Computers in Human Behavior xxx (2016) 1–14

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test

Marcos Román-González*, Juan-Carlos Pérez-González, Carmen Jiménez-Fernández

Universidad Nacional de Educación a Distancia (UNED), Faculty of Education, C/ Juan del Rosal, n° 14, C.P. 28040, Madrid, Spain

ARTICLE INFO

Article history:
Received 17 April 2016
Received in revised form
22 July 2016
Accepted 29 August 2016
Available online xxx

Keywords:
Computational thinking
Computational Thinking Test
Code literacy
Computer science education
Cognitive abilities
Cognitive assessment
Educational psychology
Primary education
Secondary education

ABSTRACT

Computational thinking (CT) is being located at the focus of educational innovation, as a set of problem-solving skills that must be acquired by the new generations of students to thrive in a digital world full of objects driven by software. However, there is still no consensus on a CT definition or how to measure it. In response, we attempt to address both issues from a psychometric approach. On the one hand, a Computational Thinking Test (CTt) is administered on a sample of 1,251 Spanish students from 5th to 10th grade, so its descriptive statistics and reliability are reported in this paper. On the second hand, the criterion validity of the CTt is studied with respect to other standardized psychological tests: the Primary Mental Abilities (PMA) battery, and the RP30 problem-solving test. Thus, it is intended to provide a new instrument for CT measurement and additionally give evidence of the nature of CT through its associations with key related psychological constructs. Results show statistically significant correlations at least moderately intense between CT and: spatial ability ($r = 0.44$), reasoning ability ($r = 0.44$), and problem-solving ability ($r = 0.67$). These results are consistent with recent theoretical proposals linking CT to some components of the Cattel-Horn-Carroll (CHC) model of intelligence, and corroborate the conceptualization of CT as a problem-solving ability.

1. Introduction

We live immersed in a digital ecosystem full of objects driven by software (Manovich, 2013). In this context, being able to handle the language of computers is emerging as an inescapable skill, a new literacy, which allows us to participate fully and effectively in the digital reality that surrounds us: it is about to *'program or be programmed'* (Rushkoff, 2010); it is about to be *'app-enabled or app-dependent'* (Gardner & Davis, 2013). The term 'code-literacy' has recently been coined to refer to the process of teaching and learning to read-write with computer programming languages (Prensky, 2008; Rushkoff, 2012). Thus, it is considered that a person is code-literate when is able to read and write in the language of computers and other machines, and to think computationally (Román-González, 2014). If code-literacy refers ultimately to a new read-write practice, computational thinking (CT) refers to the underlying problem-solving cognitive process that allows it. In other words, computer programming is the fundamental way that enables CT come alive (Lye & Koh, 2014); although CT can be transferred to various types of problems that do not directly involve programming tasks (Wing, 2008).

Given this current reality overrun by the digital, it is not surprising that there is renewed interest in many countries to introduce CT as a set of problem-solving skills to be acquired by the new generations of students; even more, CT is becoming viewed at the core of all STEM (*Science, Technology, Engineering, & Mathematics*) disciplines (Henderson, Cortina, & Wing, 2007; Weintrop et al., 2016). Although learn to think computationally has long been recognized as important and positive for the cognitive development of students (Liao & Bright, 1991; Mayer, 1988; Papert, 1980), as computation has become pervasive, underpinning communication, science, culture and business in our society (Howland & Good, 2015), CT is increasingly seen as an essential skill to create rather than just consume technology (Resnick et al., 2009). Thus, many governments around the world are incorporating computer programming into their national educational curricula. The recent decision to introduce computer science teaching from primary school onwards in the UK (Brown et al., 2013) and others European countries (European Schoolnet, 2015) reflects the growing recognition of the importance of CT.

* Corresponding author.
E-mail addresses: mroman@edu.uned.es (M. Román-González), jcperez@edu.uned.es (J.-C. Pérez-González), mjimenez@edu.uned.es (C. Jiménez-Fernández).

http://dx.doi.org/10.1016/j.chb.2016.08.047
0747-5632/© 2016 Elsevier Ltd. All rights reserved.

However, there is still little consensus on a formal definition of CT (Gouws, Bradshaw, & Wentworth, 2013; Kalelioğlu, Gülbahar, & Kukul, 2016), and disagreements over how it should be integrated in educational curricula (Lye & Koh, 2014). Similarly, there is a worrying vacuum about how to measure and assess CT, fact that must be addressed. Without attention to assessment, CT can have little hope of making its way successfully into any curriculum. Furthermore, in order to judge the effectiveness of any curriculum incorporating CT, measures that would enable educators to assess what the student has learned need to be validated (Grover & Pea, 2013).

In response, we attempt to address these issues from a psychometric approach. On the one hand, how our Computational Thinking Test (CTt) has been designed and developed is reported, as well as its descriptive statistics and reliability derived from an administration on a sample exceeding a thousand Spanish students. On the other hand, the criterion validity (Cronbach & Meehl, 1955) of the CTt is studied with respect to already standardized psychological tests of core cognitive abilities. Thus, this paper is aimed at providing a new instrument for measuring CT and additionally giving evidence of the correlations between CT and other well-established psychological constructs in the study of cognitive abilities.

1.1. *Computational thinking definitions*

We can distinguish between: a) generic definitions; b) operational definitions; c) educational and curricular definitions.

1.1.1. *Generic definitions*

One decade ago, in 2006, Jeanette Wing's foundational paper defined that CT "involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing, 2006, p. 33). Thus, CT's essence is thinking like a computer scientist when confronted with a problem. But this first generic definition has been revisited and specified in successive attempts over the last few years, still not reaching an agreement (Grover & Pea, 2013; Kalelioğlu et al., 2016). So, in 2011 Wing clarified, CT "is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Wing, 2011; on-line). One year later, this definition is simplified by Aho, who conceptualizes CT as the thought processes involved in formulating problems so "their solutions can be represented as computational steps and algorithms" (Aho, 2012, p. 832).

1.1.2. *Operational definitions*

In 2011, the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) developed an operational definition of computational thinking that provides a framework and common vocabulary for Computer Science K-12 educators: CT is a "problem-solving process that includes (but is not limited to) the following characteristics: formulating problems in a way that enables us to use a computer and other tools to help solve 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; generalizing and transferring this problem solving process to a wide variety of problems" (CSTA & ISTE, 2011; on-line).

1.1.3. *Educational-curricular definitions*

More than definitions in the strict sense, frameworks for developing CT in the classroom and other educational settings are mentioned next. So, from the UK, the organization Computing At School (CAS) states that CT involves six different concepts (logic, algorithms, decomposition, patterns, abstraction, and evaluation), and five approaches to working (tinkering, creating, debugging, persevering, and collaborating) in the classroom (CAS Barefoot, 2014). Moreover, from the United States, Brennan and Resnick (2012) describe a CT framework that involves three key dimensions: 'computational concepts' (sequences, loops, events, parallelism, conditionals, operators, and data); 'computational practices' (experimenting and iterating, testing and debugging, reusing and remixing, abstracting and modularizing); and 'computational perspectives' (expressing, connecting, and questioning). Table 1 shows a crosstab intersecting the CT framework dimensions (Brennan & Resnick, 2012) with the sampling domain of our Computational Thinking Test (CTt), which will be detailed in Sub-section 1.4.

1.2. *Computational thinking from the CHC model of intelligence*

While CT involves thinking skills to solve problems algorithmically (e.g., Brennan & Resnick, 2012; Grover & Pea, 2013), intelligence (i.e., general mental ability or general cognitive ability) involves primarily the ability to reason, plan and solve problems (Gottfredson, 1997). Even authors with alternative approaches to the conceptualization of intelligence recognize intelligence as a "computational capacity" or "the ability to process certain kinds of information in the process of solving problems of fashioning products" (Gardner, 2006, p. 503).

Within a cognitive approach, it has been recently suggested (Ambrosio, Xavier, & Georges, 2014) that computational thinking is related to the following three abilities-factors from the Cattell-Horn-Carroll (CHC) model of intelligence (McGrew, 2009; Schneider & McGrew, 2012):

- **Fluid reasoning (G_f)**, defined as: "the use of deliberate and controlled mental operations to solve novel problems that cannot be performed automatically. Mental operations often include drawing inferences, concept formation, classification, generating and testing hypothesis, identifying relations, comprehending implications, problem solving, extrapolating, and transforming information. Inductive and deductive reasoning are generally considered the hallmark indicators of G_f" (McGrew, 2009, p. 5)
- **Visual processing (G_v)**, defined as "the ability to generate, store, retrieve, and transform visual images and sensations. G_v abilities are typically measured by tasks (figural or geometric stimuli) that require the perception and transformation of visual shapes, forms, or images and/or tasks that require maintaining spatial orientation with regard to objects that may change or move through space" (McGrew, 2009, p. 5)
- **Short-term memory (G_{sm})**, defined as "the ability to apprehend and maintain awareness of a limited number of elements of information in the immediate situation (events that occurred in the last minute or so). A limited-capacity system that loses information quickly through the decay of memory traces, unless an individual activates other cognitive resources to maintain the information in immediate awareness" (McGrew, 2009, p. 5).

Therefore, it is expected that a computational thinking test should correlate with other already validated tests aimed at measuring cognitive abilities cited above.

Table 1
Crosstab intersecting CT framework (Brennan & Resnick, 2012) with the sampling domain of our CTt.

CT framework			CTt	
Dimension	Description	Components	Sampling domain	
Computational concepts	Concepts students employ as they program	Sequences	*	Computational concept addressed
		Loops	*	
		Events	–	
		Parallelism	–	
		Conditionals	*	
		Operators	*	
		Data	–	
Computational practices	Problem-solving practices that occurs in the process of programming	Experimenting and iterating	–	Required task
		Testing and debugging	/	
		Reusing and remixing	/	
		Abstracting and modularizing	/	
Computational perspectives	Students' understandings of themselves, their relationships to others, and the digital world around them	Expressing	–	–
		Connecting	–	
		Questioning	–	

*: Yes, /: Partly, -: No.

