International Journal of Child-Computer Interaction 18 (2018) 47–58

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

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

Can computational talent be detected? Predictive validity of the Computational Thinking Test☆

Marcos Román-González [a,*], Juan-Carlos Pérez-González [a], Jesús Moreno-León [b], Gregorio Robles [c]

[a] Universidad Nacional de Educación a Distancia (UNED), Faculty of Education, C/ Juan del Rosal, nº 14, C.P. 28040, Madrid, Spain
[b] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), C/ Torrelaguna, nº 58, C.P. 28027, Madrid, Spain
[c] Universidad Rey Juan Carlos (URJC), ETSI Telecomunicación, Departamental III, Camino del Molino s/n, C.P. 28943, Fuenlabrada, Madrid, Spain

HIGHLIGHTS

- Computational Thinking Test (CTt) has predictive validity with respect to academic performance.
- CTt has predictive validity with respect to coding achievement in middle school.
- CTt has predictive validity to distinguish between computational top and regular thinkers.
- To accelerate from 'block' to 'text' based programming languages might be a gender biased criterion to define computational talent.
- 'Computationally talented' students might accelerate in the CS Education standards between 1–2 years compared to the regular learners.

ARTICLE INFO

Article history:
Received 10 February 2017
Received in revised form 2 April 2018
Accepted 28 June 2018
Available online 6 July 2018

Keywords:
Computational thinking
Computational thinking test
Computational talent
Cognitive assessment
Predictive validity

ABSTRACT

Computational thinking (CT) is arising as a set of problem-solving skills that must be acquired by the new generations of students to fully understand and participate in our computer-based world. However, from a psychometric approach, we are still at an early stage regarding the definition and assessment of CT as a psychological variable. One way to advance in this area is to investigate whether 'computationally talented' students (i.e., 'computational top thinkers') can be detected even before learning to code; and, if so, how to teach them properly to fully develop their high-computational ability. This paper presents several empirical concatenated studies about the predictive validity of the Computational Thinking Test (CTt), which is administered on a sample of 314 middle school Spanish students ($n = 314$). We report the predictive validity of the CTt, conducted at the beginning of the quarter, with respect to academic performance (Informatics, Mathematics, and Language) and learning analytics in a Code.org course collected at the end of the quarter. We also analyze the predictive validity of the CTt to early distinguish between 'computational regular thinkers' and 'computational top thinkers' (i.e., those who spontaneously accelerated from the 'block-based' programming environment of Code.org to the 'text-based' one of Khan Academy). Finally, we perform a case study over two of the students categorized as 'computational top thinkers', in which one of their coding products written in Processing JavaScript is described. Our results demonstrate that 'computationally talented' students can be detected in middle school, and that these subjects have the ability to accelerate in the Computer Science Education standards between 1 and 2 years compared to the regular learners. This could have major implications on the emerging computing curricula, which should take into account these individual differences in computational ability and 'learning-how-to-code' speed to ensure an appropriate progression for every student.
© 2018 Elsevier B.V. All rights reserved.

1. Introduction

We are living in a computer-based world full of objects driven by software [1]. In this scenario, where computing has become ubiquitous and underpins every life dimension, to handle the language of computers is emerging as indispensable to fully participate and thrive in our digital societies [2]. Therefore, computer

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.ijcci.2018.06.004.

* Corresponding author.
E-mail addresses: mroman@edu.uned.es (M. Román-González), jcperez@edu.uned.es (J.-C. Pérez-González), jesus.morenol@mecd.es (J. Moreno-León), grex@gsyc.urjc.es (G. Robles).

https://doi.org/10.1016/j.ijcci.2018.06.004
2212-8689/© 2018 Elsevier B.V. All rights reserved.

programming is becoming considered as a new literacy ('code literacy') [3,4]. If code literacy refers ultimately to an emerging read–write practice, computational thinking (CT) refers to the underlying problem-solving process that allows it [5]. In other words, computer programming is the fundamental way that enables CT to come alive [6], a key tool for supporting the cognitive tasks involved in CT, and a demonstration of computational competencies [7]. However, CT can be projected on different kinds of problems that may not involve directly programming tasks [8,9].

Given this current reality, it is not surprising that CT is arising as a key set of problem-solving skills that must be acquired by the new generations of digital learners, and many countries around the world have decided to incorporate computer programming and CT in their curricula [10,11]. Nevertheless, from a psychometric approach, we find several reasons to affirm that we are still at an early stage in regards to the definition and assessment of CT as a psychological construct (i.e., as a psychological variable):

1. There is still a lack of consensus about a definition of CT [12–14].
2. There is still a worrying vacuum of standardized tests aimed to measure CT that have undergone a full validation process [15]. As it has been pointed out by [16], there is an urgent need for having standardized tools in the Computer Science (CS) Education community.
3. Related to the previous point, since there is a lack of CT validated tests, the nomological network of CT (i.e., the correlations between CT and other key psychological constructs) has not been fully defined by the researchers yet [17].
4. Finally, resulting from the above, we are missing out the benefits of having validated tools for assessing CT. For example, one benefit could be to evaluate confidently if a CS curriculum has been effective to improve the CT of the students (e.g., through a pre-post quasi-experimental research design) [6]. Another one could be to early detect students with special needs in CT ability, even before they start to learn computer programming, in order to implement proper and personalized educational interventions.

Since there is still no consensus on a CT definition, it is necessary to make explicit that we have assumed for our work the one given by Aho, who conceptualizes CT as the thought processes involved in formulating problems so "their solutions can be represented as computational steps and algorithms" [18]. It could be argued that this definition over-identifies CT with algorithmic thinking, and gives a narrow and reductionist view of CT. However, we selected Aho's general definition because it is useful to subsequently derivate specific operational ones from where CT assessment tools can be designed, which is essential for our psychometric research approach. In other words, algorithmic thinking seems to be a central part of CT, and specifically the most susceptible part for being measured. Furthermore, relevant authors have recently synthesized CT as "the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers)" [14]. We assume this statement too, and remind that, although CT ability can be mainly demonstrated by means of computer programming, it can also be expressed within unplugged contexts [19,20].

In any case, if the aforementioned issues are not addressed soon, CT is at risk of not being seriously considered in the context of educational psychology, and even more important, CT might have an unsuccessful way into the K-12 curricula [7,12]. In response, one of the major recent attempts to design and validate a CT assessment tool is the *Computational Thinking Test* (CTt) [21], which has demonstrated to be valid and reliable (internal consistency α=.80; test–retest stability r_{xx}=.70) in Spanish middle school subjects [17]. The administration of the CTt to over one thousand middle school students without prior formal programming experience showed a quasi-normal distribution with large variability [17]. Consequently, in order to design adequate and personalized educational actions, the individual differences in computational ability that seem to exist should be deeply described. In this paper, we will focus on the right tail of the distribution, that is, on students with high computational ability.

Furthermore, the criterion validity of the CTt with respect to other cognitive and non-cognitive psychological variables has been already reported. Regarding the former [17], statistically significant correlations at least moderately intense between CT and problem-solving ability (r = .67), spatial ability (r = .44), and reasoning ability (r = .44) have been found. Regarding the latter [22,23], published results show statistically significant correlations between CT and three of the dimensions from the 'Big Five' model of human personality: Openness to Experience (r = .41), Extraversion (r = .30), and Conscientiousness (r = .27). In summary, these results have contributed to start the definition of the nomological network of CT, and to empirically corroborate the conceptualization of CT as a problem-solving ability linked with general mental ability ('g'). Further description about the CTt will be given in paragraph 3.2.1.

However, the predictive validity of the CTt has not been studied yet. Then, our research questions are:

- RQ_1: *What is the predictive validity of the CTt regarding academic performance?*
- RQ_2: *What is the predictive validity of the CTt regarding coding achievement?*
- RQ_3: *What is the predictive validity of the CTt to early detect 'computationally talented' students?*

2. Background

In this section, we revise the three main issues that can contextualize our further results. Firstly, we describe other cognitive correlates of academic performance and programming proficiency. Secondly, we look in the literature for evidence supporting the existence and nature of computational talent. Finally, we consider the ways in which acceleration is specified in the field of CS education for middle school ages.

2.1. Cognitive correlates of academic performance and coding achievement

It is commonly proven that cognitive abilities (e.g., reasoning, processing speed, working memory) and executive function highly correlates (usually with r>.40) with academic performance along K-12 education [24–27]. This fact is especially evident when one studies the relations between general mental ability ('g') and general academic performance operationalized such as standardized test scores (SAT/ACT) or school grade-point average (GPA) [28–30]. Additionally, it is usually found in the literature an average correlation of .50<r<.85 approximately between cognitive abilities and academic performance when the latter refers to math and science [31,32]. In contrast, the magnitude of correlation is slightly lower but still high with respect to academic performance in language [25,33].

