TASK-BASED ASSESSMENT OF STUDENTS' COMPUTATIONAL THINKING SKILLS DEVELOPED THROUGH VISUAL PROGRAMMING OR TANGIBLE CODING ENVIRONMENTS

Takam Djambong and Viktor Freiman
Université de Moncton
Moncton, NB, Canada

ABSTRACT

While today's schools in several countries, like Canada, are about to bring back programming to their curricula, a new conceptual angle, namely one of computational thinking, draws attention of researchers. In order to understand the articulation between computational thinking tasks in one side, student's targeted skills, and the types of problems they aim to solve, we conducted a small-scale pilot case-study with two groups of students for the elementary (grade 6) and middle (grade 9) grades. While the students were working on 5-week-long curricular units in technology using robotics-based and computer-programming environments, we assessed their computational thinking abilities with 23 tasks given as pre- and post-test. Aiming to validate the tasks, namely, to see in what way it allows measuring computational things, we found a disparity between the types of the skills assessed, the easiness of the tasks, and the age groups, which makes difficult to arrive to some stable conclusion. We stated then a need for a longer and more sophisticated assessment as a subsequent research perspective to establish a stronger empirical evidence of possible relationships between related variables.

KEYWORDS

Computational thinking skills, Problem solving tasks, Technology-rich learning environment

1. INTRODUCTION: ISSUES AND PROBLEM STATEMENT

The world in which we live today, is already and will continue to be characterized by (1) increasing ubiquity of digital and computer technologies, (2) an extremely rapid changes in society and in the labor market in respect to the access to technology-rich environments,(3) increasing complexity of problems people will need to solve and challenges associated with this complexity, (4) the emergence of a networked communities marked by a strong trend towards connectivity combined to social and professional collaboration. The citizens of the 21st century are called to show great resilience capabilities and to develop specific skills, in order to adequately adapt to the constraints of an increasingly digital, complex and interconnected society.

How does today's society through teaching and learning system, prepare young citizens to adapt to these changes? For example, results of a recent survey[1] conducted in 2012, revealed that a number of Canadians, even those who belong to so-called 'born digital' generations showed the lowest levels of competence in problem solving in technology-rich environments (PS-TRE)[2]. Hence, in addition to traditionally known types of digital divides related to the access to technology and Internet and to its meaningful usage, other kinds of divide may emerge, among them the ability to use technology more productively and effectively in a larger variety of technology-rich contexts and tasks.

[1]Programme for International Assessment of Adult Competencies (PIAAC)
[2]Statistics Canada, 2013, p.13.

One of possible explanations of these new types of divides could be that many everyday activities young people naturally conduct using technology do not automatically enable them to make appropriate use of ICT in more formal (academic) contexts, especially in solving complex tasks. This somehow justifies the recommendations that UNESCO has made in its recent policy framework[3] for a sustainable development goals (SDG) putting a greater emphasis on the acquisition of high level skills, both cognitive and non-cognitive, such as critical thinking, problem-solving, decision making and teamwork. These skills are necessary to facilitate technology transfer to other contexts and the adaptation of young people to the current constraints of a (technology-richer) labor market in constant mutation.

How can we develop these skills and the abilities of transfer, remains an unsolved psychological and techno-pedagogical issue. The problem of "*how to teach effectively*" in order to enhance students to "*make learning in depth*" in the era of digital technologies, with its multiple Web-based resources and Mobile Apps must be analyzed in terms of learning environments that build on the principles of deep learning. The later require combination of (1) the ability to transfer skills from one context to another, (2) a more thoughtful understanding of academic content, and (3) the development of a variety of high-level skills. The deep learning requires exemplary teaching practices using the project-based approach and assignment of students to group tasks (Beaudoin et al., 2014, p.18-19).From this pedagogic perspective, as already mentioned in some previous research (Weintrop et al., 2015), the thoughtful use of computational tools and skillsets can deepen learning of disciplines like science, technology, engineering and mathematics (STEM).

In this respect, programming and coding has been recognized as one of the important competencies that require students to effectively use computational tools and devices, in order to solve complex real-problem today(Chao, 2016, p.202). The construct of computational thinking (CT), while not yet well defined, is often related to programming and coding while being considered as particular type of analytical thinking that employs mathematical and engineering thinking along with the abilities to understand and to solve complex problems within the constraints of the real world (Voskoglou& Buckley, 2012, p.32). The study of learning situations which are likely to favor the development of computational thinking in students through the use of programming and coding environments may therefore be an interesting line of inquiry in order to identify the most appropriate teaching practices to achieve this goal. But before we identify such practices, we need to determine key aspects of computational thinking and the set of tasks or activities that contribute to its development.