1.3. *Computational thinking assessment*

Count on validated measurement instruments is something necessary and valuable in any research area. However, for the moment, there is still a large gap of tests relating to CT that have undergone a comprehensive psychometric validation process (Mühling, Ruf, & Hubwieser, 2015). As Buffum et al. (2015) say: “developing assessments of student learning is an urgent area of need for the relatively young computer science education community as it advances toward the ranks of more mature disciplines such as physics that have established standardized assessments over time” (Buffum et al., 2015, p. 622). Anyway, we find in recent years some remarkable attempts to measure and assess CT in students from 5th to 10th grade, which are the ones of this paper's interest.

From the University of California, comes the instrument Fairy Assessment in Alice (Werner, Denner, Campe, & Kawamoto, 2012), which tries to measure the understanding and use of abstraction, conditional logic, algorithmic thinking and other CT concepts that middle school students utilize to solve problems. However, this instrument is designed *ad hoc* to be used in the context of programming learning environment Alice[1] (Graczyńska, 2010), and it has not been undergone to a psychometric validation process. The research group from Clemson University (South Carolina) provides a complementary perspective (Daily, Leonard, Jörg, Babu, & Gundersen, 2014; Leonard et al., 2015). These authors propose a kinesthetic approach to learning (‘embodied learning’) and assessment of CT with 5th and 6th grade students. To do so, they alternate activities for programming motion sequences (choreographies) in the Alice environment, with the representation of those same sequences in a physical-kinesthetic environment. The assessment tool also combines both settings, but its psychometric properties have not been reported.

Another interesting research line with middle school students is provided by the group from the University of Colorado. They work with students in the video-game programming environment AgentSheets[2] Within a first group of studies (Koh, Basawapatna, Bennett, & Repenning, 2010), these authors identify several Computational Thinking Patterns (CTP) that young programmers abstract and develop during the creation of their video-games; in this context, they design the Computational Thinking Patterns Graph, an automated tool that analyzes the games programmed by the students, and represents graphically how far each game has involved the different CTP when compared with a model. Within a second group of studies (Basawapatna, Koh, Repenning, Webb, & Marshall, 2011), the authors try to assess whether students are able to transfer the CTP acquired during video-game programming to a new context of scientific simulations programming. For this assessment, they develop CTP-Quiz instrument, whose reliability or validity have not been reported.

Similarly, from the Universidad Rey Juan Carlos (Madrid, Spain) Dr. Scratch[3] is presented (Moreno-León & Robles, 2015a, 2015b, 2014). Dr. Scratch is a free and open source web application designed to analyze, simply and automatically, projects programmed with Scratch[4] (Resnick et al., 2009), as well as it provides feedback that can be used to improve programming skills and to develop CT in middle school students (Moreno-León, Robles, & Román-González, 2015). In order to assign an overall CT score to the project, Dr. Scratch infers the programmer competence along the following seven CT dimensions: Abstraction and problem decomposition; Parallelism; Logical thinking; Synchronization; Flow control; User Interactivity; and Data representation. Therefore, Dr. Scratch is not strictly a cognitive test but a tool for the formative assessment of Scratch projects. Dr. Scratch is currently under validation process, although its convergent validity with respect to other traditional metrics of software quality and complexity has been already reported (Moreno-León, Robles, & Román-González, 2016).

Furthermore, we consider the Bebras International Contest,[5] a competition born in Lithuania in 2003 which aims to promote the interest and excellence of primary and secondary students around the world in the field of Computer Science from a CT perspective (Cartelli, Dagiene, & Futschek, 2012; Dagiene & Futschek, 2008; Dagiene & Stupuriene, 2014). Each year, the contest proposes a set of Bebras Tasks, whose overall approach is the resolution of real problems, significant for the students, through the transfer and projection of their CT over those. These Bebras Tasks are independent from any particular software or hardware, and can be administered to individuals without any prior programming experience.

[1] http://www.alice.org/index.php.
[2] http://www.agentsheets.com/.
[3] http://drscratch.org/.
[4] https://scratch.mit.edu/.
[5] http://www.bebras.org/.

For all these features, the Bebras Tasks have been pointed out as more than likely embryo for a future PISA (Programme for International Student Assessment) test in the field of Computer Science (Hubwieser & Mühling, 2014; Jašková & Kováčová, 2015). Anyway, the Bebras International Contest is, at the moment, an event for promoting CT, not a measuring instrument; among other considerations, because it is not composed by a stable and determined set of task-items, but a set that varies from year to year, with slight modifications along the countries. However, its growing expansion has aroused the interest of psychometry researchers, who have begun to investigate its possible virtues as a CT measurement instrument. Thus, descriptive studies about the student's performance on Bebras Tasks have been recently published, referred to the corresponding editions of the Bebras International Contest held in Germany (Hubwieser & Mühling, 2014, 2015), Italy (Bellettini et al., 2015), Taiwan (Lee, Lin, & Lin, 2014) or Turkey (Kalelioğlu, Gülbahar, & Madran, 2015). In all of them, and in most of the tasks studied, significantly higher performances in the male group in comparison with the female group were reported.

But strictly speaking, we only have knowledge of two tests aimed to middle/high school students which are being fully subjected to the psychometric requirements; both instruments are currently undergoing a validation process.

a. *Test for Measuring Basic Programming Abilities* (Mühling et al., 2015): it is designed for Bavarian students from 7th to 10th grade. This test is aimed at measuring the students' ability to execute a given program based on the so-called ‘flow control structures’; which are considered at the core of the CT for this age group: Sequencing (doing one step after another); Selection (doing either one thing or another); Repetition (doing one thing once and again). These control structures lead to the following CT concepts that are covered by the test: sequence of operations; conditional statement with (if/else) and without (if) alternative; loop with fixed number of iterations (repeat times); loop with exit condition (conditional loop: while or repeat until); and the nesting of these structures to create more complex programs.
b. *Commutative Assessment* (Weintrop & Wilensky, 2015): it is designed for high-school students, from 9th to 12th grade. This test is aimed at measuring students' understanding of different computational concepts, depending on whether they occur through scripts written in visual (block-based) or textual programming languages; which is a key transition to reach higher levels of code-literacy. The test has a length of 28 items, and it addresses the following CT concepts: conditionals; defined/fixed loops; undefined/unfixed loops; simple functions; functions with parameters/variables.

1.4. *Computational Thinking Test*

Overall, our Computational Thinking Test (CTt) has been developed following the practical guide to validating computer science knowledge assessments with application to middle school from Buffum et al. (2015), which is aligned with the international standards for psychological and educational testing (AERA, APA, & NCME, 2014). In addition, the CTt is consistent with other computational thinking tests under validation, aimed to middle/high school, such as the *Test for Measuring Basic Programming Abilities* (Mühling et al., 2015) or the *Commutative Assessment* (Weintrop & Wilensky, 2015), just described in Sub-section 1.3.

The CTt was initially designed with a length of 40 multiple choice items (version 1.0, October 2014). After a content validation process through twenty experts' judgement, this first version was refined to the final one (version 2.0, December 2014) of 28 items length (Román-González, 2015); which is built on the following principles:

- **Aim:** CTt aims to measure the development level of CT in the subject.
- **Operational definition of measured construct:** CT involves the ability to formulate and solve problems by relying on the fundamental concepts of computing, and using logic-syntax of programming languages: basic sequences, loops, iteration, conditionals, functions and variables.
- **Target population:** CTt is mainly designed and intended for Spanish students between 12 and 14 years old (7th and 8th grade); although it can be also used in lower grades (5th and 6th grade) and upper grades (9th and 10th grade).
- **Instrument Type:** multiple choice test with 4 answer options (only one correct).
- **Length and estimated completion time:** 28 items; 45 min.

Each item of the CTt[6] is designed and characterized according to the following five dimensions of the sampling domain:

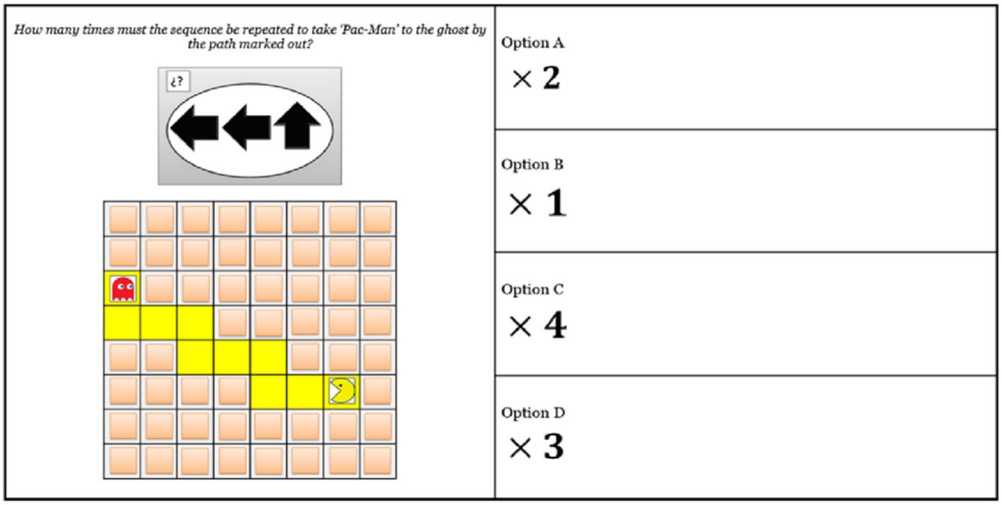
- **Computational concept addressed**: each item addresses one or more of the following seven computational concepts, ordered in increasing difficulty: Basic directions and sequences (4 items); Loops–repeat times (4 items); Loops–repeat until (4 items); If–simple conditional (4 items); If/else–complex conditional (4 items); While conditional (4 items); Simple functions (4 items). These ‘computational concepts’ are aligned with some of the CT framework (Brennan & Resnick, 2012; see Table 1) and with the CSTA Computer Science Standards for 7th and 8th grade (CSTA, 2011).
- **Environment-Interface of the item**: CTt items are presented in any of the following two environments-interfaces: ‘The Maze’ (23 items) or ‘The Canvas’ (5 items). Both interfaces are common in popular sites for learning programming such as Code.org (Kalelioğlu, 2015).
- **Answer alternatives style**: in each item, the response alternatives may be presented in any of these two styles: Visual arrows (8 items) or Visual blocks (20 items). Both styles are also common in popular sites for learning programming such as Code.org (Kalelioğlu, 2015).
- **Existence or non-existence of nesting**: depending on whether the item solution involves a script with (19 items) or without (9 items) nesting computational concepts (a concept embedded in another to a higher hierarchy level) (Mühling et al., 2015).
- **Required task:** depending on which of the following cognitive tasks is required for solving the item: Sequencing: the student must sequence, stating in an orderly manner, a set of commands (14 items); Completion: the student must complete an incomplete given set of commands (9 items); Debugging: the student must debug an incorrect given set of commands (5 items). This dimension is partially aligned with the aforementioned ‘computational practices’ from the CT framework (Brennan & Resnick, 2012; see Table 1).

