Education and Information Technologies
https://doi.org/10.1007/s10639-023-12156-y

Applying recommendation system for developing programming competencies in children from a non-weird context

Jesennia Cárdenas-Cobo[1] · Cristian Vidal-Silva[2] · Lisett Arévalo[1] · Magali Torres[1]

Received: 14 May 2023 / Accepted: 17 August 2023

Abstract

The information society is part of current life, and algorithmic thinking and programming are relevant for everybody regardless of educational background. Today's world needs professionals with computing competencies from WEIRD (Western, Educated, Industrialized, Rich, and Democratic Societies) and non-WEIRD contexts. Traditional programming languages include syntax barriers that complicate their overall adoption and usefulness for people from a non-WEIRD context. To solve it, block-based programming languages like Scratch permit the development of programming competencies without syntax restrictions in online environments. This article presents empirical evidence of the positive impact of Scratch with the CARAMBA recommendation system for lessons and exercises proposals based on collaborative filtering of personalized learning from students' experiences. Previous experiences demonstrated that students require assistance in successfully defining sub-competencies and exercises to develop programming competencies by applying Scratch. This work shows the application of Scratch and CARAMBA in a non-WEIRD school context for developing programming competencies. Obtained results show that developing exercises with Scratch and CARAMBA motivated students' autonomy, and as well, the programming learning application increased exam scores in all the analyzed grades. Those results encourage us to continue using Scratch and CARAMBA for developing programming competencies in similar non-WEIRD contexts.

Keywords Programming competencies · WEIRD · Non-WEIRD · Scratch · CARAMBA · Online education

✉ Cristian Vidal-Silva
cvidal@utalca.cl

1 Facultad de Ciencias E Ingenierías, Universidad Estatal de Milagro, Cdla. Universitaria, Km. 1.5 V´ıa Km. 26, Milagro 091706, Guayas, Ecuador

2 School of Videogame Development and Virtual Reality Engineering, Faculty of Engineering, University of Talca, Av. Lircay S/N, 3460000 Talca, Maule, Chile

Published online: 08 September 2023

1 Introduction

In today's society, most students are considered digital natives, and programming is necessary for solving problems (Kalelioğlu, 2015). Numerous educational institutions consider programming in their curricula, even at initial levels (i.e., K-12). Most of the research literature regarding developing programming competencies is in so-called WEIRD countries, nations with Western, Educated, Industrialized, Rich, and Democratic populations (Henrich et al., 2010b; Jones, 2010; Ruggeri et al., 2019). Methodologies are necessary to develop programming competencies in non-WEIRD communities.

For Henrich et al. (2010a), the experimental results in WEIRD communities are not standard for all populations because substantial variability and frequent outliers exist. Within the domains reviewed, programming learning should develop spatial reasoning, reasoning styles, motivation, categorization, and differential induction (Cárdenas-Cobo et al., 2021). As Tomczyk et al. (2019) and Martínez (2021) remark, Latin American countries, mainly Ecuador, have different social, technological, and economic features. The presence of less-developed areas from a socioeconomic point of view makes the teaching and learning process of programming a real challenge. It is a critical issue considering that students need a more substantive base for solving problems than mathematics and physics. According to (Niess, 2005; Medeiros et al., 2019; Byun et al., 2012), learning programming in a non-WEIRD context is a complex task, even in a university setting.

1.1 Problem statement, goal, and contributions

Following a sequence of steps to solve a particular problem and dividing tasks into subtasks are two main difficulties that new students of introductory programming courses at universities habitually enconunter (Medeiros et al., 2018). The usual curriculum of introductory programming courses at universities applies text-based programming languages that, for their syntax, are a significant barrier to working with and developing programming competencies. Consequently, student frustration and loss of confidence impede the acquisition of programming competencies (Bennedsen & Caspersen, 2012). This problem is even more significant for students from non-WEIRD areas because algorithmic and computing skills are rare in primary and secondary education.

After the positive results obtained in developing programming competencies in university students (Cárdenas-Cobo et al., 2019; Vidal-Silva et al., 2019), this work looks to applying Scratch and the CARAMBA recommendation system to develop programming competencies in primary school children from non-WEIRD contexts: this work applies Scratch and CARAMBA on 428 children aged eight to twelve from a primary public school in Milagro, Guayas, Ecuador. This article measures and evaluates the programming competencies' development through a validated instrument (Román-González, 2016) looking to generate a long-term impact on future university students from non-WEIRD areas in developing countries by providing a peer learning opportunity to develop programming competencies. The CARAMBA recommender system is applied for suggesting exercises and lessons. Thus, the main objective of this research is to verify the effective development of programming competencies in primary school students from a non-WEIRD area by applying block-based programming languages like Scratch. This article also examines CARAMBA's impact on university students' professional competencies when they become tutors. We considered System Engineering students for their familiarity with programming and to develop teaching abilities for class assistance at the university. In Ecuador, System Engineering is like Computer Engineering in other countries. Figure 1 summarizes the main components of this research: teachers and students of engineering, tutors, school students, and Scratch and CARAMBA. The arrow of that figure illustrates the practical supporting of teachers and tutors to develop programming competencies in school students by applying Scratch + CARAMBA.

1.2 Research questions

This project represents quantitative quasi-experimental work to answer the following research questions.

> **RQ1** [Impact of Scratch in a non-WEIRD context] How can Scratch affect the development of programming competencies in primary school students from a non-WEIRD area? Although this study includes Ecuadorian pupils, other in-developing and developed nations may benefit from the anticipated results and applied experiments.

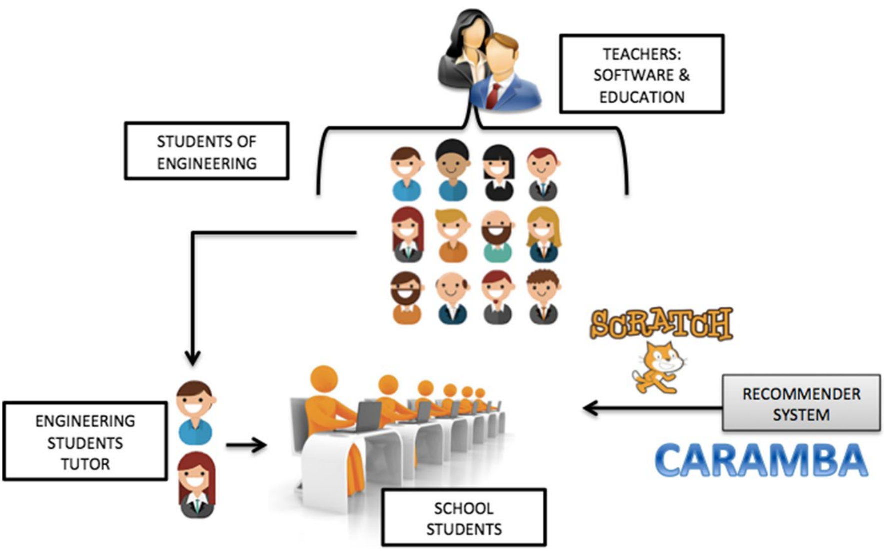


Fig. 1 The Main Participant of the Scratch and CARAMBA project

RQ2 [Impact of CARAMBA in a non-WEIRD context] What is the impact of CARAMBA (Cárdenas-Cobo et al., 2019) in the development of programming competencies in primary school students from a nonWEIRD area?
RQ3 [Tutors' Perception] How do System Engineering students as tutors perceive the benefits of Scratch, and CARAMBA (Cárdenas-Cobo et al., 2019) for effectively developing programming competencies in children?
RQ4 [Motivation to Continue Using Scratch and CARAMBA] How eager are the tutors in the experiment to learn and use more sophisticated programming skills in the future?

This study looks to answer each defined research question considering an initial test for primary school students to diagnose logical reasoning, an evaluation of computational thinking before and after applying Scratch and the CARAMBA recommendation system for the primary school students, and a final interview with university students to determine the research's impact on their academic profile.

1.3 Threats to validity

This article considers the following threats that could distort the expected results.