When it comes to the cognitive correlates of coding achievement, most of the literature has been focused on university or, to a lesser extent, on high school students in the context of learning textual programming languages (e.g., Pascal, BASIC or COBOL) [34–39]. The main cognitive predictors of programming achievement found in literature are: general cognitive ability [38]; general aptitudes [39]; level of cognitive development [35] as defined by

Table 1
Number of results in Google Scholar.

Search input term	Number of results	
	February 2017	October 2017
"musical talent"	24,300	27,600
"artistic talent"	23,100	23,900
"mathematical talent"	5,780	5,870
"sport talent"	1,330	1,510
"programming talent"	792	851
"computational talent"	15	20

Piaget; mathematics reasoning and verbal ability [35]; working-memory, some specific word problem solving abilities (i.e., problem identification and sequencing of elements), and some learning style measures (i.e., asking for hints and running programs) [36]; and spatial visualization [37]. However, there is a lack of research regarding cognitive correlates of coding achievement in middle school ages and in the context of visual programming languages (e.g., Blockly or Scratch).

2.2. Computational talent

The detection of gifted and talented students has been matter of study in general [40–42], and particularly in many disciplines, such as music [43] or mathematics [44], or for specific groups, such as minorities [45] and disadvantaged [46], among others.

However, to the knowledge of the authors, no specific research has been performed to identify gifted and talented middle school students in the field of coding or programming. Maybe the one that comes closest is a group of scholars that has devoted some efforts in the identification of talented students in areas that are relatively nearby to 'pure' programming tasks, such as three-dimensional computer graphics programming [47], User-Centered Design (UCD) methodologies [48], and design [49]. In the aforementioned studies, the critical variables to identify these talents are cognitive load (i.e., mental effort being used in the working-memory within a problem-solving context), spatial ability, and imagination personality factor (close to the 'Openness to Experience' dimension from the 'Big Five' model). Thus, it seems clear that 'computational talent' is still an area of research to be explored, as suggested by the data shown in Table 1.

Then, we wonder which kind of behaviors might characterize and distinguish 'computationally talented' students. So, in adult ages, it is widely spread that for instance the participation in Free/Open Source Software (FOSS) communities is seen as a "signaling of talent" by developers [50] that improves their careers and future employment opportunities. Nevertheless, if we descend to middle school ages, an important computational talent signal might be the ability to accelerate from visual 'block-based' to 'text-based' programming environments [51–54], as evidenced in the next subsection.

2.3. Acceleration in programming: from 'block-based' to 'text-based' environments

Educational acceleration, content or grade-based, has been claimed to be advantageous for high abilities students, because it helps to increase academic achievement of those students who were accelerated [55,56] and it saves time and frees up other resources [57]. The selection of students with the potential to receive an educational acceleration is generally based on the application of a battery of several cognitive tests in which individuals must obtain a score above the 95th percentile to be recruited [58,59]. Since there is still a lack of CT assessment tools that can already establish this cut-off points accurately, it is necessary to proceed in the opposite way: we must find a well-grounded criterion of what acceleration means in the context of CS education and CT high-abilities; and then check out if our CT assessment tool (i.e., specifically, our *Computational Thinking Test*), can be validated regarding that predictive criterion.

When we investigate how content-based acceleration is specified in the field of CS education for middle school ages, we find evidence that a faster and effective transition from 'block-based' to 'text-based' programming environments might be at the core of it (e.g., see curriculum standards from UK [60], Australia [61], USA [62], or the Region of Madrid in Spain [63]). Hence, the analysis of different national curricula and standards shows that, even if programming with visual environments is accepted and promoted in early years, the use of text-based programming languages, as the ones used by professional software developers, is the final educational goal in most countries at the end of K-12 level.

Different investigations show that 'block-based' programming languages (e.g., Blockly, Scratch, Snap) are viewed as easier than 'text-based' ones (e.g., JavaScript, Python, C++) by students due to "the ease of the drag-and-drop composition, the lack of needing to memorize commands, and fact that in the tools we used, blocks were closer to natural language than text-based programs, making them easier to read" [51]. Nonetheless, learners also highlight some issues of the 'block-based' languages, such as lack of "expressive power, and challenges in authoring larger, more sophisticated programs" [52]. Equivalently, Howland and Good identified problems that regular middle school students face when dealing with 'text-based' programming languages, such as intimidation, frustration and need for substantial help from experts [64,65]. Therefore, and according to Kölling et al., the transition from visual to 'text-based' languages "presents a significant hurdle which may not be easy to overcome, for learners and teachers alike" [53], and it seems to be a key signal to detect computationally talented students in middle school. In any case, it is assumed by relevant authors that moving faster from 'block-based' to 'text-based' programming languages might be a fundamental content-based acceleration in middle school CS Education [54].

Summarizing our research approach, we have assumed that computer programming is the main demonstration of CT ability, and we have also found evidence in literature showing that accelerating from 'block-based' to 'text-based' programming languages can be a well-grounded criterion to categorize students as 'computationally talented' in middle school. Therefore, our research question RQ_3 will be investigated in these terms along Section 4.4.

3. Method

3.1. Participants

Our initial sample is composed by $n = 314$ students who belong to eleven different Spanish middle schools. Table 2 shows the distribution of the sample regarding gender, school year[1] and age. All the students of this sample were enrolled in the elective subject of 'Informatics', and they were all recruited in the context of a broader evaluation, through a quasi-experimental design, of the *"Intro to CS Course"*[2] of Code.org. More specifically, these 314 students correspond to the experimental groups of the aforementioned evaluation.

Thus, the sample is entirely composed by individuals who were initially interested in CS, and the gender bias in favor of males that can be observed in Table 2 is due to the higher number of boys who selected 'Informatics' as an elective subject for the academic year. The recruitment process followed IRB ethical guidelines in educational research (i.e., the participants were informed about the objectives and contents of the study, both students and parents gave their informed consent to participate in the investigation, and all the data collection and processing was subsequently anonymized).

1 Following the UK Education System.
2 Available at https://studio.code.org/s/20-hour.

Table 2
Distribution of the sample ($n = 314$) by gender, school year and age.

					Gender		Total
					Boys	Girls	
Year	8	Age	12-13 y/o	Count	142	79	221
				% of Total	45.2%	25.2%	70.4%
	9		13-14 y/o	Count	71	22	93
				% of Total	22.6%	7.0%	29.6%
Total				Count	213	101	314
				% of Total	67.8%	32.2%	100.0%

3.2. Instruments

3.2.1. Computational Thinking Test (CTt)

Overall, the *Computational Thinking Test*[3] (CTt) has been designed according to the guidelines for validating CS knowledge assessments with application to middle school from Buffum et al. [16]. These guidelines are aligned with the international standards for psychological and educational testing [66]. The CTt is inspired by the general definition of Aho [18], which was already quoted in the introduction section. More specifically, the CTt is based on the following operational definition: "CT involves the ability to formulate and solve problems by relying on the fundamental concepts of computing, and using the inherent logic of programming languages: basic sequences, loops, iteration, conditionals, functions and variables" [17,21].

In another vein, the CTt is a multiple-choice test of 28 items of length, which are administered in a maximum time of 45 min. Each item of the CTt is presented either in a 'maze' or in a 'canvas' interface (see Figs. 1–3), and is designed according to the following three dimensions:

- **Computational concept(s) addressed:** each item addresses one or more of the following seven computational concepts, ordered in increasing difficulty: Basic directions and sequences; Loops—repeat times; Loops—repeat until; If—simple conditional; If/else—complex conditional; While conditional; Simple functions. These 'computational concepts' are progressively nested along the test, and are aligned with the CSTA Computer Science Standards for the Years 8 and 9 [62].
- **Style of response alternatives:** in each item, responses are presented in any of these two styles: 'visual arrows' or 'visual blocks'.
- **Required cognitive task:** depending on which cognitive task is required for solving the item: 'sequencing' ≈ stating in an orderly manner a set of commands, 'completion' of an incomplete set of commands, or 'debugging' an incorrect set of commands.

Examples of CTt items translated into English are shown in Figs. 1–3; with their specifications detailed below.

Within the current panorama of CT assessment tools, the CTt can be categorized as a summative-aptitudinal instrument. Other types that have been identified in K-12 education [67] are: CT formative–iterative tools, such as *Dr. Scratch*[4] [68] or the *Computational Thinking Patterns CTP-Graph* [69]; CT skill-transfer tools, such as the *Bebras Tasks* [70] or the *CTP-Quiz* [71]; CT learning analytics tools, such as the data-driven one from Grover [72]; CT perceptions-attitudes scales, such as the *Computational Thinking Scales* [73]; and CT vocabulary assessments [74]. The convergent validity of the CTt with respect to *Dr. Scratch* (r = .53) and the *Bebras Tasks* (r = .52) has been already reported to the scientific community [67].