The CTt is administered collectively and on-line, and it can be performed both via non-mobile or mobile electronic devices. Preliminary results about the CTt psychometric properties after its administration on a sample of 400 Spanish students (7th and 8th grade) have been already reported (Román-González, Pérez-

[6] Available at http://goo.gl/IYEKMB (Spanish version). Other forms and versions of CTt are available, free of charge, only for research purposes, from the first author.

[7] https://studio.code.org/s/20-hour.

[8] https://studio.code.org/s/course2.



How many times must the sequence be repeated to take 'Pac-Man' to the ghost by the path marked out?

Fig. 1. CTt, item 6: loops—repeat times; 'The Maze'; visual arrows; no-nesting; completion.

González, & Jiménez-Fernández, 2015). Examples of definitive CTt items translated into English are shown in Figs. 1–4; with their specifications detailed below.

2. Method

2.1. Participants

The CTt was administered on a total sample of 1,251 Spanish students, boys and girls from 24 different schools enrolled from 5th to 10th grade. The distribution of the subjects by gender, grade and age is shown in Table 2. From the total sample, 825 (65.9%) students belong to public schools, and 426 (34.1%) belong to private schools. Considering the device on which the CTt was administered, 1,001 students did it on a personal computer (80.0%) and 250 students (20.0%) did it so on a tablet. None of the subjects had prior programming formal experience when the CTt was administered.

The sampling procedure is not probabilistic and intentional. Depending on the reasons that led to sample the different subjects, these can be divided into four sub-samples.

- Sub-sample A ($n = 418$): it is composed of individuals belonging to classrooms that subsequently enrolled in the Accelerated Intro to CS Course from Code.org[7]
- Sub-sample B ($n = 48$): it is composed of individuals belonging to classrooms that subsequently enrolled in the CS Fundamentals Course 2 from Code.org[8]
- Sub-sample C ($n = 194$): it is composed of individuals belonging to classrooms that subsequently started to learn programming with Scratch.
- Sub-sample D ($n = 591$): it is composed of individuals belonging to classrooms that, although they did not subsequently start to learn programming, were interested on measuring the CT of the students.

In addition to our CTt, other standardized tests were administered concurrently to a part of the above subjects. Specifically for this paper, administrations of Primary Mental Abilities (PMA) battery ($141 \leq n \leq 166$) and RP30 problem-solving test ($n = 56$) are considered; all of these additional administrations are performed on subjects belonging to Sub-sample A. In the following Subsection 2.2, both standardized tests, PMA and RP30, are described.

2.2. Instruments

In order to address the criterion validation of the CTt, another two standardized instruments are administered: the Primary Mental Abilities (PMA) battery, and the RP30 problem-solving test; which are described next.

2.2.1. Primary Mental Abilities (PMA) battery

The PMA battery is aimed at appreciating the basic cognitive abilities through four different subtests, which allow an estimate of the main components of intelligence. This is a well-known measure of cognitive abilities (e.g., Hertzog & Bleckley, 2001; Quiroga et al., 2015) developed by Thurstone (1938). Its maximum administration time is 26 min, and it can be used from 10 years old onwards. The Spanish technical manual (TEA Ediciones, 2007) reports excellent reliability and validity coefficients about the four subtests. The PMA provides a precise measurement of the following cognitive abilities:

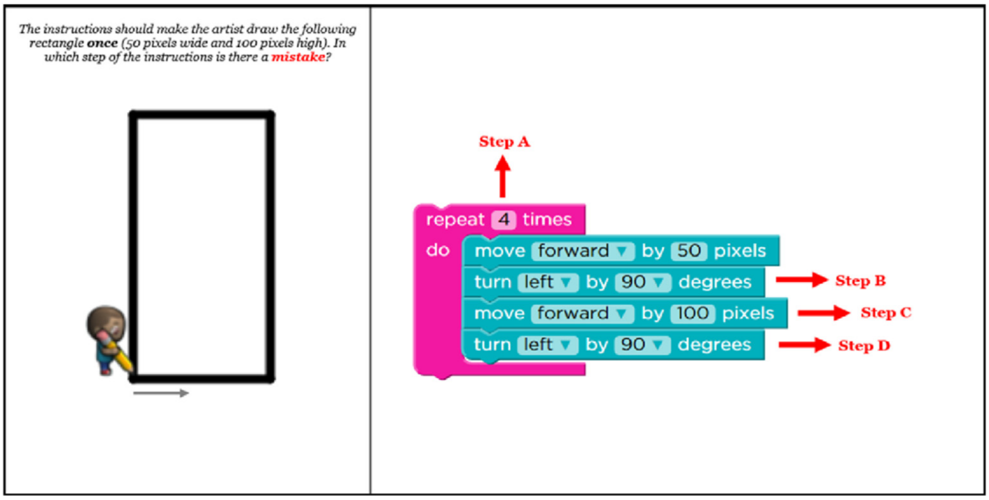
- **Verbal factor (PMA-V)**: Ability to understand and express ideas with words. PMA-V items involve selecting the accurate synonym of a word given.
- **Spatial factor (PMA-S)**: Ability to imagine and devise objects in two and three dimensions. PMA-S items involve selecting equal figures to a given model, after having been rotated.
- **Reasoning factor (PMA-R)**: Ability to solve logical problems, to understand and plan. PMA-R items involve selecting the option which continues a logical series given.
- **Numerical factor (PMA-N)**: Ability to handle numbers and quantitative concepts. PMA-N items involve checking mentally the sum of four two-digit numbers.

2.2.2. RP30 problem-solving test

RP30 problem-solving test is aimed to assess speed and flexibility in performing logical operations. Its maximum administration time is 17 min, and it can be used from 12 years old onwards. The Spanish technical manual (Seisdedos, 2002) reports excellent reliability values for RP30 ($r_{xx} > 0.90$; through the split-half method), as well as its criterion validity regarding to Changes Test of Cognitive Flexibility[9] ($r_{xy} = 0.38$) or to DAT[10]-Spatial ($r_{xy} = 0.34$).

[9] Test Cambios de Flexibilidad Cognitiva [Changes Test of Cognitive Flexibility] (Seisdedos, 1994).

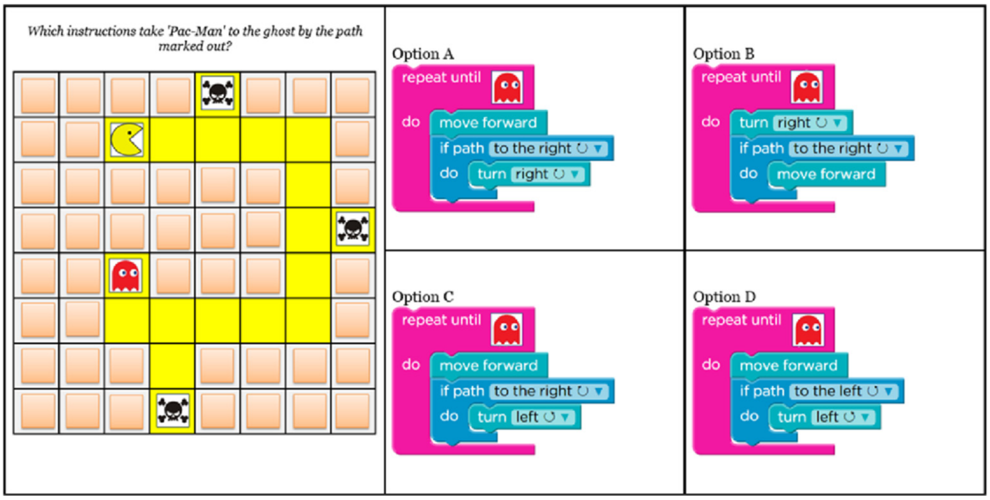
[10] DAT: Differential Aptitude Tests (Bennett, 1952).



The instructions should make the artist draw the following rectangle **once** (50 pixels wide and 100 pixels high). In which step of the instructions is there a ***mistake***?

```
repeat 4 times
do  move forward by 50 pixels
    turn left by 90 degrees
    move forward by 100 pixels
    turn left by 90 degrees
```

Fig. 2. CTt, item 7: loops—repeat times; 'The Canvas'; visual blocks; no-nesting; debugging.



Which instructions take 'Pac-Man' to the ghost by the path marked out?