2. THE RESEARCH STUDY

2.1 The Context

The present study arose from the initiative launched in Canada in 2011 by the Council for Research in Social Sciences (SSHRC), calling for identification of new ways of learning that Canadians will need to succeed in society and the labor market of tomorrow (Beaudoin et al. 2014, p.40). It is in this context, the ICT Competences Network in Atlantic (CompéTICA, Compétences en TIC en Atlantique), a partnership development team, presently conducts several case studies oriented toward the identification and measurement of the acquisition and transfer of digital literacy to transition points between the family and school, between primary and secondary, between school and post-secondary institutions, and finally, between educational institutions and the labor market. This research-and-practice-based partnership aims among others axis of research, identify best teaching practices of different educational environment and life, from the perspective of the life-long continuum of digital competences. At the first stage of the project, several aspects related to digital competences development were identified by our partners as important to develop in all citizens, starting from the young age, and among them computational thinking related to the computer programming (Gauvin et al., 2015). At the second stage, we conducted preliminary observations on how development of computational thinking occurs during programming and coding activities among students of from the elementary (Grade 6) and middle school (Grade 9) in New Brunswick, Canada. More precisely, we aimed to investigate whether it is possible to establish a link between problem-solving tasks in a computer

[3] Education 2030 framework for action (SDG4), p.17

programming and robotics-based environments and second, the targeted skills related to computational thinking. At this preliminary stage of our data analysis, we focus on validating assessment tasks, namely to see, based on the results of the pre- and post-tests, a variation according to the estimated difficulty of the task, evolution of results from the pre- to the post-tests for particular types of tasks, and what are the tasks that could be more helpful in capturing the development of computational thinking.

2.2 The Conceptual Framework

2.2.1 Computational Thinking (CT)

First postulated by Seymour Papert in 1980s and 90s through the use of LOGO programming language and the development of cognitive abilities in solving a variety of computer-based problems, computational thinking (CT) emerged as core concept being popularized by Jeannette Wing (2006), who defines it as a set of attitudes and skills that all universally applicable, not just IT professionals should learn and master. Since then, CT is considered both as: (i) an essential component of 21st century skills, (ii) a computational approach of problem-solving process, (iii) a complex construct enabling STEM[4] disciplines skills acquisition, and (iv) the fundamental aspect of digital and computer science literacy. The CT then might well be regarded as a cognitive and intellectual tool, that would most effectively foster the implementation of the problem-solving process, through the use of technology-rich environments, and this, in a both socio-constructivist and socio-cultural perspective of situated and distributed cognition. CT could then be viewed as a type of analytical thinking that employs mathematical and engineering thinking to understand and solve complex problems within the constraints of the real world (Voskoglou & Buckley, 2012).

The educational intervention conducted in the context of our study was to develop among target students some skills[5] related to computational thinking such as: (i) capture different angles of approach to a problem and its solution (abstraction) (ii) reflect on the tasks to be performed by considering a series of steps (algorithmic thinking), (iii) assess the opportunity after assessing the complexity of a given problem, whether to break it down into several simple problems (decomposition), (iv) be able to link a specific problem to other problems of the same type that has already been solved (pattern recognition) and (v) realize that the solution to a given problem, can be the basis of the resolution of a wide range of similar problems (generalization). These targeted skills justify the choice of the different solving tasks proposed to students during the pretest and posttest of our study and rely despite the lack of a consensus definition of the construct of computational thinking, on the one proposed by the Computer Science Teachers Association (CSTA) as follows: "formulating problems in a way that enable us to use a computer and other tools to help solved them; logically organizing and analyzing data; representing data through abstractions such as models and simulations; automating solutions through algorithmic thinking (a series of ordered steps); identifying, analyzing and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources; and generalizing and transferring this problem solving process to a wide variety of problems". (Barr, Harrison & Conery, 2011).It is also noted that this study is justified by the fact that the construct of in computer thinking remains unclear in its definition, as well as in its practical implementation and its conceptualization as learning object in the school system, both in primary, secondary and postsecondary level, because of the lack of a disciplinary field that is her own despite her almost natural connection to the field of computer science.

2.2.2 Technology-Rich Learning Environments (TRE)

Digital literacy is today considered as one of the key skills of the 21st century[6]. Because we live in an increasingly digital world, the use of learning environments with high technological component will be further promoted and sought. Technology-rich learning environments can be defined as educational environments that rely heavily on digital resources and have massively invested in their technology infrastructure[7].As part of this study, we are particularly focused on environments that promote the

[4]Science, Technology, Engineering and Mathematics
[5] Skills identified through the operational definition of CT released by CSTA in collaboration with The International Society for Technology in Education (ISTE) in 2011
[6]OECD, 2011
[7]OECD, 2014, p.60.