- This research was developed in a primary school with pupils between 8 and 12 years old. Regardless of the context, pupils have the minimum knowledge base to learn new technological tools (Thieman, 2008).
- The study had a longitudinal or follow-up profile to measure the evolution of programming competencies in primary school children.
- A group of university students from Systems Engineering participated in the study as tutors. Everything was controlled and advised by a group of professors from the same major participating in the research. University students were the tutors who assisted each group in the experiment.
- This research uses two reactive-type exams to measure the programming competencies evolution in primary school students following the same scheme of Román-González (2016) in a different scenario.
- To avoid the bias implied by the screening process, this work carried out a fully transparent evaluation process using multiple-choice questions with only one correct option.

The rest of this paper is organized as follows. Section 2 describes the main characteristics of WEIRD and non-WEIRD communities and details issues and current benefits of the teaching–learning in programming. Section 3 defines the material, objectives, impact, and applied tools in the research. Section 4 gives details and discusses the academic results obtained using Scratch and CARAMBA in the teaching–learning process for developing programming competencies. The paper concludes by summarizing the benefits of the educational experience and motivation of the authors for continuing to apply Scratch and CARAMBA to develop programming competencies in children anywhere, regardless of the WEIRD or non-WEIRD status of the area.

2 Background

2.1 WEIRD and non-WEIRD communities

According to Henrich et al. (2010b), scientists must stop doing most of their experimental research in a WEIRD context to understand human psychology better. Cultural psychologists have demonstrated that people from.

WEIRD communities tend to have a more individualistic and less communal outlook on life. In contrast, people from non-WEIRD backgrounds are more analytic and less holistic (Mesoudi et al., 2016).

As Roberts et al. (2020) indicate, WEIRD students are not a representative group of human beings. Using WEIRD students as a study group can result in broad and likely false assertions about what drives human behavior due to the easy access to information and technology in WEIRD populations (Jones, 2010). On the other hand, studying people living in a non-WEIRD context can offer substantially more significant research benefits (Ruggeri et al., 2019).

Arnett (2008) remark that research papers published in the top psychology journals focus on Western societies and do not represent humankind. That coincides with the outcome of (Henrich et al., 2010b) regarding the analysis of ten years of production in psychology journals suggesting a similar bias in the field of research. Henrich et al. (2010b) demonstrated a significant difference between WEIRD and non-WEIRD people cognitively and in making social decisions. Table 1 shows that Non-WEIRD communities have characteristics different from the WEIRD populations, which would allow reproducing experiments in different contexts through validated measuring instruments to ensure result quality (Pope et al., 2019; Román-González, 2016). Pope et al. (2019) suggest that by using codified rules, humans can solve many problems precisely. They also conclude that several cultural variation aspects, knowledge, environmental uncertainty, and educational training could emphasize the observed intercultural differences.

2.2 Introductory programming courses

The work of Gomes and Mendes (2007b) and Durak (2020) mention using variables, conditional structures, and repetitive cycles as essential competencies in teaching computer programming courses at the university. As the work of López-Escribano and Sánchez-Montoya (2012) indicates, introductory programming courses looking to develop those competencies at university levels have a high dropout. University students usually experience issues at the initial stages of their programming studies, leading to frustration and failure due to several factors (Cardenas-Cobo, 2020). However, the critical factor is that students need to improve in cognitive reasoning skills and applying problem-solving strategies, i.e., they have not developed the

Table 1 Differences between WEIRD and non-WEIRD populations

Characteristics	WEIRD	non-WEIRD	References
More individualist social orientation	√		(Henrich et al., 2010b; Mesoudi et al., 2016)
Less collectivist social orientation	√		(Mesoudi et al., 2016)
More analytic cognition (equity, cooperation, punishment)	√		(Henrich et al., 2010b; Mesoudi et al., 2016)
Holistic cognition (moral reasoning and independent/interdependent self-concept)		√	(Henrich et al., 2010b; Mesoudi et al., 2016)
Representative study group		√	(Jones, 2010)
Easy to access populations	√	√	(Ruggeri et al., 2019)
Nontraditional university enrolment patterns			(Byun et al., 2017; Fry, 2002; Lee & Frank, 1990)
Precise solution to problems	√		(Pope et al., 2019)
More publications in journals of Psychology	√		(Arnett, 2008)
Reasoning strategies	√		(Henrich et al., 2010b)
Implications for such traits as motivation or emotions	√		(Henrich et al., 2010b)

cognitive abilities yet to start learning programming (López-Escribano & Sánchez-Montoya, 2012).

Extensive research has examined experiences in the teaching–learning process of programming in search of the most common difficulties students face during introductory programming courses at university levels and how to address these difficulties, as this article describes as follows.

2.3 Related work

This article continues the Ph.D. work of the first author regarding applying Scratch programming language for developing programming competencies in non-WEIRD children (Cardenas-Cobo, 2020). We present development updates and evaluate the effectiveness of CARAMBA for recommending programming tasks to develop programming competencies for primary school children in non-WEIRD scenarios.

The work of Hsu et al. (2018) highlights that although developing programming competencies in education has significantly progressed in recent years, educators still need to identify how to teach it. Hence, applying recommendation tools to support children's learning process seems an adequate solution. Ali et al. (2022) propose a system architecture to build semantic recommendations regarding user requirements and preferences to seek proper courses. Because developing programming competencies is a great challenge, mainly for the lack of teaching materials and human resources, Saito and Watanobe (2020) propose a learning path recommendation system by applying a recurrent neural network with prominent results. He et al. (2020) applied recommender systems and Scratch to personalize experiences by a string input: a single program block is divided into related knowledge points to calculate distances, similarities, difficulties, and knowledge points for personalized programming experiences. The work of Cárdenas-Cobo et al. (2019) describes a recommender system (the first version of CARAMBA) for developing programming competencies in non-WEIRD students using Scratch.

Regarding the benefits and issues of using Scratch for developing programming competencies, as remarked by Rodríguez-Martínez et al. (2020), Scratch represents a free and online programming language that eases the acquisition of mathematical concepts and the development of programming competencies.

The work of Fagerlund et al. (2021) remarks that educational objectives of computational thinking are non-clear: various learning objectives exist for topics like computer science, programming, and computing but not for computational thinking. Schools must consider updating their curriculum to develop and apply computational thinking in their main subjects. That also requires adjusting the competencies of teaching majors to include programming competencies and block-based environments like Scratch and TinkerCAD for children (Tupac-Yupanqui et al., 2021).

Developing programming competencies in non-WEIRD communities is an important and necessary task. Nowadays, people live highly connected, and human dependence on technology is ever-increasing. Non-WEIRD children can feel motivated by accessing technology (Keeler & Bernstein, 2021). That motivation can help develop and improve other scientific and knowledge subjects by applying programming competencies. Different topics apply computational thinking in their experiments, and developing programming competencies can be a successful and enjoyable experience (Rojas-Valdés et al., 2022).

2.3.1 Teaching–learning initiatives

Lahtinen et al. (2005) performed a study with university students and professors to identify learning difficulties in introductory programming courses. More than half of the participant students (58.6%) had previous programming experience, and only 40.6% considered they had more or less moderate programming skills. Summarizing, that study found:

- The wide range of experience levels among programming students becomes a problem in designing exciting and challenging teaching for all students.
- The most challenging factors in programming are understanding how to design a program capable of solving a given problem, dividing functionality into steps, and locating errors in their programs.
- Self-study exercises were considered more valuable than lectures and hands-on sessions in computer classrooms.
- The rate of self-programming was more useful than self-study.
- The biggest problem among new programmers is not related to their ability to apply their new programming knowledge.

These findings highlight that student learning will improve significantly through practice and real-life situations regardless of their social source because computing and programming competencies are novel, attractive, and applicable in other areas. While theory is essential to programming, students also implement the ideas to thoroughly understand the concepts. For example, Wing (2010) refers to computational thinking as a mental activity in which people state a problem to solve with a computer.

Recent initiatives to improve the teaching–learning process of programming aim to implement strategies that foster the development of cognitive abilities in students. Those studies have prompted research into alternative methods of teaching introductory programming, such as constructive alignment (van Der Vleuten & Schuwirth, 2019). It is a student-centered education model that aims to enhance the learning of very diverse students, with various learning approaches and prior knowledge of programming (Cain & Woodward, 2012). Similarly, Gomes and Mendes (2007a) proposed a computational system that helps reduce current learning difficulties through an environment that prioritizes students' learning styles in designing different problems and tasks. Based on a constructive approach, the system provides opportunities to develop while the learning process is taking place.