Option A

```
repeat until 👻
do  move forward
    if path to the right ↻
    do  turn right ↻
```

Option B

```
repeat until 👻
do  turn right ↻
    if path to the right ↻
    do  move forward
```

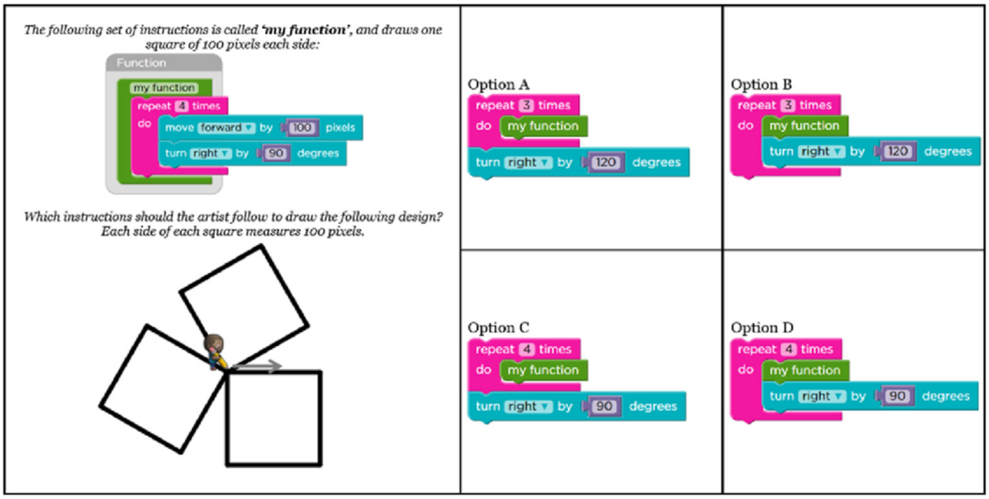
Option C

```
repeat until 👻
do  move forward
    if path to the right ↻
    do  turn left ↺
```

Option D

```
repeat until 👻
do  move forward
    if path to the left ↺
    do  turn left ↺
```

Fig. 3. CTt, item 14: loops—repeat until + If—simple conditional; 'The Maze'; visual blocks; yes-nesting; sequencing.



The following set of instructions is called ***'my function'****, and draws one square of 100 pixels each side:*

```
Function
my function
repeat 4 times
do  move forward by 100 pixels
    turn right by 90 degrees
```

Which instructions should the artist follow to draw the following design? Each side of each square measures 100 pixels.

Option A

```
repeat 3 times
do  my function
turn right by 120 degrees
```

Option B

```
repeat 3 times
do  my function
    turn right by 120 degrees
```

Option C

```
repeat 4 times
do  my function
turn right by 90 degrees
```

Option D

```
repeat 4 times
do  my function
    turn right by 90 degrees
```

Fig. 4. CTt, item 25: loops—repeat times + simple functions; 'The Canvas'; visual blocks; yes-nesting; sequencing.

Table 2
Distribution of the total sample (n = 1,251) by gender, grade and age.

					Gender		Total
					Boys	Girls	
Grade	5th	Age	10–11 y/o	Count	50	53	103
				% of Total	4.0%	4.2%	8.2%
	6th		11–12 y/o	Count	28	45	73
				% of Total	2.2%	3.6%	5.8%
	7th		12–13 y/o	Count	263	170	433
				% of Total	21.0%	13.6%	34.6%
	8th		13-14 y/o	Count	187	115	302
				% of Total	14.9%	9.2%	24.1%
	9th		14–15 y/o	Count	112	87	199
				% of Total	9.0%	7.0%	15.9%
	10th		15–16 y/o	Count	90	51	141
				% of Total	7.2%	4.1%	11.3%
Total				Count	730	521	1,251
				% of Total	58.4%	41.6%	100.0%

RP30 appreciates a high-level cognitive ability, by which a series of logical relations given are understood by the subject in order to determine whether these relations are met in several simple structures. RP30 is closely related to the non-verbal aspects of intelligence, it seems to be an important predictor to many school or professional activities, and it has been previously used as a proxy of the general mental ability (e.g., Barros, Kausel, Cuadra, & Díaz, 2014; Cáceres & Conejeros, 2011). RP30 items involve five structures in which the subject must decide whether the problem conditions are satisfied (Fig. 5). RP30 requires enough concentration as errors are penalized. It is considered that there are three cognitive abilities underlying RP30 performance (Seisdedos, 2002):

- **Reasoning**, due to the fact that the logical relations which may satisfy the structures must be previously understood by the subject.
- **Spatial ability**, as the subject must process the small circles and squares contained in each structure, in order to decide if the condition is satisfied.
- **Working memory**, which allows the subject to retain the given logical relation without need of constantly consulting it.

2.3. Procedure

Participating subjects in our research were enrolled in the elective subject of Computer Science, which is held twice a week (1 h each). Typically, the CTt was administered during the first of the two weekly classes. In the groups in which another standardized instrument was further administered, it was done during the second weekly class.

For the CTt collective administration, the Computer Science teacher followed the instructions which were sent by email in the week before, containing the URL to access the on-line test. The student's direct answers to the CTt items were stored in the Google Drive database linked with the instrument, which was subsequently downloaded as an Excel.*xls* file.

For the collective administration of the standardized instrument (PMA or RP30), students were previously signed in the on-line platform from the publishing house,[11] holder of these tests' commercial rights. Come the administration day, the subjects logged in the platform and performed the corresponding instrument, PMA battery or RP30 test (never both). Afterwards, from our administrator profile, we could download the subjects' results as an Excel.*xls* file.

Finally, all *.xls* files generated during data collection were exported to a single *.sav* file, which constitutes the data matrix under analysis with SPSS software (version 22). From this analysis arise the results exposed below.

3. Results and discussion

3.1. Descriptive statistics

Table 3 shows the main descriptive statistics of the CTt score (calculated as the sum of correct answers along the 28 items of the test) for the entire sample (n = 1,251).

In Fig. 6 (left), a histogram showing the distribution of the CTt score along the sample is depicted. As it can be seen, the aforementioned distribution fits remarkably the normal curve; although, given the very large size of the sample, the small existing maladjustments are penalized by the Kolmogorov-Smirnov test which rejects the null hypothesis of normality ($Z_{\text{k-s}} = 0.052$; $p < 0.01$).

In Fig. 6 (right), we show the success rate per item (expressed in per unit) or *item difficulty index*, that confirms empirically the progressive difficulty of the CTt; which was already anticipated by the experts during the content validation process (Román-González, 2015). The average success rate along the 28 items is $p = 0.59$ (medium difficulty); ranging from $p = 0.16$ (item 23; very high difficulty) to $p = 0.96$ (item 1; very low difficulty).

Summarizing, it can be stated that: a) the CTt score is almost normally distributed (i.e. symmetrically distributed; skewness ≈ 0), showing proper variability so that is possible to construct suitable scales (percentiles) for the target population; b) the CTt has an appropriate degree of difficulty (medium) for the target population, with an increasing difficulty along its items, as recommended in the design of abilities' tests (e.g., Carpenter, Just, & Shell, 1990; Elithorn & Telford, 1969).

3.1.1. Differences by grade

When the sample is segmented regarding to grade, the descriptive statistics shown in Table 4 are obtained. Specifically, results in Table 4 are split according to the Spanish educational system by the end of Primary Education (5th and 6th grade), the start of Secondary Education (7th and 8th grade), and the end of Secondary Education (9th and 10th grade).

Box plots for the CTt score split by aforementioned grades are shown in Fig. 7. The outlier belongs to a case from 6th grade, which obtained CTt score equal to 26 (i.e., ≈ 3 standard deviations above the mean of its reference group). The ANOVA test shows statistically significant differences in the CTt score regarding to grade ($F_{(2, 1248)} = 50.514$; $p < 0.01$). The *post-hoc* Tukey test additionally shows statistically significant differences between all possible pairs of means ($p < 0.01$).

Hence, it can be stated that the performance on the CTt increases as it does the grade; this result is consistent with our assumption that the CT is a problem-solving ability that it should be therefore linked to the cognitive development and maturity of the subjects (Ackerman & Rolfhus, 1999; Mayer, Caruso, & Salovey, 1999).

3.1.2. Gender differences

About the possible differential performance on the CTt regarding to gender, we find a statistically significant difference in the CTt score in favor of the male group ($t = 5.374$; $p < 0.01$), resulting an effect size measured through Cohen's d (Cohen, 1992) equal to 0.31 (Table 5); that can be considered as a low-moderate effect. If the aforementioned difference is analyzed along grades (Table 5), higher means in the CTt score are always found in the

[11] http://www.e-teaediciones.com/.

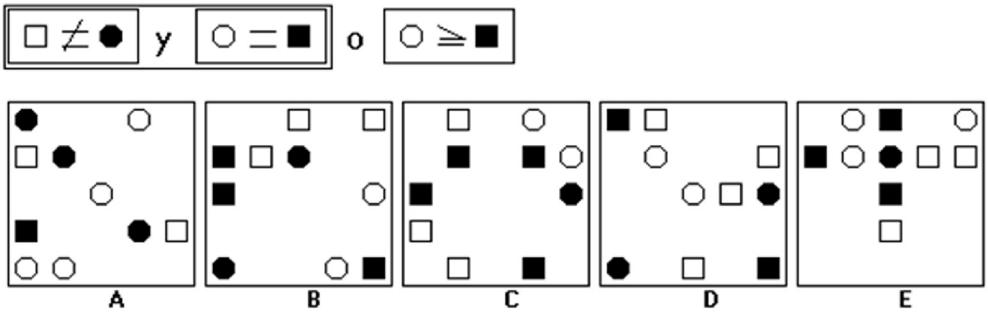


Fig. 5. Item example from the RP30 problem-solving test.

Table 3
Descriptive statistics of the CTt score for the entire sample (n = 1,251).