implementation of problem-solving activities to the development of skills related to computational thinking. Literature (Allan et al., 2014; National Research Council, 2011; Grover & Pea, 2013) identifies several types of environments associated with the implementation of learning for the development of computational thinking. These environments can be classified into three main categories: (i) simulation and modeling environments (Wilensky, 2014, Basawapatna et al., 2014), (ii) the games design environments (Repenning et al., 2014), (iii) programming (both visual and tangible) environments (Lye & Koh, 2014; Kalelioglu & Gülgahar, 2014; Berland & Wilensky, 2015; Bers et al., 2013; Chao, 2016; Leonard et al., 2016).Through the literature it appears a strong relationship between firstly, programming and computational thinking, and secondly between programming and STEM education. A recent study[8] conducted in Australia on the conceptions of teachers on computational thinking, revealed that a majority of these teachers perceived programming environments (visual and robotics) as the most appropriate to foster the development of computational thinking in their students. However, the criteria for choosing the most appropriate learning environment to support a specific learning activity for the development of these skills are not always clearly documented. However, programming could better expose students to computational thinking which involves problem-solving tasks using computer science concepts like abstraction, algorithmic thinking and decomposition (Lye & Koh, 2014, p.51). The choice of technology-rich learning environments (Scratch and EV3 Robotics kit) made in our study to support computational thinking skills acquisition taking place in a context of technology course, based on this assumption found throughout the literature, needs more strong empirical evidence through more classroom-based interventions studies. One can postulate that this choice may depend on the subject taught, the type of learning activity considered, targeted skills, the complexity of the problem (ill-defined or well-defined) that students face during the educational intervention, the nature of the problem solving tasks given to students, etc. Because intervention takes place in the context of a particular technology-rich learning environment, the ability of student may depend upon the level of complexity of problem solving tasks performed during the intervention so that there might be some relationship with their ability to perform task-based assessment during the different tests.

2.2.3 Problem Solving Tasks

Perhaps, the most important cognitive goal of education both formal and informal in every educational context is problem solving (Jonassen, 2010). The ability to solve problems is probably one of the most important manifestation of human thinking process and a critical component of intelligence, but its actual assessment is far to be obvious, especially in connection with computational thinking, a construct yet being conceptualized. Then an important question would be how to choose the most appropriate tasks to measure the ability to solve problems in connection with the development of skills associated with computational thinking? Some studies have already addressed the issue, and here are some computational thinking test assessments that have been reported in the literature: (1) tests based on standardized exercises as a collection tool (Dee Miller et al., 2012), (2) tests based on the traces of activities in an IT environment as collection tool (Koh et al. 2014), (3) assessments based on already classical cognitive validated test (Ambrosio et al. (2014), Bebras international challenge on informatics and computational thinking (Dagiené & Stupuriené, 2015; Dolgopolovas et al., 2015), (4) the Google for Education Exploring CT problems[9] and (5) Computer Science Unplugged[10] tests activities. Our study was globally based on the test proposed by the Bebras[11] contests which consist of a set of tasks in a form of short questions or quizz. Each Bebras task can demonstrate an aspect of computer science (CS) and test the aspects of CT of the participant (Dolgopolovas, Jevsikova, Savulioniené & Dagiené, 2015). Additionnaly, each Bebras task has at least some component of CT: abstraction (AB), decomposition (DE), algorithmic thinking (AL), pattern recognition (PR), or generalization (GE). In order to solved these tasks, students have to mobilize abilities related to CS, discrete strutures, computation, data processing, data visualisation, algorithmic and programming operations (Diagene & Stupuriene, 2014). Some criteria for good Bebras taks are presented in table 1 below:

[8]Bower et al., 2015
[9]https://www.google.com/edu/resources/programs/exploring-computational-thinking/
[10]http://tabs.chalifour.fr/la-science-informatique-a-lecole/cs-unplugged/
[11]http://www.bebras.org/

Table 1. Criteria of good Bebras tasks (Diagene & Futschek, 2008, p.22; Diagene &Stupuriené, 2015, p.22-27)

Good tasks	Explanation
Are related to computer science and computational thinking	Bebras contest is a competition on CS and CT
Student could achieve 18 to 24 tasks within 45 55 minutes	Three minutes is approximately the average time to solve a task
Have three levels of difficulty	*Level A* (easy): all pupils of target group should be able to solved *Level B* (medium or intermediate): challenging tasks need some thinking to solve *Level C* (hard): only the best pupils can solved these tasks
Are adequate for the age of participants	Bebras contest has five age groups: Little beaver (grade 3-4; age 8-10) Benjamin (grade 5-6; age 11-12) Cadet (grade 7-8; age 13-14) Junior (grade 9-10 age 15-16) Senior (grade (11-12; age 17-19)
Are independent from any curriculum	The International Bebras contest cannot support all curricula of a large number of countries