Currently, learners must develop a self-learning capacity according to the requirements of society. Hernán et al. (2006) propose changing the teaching focused on transmitting formal contents (expository classes) to learning focused on competence development and solving problems related as closely as possible to reality. Problem-based learning improves retention in the long term by forcing the learner not only to understand but to apply or reflect on the learned concept. As a current problem-solving technique, programming and programmers require creativity and being creative, respectively (Ramírez-Montoya et al., 2021).

According to Sun and Liu (2022), the number of teaching methods is getting closer to the creative, and the potential role of programming needs to be higher in computer science research. Hence, educators should foster the students' motivation to improve their creative skills (Sánchez et al., 2022). Usually, researchers and computer education teachers do not emphasize creativity; they focus more on syntax rules and the program's goal. In that context, beginner programmers must interpret and give algorithmic solutions making algorithm programming a real challenge.

3 Methodology and research design

3.1 Methodology

This study worked with 428 children, 856 parents or relatives, 24 teachers from the educational unit, and 48 systems engineering students from the university. This study took place in a computer lab at the primary school. The primary school computer lab contained 15 computers fitted out for the research intervention. Although the primary school is located downtown in the city, there were children from the city outskirts and countryside.

This work diagnosed some problems in the initial curriculum of the computer course at primary school regarding essential programming competencies (Durak, 2020; Gomes & Mendes, 2007b): defining and using variables, applying conditional structures, and using repeating operations. The initial curriculum of the computer subject was limited to dominating basic office tool concepts and recognizing computer devices. Hence, we defined the following teaching–learning activities: 1. Definition of basic computer concepts. 2. Diagnostic evaluation of logical thinking. 3. Teaching Scratch. 4. Evaluation of Scratch Intervention. 5. Applying CARAMBA. 6. Final Evaluation of computational thinking in the verification results stage.

The diagnostic evaluation results highlighted low access to technology, especially computers at home. The first activity was teaching basic computing concepts, using the mouse and the operating system to create folders and store files. This research applied Scratch and CARAMBA in two phases: the first phase of five months and the second phase of the remaining three months. At the end of each phase, the 'Computational Thinking Test' version 2 (CTT V2) was applied (Román-González, 2016), an objective multiple choice test with four response options (only one correct), to characterize each item in one of five design axes: i) Treated computational concept, ii) Interface-environment of the item, iii) Style of response alternatives, iv) Existence or no existence of nesting, v) Required task.

Table 2 describes the fundamental characteristics of the student population at the academic unit. We observed that most students were younger than 15 years old. The male group is more significant than the female group, and 8% of the children have special needs. Students and professors of Systems Engineering collaborate in each working group with the course professor. Each Systems Engineering professor specialized in

Table 2 Equality-population approach matrix

Focus	Description	Beneficiaries
Gender	Male	236
	Female	192
	Total	428
Age	Under 8 years of age	426
	From 8 to 12 years old	2
	From 12 years to more	0
	Total	428
Disabilities	Physics	13
	Psychological	1
	Mental	2
	Auditory	2
	Visual	14
	Total	32
Mobility	Ecuadorian	419
	Foreigners	9
	Total	428

a technical topic of the research. Table 3 shows each group and describes its role and functions within the research.

3.2 Materials

Appendix A (Figs. 8, 9, 10, 11, and 12) shows the questions for the initial diagnosis to students. From this data, we obtained informative data, ethnic distribution, gender, level of access to technology, and essential questions of mathematical and logical reasoning, keeping in mind that in the school population, there are students who have never used a computer. Figures 13, 14, and 15 in the Appendix B show the second questionnaire that aims to assess the computational thinking of children. Students answered that questionnaire before and after applying Scratch with CARAMBA support.

Table 4 shows the third questionnaire used to understand the attitude of System Engineering students after participating in the research to learn about their motivation, knowledge of the professional world, generic competencies, specific competencies, and conclusions. That permitted determining social responsibility after they participated in the research.

The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Table 3 Organization of work groups

Roles	Participants	Functions
A) Teams of Engineering and Education Professors	Project director Expert in the tools used, Professor of Education, Associate supporting staff	Guide and train the students of Systems Engineering besides they make the follow up of the project
B) Teams Engineering Students –Tutor	Systems Engineering Students	Trained in pedagogic methods for teaching children and in the tools Scratch and CARAMBA. Their role is being teachers of basic concepts of programming and using the tools
C) Teams Engineering Students –Tutor	Systems Engineering Students	Trained in methodology for teaching and in the tools Scratch and CARAMBA, their role is to advise children's work
D) Teams Engineering Students—Evaluators	Systems Engineering Students	Their role is to apply tests for evaluating computational thinking

Table 4 Survey for interviewing engineering students

QUESTIONS	VARIABLES
(2 min) Students will receive an explanation of the methodology used and the objectives of the survey (5 min) The aim is to measure the student's motivation in two areas: motivation to offer a service	Preamble: Introduction to Project
to the community and motivation for the subject (Programming) " Is it harder for you to miss programming classes than for other subjects"? "Were you more motivated to program?" (5 min) Through questions:	Motivation
"Do you think the work done on the project is similar to what you will do in your future professional? Do you see this approach as interesting? We have worked with a public institution. ¿Do you find it interesting?" (5 min) They were asked the following question "Another objective of this methodology is to create a working environment that allows the student to develop generic or transversal competencies. Examples are:	Approach to professional world
autonomous learning, information search, teamwork, communication skills, planning, etc What do you think? In particular, how have you experienced this project? What did cohabitation and working with children in the public sector seem like? Did you perceive any problems (affective, emotional, learning, economic) in the children with whom you worked?" (5 min) The students were asked the question: Do you think you have learned more deeply the	Generic competences
technical knowledge of the subject (programming)? (20 min) It intends to obtain additional comments to improve the project. Do you think your programming learning has been better or worse because of the project?	Specific competences
(5 min) What was the best thing? (5 min) What improvements do you suggest? (5 min) Have you left anything unsaid?	Other

4 Scratch & CARAMBA

4.1 Scratch

The Scratch platform proposes a programming language based on blocks with visual grammar and combination rules that have the same role as the syntax in text-based programming languages such as C, Java, or Python (Cabo, 2019; Tupac-Yupanqui et al., 2021). As indicated by Resnick et al. (2009), the original goal of Scratch was to develop a programming approach that would attract people, regardless of age, social background, or educational background, to develop algorithmic solutions. Without the complexities of syntax and semantics of traditional programming languages, that makes Scratch a language for programming interactive stories, games, animations, and simulations easy for all its users, who can also share their creations with others. Thus, the primary goal of Scratch is not to prepare people for professional or technical careers in programming but to nurture a new generation of creative and systematic thinkers using programming to express their ideas. These researchers used Scratch to create educational activities to teach programming for various reasons: highly developed platform, the social component, collaborative work, ease of use, and variety of bibliography (Vidal Silva & Pavesi Farriol, 2005).

In WEIRD contexts, school children have more significant opportunities to access technology and develop computational thinking; as this work highlights, the literature demonstrates the application of various Scratch interventions in a WEIRD school context.

4.2 CARAMBA

CARAMBA is a recommendation system that allows students to suggest exercises to develop in Scratch. Those exercises are for comparing profiles with other users that permit recommending activities according to their level of knowledge, which can increase in complexity as they solve new exercises. Each student evaluates the style and complexity at the end of developing solutions. These two variables enable comparing students with similar backgrounds who have evaluated the exercises that rest in a knowledge base managed by the CARAMBA recommendation system. The works of Cárdenas et al. (2017); Cárdenas-Cobo et al. (2019) detail experiments and the academic results of using CARAMBA on university students in Ecuador.

CARAMBA was developed to improve learning in university contexts (Cárdenas et al., 2017). As an intermediary and motivator Scratch supporting tool can assist new students in developing programming competencies. After demonstrating the CARAMBA effectiveness with university students, it was scaled to a primary school context to propose a sustainable learning process in a non-WEIRD context. That is, to ensure that children in their primary school stage develop computational thinking and reduce difficulties in learning programming and decision-making regardless of their social setting. It is sustainable because school students can return to the learning of the development of computational thinking in other subjects in a school context by developing applied soft skills with their linking practices.