Mean		16.38
Std. Error of Mean		.136
Median		16.00
Mode		17
Std. Deviation		4.824
Variance		23.271
Skewness		.058
Kurtosis		−.446
Minimum		3
Maximum		28
Percentiles	10	10.00
	20	12.00
	25	13.00
	30	14.00
	40	15.00
	50	16.00
	60	17.00
	70	19.00
	75	20.00
	80	21.00
	90	23.00

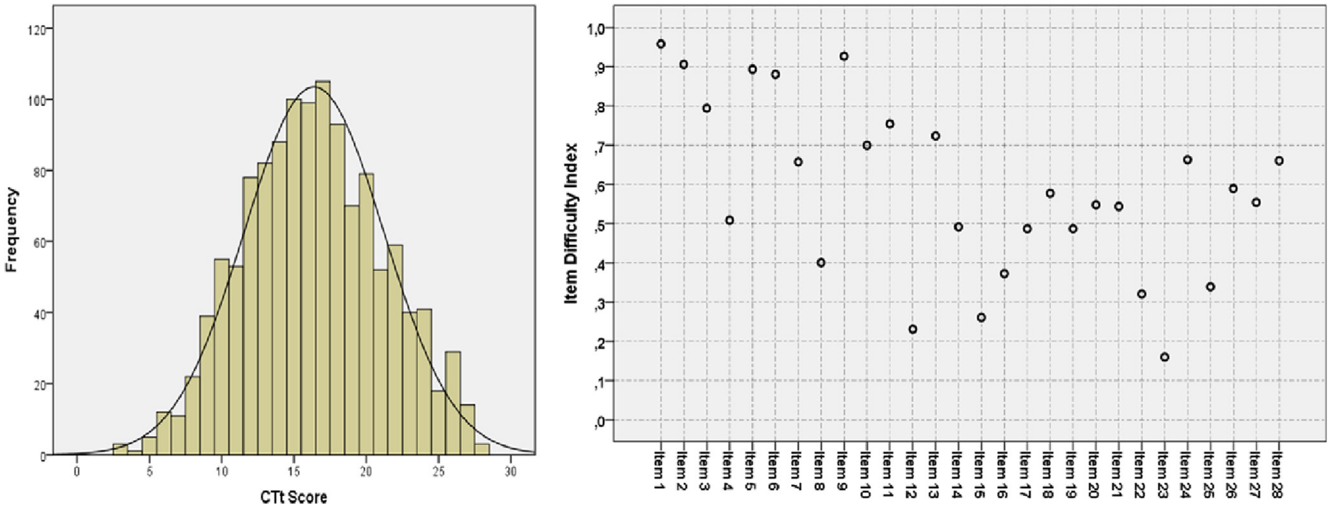


Fig. 6. Histogram of the CTt score (left); *Item Difficulty Index* for each item of the CTt (right).

male group, although these gender differences are only statistically significant from 7th and 8th grade ($t = 2.928$; $p < 0.01$) onwards; being more intense in 9th and 10th grade ($t = 3.451$; $p < 0.01$). Hence, it seems that there is a progressive gender gap over the CTt performance, as we advance along the grades (Fig. 8).

These gender differences are consistent with those found in previous research with Bebras Tasks, on which most of the investigations report higher yields of the male group, as described in Sub-section 1.3.

Table 4
Descriptive statistics of the CTt score split by grades.

		Grades		
		5th & 6th	7th & 8th	9th & 10th
n		176	735	340
Mean		13.76	16.24	18.05
Std. Error of Mean		.326	.167	.274
Median		14.00	16.00	18.00
Mode		15	18	17
Std. Deviation		4.330	4.519	5.049
Variance		18.746	20.419	25.496
Skewness		.125	.018	−.097
Kurtosis		−.148	−.453	−.577
Minimum		3	3	3
Maximum		26	27	28
Percentiles	10	8.00	10.00	12.00
	20	10.00	12.00	13.20
	25	11.00	13.00	14.00
	30	11.00	14.00	15.00
	40	13.00	15.00	17.00
	50	14.00	16.00	18.00
	60	15.00	17.00	19.00
	70	16.00	19.00	21.00
	75	16.75	20.00	22.00
	80	17.00	20.00	23.00
	90	20.00	22.00	25.00

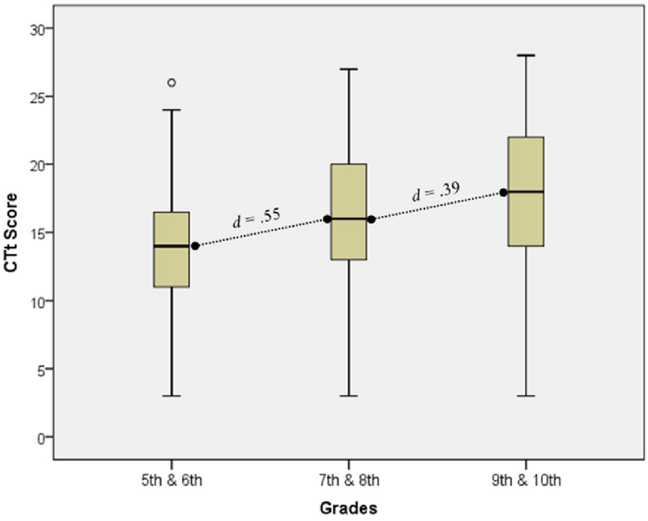


Fig. 7. Box plots for the CTt score split by grades.

3.2. Reliability

Reliability as internal consistency of the CTt, measured by Cronbach's Alfa is $\alpha = 0.793 \approx 0.80$; that can be considered as good reliability (Nunnally & Bernstein, 1994). When reliability is studied regarding to grade and administration's device (Table 6) we find that: a) reliability increases as it does the grade; which is coherent with the greater accuracy and consistency often shown by the answers coming from the upper grades' students (Anastasi, 1968); b) reliability increases when CTt is administered through mobile devices such as tablets, perhaps because these devices allow the subject to rotate the screen to one side and another, reducing the spatial cognitive load of the items and avoiding that the subjects commit unexpected errors on the same. This interpretation is supported by the results obtained when comparing the average yield in the CTt between subjects who performed it on a computer and subjects who did it so on a tablet: for instance, in 7th and 8th grade, $\overline{X}_{\text{computer}} = 16.01$ *vs.* $\overline{X}_{\text{tablet}} = 18.24$ ($t = 4.116$; $p < 0.01$; $d = 0.50$). In the future, if we achieve a larger sample of subjects who perform the CTt on tablet, and if these aforementioned significant differences between devices continue, it will be necessary to construct different scales for the CTt depending on the administration device.

3.3. Criterion validity

3.3.1. Relative to Primary Mental Abilities (PMA) battery

Correlations between the CTt and the various tests of the PMA battery are shown in Table 7. As it can be seen, the CTt has a positive statistically significant correlation ($p < 0.01$), moderately intense with PMA-R (reasoning factor) and PMA-S (spatial factor), and slightly intense with PMA-V (verbal factor). There is no statistically significant correlation between CTt and PMA-N (numerical factor). Corresponding scatter plots are shown in Fig. 9.

At this point, we perform a multiple linear regression over the CTt score (considered as the dependent variable) based on the PMA-V, PMA-S, PMA-R and PMA-N scores (considered as predictors). Table 8 summarizes de regression model, which is calculated through the 'enter' method. This regression model, based on the PMA battery, correlates $r = 0.540$ with the CTt; which means an adjusted $R^2 = 0.27$. That is, the 27.0% of the CTt scores' variance is explained from a linear combination of the primary mental abilities measured through the PMA battery. Normality of the regression model residuals was verified.

The regression model is able to explain, statistically significant, the differences in the CTt scores, as $F_{(4,\ 131)} = 13.457$ ($p < 0.01$). However, as shown in following Table 9 which contains the coefficients of the regression model, only PMA-S (spatial factor) and PMA-R (reasoning factor) are capable, specifically and statistically significant ($p < 0.01$), to explain differences in the dependent variable (CTt). The standardized coefficients of the model are, from highest to lowest value, $\beta_{(\text{PMA-S})} = 0.308$; $\beta_{(\text{PMA-R})} = 0.265$; $\beta_{(\text{PMA-V})} = 0.134$; $\beta_{(\text{PMA-N})} = -0.051$.

From our perspective, these results point out two important issues:

- Firstly, there is still a 73.0% of the CTt scores' variance that is not explained by the primary mental abilities measured through the PMA battery; which suggests certain independence of CT as a psychological construct, distinct from the traditional aptitudes.
- Secondly, the cognitive abilities with higher explanatory power about CT are reasoning ability and spatial ability; from both there is abundant evidence in the literature that reports certain male superiority. Regarding to the former, Kuhn and Holling (2009) recently report gender differences in reasoning ability favoring males in German students from 7th to 10th grade. Regarding to the latter, there are some meta-analysis that demonstrate higher male spatial ability, especially in tasks that involve mentally rotation of figures (Linn & Petersen, 1985; Voyer, Voyer, & Bryden, 1995). All the above could explain the higher yield of the boys in the CTt seen in Sub-section 3.1.2.

3.3.2. Relative to RP30 problem-solving test

Correlation between CTt and RP30 problem-solving test is shown in Table 10. As it can be seen, we find a positive, statistically significant, and moderately-strongly intense correlation ($r = 0.669$; $p < 0.01$) between both instruments. Corresponding scatter plot is shown in Fig. 10, such as the coefficient of determination $R^2 = 0.447$ (i.e., 44.7% of shared variance between both scores). Recall that RP30 test appreciates a high-level cognitive ability and it has been previously used as a proxy of the general mental ability. Our results

Table 5
Gender differences in CTt score.

			n	Mean	Std. Deviation	Student's *t*	Effect size Cohen's *d*
Entire sample		Boys	730	16.99	4.802	5.374^{**}	0.31
		Girls	521	15.52	4.727		
Grades	5th & 6th	Boys	78	14.40	4.185	1.765	0.27
		Girls	98	13.24	4.396		
	7th & 8th	Boys	450	16.62	4.463	2.928^{**}	0.22
		Girls	285	15.63	4.547		
	9th & 10th	Boys	202	18.82	5.115	3.451^{**}	0.38
		Girls	138	16.93	4.749		

$^{**}p < 0.01$.