2.3 Methodology

Our exploratory study was conducted during 2015-2016 school year through three stages. We began meeting with our partners in October to set up a research procedure. The first stage consisted in the literature review on computational thinking, its definition, development and assessment (November – January). The results of this work were communicated elsewhere (Djambong, 2016). Then, at the second stage, we started building up a questionnaire while making the first selection of tasks, their validation first by the partners, then in one Grade 10 classroom (February – April). Finally, due to the time restriction by the end of the school year, we opted for a small pilot study over 6 weeks (May-June) in which we conducted pre- and post- tests using selected tasks from the Stage 2 along with in-class observations and interviews with the teacher and students. While the data analysis is still underway, we present in this paper preliminary data from the pre- and post- tests. We mainly seek to investigate two research questions:

1) How do the scores vary according to the estimated difficulty of the task between the pre- and the post-test?
2) How do the scores vary according to the elements (or their combination) of computational thinking of each task?

2.3.1 Participants and Context

This pilot study took place in one school (with approximately 310 students enrolled in grades 6 through 12) in New-Brunswick, Canada. Two groups of students (one from Grade 6 and one from Grade 9, for a total of 24 students, 15 females and 9 males) took part in the study. Each group has achieved a variety of programming activities over a period of five weeks (the pre-test was given during the week 0, and the post-test and interviews during the week 6). Grade 6 students (n=10) were working mainly with the Scratch programming environment on a weekly basis (on hour per week) whereas Grade 9 students were doing activities with LEGO EV3 Robotics Kit, a tangible programming environment on a daily basis (1 hour per day). The choice of the programming environment was not prescribed by the curriculum and was made by the teacher.

2.3.2 Data Collection Instruments and Procedure

The preparation of the questionnaire consisted of several steps:

First, 19 Bebras multiple-choice tasks were selected from available online materials which were combined with 11 tasks designed by team researchers were selected. The main selection criteria were:

- The presence of at least one well-expressed CT concept in the task;

- The focus on algorithmic thinking because, intervention activities were base on programming and coding through visual or tangible environments.

The second step included analysis of selected tasks by one classroom teacher and her students from Grade 10. After this validation phase, 14 Bebras tasks and 9 tasks designed by the research team were finally selected to form the final 23 paper-pencil tasks-based assessment test.

After their parents have completed consent forms as required by the Ethics Committee, we have submitted participating students to a series of coding tasks using a questionnaire before the beginning of classroom learning activities (pre-test) and after five weeks of learning (post-test).We have also made observations (including video-recording) during 5-weeks long learning period .Just to remind, in this particular paper, we aim to investigate the tasks using only pre- and post-test results. Other data will be presented later upon the completion of their analysis.

2.3.3 Data Analysis

During this pilot case study, we use descriptive quantitative methods to analyze the data collected during the pre-test and the posttest. Because of a very small sample, we were not able to conduct more sophisticated analysis which we hope to be able to do next year, when we plan to extend our sample and observation period.

2.4 Results

We recall that our first research question was to investigate in what way the set of tasks (the same for Grade 6 and Grade 9) reflect (the changes in) computational thinking while comparing the results before the classroom learning activity and after five week of programming classes.

2.4.1 Question 1: Global Performance on the Pre- and Post-Tests

We begin with presenting general score on the pre- and post-tests for both Grades (6 and 9) (Table 2).

Table 2. Average Scores of Solving Computational Tasks by Grade

Grade	Average score	
	Pre-test	Post-test
Grade 9 (n=14)	44.9%	48.2%
Grade 6 (n=10)	29.6%	33.0%

As we notice, for the Grade 9, the global results from the pre-test to the post-test have only slightly increased (from 44.9% on the pre-test to the 48.2%) whereas for Grade 6, the increase was slightly at the same extent (from 29.6% to 33.0%). We also observed that the results in Grade 9 are significantly higher than in Grade 6.

While presenting the results according to the difficulty level of the task (as being attributed by the Bebras team for the Bebras tasks and by our research tem for other tasks, Table 3), we take into account that the difficulty level was not the same for Grade 9 and Grade 6, there is why, we mentioned the number of problems for each Grade.