[3] A sample copy of the test is available at https://goo.gl/xHfr5V.
[4] http://www.drscratch.org/.

3.2.2. Academic performance

In Spanish middle schools, the academic performance of the students is expressed through a grading system that ranges from 0 to 10 in each of the subjects. The academic grades that we collected were gathered and submitted directly by the teachers, not self-reported by the students (i.e., we avoided the probable bias that entails self-reported data). Specifically, we required the teachers to submit the grades of the students in three subjects: Informatics, Mathematics, and Language. From these three inputs, we calculated the grade-point average (GPA).

3.2.3. Learning analytics of Code.org and Khan Academy

The courses of Code.org, such as the *"Intro to CS Course"*, are mainly composed by a learning path of programming puzzles (i.e., close-ended coding problems). The platform provides virtual classrooms that track the progression of the students. This learning analytics system allowed us to download two achievement indicators for each participating student: 'Completed Levels' and 'Lines of Code'.

Furthermore, as it will be detailed in Section 4.5, during the case study over two 'computationally talented' students within the *"Intro to JS: Drawing & Animation"*[5] course of Khan Academy, we collected the following learning analytics that can be considered as achievement indicators: 'Total minutes' (i.e., total of minutes spent by the student in the course, which is split in 'Video minutes' ≈ spent on viewing tutorials, and 'Skill minutes' ≈ spent on coding tasks); 'Badges earned'; and 'Points earned'.

3.3. Procedure

At the beginning of the quarter (T_1), we administered the CTt to our initial sample ($n = 314$) in pre-test condition (i.e., any of the subjects had prior formal programming experience). Then all students, within the subject of Informatics, enrolled in the *"Intro to CS Course"* of Code.org during 12 weeks. At the end of the quarter (T_2), we conducted the CTt again in post-test condition[6] and we collected the academic grades of the students for the ending trimester. It was only possible to retrieve the academic performance of 138 students (43.95% of the total sample), because not all the parents authorized the use of such information for our study (as mandated by the IRB ethical guidelines in educational research). In addition, we could download the learning analytics of Code.org for 289 students (92.04%).

Moreover, from the eleven middle schools involved in our research, nine of them ≈ 269 students conducted the Code.org course following a 'teacher-paced' methodology (i.e., all students progressing along the learning path at the same pace, which was fixed by the teacher), and the other two schools ≈ 45 students did it through a 'self-paced' methodology (i.e., each student had the possibility to progress at his/her own pace) (Table 3). In the latter context, 'computational top thinkers' could emerge as these students moved along the Code.org course much faster than their regular peers did, and requested the teacher to accelerate to a 'text-based' programming environment. Specifically, seven from the 45 students (15.6%), all males, were cataloged as 'computationally talented' subjects following this criterion, which will be used to determinate the predictive validity of the CTt in Section 4.4. Finally, it was possible to study in detail two of the seven 'computationally talented' students within the *"Intro to JS: Drawing & Animation"* course of Khan Academy (Section 4.5). Overall, IRB ethical guidelines were followed throughout the entire research procedure.

[5] Available at https://www.khanacademy.org/computing/computer-programming/programming.
[6] This pre-post evaluation with the CTt is out of the scope of this paper.

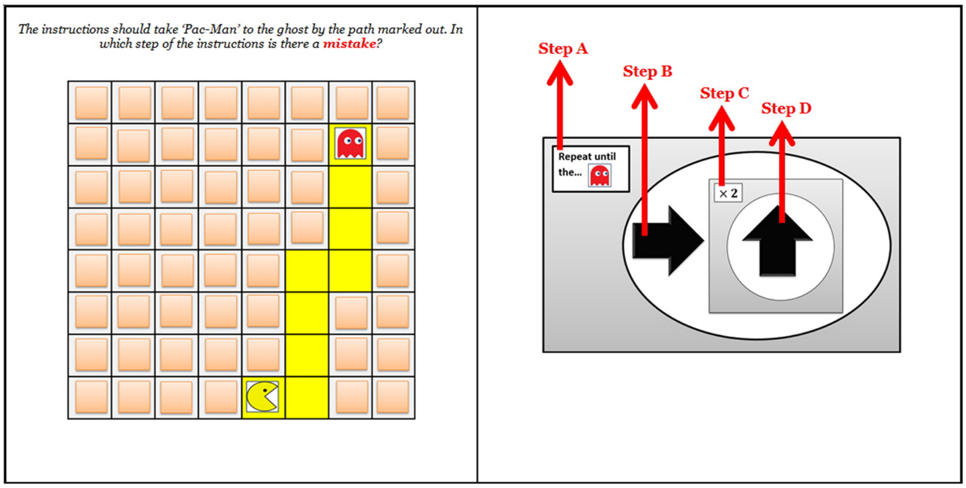

Fig. 1. CTt, item #11 ('maze'): loops 'repeat until + repeat times' (nested); 'visual arrows'; 'debugging'.

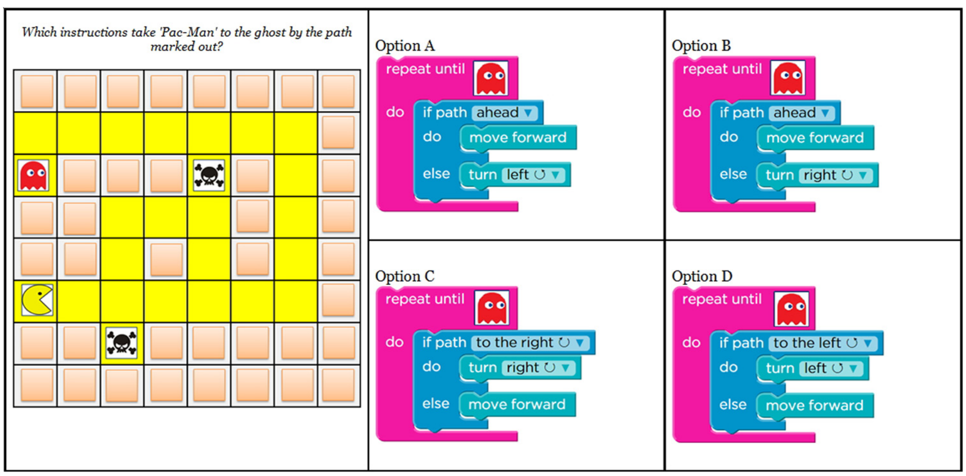

Fig. 2. CTt, item #18 ('maze'): loops 'repeat until' + if/else conditional (nested); 'visual blocks'; 'sequencing'.

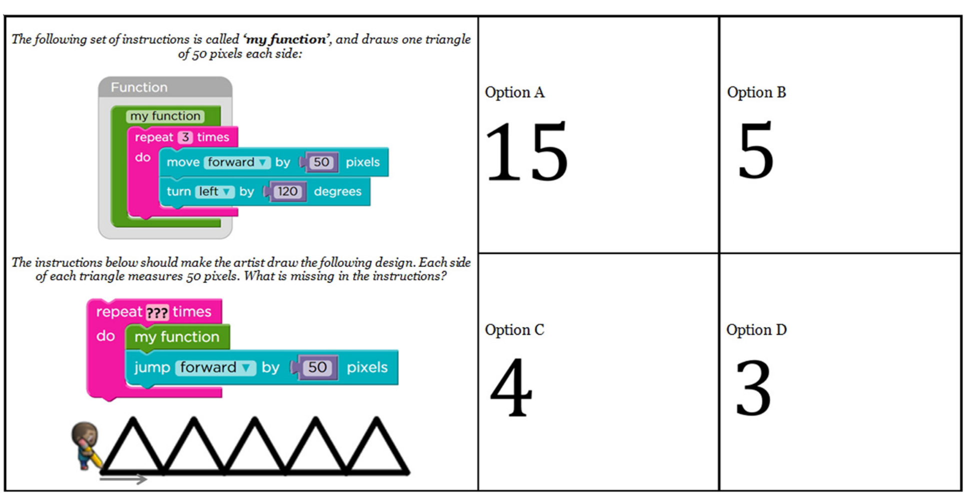

Fig. 3. CTt, item #26 ('canvas'): loops 'repeat times' + simple functions (nested); 'visual blocks'; 'completing'.

Table 3
Distribution of the initial sample ($n = 314$) by pace (self-paced vs. teacher-paced), school, and gender.

Pace				Gender		Total
				Boys	Girls	
Self-paced	School name	IES María Blasco (A)	Count	12	5	17
		IES Andreu Sempere		17	11	28
	Total			29	16	45
			% of Total	64.4%	35.6%	100.0%
Teacher-paced	School name	IES Penyagolosa	Count	23	6	29
		IES L'Almadrava		16	6	22
		IES Camp de Morvedre		44	13	57
		IES Juan de Garay		7	10	17
		IES Dr. Lluis Simarro		9	6	15
		IES María Blasco (B)		30	13	43
		IES L'Eliana		27	8	35
		Escola La Masía		15	13	28
		IES El Plá		13	10	23
	Total			184	85	269
			% of Total	68.4%	31.6%	100.0%

Table 4
Descriptive statistics of the CTt score (T_1) for the entire sample ($n = 314$).