CARAMBA is not synonymous with Scratch (Cardenas-Cobo, 2020). It complements those interventions allowing students' autonomous learning by recommending exercises. The number of teachers and tutors required in a classroom to accompany the students is minimal. This study shows that students obtained better results on the test after the guided learning intervention with CARAMBA since they learned at their own pace in regard to that set by a teacher. Each application of CARAMBA begins with user registration to get the age and define new level exercises for the user. After identifying a user, CARAMBA recommends exercises from similar users regarding age, taste, suggested programming competencies, and difficulty level.

5 Data analysis & results

5.1 Initial evaluation of logical thinking

Before initiating Phase I, we applied a reasoning test to the students regarding their previous algorithmic and programming competencies (see Appendix A). From a population distributed in 12 parallels, morning and evening sessions, 5th, 6th, and 7th year of primary school, the test consisted of 17 questions to determine access to technology before intervention classified in abstraction, logical reasoning, successions, and series. Concerning the test and results, we observed the following.

- Questions 3, 14, and 15 (see Appendix A: Figs. 9, 12) are the ones that record the most errors at the time of application—above 85%.
- Question 3 allows the evaluation of abstraction.
- Questions 14 and 15 allow the evaluation of logical-mathematical thinking.
- In the questions of successions and series, the results have been satisfactory.

We can appreciate that questions about logical reasoning, mathematical thinking, and abstraction represent the main issues for students before starting the programming course.

5.2 Phase I: applying scratch

After the diagnosis, we proceeded to work with children on Scratch. One of the Systems Engineering students plays the role of tutor and explains the main options of Scratch. After reviewing variable initialization, conditioning factors, and repetitive cycles, this research applied the CTT V2 (Román-González, 2016).

Figure 2 shows the successes and failures for each group and section in the first exam test assessment. We can see that the percentage of successes and failures is very similar in each group in the morning and afternoon. The group with the most hits was 5th A (5A) in the afternoon.

Figure 3 summarizes the study's results showing that the failures are slightly superior to the successes in terms of the median. When applying the paired t-test statistic for independent samples with a normal distribution (Shapiro–Wilk, p-values = 0.945) and homogeneous variances (F-test, p-value = 1), we noted that these differences are not significant (t-test, pvalue = 0.982). Hence, we conclude that the results achieved in the first test were unsatisfactory because the number of hits and failures are similar. Thus, as an answer for RQ1, only using Scratch is insufficient for developing programming competencies in non-WEIRD child students. On this basis, this work identified students' weaknesses, and the work with CARAMBA began.

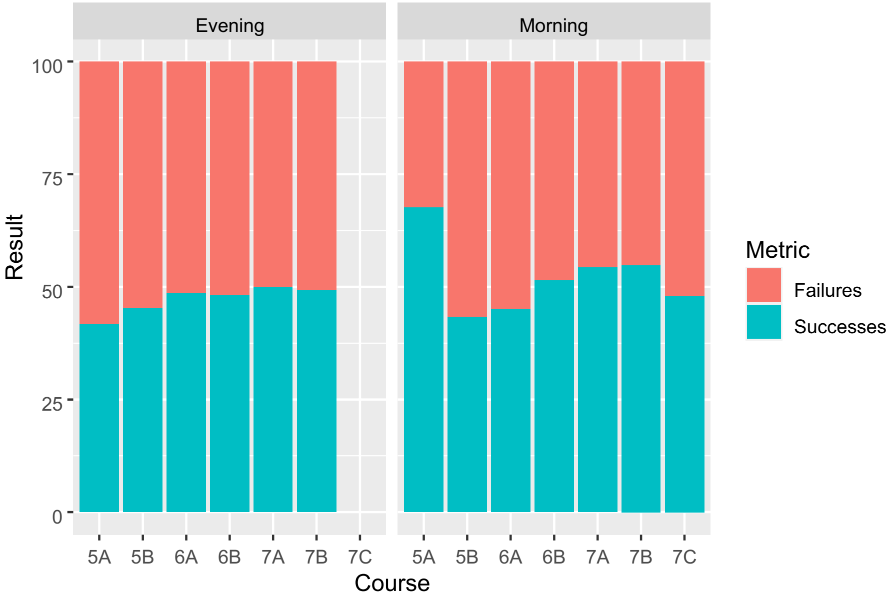


Fig. 2 Successes and failures by course. First exam test assessment

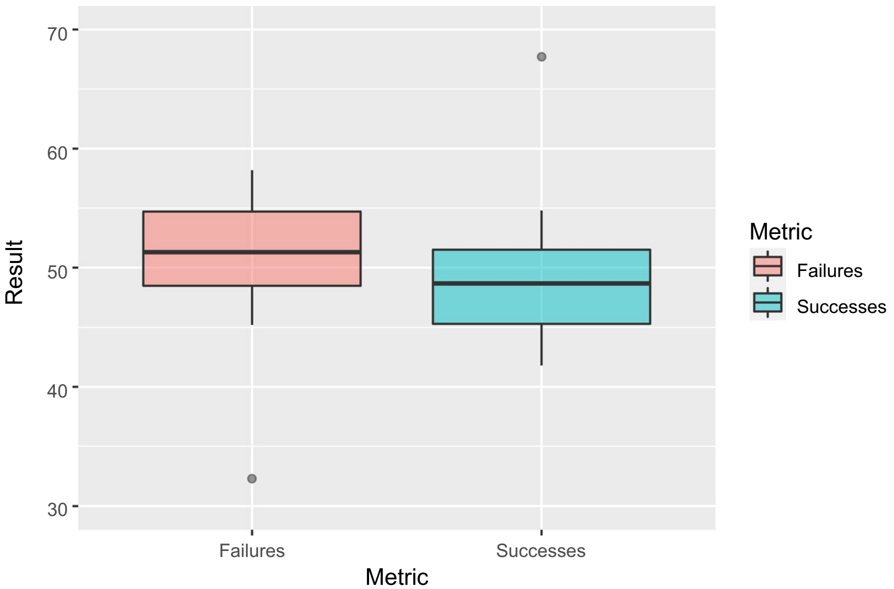


Fig. 3 Summary of successes and errors. First exam test assessment

5.3 Phase II: applying (Scratch + CARAMBA)

Unlike the first application of the CTT V2 (Román-González, 2016), the successes largely surpassed the failures in the test post-CARAMBA on the same students. Figures 4 and 5 show those results. We applied a nonparametric Wilcoxon test after proving that the data does not distribute normally (Shapiro–Wilk test, p-values = 0.00973). The results (p-value = 1.923e-08) show significant differences between failures and successes.

To finish with the analysis, Fig. 6 summarizes the successes achieved between the first and second phases of the project. A substantial improvement exists in the earned hits in the second test. According to the t-test (normality [p-value = 0.1659 and p-value homogeneity = 0.2465]), these differences resulted in significant (p-value = 1.892e-05). These results show that CARAMBA contributed to the learning process for developing programming and computing competencies of students from a non-WEIRD context, answering RQ2.

Additionally, Fig. 7 evidences that the students at the academic unit in the afternoon session present a noticeable improvement compared to those in the morning session, except for the 5A afternoon session. The education schedule of primary schools in Ecuador presents a separation between morning and evening courses. It was not easy to control autonomous learning in this group, as students did not always complete the exercises recommended by CARAMBA. Furthermore, guided education prevails due to various factors such as age, group of tutors, and schedule. The control of those variables should be part of future work. The growth rate of the evening session, concerning course success, is above 32%.