Table 3. Average Scores of Solving Computational Tasks by Level of Task Difficulty

Level of task difficulty				Average score Grade-6		Average score Grade-9	
		K-6	K-9	Pre-test	Post-test	Pre-test	Post-test
Easy	(A)	2*	9*	40.0%	55.0%	46.7%	48.4%
Medium	(B)	8*	8*	25.0%	28.7%	44.2%	41.1%
Hard	(C)	13*	6*	26.9%	32.3%	36.7%	34.5%

(*****): *number of tasks by level of difficulty in each grade*

We observed that, for the Grade 6, the highest score was obtained for the Easy task (39.4% for the pre-test and 55.0% for the post-test). For this Grade and difficulty level (Grade 6, level Easy), we also have the (significantly) highest increase from the pre-test to the post-test. For the problems of the difficulty levels Medium and Hard, the scores are significantly lower, 25.0% and 28.7% for Medium and 26.9% and 32.3% for High. We also noticed that for both levels, Medium and High, the increase between the scores for the pre-test and the post-test is much smaller than for the level Easy. Also, we can see that the hardest problems were solved slightly with a better score than those for the Medium level. As for the Grade 9, the highest score was also obtained for the Easy level, although the difference between pre- and post-test scores is lower (46.7% vs 48.4%. Yet, we see that the Medium problems were solved with only slightly smaller score (44.2% and 41.1%). For the hard problems, it the scores are significantly lower (36.7% and 34.5%). We also notice that for this Grade (9), there was actually a slight decrease in scores from the pre- to the post-test.

Table 4. Average scores of solving computational tasks by type of computational thinking skill involved

Type of CT skill involved	**(*)**	Average score Grade-6		Average score Grade-9	
		Pre-test	Post-test	Pre-test	Post-test
AB[12]	(1)	20.0%	50.0%	86.7%	71.4%
AL[13]	(3)	40.0%	36.7%	55.6%	42.9%
PR[14]	(1)	90.0%	100%	73.3%	85.7%
AB+DE[15]	(1)	60.0%	80.0%	86.7%	71.4%
AB+PR	(4)	25.0%	20.0%	23.3%	21.4%
AL+DE	(1)	60.0%	70.0%	93.3%	71.4%
AL+PR	(2)	10.0%	10.0%	13.3%	25.0%
AB+AL+DE	(5)	14.0%	24.0%	38.7%	27.2%
AB+DE+PR	(1)	10.0%	30.0%	60.0%	71.4%
AL+DE+PR	(1)	40.0%	10.0%	20.0%	42.9%
AB+AL+DE+PR	(3)	33.3%	30.0%	44.4%	52.4%

(*):*Number of tasks for each type*

The best average scores in grade 6(90.0% for the pre-test and 100% for the posttest) were obtained for the only task involving PR as the only CT skill to be assessed, while in grade 9, best average scores were obtained in pre-test (93.3%) for the task involving a combination of AB and De CT Skills, and in post-test (85.7%) for the only task involving PR as the only CT skill to be assessed. The lowest average score were obtained for four tasks involving AL and PR as CT skills taken together (10.0% for the pre-test and 10.0% for the posttest in grade 6; and 13.3% in the pre-test and 25.0% for the post-test in grade 9). Interestingly, all types of tasks which included AL skills seemed to produce lower results with some scored having actually decreased on the post-test compared to the pre-test (like in both Grades for isolated AL tasks). In this respect we also noticed that the only combination of skills that did not have this component (AB+DE+PR) has produced and increase for both Grades whereas Grade 9 students have outperformed their Grade 6 peers for this same combination.

[12] Abstraction
[13] Algorithmic thinking
[14] Pattern recognition
[15] Decomposition

ISBN: 978-989-8533-55-5 © 2016

2.4.2 Question 2: Scores for Each of Proposed Tasks

Table 5. Average scores of solving computational tasks per each proposed task

Task N°	CT skill involved	Level of difficulty		Average score Grade-6		Average score Grade-9	
		Grade 6	Grade 9	Pre-test	Post-test	Pre-test	Post-test
Task 1	PR	A	A	90.0%	100%	73.3%	85.7%
Task 2	AL	C	B	80.0%	80.0%	73.3%	64.3%
Task 3	AL+DE	B	A	60.0%	70.0%	93.3%	71.4%
Task 4	AB+PR	C	B	70.0%	30.0%	33.3%	21.4%
Task 5	AB	C	B	20.0%	50.0%	86.7%	71.4%
Task 6	AB+AL+DE+PR	B	A	50.0%	40.0%	60.0%	64.3%
Task 7	AB+DE	C	C	60.0%	80.0%	86.7%	71.4%
Task 8	AL+PR	B	A	20.0%	10.0%	26.7%	42.9%
Task 9	AL+PR	C	C	0.0%	10.0%	0.0%	7.1%
Task 10	AB+PR	C	B	0.0%	10.0%	6.7%	7.1%
Task 11	AB+AL+DE+PR	C	C	10.0%	10.0%	13.3%	35.7%
Task 12	AB+PR	B	A	10.0%	10.0%	46.7%	35.7%
Task 13	AB+DE+PR	B	B	10.0%	30.0%	60.0%	71.4%
Task 14	AL+DE+PR	C	B	40.0%	10.0%	20.0%	42.9%
Task 15	AB+AL+DE+PR	C	C	40.0%	40.0%	60.0%	57.1%
Task 16	AL	B	A	20.0%	10.0%	46.7%	28.6%
Task 17	AL	B	A	20.0%	20.0%	46.7%	35.7%
Task 18	AB+AL+DE	C	B	10.0%	10.0%	46.7%	28.6%
Task 19	AB+AL+DE	B	A	10.0%	40.0%	26.7%	42.9%
Task 20	AB+AL+DE	A	A	30.0%	10.0%	60.0%	28.6%
Task 21	AB+PR	C	B	10.0%	30.0%	6.7%	21.4%
Task 22	AB+AL+DE	C	C	20.0%	40.0%	33.3%	21.4%
Task 23	AB+AL+DE	C	C	0.0%	20.0%	26.7%	14.3%

