points each) devised to evaluate the following abilities: association, correlation, count, application, identification, memorization, comparison and sorting. The conception of the assessment instrument was based on these skills and the weights of the questions have been distributed according to the abilities involved. In particular, Q1 required from students the skills to correlate and count; Q2 to associate, correlate and apply; Q3 to identify, memorize and compare; and Q4 to correlate, count, apply, memorize, compare and sort. Based on these skills, Q1 and Q2 evaluated activities BN and CN, Q3 evaluated BN, and Q4, activity SA. Students' total scores and by question, in pre- and post-test, including the gains and the results of the test of significance are provided in Table 1.

Table 1. Description of students scores. Total and by question scores in the pre- and post-test including gain and results of significance test.

Group	Question (weight)	Before		After		Gain		t-test	
		Mean	SD	Mean	SD	Mean	SD	T	p
Morning n=16	1 (2,0)	1,1	0,3	1,1	**0,6**	0,0	0,5	0,24	0,8172
	2 (2,0)	1,6	0,8	1,5	**0,9**	-0,1	0,5	-0,72	0,4847
	3 (3,0)	2,3	0,9	2,6	**0,7**	0,4	1,3	1,14	0,2708
	4 (3,0)	1,2	1,2	2,4	**0,9**	1,2	1,6	2,92	0,0105
	Total	**6,1**	**1,8**	**7,6**	**1,6**	**1,5**	**1,6**	**3,67**	**0,0023**
Afternoon n=19	1 (2,0)	1,1	0,5	1,3	**0,5**	0,2	0,5	1,37	0,1868
	2 (2,0)	1,5	0,8	1,7	**0,8**	0,2	1,0	0,84	0,4141
	3 (3,0)	2,5	0,7	2,7	**0,5**	0,2	0,7	1,25	0,2284
	4 (3,0)	1,6	1,1	2,1	**1,0**	0,5	1,4	1,64	0,1185
	Total	**6,7**	**2,1**	**7,8**	**1,1**	**1,1**	**1,8**	**2,59**	**0,0186**

n = number of students; SD = standard deviation

The pre-test score describes the initial condition of the student in relation to the skills worked in the activities. The gain with the development of these activities is measured by the difference between the scores of students in the post-test and pre-test. The significance of these gains is verified by the t-test for paired samples, with $\alpha = 0.05$. The hypothesis under verification in this test assumes that the average gain of the students (μ) does not differ from zero, i.e. $H0 : \mu = 0$.

It is possible to verify for both groups that the average performance of students in the pre-test was lower than 7.0, but the afternoon class (average 6.7) was slightly better than the morning class (average 6.1). In the post-test, the gain from the morning class (average 1.5) was higher than the one from the afternoon class (average 1.1). Although they are both low, these average gains were significant. However, when one analyses the gain by question, it is possible to note that only in Q4, and only for the morning class, the average gain was significantly higher than zero ($p = 0.0105$). This result suggests that the morning class responded better to the skills worked in the SA activity. This analysis also reveals that questions that the students could not solve in the pre-test are still incorrect in the post-test. For example, only 11% of students got the first question fully right in the pre-test. Although the percentage increased to 23% in the post-test, most students could not answer the question that asked to complete a numerical sequence. This means that activities NB and CN, did not help the ability to correlate and count that were needed to solve the question. Questions 2 and 3 required six of the eight abilities mentioned, and students got at least 75% of it right already in the pre-test, with no significant increase in the post-test. Only in question 4 there was a major increase in the result from 40% in the pre-test to 80% percent in the post-test. This question required different abilities such as to correlate, count, apply, memorize and sort.

These results may suggest that the pre and post-tests used were not a good instrument to measure the gain achieved by the students. This gain was clearly observable in the class assignments. This observation is important so we can seek a better mechanism to measure the knowledge and abilities of students in the next years.

5. Final remarks

This paper contributes with a detailed description, from modeling to development of appropriate methodology to teach some concepts of CT. Specially, we focus on four activities related to data and image representation, sorting and searching algorithms, developing deductive logical reasoning for fourth grade students in a public elementary school. The students achievement assessment is obtained by analysis of pre- and post-tests. Case studies and results of our research illustrate how schools and academia can be integrated to plan for instructional improvement in basic education, especially in early school years.