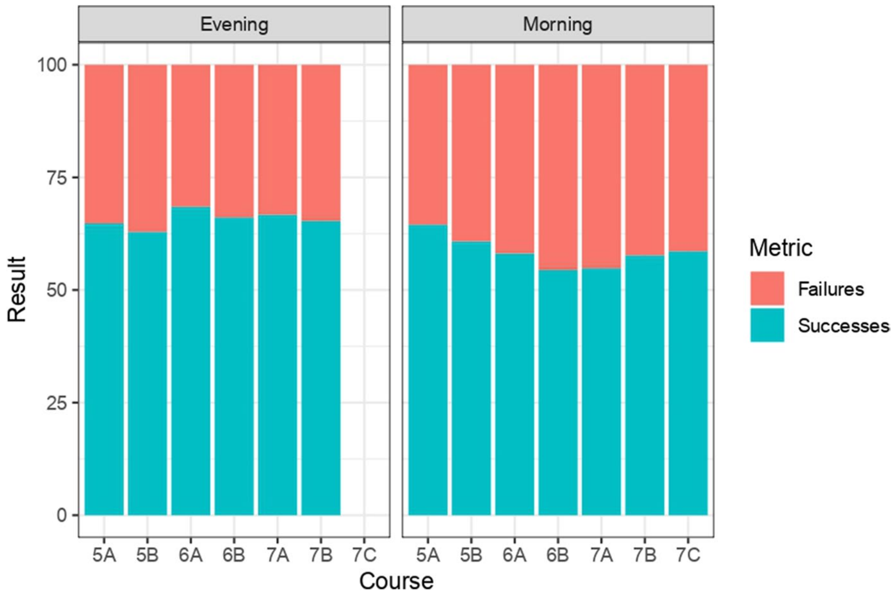


Fig. 4 Successes and failures by course. Second exam test assessment

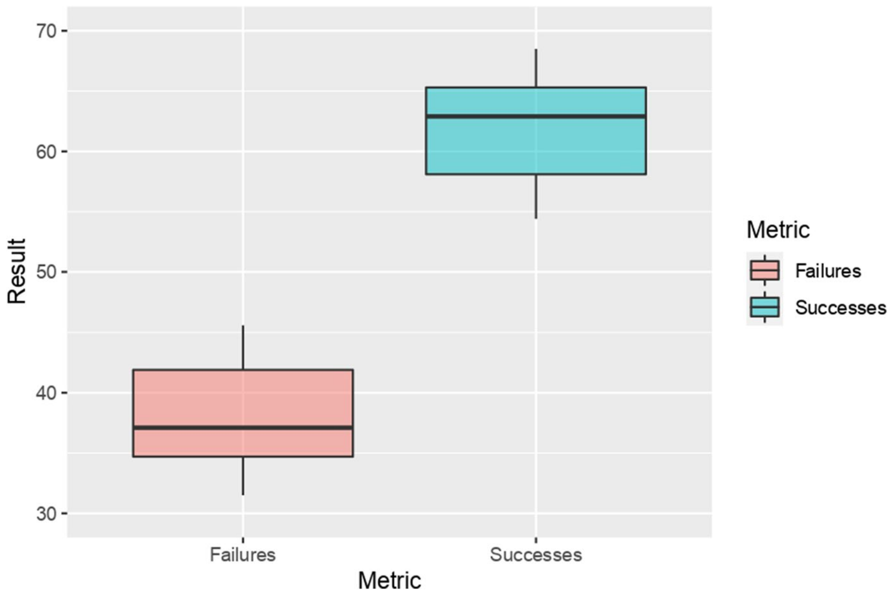


Fig. 5 Summary of successes and failures. Second exam test assessment

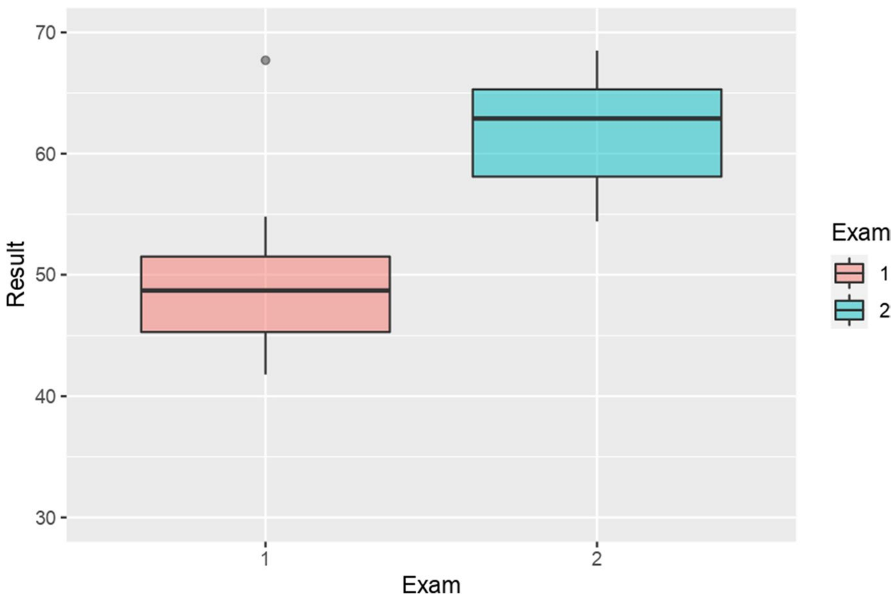


Fig. 6 Summary of successes and failures. Second exam test assessment

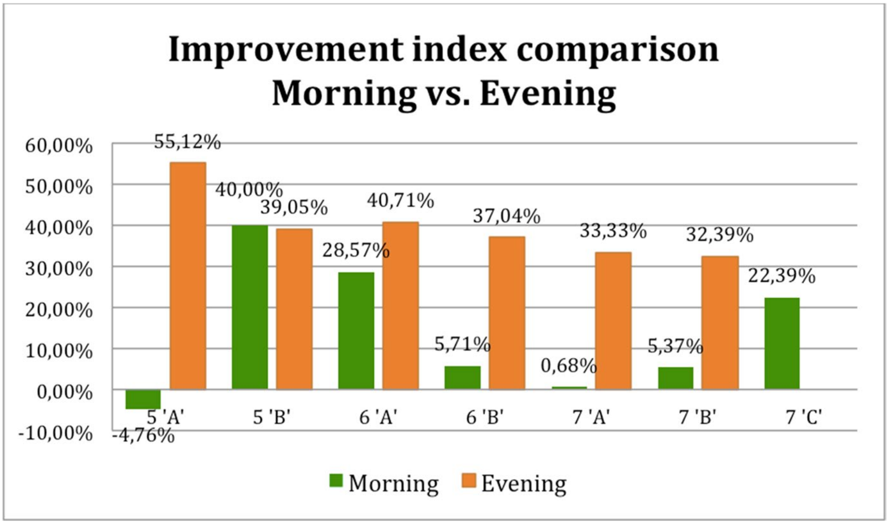


Fig. 7 Comparison of improvement for course and session

5.4 Social responsibility in engineering students

Ultimately, this research interviewed System Engineering students who participated as tutors. Table 4 shows the applied questions and research variables. The following lines summarize part of their answers.

- **Motivation:** 54% of the System Engineering students replied that it would be harder to miss the project than the classes in a normal subject. Hence, half of the engineering students consider it essential to assist in developing the project and avoid failing their responsibility. This percentage does not conclude as satisfactory the intervention in terms of responsibility; however, the priority given to his subjects contributes indirectly to better academic preparation for their professional development. Regarding this aspect, 96% of the engineering students confirm that they have been motivated to prepare themselves better in the fundamentals of programming before project participation, which helps improve their programming skills.
- **Approach to the professional world:** 62% of the System Engineering students consider that the activities developed will contribute to their professional life, and 35% perhaps, only 4% feel that they will not. Additional comments of students who responded with a “maybe or not” was because they focused on the question asking for their professional life as teachers in the future, a profile in which they are not interested at the moment. However, those who said “yes” contemplated the beneficial experience of developing

skills such as public speaking, self-learning, and acquired patience, which they consider can help them in their professional future.

- **Generic competencies:** 96% of the System Engineering students confirm that they have developed social responsibility by participating in the project. The adjectives used to qualify to live with the children of the public school are, in summary: excellent, good, enriching, wonderful, pleasant, rewarding, and unique. All comments suggest a favorable response to participation with children and the results obtained.

The coexistence also allowed System Engineering students to detect learning problems, family problems, low economic level, lack of affection, poor knowledge of computers before intervention due to the lack of access to technology in their homes and due to lack of economic resources, disability (visual, hearing, behavioral disorder, autism), in children in the academic unit. That resulted in increased perception and observation, essential competencies for a system engineering student.