We note that the best average scores were obtained for the tasks 1(90.0% for the pre-test and 100% for the post-test in grade 6; 73.3% for the pre-test and 85.7% for the post-test in grade 9), 2(80.0% for the pre-test and 80.0% for the post-test in grade 6; 73.3% for the pre-test and 64.3% for the post-test in grade 9) ,3 (60.0% for the pre-test and 70% for the post-test in grade 6; 93.3% for the pre-test and 71.4% for the post-test in grade 9) , 5 (86.6% for the pre-test and 71.4% for the post-test in grade 9)and 7 (60.0% for the pre-test and 80.0% for the post-test in grade 6; 86.6% for the pre-test and 71.4% for the post-test in grade 9) while the lowest ones were obtained for the task 9 (0.0% for the pre-test and 10.0% for the post-test in grade 6; 0.0% for the pre-test and 7.1% for the post-test in grade 9)and 10(0.0% for the pre-test and 10.0% for the post-test in grade 6; 6.7% for the pre-test and 7.1% for the post-test in grade 9). While more detailed and sophisticated analysis of this table is yet to be conducted, we can notice that that the task 7 labeled as difficult (Level C) for both Grades, 6 and 9, the results are quite good whereas for the task 9 (also Level C) they near 0% of success for both Grades. We also observe from the table that generally for the tasks that require a combination of abilities, the results of Grade 9 students are generally higher than for Grade 6 students. We also see the biggest increase from the pre- to the post-test for the tasks 19 and 21 whereas for the tasks 4 and 20, the score have rather dropped for both Grades.

2.5 Discussion

The results displayed in the above table suggest that: (i) the type of programming environment in which the students worked during the intervention phase may have some influence in their ability to solve tasks; (ii) the type of problems solving activities, in terms of complexity[16] and structuredness[17] the students faced during

[16] Complexity is more concerned with how many components are represented implicitly or explicitly in the problem and how much they interact and how much the students understand those components (*J.M. Spector et al., 2014)*

[17] In general, ill-defined problems tend to be more complex then well-defined problems (*Ibid.*)

the intervention phase may have some influence in their ability to solve tasks; (iii) generally, we can observe that, the more difficult is a task, the lowest are the average scores. It then appears that, the ability to correctly solve a task seems to decrease with the level of difficulty of the task. But this observation needs more strong empirical evidence; (iv) these results suggest that, the CT competence involved in a task may have some influence on the average score obtained during the assessment; (v) it seems to appear no clear or apparent relationship between task competence composition and average scores obtained.

The Bebras International Contest is considered as homogenous test for evaluating students' implicit ability of computational thinking in the context of the process of solving tasks which implicitly involve cognitive procedure of CT and problem solving (Dolgopolovas, Jevsikova, Savulioniené & Dagiené, 2015).The development of computational thinking skills then, suggests the implementation of a more or less complex problem-solving strategy involving several mental processes as stated in its operational definition. Student results obtained during pre-test and post-test, assuming that they are valid performance indicators, do they allow the effective implementation of such a strategy? We can just see that the average scores obtained by students in 6th Grade and 9th Grade at both pre-test, post-test, does not allow to obtain an objective picture of the evolution of students mental representations and schemes in tasks problem solving process during the test and during the learning activities in class between two tests. This finding could justify the need to develop measurement tools that have not only the characteristics of a classical psychometric test, but are also able to allow the capture of changes in implementation of thinking processes through a problem-solving activity for the development of cognitive and non-cognitive skills related to computational thinking. It therefore seems difficult to affirm that the set of tasks we used is a valid instrument for evaluating or measuring computational thinking as stated in recent study (Ibid.). It would be wise to achieve this end, to undertake further studies combining the tasks from our set with additional and eventually more appropriate data collection tools, such as in-depth interviews and analysis of students' 'thinking-aloud' discourse.