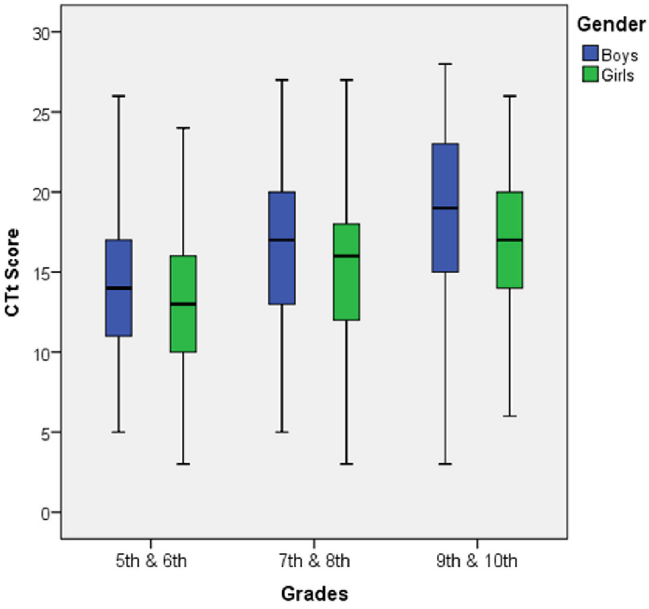


Fig. 8. Box plots for the CTt score split by gender and along grades.

Table 6
Reliability as internal consistency of the CTt.

		Cronbach's alpha	n		Cronbach's Alpha	n
Entire sample		0.793	1,251	Computer	0.786	1,001
				Tablet	0.817	250
Grades	5th & 6th	0.721	176	Computer	0.719	66
				Tablet	0.712	110
	7th & 8th	0.762	735	Computer	0.744	659
				Tablet	0.836	76
	9th & 10th	0.824	340	Computer	0.824	276
				Tablet	0.825	64

Table 7
Correlations (Pearson's *r*) between CTt and PMA battery.

	PMA-V	PMA-S	PMA-R	PMA-N
CTt	0.273^{**}	0.439^{**}	0.442^{**}	−0.157
PMA-V		0.225^{**}	0.334^{**}	−0.020
PMA-S			0.356^{**}	-0.164^{*}
PMA-R				−0.030

$141 \leq n \leq 166$; $^{**}p < 0.01$; $^{*}p < 0.05$.

indicate that CTt correlate more intensely with RP30 than with any of the primary mental abilities measured through PMA battery (Table 11). Hence, it seems that computational thinking could be fundamentally linked with general mental ability (particularly with fluid intelligence); and to a lesser extent with different cognitive aptitudes, such as logical reasoning and spatial ability.

When results of preceding Sub-sections 3.3.1 and 3.3.2 are triangulated, we find a clear consistency between the magnitude of the correlations CTt*PMA and CTt*RP30, and the expected composition of computational thinking from the CHC model of intelligence exposed in Sub-section 1.2 (Table 11). From our point of view, this is a powerful evidence of the criterion concurrent validity of our CTt, as well as an empirical confirmation of the computational thinking construct's composition proposed by Ambrosio et al. (2014).

4. Implications and limitations

The CTt has some strengths like: it can be administered in pretest conditions to measure the initial development level of CT in students without prior programming experience from 5th to 10th grade; it can be collectively administered so it could be used in massive screenings and early detection of students with high abilities (or special needs) for programming tasks; it can be utilized for collecting quantitative data in pre-post evaluations of the efficacy of curricula or programs aimed at fostering CT, which would be a desirable practice versus the qualitative approach that has been mostly used in the literature so far (Lye & Koh, 2014); and it could be used along academic and professional guidance processes towards STEM disciplines. However, the CTt also has obvious limitations and weaknesses:

- The CTt provides a static and decontextualized assessment. Therefore, we recommend to complement its use with other CT assessment tools designed from a formative perspective, such as Dr. Scratch (Moreno-León et al., 2015)
- In terms of CT framework (Brennan & Resnick, 2012), the CTt is overly focused on 'computational concepts', only covers 'computational practices' partly, and ignores 'computational perspectives'.
- The CTt only demands the projection of computational thinking over logical and visuospatial problems, such as solving mazes or designing geometric patterns. This implies a clear bias of the CTt, as computational thinking can also be projected over problems with different features, such as: modeling scientific simulations (Weintrop et al., 2016); algorithmic composition of computational music (Edwards, 2011); or digital interactive storytelling (Burke, 2012; Howland & Good, 2015). The latter authors report significantly higher values in the computational complexity of scripts written by girls from 7th and 8th grade in comparison with their male peers within narrative tasks; this

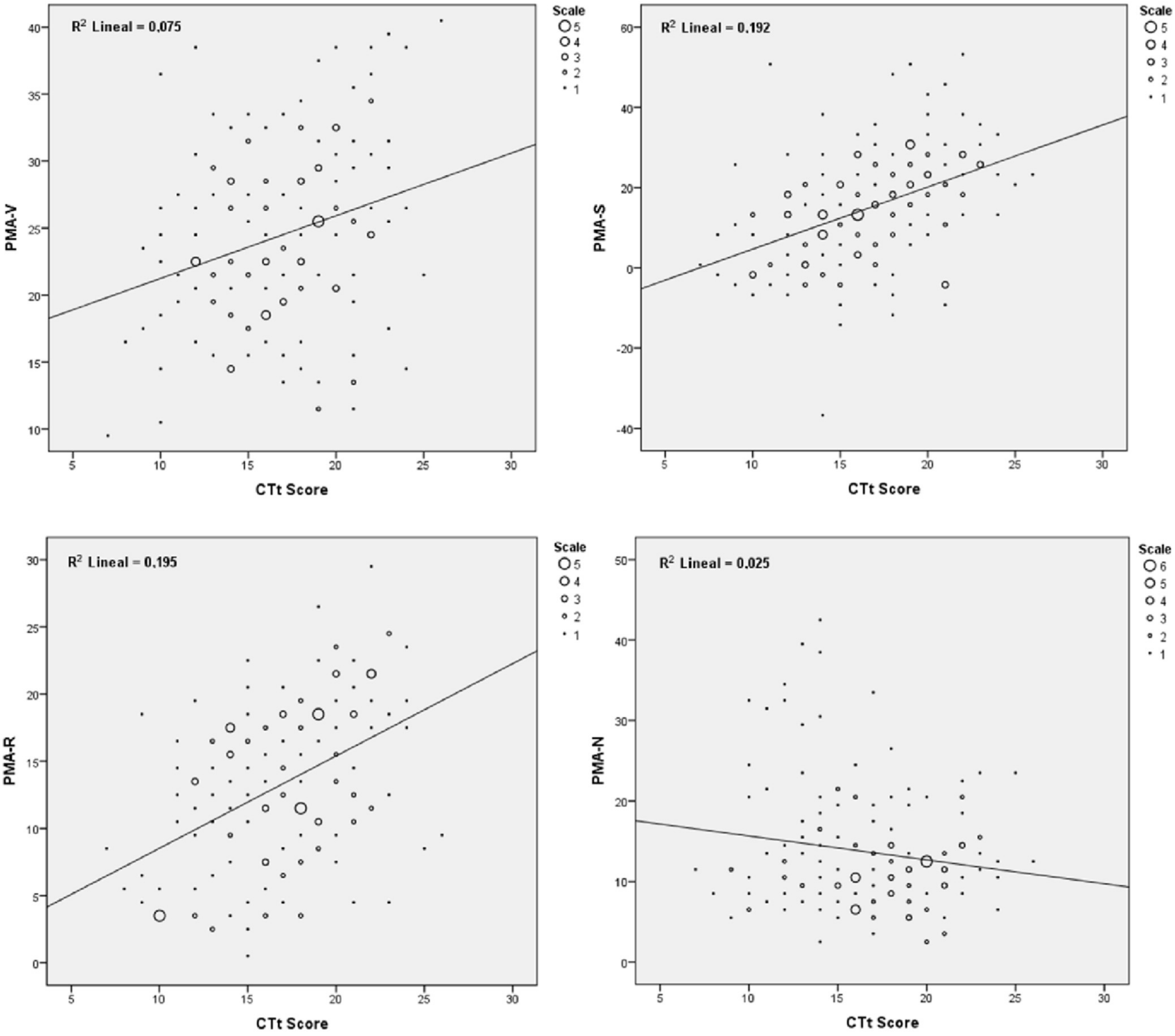


Fig. 9. Scatter plots between CTt and PMA battery.

result is consistent with the (slight) female superiority in tasks involving verbal ability reported in the classical literature (Hyde & Linn, 1988). It seems, therefore, that the direction of gender differences in CT may vary depending on the type of problems on which such ability is projected.

- Finally, as the CTt is entirely designed with multiple choice items, it might be measuring CT at its lower cognitive complexity levels ('recognize' and 'understand') (Gouws et al.,

Table 8
Summary of the regression model of the CTt onto the PMA subtests.

Model	*R*	*R* square	Adjusted *R* square	Std. Error of the estimate
1	0.540[a]	0.291	0.270	3.391

[a] Predictors: (Constant), PMA-V, PMA-S, PMA-R, PMA-N.

Table 9
Standardized coefficients[a] of the regression model of the CTt onto the PMA subtests.

Model		β Standardized coefficients	Student's *t*
1	(Constant)		9.006**
	PMA-V	0.134	1.715
	PMA-S	0.308	3.865**
	PMA-R	0.265	3.253**
	PMA-N	−0.051	−0.688

**p < 0.01.
[a] Dependent variable: CTt.

Table 10
Correlation between CTt and RP30 problem-solving test.