Mean		16.39
Std. Error of mean		.244
Median		16.00
Mode		16
Std. Deviation		4.332
Variance		18.762
Skewness		.162
Kurtosis		−.292
Minimum		6
Maximum		27
Percentiles	10	11.00
	20	13.00
	25	13.00
	30	14.00
	40	15.00
	50	16.00
	60	17.00
	70	18.00
	75	19.00
	80	20.00
	90	22.00
	95	24.00
	97	25.55
	99	26.00
	99.5	27.00

Table 5
Non-parametric correlations (Spearman's r) between CTt (T_1) and academic performance (T_2).

		Academic performance (T_2)			
		Informatics	Mathematics	Language	GPA
(T_1)	CTt	.432[a]	.355[a]	.421[a]	.472[a]
(T_2)	Informatics		.596[a]	.494[a]	.796[a]
	Mathematics			.629[a]	.891[a]
	Language				.829[a]

$n = 138$.
[a] p<.01.

4. Results and discussion

4.1. Descriptive statistics

Table 4 shows the main descriptive statistics of the CTt score, which may range from 0 to 28, relative to the administration of the test at the beginning of the quarter (T_1) to the entire sample ($n = 314$). As it can be seen, the CTt score presents a large variability that enables to build accurate and discriminant percentiles all along the range of the variable (e.g., if we focus on the right tail of the distribution, different CTt scores are established for percentiles 90, 95, 97, 99 and 99.5). Although mean, median and mode have very similar values, and skewness is close to zero, the normality of the CTt score distribution is rejected ($Z_{k\text{-}s}$=.075; p<.01).

4.2. Predictive validity of the CTt regarding academic performance

Concerning the predictive validity of the CTt, conducted in T_1, with respect to academic performance, collected in T_2, the correlation values shown in Table 5 are obtained. As the normality assumption was also not strictly met by the academic performance variables (.12<$Z_{k\text{-}s}$<.16; p <.01 in all variables), non-parametric correlations (Spearman's r) are calculated.

As it can be seen (Table 5; Fig. 4), the correlation between the CTt and the grade-point average (GPA) (r = .47) is statistically significant, positive and moderately intense; and its value is in the expected range (r>.40) of other well-established cognitive correlates of academic performance, as described in Section 2.1. These results lead us to affirm the overall predictive validity of the CTt regarding academic performance.

Moreover, when we focus on the correlations of the CTt with the academic performance in each of the subjects, we find that the higher value corresponds to Informatics (r = .43), which is consistent with the fact that the Code.org course was followed within that subject along the quarter. However, the correlation value CTt (T_1) * Mathematics (T_2) (r = .36) is the lowest of the three subjects, and it seems to be lower than expected according to the literature. This result may have two explanations: on the one hand, 12 weeks elapse between both measurements so the correlation is more than likely reduced. On the other hand, the fact that the aforementioned correlation value is even lower than the one regarding Language (r = .42) is consistent with prior results found in [17], where the CTt correlated in a statistically significant way with verbal ability (r = .27; p<.01), but it did not with numerical ability (r =-.16; p>.05). In other words, some evidence is gathered that supports the statement that the syntactic–linguistic structural aspects might be more important than the numerical ones for thinking computationally, as suggested before in [64].

4.3. Predictive validity of the CTt regarding coding achievement

Regarding the predictive validity of the CTt, conducted in T_1, with respect to coding achievement in the Code.org course, retrieved in T_2, the correlation values shown in Table 6 are obtained.

Table 6
Non-parametric correlations (Spearman's r) between CTt (T_1) and coding achievement (T_2).

		Coding achievement through learning analytics in Code.org (T_2)	
		Completed levels	Lines of code
(T_1)	CTt	.444[a]	.357[a]
(T_2)	Completed levels		.836[a]

$n = 289$.
[a] p<.01.

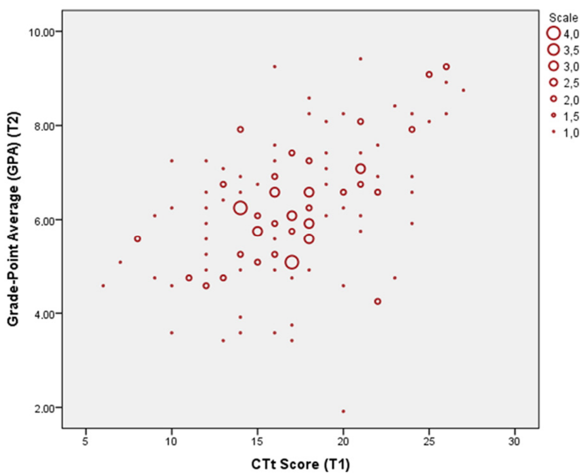

Fig. 4. Scatterplot between CTt (T_1) and grade-point average (GPA) (T_2).

Non-parametric correlations (Spearman's r) are calculated again, since coding achievement variables did not reach normality either (.09<$Z_{k\text{-}s}$<.12; p <.01 in both variables).

As it can be seen (Table 6; Fig. 5), the correlation between the CTt and the number of completed levels in Code.org (r = .44) is statistically significant, positive and moderately intense; and its value is also in the same expected range (r>.40) than before. These results lead us to affirm the overall predictive validity of the CTt regarding coding achievement.

Furthermore, 'completed levels' seems to be a more confident indicator of coding achievement than 'lines of code', since the same puzzle (≈ level) can be solved using a diverse number of lines of code, often inefficiently (i.e., high computational ability does not correlate necessarily with more lines of code written by the subject, see Fig. 6). This fact is expressed in the non-perfect correlation between both indicators (r = .84).

4.4. Predictive validity of the CTt to early detect 'computationally talented' students

As previously justified, the criterion used for categorizing the students as 'computationally talented' is to have accelerated from the 'block-based' environment of Code.org to the 'text-based' one in Khan Academy. This criterion was only applied in the classrooms with a 'self-paced' approach during the quarter ($n = 45$) so that the seven 'computationally talented' students, all males, could spontaneously emerge (i.e., indeed, in the 'teacher-paced' classrooms none of the students could progress to Khan Academy).

Thus, we wonder if the CTt conducted in T_1 condition could already detect significant differences between the subjects later categorized as 'computationally talented' (≈ *computational top thinkers*) and the 'regular learners' (≈ *computational regular thinkers*, i.e., those who did not accelerate). This question is addressed through the analysis shown in Table 7.

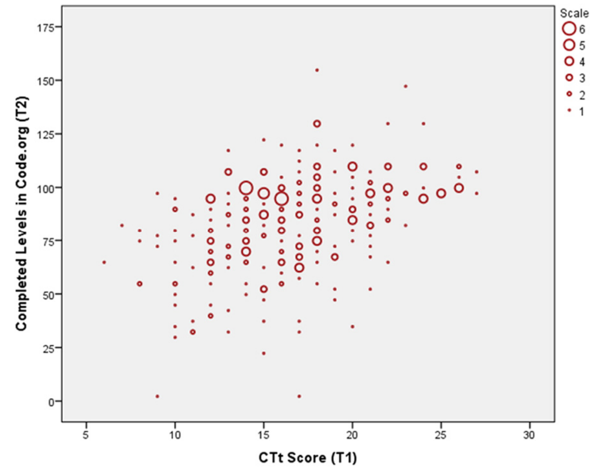

Fig. 5. Scatterplot between CTt (T_1) and coding achievement in terms of completed levels in Code.org (T_2).

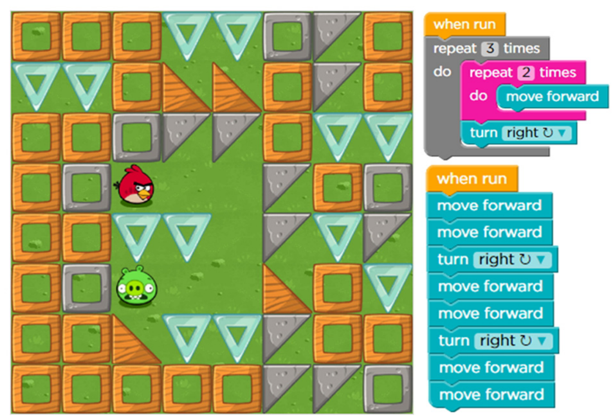

Fig. 6. Puzzle #9[7] of the *'Intro to CS Course'* (left), and two possible solutions to complete the level (efficient on right-top ≈ 5 lines of code, inefficient on right-bottom ≈ 9 lines of code).