The results at both pretest at post-test, highlight a certain balance between the average scores obtained by task, with the difficulty level predicted by designers for each proposed task. This complies with the claim of the validity of the prediction criterion of Bebras tasks, which stipulates that the success rate for a particular task can be used to describe the difficulty level of that task (Vegt, 2013, p.133). However, one cannot ignore the case of some tasks for which, the average scores have not reflected the difficulty levels predicted by the designers. What seems to be in agreement with the findings of previous studies[18] indicating that, the prediction of the level of difficulty of a task is an inexact science, because its estimation is the result of task designer intuition, which can be mistaken, hence the need to properly conduct the validation process tasks for large-scale studies.

The tasks proposed in this study, differ from one another both by type (AB, AL, DE or PR) of skills related to computational thinking involved, their number per task (which can be 1, 2, 3 or 4), the level of difficulty (easy, medium or hard) of the task, and the nature of the task to be solved (checking or performing a procedure). The complexity of a task generally is the result of a joint of several parameters among which we can mention: (i) the breadth of knowledge required to solve the task, (ii) the level of prior knowledge, (iii) the intricacy for the problem-solutions procedures, and (iv) the number of relations that need to be processed in parallel during the problem solving process (Jonassen, 2010, p.3).

In our study, the level of complexity of the proposed tasks, could therefore combine parameters mentioned above. However, taking into account only those parameters as complexity indicators, it would be difficult to establish and clearly define a scale comparison of the complexity of the tasks proposed in the quiz, in the light of the difference in average scores observed for certain tasks (tasks 11 and 15, for example, which appear to have the same complexity indicators). It is therefore possible that other intervening factors such as complexity indicators and not highlighted in our study are likely to better explain the results.

The results obtained in our study need to be further supported by empirical evidence, given the following limits: (1) the small sample size does not allow analysis that could lead to a generalization of observations; (2) the choice of participants who has not subject to random assignment was able to influence and bias the results. Other limitations are inherent to the format of the tests: paper-and-pencil and multiple-choice. For instance, the teacher mentioned to us that the fact of switching from working at computers during the class hours to filling-in paper forms could be demotivating for some students. Moreover, the multiple-choice does not allow to get a track of the work done by the students leading her or him to the selection of an answer; which could also be done randomly in some cases. As we said before, it would be important to conduct

[18] Ibid. p.134

interviews with students to learn how they arrive to the answer; we also think of constructing a virtual form of this test. Our final remark concerns that period of the post-test – only few weeks before summer vacations – this could be also a factor of certain decrease in some post-test scores. However, our results point that this is not the case for all tasks, hence, our observations at this stage indicate of possibility of tasks to discriminate CT abilities and tasks-complexity levels.

3. CONCLUSION

In conclusion, this study shows that there could be a link between the students' ability to solve the tasks proposed, the type of targeted skills related to computational thinking, and the degree of difficulty or complexity of the proposed tasks. The influence of the programming environment to which the students were exposed in the context of problem solving tasks during the intervention, is difficult to demonstrate given the limitations associated with the experiment (small size of sample, non-randomized sample, lack of a control group). However, this study justifies the need for further studies to establish the validation of the proposed tasks based on more solid empirical evidence. It could thus be useful to look at the effect that the nature of the pedagogical intervention in programming environments (visual versus tangible) could have on the validation of the proposed set of tasks. For this purpose, more subtle research design for the study and instruction design for the problem-solving tasks are needed.

ACKNOWLEDGEMENT

This ongoing study is being conducted with the help of the Canadian Social Sciences and Humanities Research Council (Partnership Development Grant #890-2013-0062), New Brunswick Innovation Foundation (2016 Research Assistantship Program) and le Secrétariat aux Affaires Intergouvernementales Canadiennes du Québec (Programme de soutien à la Francophonie Canadienne). CRSH (Social Sciences and Humanities Research Council of Canada).

REFERENCES

Book

Beaudoin, J. et al, 2014. *Usages du numérique dans les écoles québécoises. L'apport des technologies et des ressources numériques à l'enseignement et à l'apprentissage. Recension des écrits*. Centre de Recherche et d'intervention sur la réussite scolaire, Montréal, Québec.

Committee for the Workshops on Computational Thinking; National Research Council, 2010. *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press, Washington, USA

Dagiene, V., Futschek, G., 2008. Bebras Intetnational Contest on Informatics and Computer Literacy: Criteria for Good Tasks. *In Informatics Education-Supporting Computational Thinking*, Springer Berlin Heidelberg pp. 19-30

Godefroid, J., 2008. *Psychologie : Science humaine et science cognitive*. De Boeck, Bruxelles, Belgique, pp 420-425

Jonassen, D., 2014. Assessing Problem Solving in Spector, J.M. et al, (Eds.) *Handbook of Research on Educational Communications and Technology*, Springer Science + Business Media, New York, USA

OCDE, 2014. *Environnements pédagogiques et pratiques novatrices, La recherche et l'innovation dans l'enseignement*, Éditions OCDE