- **Specific competencies:** 77% of the System Engineering students consider that they have learned more from programming than in their regular classes through self-learning and preparation for the course at the public school. 15% show a different opinion by answering that they had already mastered programming concepts when the research started, and 8% considered that it might have contributed to its training.
- **Others (to extract ideas for improvement)** Words taken from the comments of System Engineering students: teaching (love, like, joy, boys, and girls), motivation, encouragement (boys and girls), help, learning, living together, experience, commitment, and opportunity.

Previously described items permit answering RQ3 and RQ4; System Engineering students positively perceived the effects of using (Scratch + CARAMBA) for developing programming competencies in non-WEIRD contexts. Considering the results, the research team wants to continue using (Scratch + CARAMBA) to benefit non-WEIRD students in other places.

It is necessary to consider the proposed improvement ideas to rethink the project and continue the intervention to measure its long-term impact. For this, the research team must consider the following: i) to develop a structured lesson plan for children who have experienced difficulties in independent learning, ii) to increase the number of engineering students participating as tutors, iii) to feed the CARAMBA database with more straightforward exercises for the initialization level, iv) to increase the number of computers, v) to train more engineering students to develop programming competencies outside, vi) to increase class time in phase I, vii) to extend the introductory program for children who have no prior knowledge of computer science (peripheral device management, operating system) and, viii) to plan the classes of phase I according to age.

In summary, the followings were the main contributions of this paper:

1. To confirm the applicability of (Scratch + CARAMBA) for developing programming competencies in students regardless of their study level and social status.
2. To demonstrate the usefulness of (Scratch + CARAMBA) in developing algorithmic thinking and programming competencies in children from non-WEIRD areas.
3. To project the applicability of (Scratch + CARAMBA) for developing algorithmic thinking and programming competencies in children anywhere: children from WEIRD and non-WEIRD areas.

6 Conclusions

Following the initial diagnostic of reasoning, we can draw the following conclusions:

1. According to our research on using (Scratch + CARAMBA) to develop programming competencies, considering the obtained results, we conclude that both tools represent excellent options in the non-WEIRD context.
2. Our work successfully answers the four research questions: 1. Scratch positively affects children's learning process in non-WEIRD schools. 2. CARAMBA is a great recommender tool for assisting the development of programming competencies of children in non-WEIRD contexts. 3. Tutor's perception, university students, perceived the usefulness of Scratch + CARAMBA for developing programming competencies of children in non-WEIRD areas. 4. We want to motivate the use and continue using (Scratch + CARAMBA) for developing programming competencies in all contexts.
3. We were able to pinpoint variables that may link to the social responsibility acquired when interacting with school-age children thanks to the findings of the engineering students' final survey.
4. Due to participant exchanges of experiences that allowed for peer learning, the initiative helped engineering students acquire social responsibility.

Our work accomplished its primary goal of applying Scratch + CARAMBA to successfully develop programming competencies in children of non-WEIRD areas from an in-development country. Although other recommender tools exist with Scratch, such us (He et al., 2020), this research applied CARAMBA to highlight the research work of the first author in her Ph.D. studies. This work aligns with existing work like Cárdenas-Cobo et al. (2019), Ansari et al. (2016), He et al. (2020) and Saito and Watanobe (2020) regarding the usefulness and positive implications of using recommendation system for developing computing competencies in the education process of young students.

Appendix A Questions for the initial diagnosis

UNIVERSIDAD ESTATAL DE MILAGRO
FACULTY OF SCIENCE AND ENGINEERING
SYSTEMS ENGINEERING CAREER
COMMUNITY OUTREACH PROJECT
"COMPUTER ACADEMY"

SURVEY ADDRESSED TO CHILDREN FROM 8 TO 12 YEARS OF AGE IN THE DISTRICT OF MILAGRO

CODE: COURSE-SCHOOL-SECTION-SECTION-No. LIST **DATE:**________

INFORMATIVE DATA:

1.- Gender:
- a) Female
- b) Male

2.- People and Nationality:
- a) Indigenous
- b) Mestizos
- c) African American
- d) White
- e) Montubios
- f) Others

3.- Age group (age):
- a) Under 8 years old
- b) From 8 to 12 years old
- c) From 12 to 15 years old
- d) 15 and over

4.- Type of disability:
- a) Physical
- b) Psychological
- c) Mental
- d) Hearing
- e) Visual
- f) None

5.- Nationality:
- a) Ecuadorian
- b) Foreigner

Indicates the degree of use of the following services offered by the Internet	**YES**	**NO**	**SOMETIMES**
Do you have a computer at home?			
Do you have Internet at home?			
Do you download music?			
Do you download documents?			
Do you download images?			
Do you download videos?			
Do you log in to Facebook?			
Do you downloads research?			
The knowledge you have about computers, be it a lot or a little, who taught you?	**YES**	**NO**	**SOMETIMES**
Friends			
Father			
Mother			
Brother / Sister			
Himself / Herself			
Training course			
School			
Cyber			

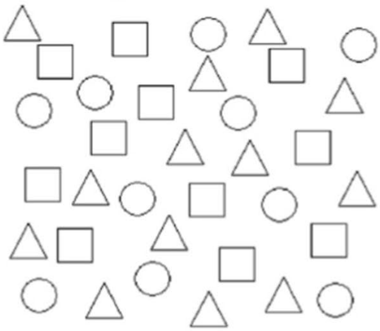
Fig. 8 Initial diagnostic test for children -Informative data

REASONING TEST FOR CHILDREN FROM 8 TO 12 YEARS OF AGE

Find out what logical reasoning ability your children have by answering the questions in this test.

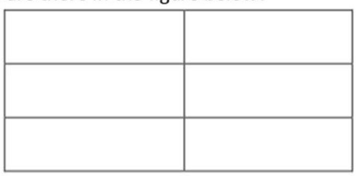
1. SACO is to ASCO as 7683 is to:
   a) 3678
   b) 3867
   c) 6783
   d) 8376

2. ¿ How many circles do you see in this picture?



   a) 7
   b) 8
   c) 9
   d) 10

3. ¿ How many rectangles are there in the figure below?



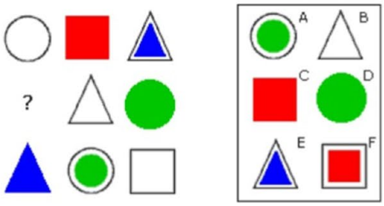
   a) 6
   b) 18
   c) 15
   d) 10

4. If 4 apples out of a dozen are rotten, how many are good?

   a) 2
   b) 4
   c) 8
   d) 6

Fig. 9 Initial diagnosis test for children—Questions part I

5. Find among the six figures on the right which one is missing in the set on the left.



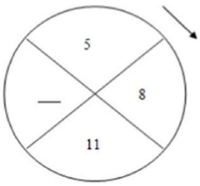
a) A
b) B
c) C
d) D
e) E
f) F

6. ¿ What is the number that completes the series?

6 - 12- 18- 24 - -36

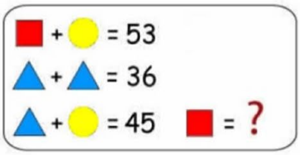
a) 34
b) 28
c) 30
d) 32

7. ¿ What is the missing number?



a) 3
b) 33
c) 13
d) 14

8. Look at the following image, think and calculate the value of the square.



a) 24
b) 26
c) 28
d) 30

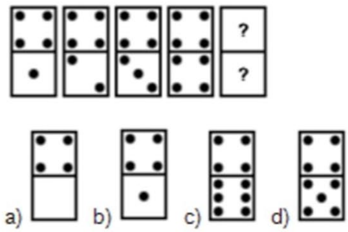
Fig. 10 Initial diagnosis test for children—Questions part II

9. Today I went to buy oranges, the sales clerk gave me 6, I ate 1 and my father ate 2, another one fell and spoiled. How many oranges do I have left?
    - a) 2
    - b) 3
    - c) 4
    - d) 5

10. What value is next in this series?
    4 - 6 - 8 - 10 - ?
    - a) 9
    - b) 11
    - c) 12
    - d) 14

11. What letter follows in this series?
    c - e - g - i - ?
    - a) k
    - b) a
    - c) l
    - d) j

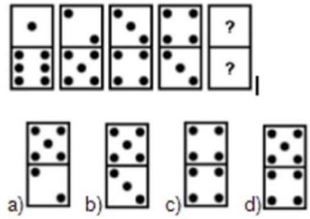
12. Which card follows in the next series?