As it can be seen, the CTt could already detect in pre-test condition (T_1) a statistically significant difference (t =-2.300; p=.026<.05; or the equivalent non-parametric test $U_{\text{Mann-Whitney}} =$ 66.500; p=.035<.05), with a large effect size associated (d = .95) [75], between the subjects who were categorized or not as 'computationally talented' several weeks later. Hence, the predictive validity of the CTt to early distinguish between computational *top vs. regular* thinkers, as these categories have been defined, is demonstrated.

Another way to prove the predictive power of the CTt to early distinguish between the two types of subjects is to calculate the corresponding *receiver operating characteristic* (ROC) curve [76], which depicts the false positive rate (i.e., also called 'specificity') and the true positive rate (i.e., also called 'sensitivity') of a dichotomous variable (i.e., yes ≈ 'computationally talented' vs. no ≈ 'regular learner') along the range of a continuous variable (i.e., the CTt Score at T_1). Results are shown in Tables 8–9 and Fig. 7.

As it can be seen, the area under the ROC curve is A = .750. This result is statistically significant (p=.037<.05) and can be interpreted as an indicator of 'good' predictive power [76]. Furthermore, we have detected the three cut-off points along the range of the CTt score with highest predictive power (17.5, 18.5, and 24.5). Focusing

7 Available at https://studio.code.org/s/20-hour/stage/2/puzzle/9.

Table 7
t-test for independent samples ('regular learners' vs. 'computationally talented') and effect size (d).

		n	Mean	Std. Deviation	Student's t	p-value$_{(t)}$	Cohen's d
CTt (T_1)	'Regular learners'	38	17.45	4.360	-2.300[a]	.026	.95
	'Computationally talented'	7	21.57	4.353			

[a] p<.05.

Table 8
Area under the ROC Curve regarding the predictive power of the CTt (T_1) to early detect 'computationally talented' students.

Test Result Variable: CTt Score (T_1)				
Area	Std. Error[a]	Asymptotic Sig.[b]	Asymptotic 95% confidence interval	
			Lower bound	Upper bound
.750	.092	.037	.569	.931

[a] Under the nonparametric assumption.
[b] Null hypothesis: true area = .50.

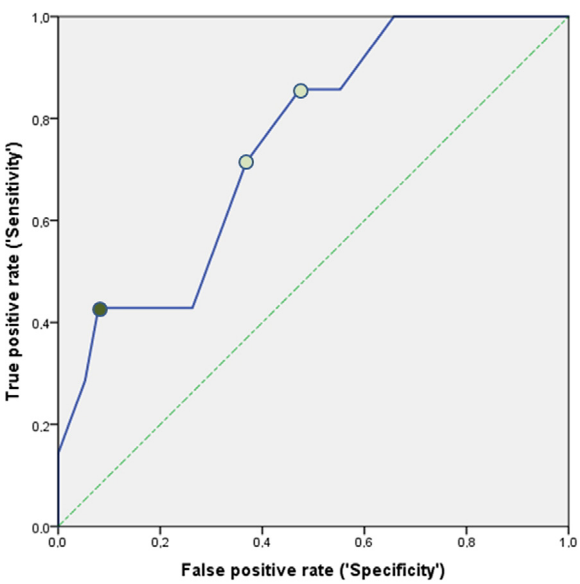

Fig. 7. ROC Curve regarding predictive power of the CTt (T_1) to early detect 'computationally talented' students.

on the latter, from the $n = 45$ students involved in this predictive study, six of them scored $\geq$24.5 in the CTt (T_1), whose gender is distributed as follows:

- Three boys, all of them 'true positives' (i.e., later categorized as 'computationally talented'). Therefore, 'sensitivity'$_{(CTt\geq 24.5)}$=3/7=.429 (42.9% of 'true positives' detected).
- Three girls, all of them 'false positives' (i.e., later categorized as 'regular learners'). Therefore, 'specificity'$_{(CTt\geq 24.5)}$=3/38 = .079 (7.9% of 'false positives' detected).

In other (critical) words, we find girls who, even having a high-computational ability, did not request for a coding acceleration (i.e., they were not subsequently categorized as 'computationally talented' students). This implies that our criterion for categorizing 'computationally talented' students might be gender biased (i.e., it might be only 'male sensitive'), being at the risk of hiding female talent in computing. In addition, these results are consistent with a large body of literature that describes: the female resistance, especially in middle school ages, to acceleration due to gender stereotypes and low self-efficacy perceptions [77]; the lower engagement of girls to acceleration programs [78]; or the less positive peer relations reported by girls after accelerating, specifically when skipping a grade [79]. Hence, it seems that computational talent should be described in additional terms so that the female perspective is included, as it will be suggested in Section 6.

4.5. *Case study over two 'computationally talented' subjects*

At last, we could study in detail two of the seven boys categorized as 'computationally talented' within the *"Intro to JS: Drawing & Animation"* course of Khan Academy. Both talented students were part of the same classroom of Year 8 (12–13 y/o). First, in Table 10, all the aptitudinal and performance results of the two subjects are shown: CTt score (T_1); RP30 problem-solving test [80] Score (T_1), which was conducted particularly in this classroom as part of the criterion validity of the CTt (further details of the RP30 test, as well as an example of one of its items, can be found at [17]); and academic performance (T_2) in Informatics, Mathematics and Language.

As it can be seen, both cases scored extremely high in the CTt (T_1), above the optimal cut-off point just found in previous subsection ($\geq$24.5; i.e., they were 'true positives'). However, their scores in the RP30 problem-solving test were just moderate. This latter result is unexpected, since the RP30 test was the one most correlated with the CTt (r = .67) during its criterion validation, already reported in [17]. In other words, although the RP30 test and the CTt are convergent measures, this convergence tends to decrease in the right tail of the CTt score distribution. Hence, these findings suggest that 'computationally talented' subjects might need specific tests to be detected, such as the CTt, given that a usual problem-solving test seems to be insensitive to them.

Furthermore, Table 11 shows the accumulated learning analytics of the subjects after following the *"Intro to JS: Drawing & Animation"* course of Khan Academy. As it can be seen, although both cases are considered 'computationally talented', each of them has a different interaction style with the on-line platform: Case 1 seems to be more active (more time spent on coding tasks), while Case 2 seems to be more reflexive (more time spent on viewing video tutorials).

Finally, one of the programming projects coded by Case 1 is depicted in Fig. 8. When the code is analyzed, it can be stated that the subject could write a Processing JavaScript program that involves: three variables included in a complex function, which is executed inside an infinite loop. These computational concepts written in a 'text-based' programming language are aligned with the curricular standards at the end of Year 10 in the Spanish curriculum [63]. In other words, our subject accelerated around 1-2 years in curricular terms.

5. Limitations and threats to validity

Three types of limitations can be noted for our work. Firstly, we recognize some limitations regarding the sampling procedures along the research. In this vein, the initial sample ($n = 314$) was composed by individuals who had previously selected 'Informatics' as an elective subject for the academic year, and due to this circumstance the sample was, at least, gender biased. Moreover, part

Table 9
Coordinates of the ROC curve and cut-off points with highest predictive power.

Corresponding percentile in the entire sample	Test Result Variable: CTt Score (T_1) Positive if Greater Than or Equal To	Sensitivity	Specificity
	7.00	1.000	1.000
	9.00	1.000	.974
	11.00	1.000	.947
	13.00	1.000	.842
	14.50	1.000	.763
	15.50	1.000	.658
	16.50	.857	.553
70	**17.50**	**.857**	**.474**
75	**18.50**	**.714**	**.368**
	19.50	.571	.316
	20.50	.429	.263
	22.50	.429	.132
95	**24.50**	**.429**	**.079**
	25.50	.286	.053
	26.50	.143	.000
	28.00	.000	.000

Table 10
Aptitudinal and academic performance results of the two 'computationally talented' subjects.

	Aptitudinal tests (T_1)				Academic performance (T_2)		
	CTt		RP30		Informatics	Mathematics	Language
	Score	Percentile	Score	Percentile			
Case 1	27	99.5	42	61	10	9	7
Case 2	26	99	36	44	10	9	8

of this initial sample was lost from T_1 to T_2 (especially with regard to academic performance variable), which supposes a threat to the validity of our concatenated results.

Secondly, we find other limitations regarding the instruments and indicators used in our research. Thus, the CTt has itself some limitations such as being heavily focused on computational 'concepts', ignoring computational 'practices' and 'perspectives', in terms of Brennan and Resnick [81]. Moreover, the CTt relies on a narrow CT definition that is closely tied to algorithmic thinking. This decision was already justified along the introduction, since algorithmic thinking seems to be a central part of CT, and especially prone to be measured. Nevertheless, the CTt might be ignoring other key parts of the construct, which constitutes a potential limitation of our study. Then, other CT assessment instruments, and other academic subjects or coding achievement indicators, could have been chosen, so other predictive validity values could have resulted to answer RQ_1 and RQ_2.