		RP30
CTt	Pearson correlation	0.669**
	Sig. (2-tailed)	0.000
	n	56

**p < 0.01.

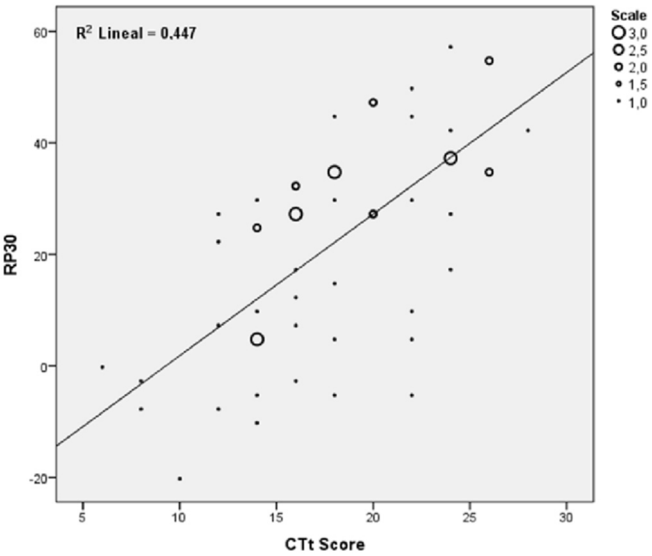


Fig. 10. Scatter plot between CTt and RP30.

Table 11
Correlations CTt*PMA and CTt*RP30, and contingency with G_f, G_v, and G_{sm} components of CHC model.

			PMA-N	PMA-V	PMA-S	PMA-R	RP30
CTt			−0.157	0.273**	0.439**	0.442**	0.669**
Selected components of the CHC model of intelligence	G_f	*Is it measured in the following instruments?*	No	No	No	Yes	Yes
	G_v		No	No	Yes	No	Yes
	G_{sm}		No	No	No	No	Yes

**p < 0.01.

2013). An instrument intended to measure CT also at higher levels of complexity ('Apply' and 'Assimilate') should include items which require not only recognize but also evoke the correct algorithm; as well as open complex problems whose resolution demands students to creatively transfer CT towards different domains.

5. Conclusions and further research

In this paper we have provided evidences of reliability and criterion validity of a new instrument for the assessment of CT and additionally we expanded our understanding of the CT nature through the theory-driven exploration of its associations with other established psychological constructs in the cognitive sphere. We have found expected positive small or moderate significant correlations ($0.27 < r < 0.44$) between CT and three of the four primary mental abilities of the Thurstone (1938) model of intelligence, as well as a high correlation ($r = 0.67$) between CT and problem-solving ability as a proxy of general mental ability. Our findings are consistent with recent theoretical proposals by Ambrosio et al. (2014) linking CT with some core elements of the CHC model of intelligence (McGrew, 2009), especially with respect to G_f (fluid intelligence) and G_v (visual processing). Furthermore, our results support the statement that CT is fundamentally linked with general mental ability; and also, though to a lesser extent, with specific cognitive aptitudes, such as inductive reasoning, spatial and verbal abilities. This corroborates the conceptualization of CT as a problem-solving ability (e.g., Brennan & Resnick, 2012; Lye & Koh, 2014; Wing, 2006, 2008); and it is consistent with the framework recently described by Kalelioğlu et al. (2016), in which CT is defined as a complex and high-order thinking skill involved in problem-solving processes.

Overall, it should be noted that this paper contributes to the establishment of the nomological net (Cronbach & Meehl, 1955) of computational thinking as an emergent scientific construct. Future research might expand this nomological net exploring how CT is related to other cognitive and computational variables, such as working memory, executive functions, or specific programming skills, among others. Finally, we plan the following further research lines concerning the CTt: a) convergent validity studies between CTt and other alternative CT assessment tools, such as Dr. Scratch (Moreno-León & Robles, 2015b, 2015a), Bebras Tasks (Dagiene & Stupuriene, 2014), the Test for Measuring Basic Programming Abilities (Mühling et al., 2015), or the Commutative Assessment (Weintrop & Wilensky, 2015); b) CTt adaptation and translation into other languages (already underway adaptations-translations into English and Portuguese), and replications of our psychometric studies in other populations; c) enhanced CTt versions including items that require the subject the evocation of algorithms and/or items that demand to project and transfer CT on scientific, narrative and musical relevant problems.

Acknowledgements

We thank Professor Dr. Kate Howland (University of Sussex) for collaborating in the adaptation and translation of CTt items from the Spanish language to the English language.

References

Ackerman, P. L., & Rolfhus, E. L. (1999). The locus of adult intelligence: Knowledge, abilities, and nonability traits. *Psychology and Aging, 14*(2), 314–330. http://dx.doi.org/10.1037/0882-7974.14.2.314.

AERA, APA, & NCME. (2014). *Standards for educational and psychological testing.* Washington, DC: AERA.

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

Ambrosio, A. P., Xavier, C., & Georges, F. (2014). Digital ink for cognitive assessment of computational thinking. In *Frontiers in education conference (FIE), 2014 IEEE* (pp. 1–7). http://dx.doi.org/10.1109/FIE.2014.7044237.

Anastasi, A. (1968). *Psychological testing* (3rd ed.). Oxford, England: Macmillan.

Barefoot, C. A. S. (2014). *Computational thinking [web page]*. Retrieved from. http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/.

Barros, E., Kausel, E. E., Cuadra, F., & Díaz, D. A. (2014). Using general mental ability and personality traits to predict job performance in three Chilean organizations. *International Journal of Selection and Assessment, 22*(4), 432–438. http://dx.doi.org/10.1111/ijsa.12089.

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 245–250. http://dx.doi.org/10.1145/1953163.1953241.

Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., & Torelli, M. (2015). How challenging are Bebras tasks? An IRT analysis based on the performance of Italian students. In *Proceedings of the 2015 ACM conference on innovation and technology in computer science education* (pp. 27–32). http://dx.doi.org/10.1145/2729094.2742603.

Bennett, G. K. (1952). *Differential aptitude tests [technical manual]*. New York: Psychological Corporation.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the american educational research association (vancouver: Canada)*. Retrieved from: http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf.

Brown, N. C. C., Kölling, M., Crick, T., Peyton Jones, S., Humphreys, S., & Sentance, S. (2013). Bringing computer science back into schools: Lessons from the UK. In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 269–274). http://dx.doi.org/10.1145/2445196.2445277.

Buffum, P. S., Lobene, E. V., Frankosky, M. H., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2015). A practical guide to developing and validating computer science knowledge assessments with application to middle school. In *Proceedings of the 46th ACM technical symposium on computer science education* (pp. 622–627). http://dx.doi.org/10.1145/2676723.2677295.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *The Journal of Media Literacy Education, 4*(2), 121–135. Retrieved from:http://eric.ed.gov/?id=EJ985683.

Cáceres, P. A., & Conejeros, M. L. (2011). Efecto de un modelo de metodología centrada en el aprendizaje sobre el pensamiento crítico, el pensamiento creativo y la capacidad de resolución de problemas en estudiantes con talento académico. *Revista Española De Pedagogía, 69*(248), 39–55. Retrieved from: http://www.jstor.org/stable/23766382.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the raven progressive matrices test. *Psychological Review, 97*(3), 404–431. http://dx.doi.org/10.1037/0033-295X.97.3.404.

Cartelli, A., Dagiene, V., & Futschek, G. (2012). Bebras contest and digital competence assessment: Analysis of frameworks. In A. Cartelli (Ed.), *Current trends and future practices for digital literacy and competence* (pp. 35–46). Hershey, PA: IGI Global.

Cohen, J. (1992). A power primer. *Psychological Bulletin, 112*(1), 155–159. http://dx.doi.org/10.1037/0033-2909.112.1.155.

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*(4), 281–302. http://dx.doi.org/10.1037/h0040957.

CSTA. (2011). *K–12 computer science standards*. Retrieved from http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf.

CSTA, & ISTE. (2011). *Operational definition of computational thinking for K–12 education*. Retrieved from: http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf.

Dagiene, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In R. T. Mittermeir, & M. M. Sysło (Eds.), *Informatics education-supporting computational thinking* (pp. 19–30). Berlin: Springer.

Dagiene, V., & Stupuriene, G. (2014). Informatics education based on solving attractive tasks through a contest. In *Proceedings of KEYCIT 2014–Key competencies in informatics and ICT* (pp. 51–62). Retrieved from: http://www.bebras.org/sites/default/files/documents/publications/Dagiene%2C%202014.pdf.

Daily, S. B., Leonard, A. E., Jörg, S., Babu, S., & Gundersen, K. (2014). Dancing Alice: Exploring embodied pedagogical strategies for learning computational thinking. In *Proceedings of the 45th ACM technical symposium on computer science education* (pp. 91–96). http://dx.doi.org/10.1145/2538862.2538917.

Ediciones, T. E. A. (2007). *PMA: Aptitudes Mentales Primarias (manual técnico)* [*PMA: Primary Mental Abilities. (Technical manual)*]. Madrid: TEA Ediciones.

Edwards, M. (2011). Algorithmic composition: Computational thinking in music. *Communications of the ACM, 54*(7), 58–67. http://dx.doi.org/10.1145/1965724.1965742.

Elithorn, A., & Telford, A. (1969). Computer analysis of intellectual skills. *International Journal of Man-Machine Studies, 1*(2), 189–209. http://dx.doi.org/10.1016/S0020-7373(69)80021-0.

European Schoolnet. (2015). *Computing our future*. Computer programming and coding: priorities, school curricula and initiatives across Europe [Technical report]. Retrieved from:http://www.eun.org/publications/detail?publicationID=661.