- a) a
- b) b
- c) c
- d) d

Fig. 11 Initial diagnosis test for children—Questions part III

13. ¿ Which card follows in the next series?



a) a
b) b
c) c
d) d

14. If a fly lives 5 days and in one day it travels 12 meters, how far will it travel in 7 days?

a) 60 meters
b) 84 meters
c) 77 meters

15. If 7 cats catch 7 mice in 7 minutes, how many minutes will it take 1 cat to catch 1 mouse?

a) 7
b) 1
c) 5

16. What number is the car parked at?

a) 78
b) 87
c) 89
d) 86

17. DIDIIDIDID is to 49499494 as DIIDIIDD is to:

a) 94494499
b) 49949944
c) 49499494
d) 94944949
e) 49944949

Fig. 12 Initial diagnosis test for children—Questions part IV

Appendix B Questions to assess the computational thinking of children

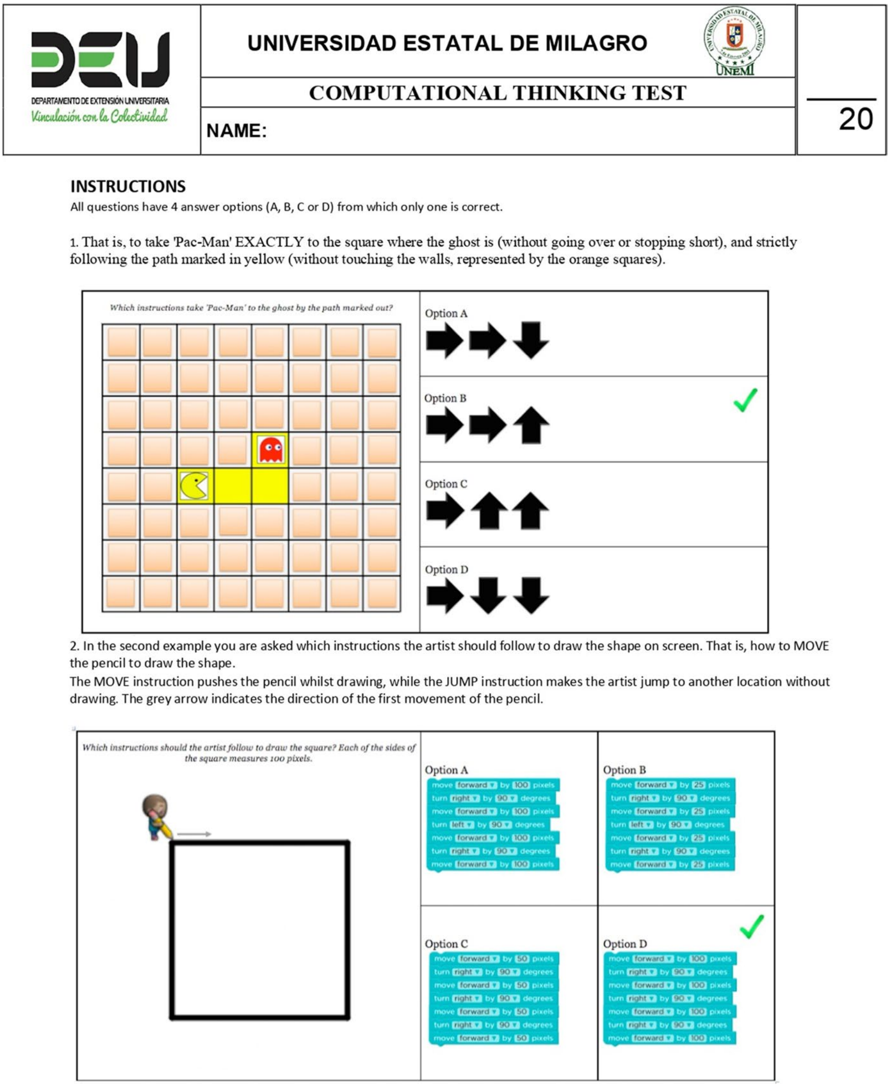
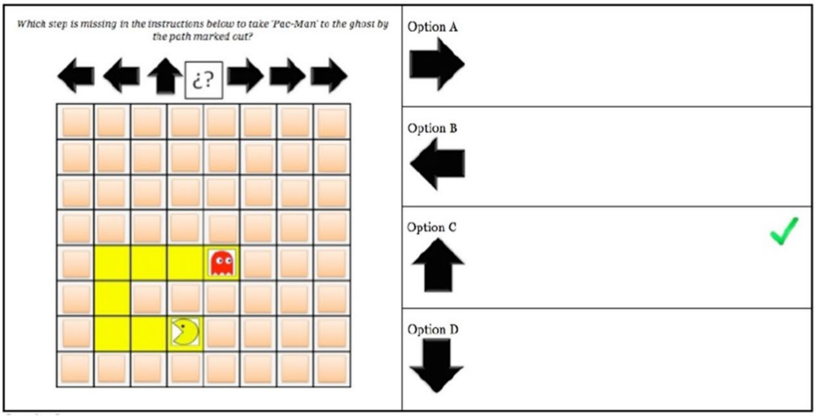
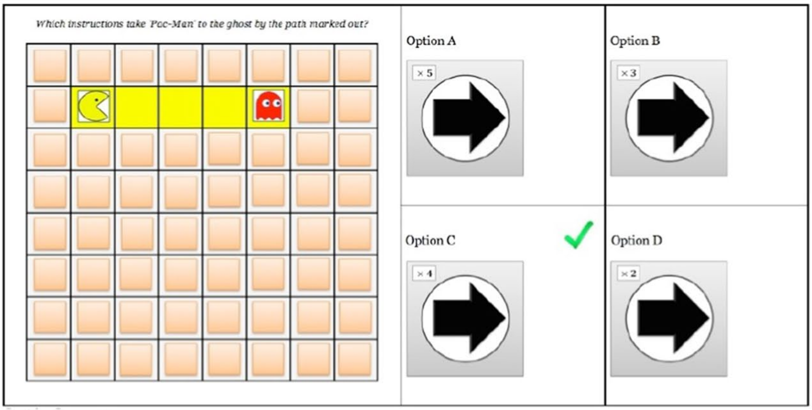