Journal

Barr, D., et al, 2011. Computational Thinking: A Digital Age Skill for Everyone. *In Learning & Leading with Technology*, Vol. 38, No 6, pp. 20-23

Berland, M., Wilenski, U., 2015. Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking. *In Journal of. Science Education and Technology*, (2015) 24, pp. 628-647

Bers, M.U., et al., 2014. Computattional thinking and thinkering : Exploration of an early chilhood robotics curriculum. *In Computer & Education*, 72 (2014) pp. 145-157

Chao, P., 2016. Exploring Students' computational practice, design and performance of problem-solving through a visual programming environment. *In Computer & Education*, 95 (2016) pp. 202-2015

Dagiene, V., Stupuriené, G., 2015. Bebras – a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *In Informatics in Education*, Vol. 15, No. 1, pp. 25-44

Dolgopolovas, V., et al, 2016. Exploration of Computational Thinking of Software Engineering Novice Students Based on Solving Computer Science Tasks. *In International Journal of Engineering Education*, Vol. 32, No. 3(A), pp. 1-10

Grover, S., Pea, R., 2013. Computational Thinking in K-12: A Review of the State of the Field. In Educational Researcher, Vol. 42, No. 1, pp. 38-43

Kalelioglu, F., Gülbahar, Y., 2014. The Effects of Teaching Programming via Scratch on Problem Solving Skills : A Discussion from Learner's Perspective. *In Informatics in Education*, 2014, Vol. 13, No. 1, pp. 33-50

Lee, I., et al., 2011. Computational Thiking for Youth in Practice. *In ACM Inroads*, Vol. 2, No. 1, pp. 32-37

Leonard, J. et al., 2016. Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. *In Journal of. Science Education and Technology*, (2016). Doi:10.1007/s10956-016-9628-2

Lye, S., Koh, J., 2014. Review on teaching and learning of computational thinking through programming: What is next for K-12. *In Computers in Human Behavior*, Vol. 4, No. 41, pp. 51-61

Repenning, A., et al, 2014. Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation. *In ACM Transactions on Computing Education*, Vol. 15(2), No. 11, pp. 1-34

Vegt, W., 2013. Predicting the Difficulty Level of a Bebras Tasks. In Olympiads in Informatics, Vol. 7, pp 132-139

Voskoglou, M., Buckley, S., 2012. Problem Solving and Computers in a Learning Environment. In Egyptian Computer Science Journal ECS, Vol. 36, No. 4 pp. 28-37

Weintrop, D. et al., 2015. Defining Computational Thinking for Mathematics and Science Classrooms. *In Journal of. Science Education and Technology*, Vol. 2016, No. 25, pp. 127-147

Wing, J. M., 2006. Computational Thinking. *In Communications of the ACM*, Vol. 49, No. 3, pp. 33-35

Wing, J. M., 2008. Computational thinking and thinking about computing. *In Philosophical Transactions of the Royal Society a: Mathematical, Physical and Engineering Sciences*, Vol. 366, No. 1881, pp. 3117-3725

Conference paper or contributed volume

Ambrosio, A.P., et al, 2014. Exploring Core Cognitive Skills of Computational Thinking. Proceedings of PPIG, University of Sussex, Sussex, England

Basawapatna, A., et al, 2014. The Consume-Create Spectrum: Balancing Convenience and Computational Thinking in STEM Learning. *Proceedings of SIGCSE'14 Conference*, Atlanta, USA, pp. 658-664

Djambong, T., 2016. Computational Thinking in Connection with the Acquisition of Key Skills for the 21st Century: Reflection from Literature Review. *Proceedings of 8th International Conference on Education and New Learning Technologies*. Barcelona, Spain, pp.500-509

Dolgopolovas, V., et al, 2015. On Evaluation of Computational Thinking of Software Engineering Novice Students. *Proceedings* of IFIP *TC3 Working Conference "A New Culture of Learning: Computing and next Generations*, Vilnius, Lithuania

Gauvin, S., et al, 2015. Vizwik – visual data flow programming and its educational implications. In S. Carliner, C. Fulford & N. Ostashewski (Eds.), *Proceedings of EdMedia: World Conference on Educational Media and Technology* Montréal, Canada, pp. 1602-1608

Jonassen, D.H., 2010. Research Issues in Problem Solving. Proceedings of the 11th International Conference on Educational Research, New Educational Paradigm for Learning and Instruction, pp. 1-15

Koh, K.H., et al, 2014. Early Validation of Computational Thinking Pattern Analysis. *Proceedings of ITIGSE14*, Uppsala, Sweden

Miller, L.D., et al, 2014. Integrating Computational and Creative Thinking to Improve Learning and Performance in CS1. *Proceedings of SIGCSE'14*, Atlanta, Georgia, USA