Lastly, some limitations can be pointed out regarding how the criterion to define 'computationally talented' students was stated. This criterion has a 'self-selection' bias (i.e., to be categorized as 'computationally talented' depends in first instance on whether the student decides to explicitly request the acceleration), so the aforementioned criterion might be indicating not only high computational ability but also high computational motivation and/or high computational self-efficacy, which constitutes a threat to its internal validity. The implications of this 'self-selection' bias with respect to gender have been already discussed in Section 4.4. Consequently, other criteria could have been stated to define 'computationally talented' students, so other predictive validity values (and another gender distribution of computational talent) could have been obtained to answer RQ_3.

6. Conclusions, implications and further research

Based on the previous results and discussions, some conclusions can be stated:

- There are large individual differences in computational ability (i.e., in CT aptitude) among middle school students, even before they start to learn coding.

Table 11
Learning analytics of the two 'computationally talented' subjects studied in Khan Academy.

	Video minutes	Skill minutes	Badges earned	Points earned
Case 1	169.45	795.01	16	95,222
Case 2	393.41	155.81	13	78,005

- The CTt has predictive validity with respect to the grade-point average (GPA) in Informatics, Mathematics and Language. This empirical evidence supports the statement that CT is a 'new' cognitive correlate of academic performance.
- The CTt has predictive validity with respect to the number of completed levels in Code.org. This empirical evidence supports the statement that CT is a 'new' cognitive correlate of programming achievement, at least in 'block-based' environments.
- The CTt has predictive validity to early distinguish between 'computational top thinkers' (i.e., 'computationally talented' students) and 'computational regular thinkers', when the criterion followed for stating these categories is to accelerate or not from 'block-based' to 'text-based' programming languages within a 'self-paced' learning methodology in the classroom.
- The prior criterion is potentially gender biased as middle school girls may not feel like accelerating even if they have the potential ability for it.
- 'Computationally talented' students might need specific tests to be detected, such as the CTt, given that usual problem-solving tests seem to be insensitive to them.
- Different individuals cataloged as 'computationally talented' can have different styles when they learn 'text-based' programming languages through on-line platforms, such as Khan Academy.
- 'Computationally talented' students detected in middle school might have the ability to accelerate around 1 or 2 years in terms of curricular standards.

Then, we consider several implications that derivate from the conclusions above. Firstly, given the predictive validity of the CTt

```
var xPos = 200;
var yPos = 200;
var e=10;
draw = function() {
    background(29, 40, 115);
    fill(255, 242, 0);
    ellipse(xPos, yPos, e, e);
    e+=1;

};


```

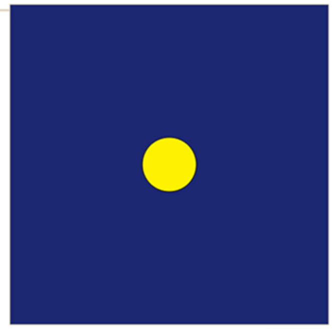

Fig. 8. Programming project coded by Case 1, available at https://www.khanacademy.org/computer-programming/spin-off-of-project-shooting-star/6674640557309952.

regarding academic performance and programming achievement, it is possible and valuable to administer the test at the beginning of middle school level to early detect students that might probably: (a) be at risk of school failure; (b) have further 'learning to code' difficulties. Then, it would be possible to design preventive educational actions for them. Secondly, given the predictive validity of the CTt regarding the detection of 'computationally talented' students, it is also valuable to administer the test when starting middle school to early detect students with high-computational ability, who probably will learn faster and could be beneficiated of being accelerated through the computing–coding curriculum. Thus, their extraordinary potential would be reached, and their talent would not be lost.

However, given the gender bias of the criterion followed for stating computational talent, additional approaches should be included to recognize and promote the top female computational thinkers, and to consequently encourage their interest in advancing to 'text-based' programming languages. These female approaches should go beyond the mere acceleration in 'self-paced' on-line learning contexts, and some of them have been already suggested in the literature specifically for middle school girls, such as the use of: 'pair programming' methodology [82,83]; tangible interfaces for learning programming [84]; embodied pedagogical strategies [85]; story-telling contexts for developing CT [64]; or conversational chat-bots [86,87].

Moreover, this paper contributes to the validation process of the CTt, already started in [17,21–23] and, ultimately, to reinforce the solidity of CT as a psychological construct. In summary, this work provides a validated assessment tool such as the CTt, which will permit to measure, make visible, and attend the natural individual differences in computational ability. Consequently, each and every of the students will be in better conditions to reach as far as their aptitude and motivation allow them. Further investigations should lead the CT research community to deepen in the definition, detection and intervention not only of high-computational abilities (i.e., the right tail), but also of low-computational abilities. All of them must be considered in order to ensure an inclusive CS education.

References

[1] L. Manovich, Software Takes Command, Bloomsbury Academic, UK, 2013.

[2] D. Rushkoff, Program or Be Programmed: Ten Commands for a Digital Age, OR Books, New York, 2010.

[3] M. Prensky, Programming is the new literacy, Edutopia, 2008. https://www.edutopia.org/literacy-computer-programming.

[4] D. Rushkoff, Code literacy: A 21st-century requirement, Edutopia, 2012. https://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff.

[5] M. Román-González, Aprender a programar "apps" como enriquecimiento curricular en alumnado de alta capacidad [To learn programming "apps" as curriculum enrichment on gifted students], Bordón. Rev. Pedagog. 66 (2014) 135–155. http://dx.doi.org/10.13042/Bordon.2014.66401.

[6] S.Y. Lye, J.H.L. Koh, Review on teaching and learning of computational thinking through programming: What is next for K-12? Comput. Hum. Behav. 41 (2014) 51–61. http://dx.doi.org/10.1016/j.chb.2014.09.012.

[7] S. Grover, S. Cooper, R. Pea, Assessing computational learning in K-12, in: Proc. 2014 Conf. Innov. Technol. Comput. Sci. Educ., 2014, pp. 57–62. http://dx.doi.org/10.1145/2591708.2591713.

[8] J.M. Wing, Computational thinking, Commun. ACM 49 (2006) 33–35. http://dx.doi.org/10.1145/1118178.1118215.

[9] J.M. Wing, Computational thinking and thinking about computing, Phil. Trans. R. Soc. A 366 (2008) 3717–3725. http://dx.doi.org/10.1098/rsta.2008.0118.

[10] A. Balanskat, K. Engelhardt, Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe, European Schoolnet, Brussels, 2014. http://www.eun.org/c/document_library/get_file?uuid=521cb928-6ec4-4a86-b522-9d8fd5cf60ce&groupId=43887.

[11] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, et al., Developing Computational Thinking in Compulsory Education-Implications for policy and practice, Join Research Center (European Commission), Seville, 2016. http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf.

[12] S. Grover, R. Pea, Computational thinking in K–12 a review of the state of the field, Educ. Res. 42 (2013) 38–43. http://dx.doi.org/10.3102/0013189X12463051.

[13] F. Kalelioglu, Y. Gülbahar, V. Kukul, A framework for computational thinking based on a systematic research review, Balt. J. Mod. Comput. 4 (2016) 583–596. http://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf.

[14] V.J. Shute, C. Sun, J. Asbell-Clarke, Demystifying computational thinking, Educ. Res. Rev. 22 (2017) 142–158. http://dx.doi.org/10.1016/j.edurev.2017.09.003.

[15] A. Mühling, A. Ruf, P. Hubwieser, Design and first results of a psychometric test for measuring basic programming abilities, in: Proc. Work. Prim. Second. Comput. Educ., 2015, pp. 2–10. http://dx.doi.org/10.1145/2818314.2818320.

[16] P.S. Buffum, E.V. Lobene, M.H. Frankosky, K.E. Boyer, E.N. Wiebe, J.C. Lester, A practical guide to developing and validating computer science knowledge assessments with application to middle school, in: Proc. 46th ACM Tech. Symp. Comput. Sci. Educ., 2015, pp. 622–627. http://dx.doi.org/10.1145/2676723.2677295.

[17] M. Román-González, J.-C. Pérez-González, C. Jiménez-Fernández, Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test, Comput. Hum. Behav. 72 (2017) 678–691. http://dx.doi.org/10.1016/j.chb.2016.08.047.

[18] A.V. Aho, Computation and computational thinking, Comput. J. 55 (2012) 832–835. http://dx.doi.org/10.1093/comjnl/bxs074.

[19] B. Rodriguez, S. Kennicutt, C. Rader, T. Camp, Assessing computational thinking in CS unplugged activities, in: Proc. 2017 ACM SIGCSE Tech. Symp. Comput. Sci. Educ., 2017, pp. 501–506. http://dx.doi.org/10.1145/3017680.3017779.