Gardner, H. (2006). On failing to grasp the core of MI theory: A response to Visser et al. *Intelligence, 34*(5), 503–505. http://dx.doi.org/10.1016/j.intell.2006.04.002.

Gardner, H., & Davis, K. (2013). *The App Generation: How today's youth navigate identity, intimacy, and imagination in a digital world*. New Haven: Yale University Press.

Gottfredson, L. S. (1997). Why g matters: The complexity of everyday life. *Intelligence, 24*(1), 79–132. http://dx.doi.org/10.1016/S0160-2896(97)90014-3.

Gouws, L. A., Bradshaw, K., & Wentworth, P. (2013). Computational thinking in educational activities: An evaluation of the educational game light-bot. In *Proceedings of the 18th ACM conference on innovation and technology in computer science education* (pp. 10–15). http://dx.doi.org/10.1145/2462476.2466518.

Graczyńska, E. (2010). ALICE as a tool for programming at schools. *Natural Science, 2*(2), 124–129. http://dx.doi.org/10.4236/ns.2010.22021.

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

Henderson, P. B., Cortina, T. J., & Wing, J. M. (2007). Computational thinking. *ACM SIGCSE Bulletin, 39*(1), 195–196. http://dx.doi.org/10.1145/1227504.1227378.

Hertzog, C., & Bleckley, M. K. (2001). Age differences in the structure of intelligence: Influences of information processing speed. *Intelligence, 29*(3), 191–217. http://dx.doi.org/10.1016/S0160-2896(00)00050-7.

Howland, K., & Good, J. (2015). Learning to communicate computationally with flip: A bi-modal programming language for game creation. *Computers & Education, 80*, 224–240. http://dx.doi.org/10.1016/j.compedu.2014.08.014.

Hubwieser, P., & Mühling, A. (2014). Playing PISA with bebras. In *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 128–129). http://dx.doi.org/10.1145/2670757.2670759.

Hubwieser, P., & Mühling, A. (2015). Investigating the psychometric structure of Bebras contest: Towards measuring computational thinking skills. In *International conference on learning and teaching in computing and engineering (LaTiCE)* (pp. 62–69). http://dx.doi.org/10.1109/LaTiCE.2015.19.

Hyde, J. S., & Linn, M. C. (1988). Gender differences in verbal ability: A meta-analysis. *Psychological Bulletin, 104*(1), 53–69. http://dx.doi.org/10.1037/0033-2909.104.1.53.

Jašková, L., & Kováčová, N. (2015). Bebras contest for blind pupils. In *Proceedings of the 10th workshop in primary and secondary computing education* (pp. 92–95). http://dx.doi.org/10.1145/2818314.2818324.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior, 52*, 200–210. http://dx.doi.org/10.1016/j.chb.2015.05.047.

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing, 4*(3), 583–596. Retrieved from http://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf.

Kalelioğlu, F., Gülbahar, Y., & Madran, O. (2015). A snapshot of the first implementation of Bebras international informatics contest in Turkey. In A. Brodnik, & J. Vahrenhold (Eds.), *Informatics in schools. Curricula, competences, and competitions* (pp. 131–140). Berna: Springer. http://dx.doi.org/10.1007/978-3-319-25396-1_12.

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. In *Visual languages and human-centric computing (VL/HCC), 2010 IEEE symposium* (pp. 59–66). http://dx.doi.org/10.1109/VLHCC.2010.17.

Kuhn, J., & Holling, H. (2009). Gender, reasoning ability, and scholastic achievement: A multilevel mediation analysis. *Learning and Individual Differences, 19*(2), 229–233. http://dx.doi.org/10.1016/j.lindif.2008.11.007.

Lee, G., Lin, Y., & Lin, J. (2014). Assessment of computational thinking skill among high school and vocational school students in Taiwan. In *World conference on educational multimedia, hypermedia and telecommunications* (pp. 173–180). Retrieved from: http://www.editlib.org/p/147499/.

Leonard, A. E., Dsouza, N., Babu, S. V., Daily, S. B., Jörg, S., Waddell, C., et al. (2015). Embodying and programming a constellation of multimodal literacy practices: Computational thinking, creative movement, biology, & virtual environment interactions. *Journal of Language and Literacy Education, 11*(2), 64–93. Retrieved from http://jolle.coe.uga.edu/wp-content/uploads/2015/10/Leonard_Template-Final-fixed-links.pdf.

Liao, Y. C., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research, 7*(3), 251–268. http://dx.doi.org/10.2190/E53G-HH8K-AJRR-K69M.

Linn, M. C., & Petersen, A. C. (1985). Emergence and characterization of sex differences in spatial ability: A meta-analysis. *Child Development, 56*(6), 1479–1498. http://dx.doi.org/10.2307/1130467.

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

Manovich, L. (2013). *Software takes command*. New York: Bloomsbury.

Mayer, R. E. (1988). *Teaching and learning computer programming: Multiple research perspectives*. New York: Routledge.

Mayer, J. D., Caruso, D. R., & Salovey, P. (1999). Emotional intelligence meets traditional standards for an intelligence. *Intelligence, 27*(4), 267–298. http://dx.doi.org/10.1016/S0160-2896(99)00016-1.

McGrew, K. S. (2009). CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence, 37*(1), 1–10. http://dx.doi.org/10.1016/j.intell.2008.08.004.

Moreno-León, J., & Robles, G. (2014). Automatic detection of bad programming habits in scratch: A preliminary study. *Frontiers in Education Conference (FIE), 2014 IEEE*, 1–4. http://dx.doi.org/10.1109/FIE.2014.7044055.

Moreno-León, J., & Robles, G. (2015a). *Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills. Scratch Conference* (pp. 12–15). Retrieved from http://jemole.me/replication/2015scratch/InferCT.pdf.

Moreno-León, J., & Robles, G. (2015b). Dr. Scratch: A web tool to automatically evaluate scratch projects. In *Proceedings of the 10th workshop in primary and secondary computing education* (pp. 132–133). http://dx.doi.org/10.1145/2818314.2818338.

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. RED. *Revista de Educación a Distancia, 46*. Retrieved from http://www.um.es/ead/red/46/moreno_robles.pdf.

Moreno-León, J., Robles, G., & Román-González, M. (2016). Comparing computational thinking development assessment scores with software complexity metrics. In *2016 IEEE global engineering education conference (EDUCON)* (pp. 1040–1045). http://dx.doi.org/10.1109/EDUCON.2016.7474681.

Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the 10th workshop in primary and secondary computing education* (pp. 2–10). http://dx.doi.org/10.1145/2818314.2818320.

Nunnally, J. C., & Bernstein, I. H. (1994). *Psychometric theory* (3rd ed.). New York: McGraw-Hill.

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

Prensky, M. (2008, January 13). *Programming is the new literacy [blog post]*. Retrieved fromhttp://www.edutopia.org/literacy-computer-programming.

Quiroga, M.Á., Escorial, S., Román, F. J., Morillo, D., Jarabo, A., Privado, J., & Colom, R. (2015). Can we reliably measure the general factor of intelligence (g) through commercial video games? Yes, we can! *Intelligence, 53*, 1–7. http://dx.doi.org/10.1016/j.intell.2015.08.004.

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

Román-González, M. (2014). Aprender a programar 'apps' como enriquecimiento curricular en alumnado de alta capacidad. Bordón. *Revista de Pedagogía, 66*(4), 135–155. http://dx.doi.org/10.13042/Bordon.2014.66401.

Román-González, M. (2015). Computational thinking Test: Design guidelines and content validation. In *7th annual international conference on education and new learning technologies (Barcelona: Spain)*. http://dx.doi.org/10.13140/RG.2.1.4203.4329.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2015). Test de Pensamiento computacional: Diseño y psicometría general [computational thinking test: Design & general psychometry]. In *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad, CINAIC2015 (Madrid: Spain)*. http://dx.doi.org/10.13140/RG.2.1.3056.5521.

Rushkoff, D. (2012, November 13). *Code literacy: A 21st-Century requirement [blog post]*. Retrieved from http://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff.

Rushkoff, D. (2010). *Program or be programmed*. New York: OR Books.

Schneider, W. J., & McGrew, K. S. (2012). The Cattell-Horn-Carroll model of intelligence. In D. Flanagan, & P. Harrison (Eds.), *Contemporary intellectual assessment: Theories, tests, and issues* (3rd ed., pp. 99–144). New York: Guilford.

Seisdedos, N. (1994). *CAMBIOS: Test de flexibilidad cognitiva (Manual técnico)* [*CHANGES: Cognitive Flexibility Test (Technical manual)*]. Madrid: TEA Ediciones.

Seisdedos, N. (2002). *RP30: Test de Resolución de Problemas (Manual técnico)* [*RP30: Problem-solving Test (Technical manual)*]. Madrid: TEA Ediciones.

Thurstone, L. L. (1938). *Primary mental abilities*. Chicago: University of Chicago Press.

Voyer, D., Voyer, S., & Bryden, M. P. (1995). Magnitude of sex differences in spatial abilities: A meta-analysis and consideration of critical variables. *Psychological Bulletin, 117*(2), 250–270. http://dx.doi.org/10.1037/0033-2909.117.2.250.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. http://dx.doi.org/10.1007/s10956-015-9581-5.

Weintrop, D., & Wilensky, U. (2015). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In *Proceedings of the eleventh annual international conference on international computing education research, ICER15* (pp. 101–110). http://dx.doi.org/10.1145/2787622.2787721.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on computer science education* (pp. 215–220). http://dx.doi.org/10.1145/2157136.2157200.

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

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions. Series A, Mathematical. *Physical, and Engineering Sciences, 366*(1881), 3717–3725. http://dx.doi.org/10.1098/rsta.2008.0118.

Wing, J. M. (2011). *Research Notebook: Computational thinking—what and Why? The link*. The magazine of the Carnegie Mellon University School of Computer Science. Retrieved from http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.