Further work considers improvements in the evaluation tests and development of new activities promoting other potential abilities of logical reasoning in elementary school students. We hope that this project can be one step forward in consolidating CT in primary schools in Brazil in the future.

References

Andrade, D. et al. (2013). Proposta de atividades para o desenvolvimento do pensamento computacional no ensino fundamental. In *WIE 2013*, pages 169–178.

Barr, V. and 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.

Bell, T., Witten, I. H., and Fellows, M. F. (2010). *Computer Science Unplugged*.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1:67–69.

Caetano, G. and Oliveira, O. (2015). Computational thinking and child performance in a preschool. In Dagiene, V., Schulte, C., and Jevsikova, T., editors, *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCS 2015, Vilnius, Lithuania, July 4-8, 2015*, pages 57–62. ACM.

Carnegie Mellon. Center for Computational Thinking. http://www.cs.cmu.edu/ CompThink/. Accessed March, 2016.

CSTA, ISTE, and NSF (2010). Computational Thinking Leadership Toolkit. http://www.csta.acm.org/Curriculum/sub/CompThinking.html. Acessed March, 2016.

de Francca, R. S. and Tedesco, P. (2014). Um modelo colaborativo para à aprendizagem do pensamento computacional aliado a autorregulação. In *Anais do SBIE 2014*, pages 1133–1142.

Deng, Z., Huang, W., and Dong, R. (2009). Discussion of ability cultivation of computational thinking in course teaching. In *Proc. ICETC'09*, pages 197–200.

Denning, P. J. (2009). The profession of it: Beyond computational thinking. *Commun. ACM*, 52(6):28–30.

França, R. S. and Amaral, C. J. H. (2011). Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. In *XIX Workshop de Informática na Escola*.

França, R. S. and Amaral, C. J. H. (2013). Ensino de computação na educação básica no brasil: Um mapeamento sistemático. In *XXI Workshop de Informática na Escola*.

Google (2013). Exploring Computational Thinking. http://www.google.com/ edu/computational-thinking/. Acessed March, 2016.

Hambrusch, S. et al. (2009). A multidisciplinary approach towards computational thinking for science majors. In *SIGCSE 2009*, pages 183–187. ACM.

ISTE (2013). Operational Definition of Computational Thinking. https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf. Acessed March, 2016.

Kafura, D. and Tatar, D. (2011). Initial experience with a computational thinking course for computer science students. In *SIGCSE 2011*, pages 251–256. ACM.

Lee, I. et al. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1):32–37.

PUC-Rio (2014). Scalable Game Design Brasil. http://www.serg.inf.puc-rio.br/wiki. Acessed March, 2016.

Ramos, F. and da Silva Teixeira, L. (2015). Significação da aprendizagem através do pensamento computacional no ensino médio: uma experiência com scratch. In *Anais do WIE 2015*, pages 207–217.

Repenning, A., Webb, D., and Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *41st Tech. Symp. on Comp. Sci. Education*, pages 265–269, New York, USA. ACM.

Rodrigues, R., Andrade, W., Guerrero, D., and Sampaio, L. (2015). Análise dos efeitos do pensamento computacional nas habilidades de estudantes no ensino básico: um estudo sob a perspectiva da programação de computadores. In *Anais do SBIE 2015*, pages 111–121.

Rodriguez, C., Zem-Lopes, A. M., Marques, L., and Isotani, S. (2015). Pensamento computacional: transformando ideias em jogos digitais usando o scratch. In *Anais do WIE 2015*, pages 62–71.

Scaico, P. D. et al. (2012). Um Relato de Experiências de Estagiários da Licenciatura em Computação com o Ensino de Computação para Crianças. *RENOTE*, 10(3).

UFPel (2014). EXP-PC - Explorando o Pensamento Computacional para a Qualificação do Ensino Fundamental. `http://wp.ufpel.edu.br/pensamentocomputacional`. Acessed May, 2016.

Yinnan, Z. and Chaosheng, L. (2012). Training for computational thinking capability on programming language teaching. In *Proc ICCSE'12*, pages 1804–1809.