[20] C.P. Brackmann, M. Román-González, G. Robles, J. Moreno-León, A. Casali, D. Barone, Development of computational thinking skills through unplugged activities in primary school, in: Proc. 12th Work. Prim. Second. Comput. Educ., ACM, New York, NY, USA, 2017, pp. 65–72. http://dx.doi.org/10.1145/3137065.3137069.

[21] M. Román-González, Computational thinking test: Design guidelines and content validation, in: Proc. 7th Annu. Int. Conf. Educ. New Learn. Technol. (EDULEARN 2015), 2015, pp. 2436–2444. http://dx.doi.org/10.13140/RG.2.1.4203.4329.

[22] M. Román-González, J.-C. Pérez-González, J. Moreno-León, G. Robles, Does computational thinking correlate with personality?: The non-cognitive side of computational thinking, in: Proc. Fourth Int. Conf. Technol. Ecosyst. Enhancing Multicult., ACM, New York, 2016, pp. 51–58. http://dx.doi.org/10.1145/3012430.3012496.

[23] M. Román-González, J.-C. Pérez-González, J. Moreno-León, G. Robles, Extending the nomological network of computational thinking with non-cognitive factors, Comput. Hum. Behav. 80 (2018) 441–459. http://dx.doi.org/10.1016/j.chb.2017.09.030.
[24] J.R. Best, P.H. Miller, J.A. Naglieri, Relations between executive function and academic achievement from ages 5 to 17 in a large, representative national sample, Learn. Individ. Differ. 21 (2011) 327–336. http://dx.doi.org/10.1016/j.lindif.2011.01.007.
[25] A.S. Finn, M.A. Kraft, M.R. West, J.A. Leonard, C.E. Bish, R.E. Martin, M.A. Sheridan, C.F.O. Gabrieli, J.D.E. Gabrieli, Cognitive skills, student achievement tests, and schools, Psychol. Sci. 25 (2014) 736–744. http://dx.doi.org/10.1177/0956797613516008.
[26] F. Gagné, F. St Père, When IQ is controlled, does motivation still predict achievement? Intelligence 30 (2001) 71–100. http://dx.doi.org/10.1016/S0160-2896(01)00068-X.
[27] K. Laidra, H. Pullmann, J. Allik, Personality and intelligence as predictors of academic achievement: A cross-sectional study from elementary to secondary school, Pers. Individ. Differ. 42 (2007) 441–451. http://dx.doi.org/10.1016/j.paid.2006.08.001.
[28] J. Gottschling, M. Spengler, B. Spinath, F.M. Spinath, The prediction of school achievement from a behavior genetic perspective: Results from the German twin study on Cognitive Ability, Self-Reported Motivation, and School Achievement (CoSMoS), Pers. Individ. Differ. 53 (2012) 381–386. http://dx.doi.org/10.1016/j.paid.2012.01.020.
[29] A.R. Jensen, The relationship between learning and intelligence, Learn. Individ. Differ. 1 (1989) 37–62. http://dx.doi.org/10.1016/1041-6080(89)90009-5.
[30] S.B. Kaufman, M.R. Reynolds, X. Liu, A.S. Kaufman, K.S. McGrew, Are cognitive g and academic achievement g one and the same g? An exploration on the Woodcock–Johnson and Kaufman tests, Intelligence 40 (2012) 123–138. http://dx.doi.org/10.1016/j.intell.2012.01.009.
[31] R.C. Eaves, L. Mann, R.H. Vance, A. Parker-Bohannon, Cognition and academic achievement: The relationship of the cognitive levels test, the keymath revised, and the woodcock reading mastery tests-revised, Psychol. Sch. 27 (1990) 311–318. http://dx.doi.org/10.1002/1520-6807(199010)27:4<311::AID-PITS2310270406>3.0.CO;2-K.
[32] M.L. Fleming, M.R. Malone, The relationship of student characteristics and student performance in science as viewed by meta-analysis research, J. Res. Sci. Teach. 20 (1983) 481–495. http://dx.doi.org/10.1002/tea.3660200510.
[33] A. Lieury, S. Lorant, B. Trosseille, F. Champault, R. Vourc'h, Video games vs. reading and school/cognitive performances: a study on 27000 middle school teenagers, Educ. Psychol. 36 (2016) 1560–1595. http://dx.doi.org/10.1080/01443410.2014.923556.
[34] R.J. Koubek, W.K. LeBold, G. Salvendy, Predicting performance in computer programming courses, Behav. Inf. Technol. 4 (1985) 113–129. http://dx.doi.org/10.1080/01449298508901793.
[35] R. Cafolla, Piagetian formal operations and other cognitive correlates of achievement in computer programming, J. Educ. Technol. Syst. 16 (1987) 45–55. http://dx.doi.org/10.2190/GW1P-7BBK-DVWR-W3TU.
[36] V.J. Shute, Who is likely to acquire programming skills? J. Educ. Comput. Res. 7 (1991) 1–24. http://dx.doi.org/10.2190/VQJD-T1YD-5WVB-RYPJ.
[37] S. Fincher, A. Robins, B. Baker, I. Box, Q. Cutts, M. de Raadt, P. Haden, J. Hamer, M. Hamilton, R. Lister, et al., Predictors of success in a first programming course, in: Proc. 8th Australas. Conf. Comput. Educ. Vol. 52, 2006, pp. 189–196. https://dl.acm.org/citation.cfm?id=1151894.
[38] C.G. Cegielski, D.J. Hall, What makes a good programmer? Commun. ACM 49 (2006) 73–75. http://dx.doi.org/10.1145/1164394.1164397.
[39] Y. Erdogan, E. Aydin, T. Kabaca, Exploring the psychological predictors of programming achievement, J. Instr. Psychol. 35 (2008) 264–270. https://www.learntechlib.org/p/101475/.
[40] G.A. Davis, S.B. Rimm, Education of the Gifted and Talented, Prentice-Hall, Inc, 1989.
[41] S.W. Brown, J.S. Renzulli, E.J. Gubbins, D. Siegle, W. Zhang, C.-H. Chen, Assumptions underlying the identification of gifted and talented students, Gift. Child Q. 49 (2005) 68–79. http://dx.doi.org/10.1177/001698620504900107.
[42] D. Lubinski, From terman to today: A century of findings on intellectual precocity, Rev. Educ. Res. 86 (2016) 900–944. http://dx.doi.org/10.3102/0034654316675476.
[43] S.M. Baum, S.V. Owen, B.A. Oreck, Talent beyond words: Identification of potential talent in dance and music in elementary students, Gift. Child Q. 40 (1996) 93–101. http://dx.doi.org/10.1177/001698629604000206.
[44] V. Koshy, P. Ernest, R. Casey, Mathematically gifted and talented learners: theory and practice, Internat. J. Math. Ed. Sci. Tech. 40 (2009) 213–228. http://dx.doi.org/10.1080/00207390802566907.
[45] C.J. Maker, Identification of gifted minority students: A national problem, needed changes and a promising solution, Gift. Child Q. 40 (1996) 41–50. http://dx.doi.org/10.1177/001698629604000106.
[46] E.S. Richert, Rampant problems and promising practices in the identification of disadvantaged gifted students, Gift. Child Q. 31 (1987) 149–154. http://dx.doi.org/10.1177/001698628703100403.
[47] F. Anvari, H.M.T. Tran, M. Kavakli, Using cognitive load measurement and spatial ability test to identify talented students in three-dimensional computer graphics programming, Int. J. Inf. Educ. Technol. 3 (2013) 94–99. http://dx.doi.org/10.7763/IJIET.2013.V3.241.
[48] F. Anvari, D. Richards, Using personality traits and a spatial ability test to identify talented aspiring designers in User-Centred Design methodologies, in: Eval. Nov. Approaches to Softw. Eng. (ENASE), 2015 Int. Conf., 2015, pp. 90–101. http://ieeexplore.ieee.org/document/7320341/.
[49] F. Anvari, D. Richards, A method to identify talented aspiring designers in use of personas with personality, in: Int. Conf. Eval. Nov. Approaches to Softw. Eng., 2015, pp. 40–61. http://dx.doi.org/10.1007/978-3-319-30243-0_3.
[50] J. Lerner, J. Tirole, Some simple economics of open source, J. Ind. Econ. 50 (2002) 197–234. http://dx.doi.org/10.1111/1467-6451.00174.
[51] D. Weintrop, Blocks, text, and the space between: The role of representations in novice programming environments, in: Vis. Lang. Human-Centric Comput. (VL/HCC), 2015 IEEE Symp., 2015, pp. 301–302. http://dx.doi.org/10.1109/VLHCC.2015.7357237.
[52] D. Weintrop, U. Wilensky, To block or not to block, that is the question: students' perceptions of blocks-based programming, in: Proc. 14th Int. Conf. Interact. Des. Child., 2015, pp. 199–208. http://dx.doi.org/10.1145/2771839.2771860.
[53] M. Kölling, N.C.C. Brown, A. Altadmri, Frame-based editing: Easing the transition from blocks to text-based programming, in: Proc. Work. Prim. Second. Comput. Educ., 2015, pp. 29–38. http://dx.doi.org/10.1145/2818314.2818331.
[54] M. Armoni, O. Meerbaum-Salant, M. Ben-Ari, From Scratch to ''real'' programming, ACM Trans. Comput. Educ. 14 (2015) 25:1–25:15. http://dx.doi.org/10.1145/2677087.
[55] J.A. Kulik, C.-L.C. Kulik, Effects of accelerated instruction on students, Rev. Educ. Res. 54 (1984) 409–425. http://dx.doi.org/10.3102/00346543054003409.
[56] S. Steenbergen-Hu, M.C. Makel, P. Olszewski-Kubilius, What one hundred years of research says about the effects of ability grouping and acceleration on K–12 students' academic achievement: Findings of two second-order meta-analyses, Rev. Educ. Res. 86 (2016) 849–899. http://dx.doi.org/10.3102/0034654316675417.
[57] M. Stephen, I. Warwick, Educating the More Able Student: What Works and why, SAGE, 2015, http://dx.doi.org/10.4135/9781473922204.
[58] M.R. Maggio, M. Sayler, Trying out acceleration for mathematically talented fifth graders, Gift. Child Today 36 (2013) 20–26. http://dx.doi.org/10.1177/1076217512465284.
[59] J. Wai, D. Lubinski, C.P. Benbow, J.H. Steiger, Accomplishment in science, technology, engineering, and mathematics (STEM) and its relation to STEM educational dose: A 25-year longitudinal study, J. Educ. Psychol. 102 (2010) 860. http://dx.doi.org/10.1037/a0019454.
[60] U.K. Department of Education, National Curriculum in England: Computing Programmes of Study, 2013. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study.
[61] K. Falkner, R. Vivian, N. Falkner, The Australian digital technologies curriculum: challenge and opportunity, in: Proc. Sixt. Australas. Comput. Educ. Conf. Vol. 148, 2014, pp. 3–12. https://dl.acm.org/citation.cfm?id=2667491.
[62] D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O'Grady-Cunniff, B.B. Owens, C. Stephenson, A. Verno, CSTA K–12 Computer Science Standards: Revised 2011, 2011. http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf.
[63] J.V. Berrocoso, M.R.F. Sánchez, M. del C.G. Arroyo, El pensamiento computacional y las nuevas ecologías del aprendizaje, Rev. Educ. A Distancia (2015). http://dx.doi.org/10.6018/red/46/3.
[64] K. Howland, J. Good, Learning to communicate computationally with Flip: A bi-modal programming language for game creation, Comput. Educ. 80 (2015) 224–240. http://dx.doi.org/10.1016/j.compedu.2014.08.014.
[65] J. Good, K. Howland, Programming language, natural language? Supporting the diverse computational activities of novice programmers, J. Vis. Lang. Comput. (2016). http://dx.doi.org/10.1016/j.jvlc.2016.10.008.
[66] A.E.R. Association, A.P. Association, N.C. on Measurement in Education, et al., Standards for Educational and Psychological Testing, American Educational Research Association, 1999.
[67] M. Román-González, J. Moreno-León, G. Robles, Complementary tools for computational thinking assessment, in: S.C. Kong, J. Sheldon, K.Y. Li (Eds.), Proc. Int. Conf. Comput. Think. Educ., (CTE 2017), The Education University of Hong Kong, 2017, pp. 154–159. http://www.eduhk.hk/cte2017/doc/CTE2017Proceedings.pdf.
[68] J. Moreno-León, G. Robles, M. Román-González, Dr. Scratch: automatic analysis of scratch projects to assess and foster computational thinking, RED. Rev. Educ. A Distancia 15 (2015) 1–23. http://www.um.es/ead/red/46/moreno_robles.pdf.
[69] K.H. Koh, A. Basawapatna, V. Bennett, A. Repenning, Towards the automatic recognition of computational thinking for adaptive visual language learning, in: Vis. Lang. Human-Centric Comput. (VL/HCC), 2010 IEEE Symp., 2010, pp. 59–66. http://dx.doi.org/10.1109/VLHCC.2010.17.
[70] V. Dagiene, G. Futschek, Bebras international contest on informatics and computer literacy: Criteria for good tasks, in: Int. Conf. Informatics Second. Sch.

Perspect., Springer, Berlin, Germany, 2008, pp. 19–30. http://dx.doi.org/10.1007/978-3-540-69924-8_2.

[71] A. Basawapatna, K.H. Koh, A. Repenning, D.C. Webb, K.S. Marshall, Recognizing computational thinking patterns, in: Proc. 42nd ACM Tech. Symp. Comput. Sci. Educ., 2011, pp. 245–250. http://dx.doi.org/10.1145/1953163.1953241.

[72] S. Grover, S. Basu, M. Bienkowski, M. Eagle, N. Diana, J. Stamper, A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments, ACM Trans. Comput. Educ. 17 (2017) 14:1–14:25. http://dx.doi.org/10.1145/3105910.

[73] Ö. Korkmaz, R. Çakir, M.Y. Özden, A validity and reliability study of the Computational Thinking Scales (CTS), Comput. Hum. Behav. (2017). http://dx.doi.org/10.1016/j.chb.2017.01.005.

[74] S. Grover, Robotics and engineering for middle and high school students to develop computational thinking, in: Annu. Meet. Am. Educ. Res. Assoc. New Orleans, LA, 2011. https://pdfs.semanticscholar.org/69a7/c5909726eed5bd66719aad69565ce46bbdcc.pdf.

[75] J. Cohen, A power primer, Psychol. Bull. 112 (1992) 155–159. http://dx.doi.org/10.1037/0033-2909.112.1.155.

[76] J.A. Hanley, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29–36. http://dx.doi.org/10.1148/radiology.143.1.7063747.

[77] J.E. Jacobs, V. Weisz, Gender stereotypes: Implications for gifted education, Roeper Rev. 16 (1994) 152–155. http://dx.doi.org/10.1080/02783199409553562.

[78] M.A. Swiatek, C.P. Benbow, Ten-year longitudinal follow-up of ability-matched accelerated and unaccelerated gifted students, J. Educ. Psychol. 83 (1991) 528. http://dx.doi.org/10.1037/0022-0663.83.4.528.

[79] J. Kretschmann, M. Vock, O. Lüdtke, A. Gronostaj, Skipping to the bigger pond: Examining gender differences in students' psychosocial development after early acceleration, Contemp. Educ. Psychol. 46 (2016) 195–207. http://dx.doi.org/10.1016/j.cedpsych.2016.06.001.

[80] N. Seisdedos, RP-30: Test de Resolución de problemas. Manual Técnico [RP-30: Problem-Solving Test. Technical Manual], TEA Ediciones, 2002.

[81] K. Brennan, M. Resnick, New frameworks for studying and assessing the development of computational thinking, in: Proc. 2012 Annu. Meet. Am. Educ. Res. Assoc. Vancouver, Canada, 2012, pp. 1–25. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf.

[82] L.L. Werner, J. Denner, S. Bean, Pair programming strategies for middle school girls, in: CATE, 2004, pp. 161–166. https://pdfs.semanticscholar.org/c143/ef2b9060bcd125f959bbb34223a29d970bef.pdf.

[83] L. Werner, J. Denning, Pair programming in middle school: What does it look like? J. Res. Technol. Educ. 42 (2009) 29–49. http://dx.doi.org/10.1080/15391523.2009.10782540.

[84] M.S. Horn, E.T. Solovey, R.J. Crouser, R.J.K. Jacob, Comparing the use of tangible and graphical programming languages for informal science education, in: Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2009, pp. 975–984. http://dx.doi.org/10.1145/1518701.1518851.

[85] S.B. Daily, A.E. Leonard, S. Jörg, S. Babu, K. Gundersen, Dancing alice: Exploring embodied pedagogical strategies for learning computational thinking, in: Proc. 45th ACM Tech. Symp. Comput. Sci. Educ., 2014, pp. 91–96. http://dx.doi.org/10.1145/2538862.2538917.

[86] L. Benotti, M.C. Martinez, F. Schapachnik, A tool for introducing computer science with automatic formative assessment, IEEE Trans. Learn. Technol. 11 (2018) 179–192. http://dx.doi.org/10.1109/TLT.2017.2682084.

[87] L. Benotti, M.C. Martínez, F. Schapachnik, Engaging high school students using chatbots, in: Proc. 2014 Conf. Innov. Technol. Comput. Sci. Educ., 2014, pp. 63–68. http://dx.doi.org/10.1145/2591708.2591728.