Fig. 13 Computational thinking test for children—Questions part I

3. Select the correct answer



4. Select the correct answer



5. Select the correct answer

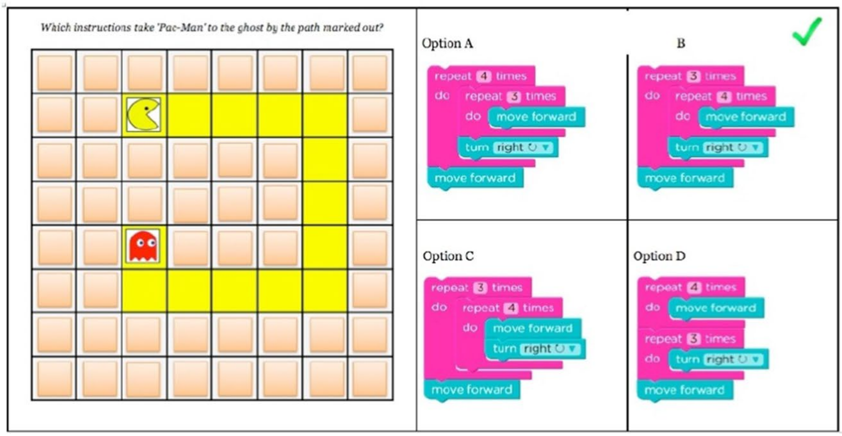
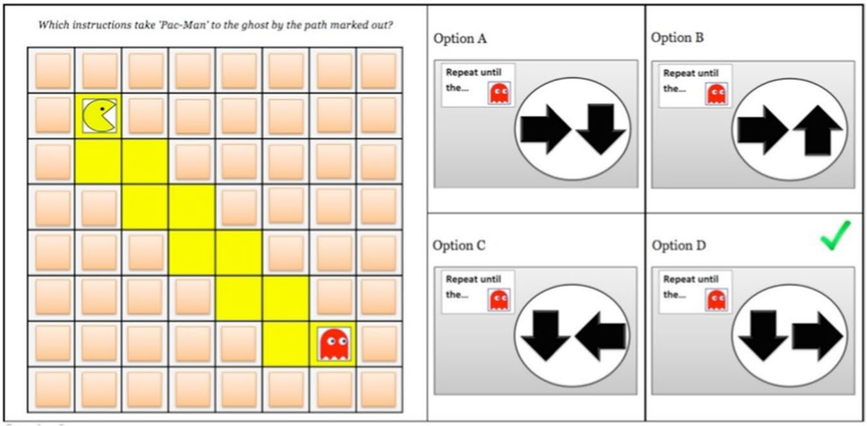
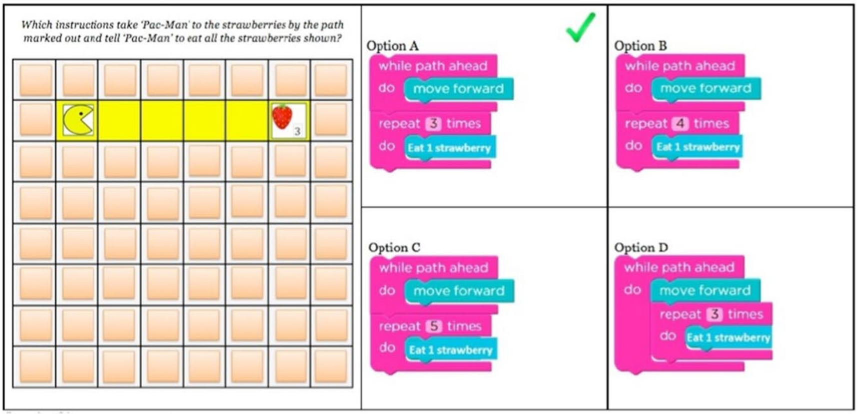


Fig. 14 Computational thinking test for children—Questions part II

6. Select the correct answer



7. Select the correct answer



8. Select the correct answer

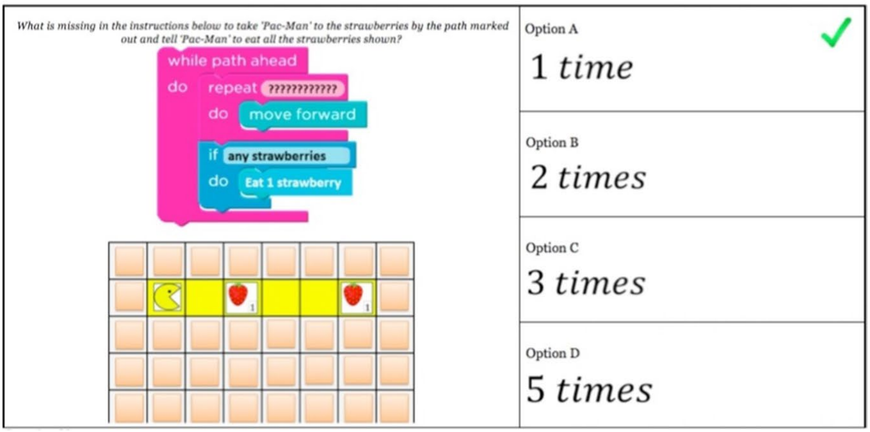


Fig. 15 Computational thinking test for children—Questions part III

Funding The authors did not receive support from any organization for the submitted work.

Data Availability The data supporting this study's findings are available from the corresponding author upon reasonable request.

Declarations

The authors declare that they have received no funding for preparing this article. The authors declare that they have no competing interests in this work. The authors contribute equally to the preparation of this article.
Editorial Policies for:
Springer journals and proceedings: https://www.springer.com/gp/editorial-policies Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies *Scientific Reports*: https://www.nature.com/srep/journal-policies/editorial-policies
BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

Conflicts of interests/Competing interests None of the authors receive compensation for participating in the article.

References

Ali, S., Hafeez, Y., Humayun, M., Jamail, N. S. M., Aqib, M., Nawaz, A. (2022). Enabling recommendation system architecture in virtualized environment for e-learning. *Egyptian Informatics Journal, 23*, 33–45. URL: https://www.sciencedirect.com/science/article/pii/S111086652100030X. https://doi.org/10.1016/j.eij.2021.05.003

Ansari, M. H., Moradi, M., NikRah, O., Kambakhsh, K. M. (2016). Coders: A hybrid recommender system for an e-learning system, in: 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), pp. 1–5. https://doi.org/10.1109/ICSPIS.2016.7869884

Arnett, J. J. (2008). The neglected 95%: Why american psychology needs to become less american. *American Psychologist, 63*, 602.

Bennedsen, J., & Caspersen, M. E. (2012). Persistence of elementary programming skills. *Computer Science Education, 22*, 81–107.

Byun, S. Y., Meece, J. L., & Irvin, M. J. (2012). Rural-nonrural disparities in postsecondary educational attainment revisited. *American Educational Research Journal, 49*, 412–437.

Byun, S. Y., Meece, J. L., & Agger, C. A. (2017). Predictors of college attendance patterns of rural youth. *Research in Higher Education, 58*, 817–842.

Cabo, C. (2019). Student progress in learning computer programming: Insights from association analysis. IEEE Frontiers in Education Conference (FIE), Covington, KY, pp. 1-8, https://doi.org/10.1109/FIE43999.2019.9028691

Cain, A., Woodward, C. J. (2012). Toward constructive alignment with portfolio assessment for introductory programming. Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012, Hong Kong, China, pp. H1B-11-H1B-17, https://doi.org/10.1109/TALE.2012.6360322

Cardenas-Cobo, J. (2020). *Enhancing the learning of programming using Scratch: a recommender-systems-based approach in non WEIRD communities*. Spain: Ph.D. thesis. University of Seville.

Cárdenas, J., Benavides Cuevas, D. F., D'Armas, M., Vinueza Morales, M. (2017). Programación con la herramienta Scratch+Caramba: Una experiencia de aprendizaje significativo. 15th LACCEI International MultiConference for Engineering, Education, and Technology: "Global Partnerships for Development and Engineering Education" (2017), pp. 1–6., Latin American and Caribbean Consortium of Engineering Institutions. Global Partnerships for Development and Engineering Education: Proceedings of the 15th LACCEI International Multi-Conference for Engineering, Education and Technology. pp. 107–144.

Cárdenas-Cobo, J., Puris, A., Novoa-Hernández, P., Galindo, J. A., Benavides, D. (2019). Recommender systems and scratch: An integrated approach for enhancing computer programming learning. IEEE Transactions on Learning Technologies, 13(2), 387–403. https://doi.org/10.1109/TLT.2019.2901457

Cárdenas-Cobo, J., Puris, A., Novoa-Hernández, P., Parra-Jiménez, Á., Moreno-León, J., & Benavides, D. (2021). Using scratch to improve learning programming in college students: A positive experience from a non-weird country. *Electronics, 10*, 1180.

Durak, H. Y. (2020). Modeling different variables in learning basic concepts of programming in flipped classrooms. *Journal of Educational Computing Research, 58*, 160–199.

Fagerlund, J., Häkkinen, P., Vesisenaho, M., Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education, 29*, 12–28.

Fry, R. (2002). Latinos in higher education: Many enroll, too few graduate. Pew Hispanic Center. https://fordhaminstitute.org/national/commentary/latinos-higher-education-many-enroll-too-few-graduate. Accessed 25 Aug 2023.

Gomes, A., Mendes, A. J. (2007a). An environment to improve programming education, in: Proceedings of the 2007a International Conference on Computer Systems and Technologies, Association for Computing Machinery, New York, NY, USA. URL, p. 1330691, https://doi.org/10.1145/1330598.1330691

Gomes, A., Mendes, A. J. (2007b). Learning to program-difficulties and solutions, in: International Conference on Engineering Education–ICEE, University of Coimbra, Portugal, pp. 283–287.

He, Y. T., Guo, B. J., Lu, J., Xu, Y. P., Gong, M. (2020). Research of scratch programming recommendation system based on med and knowledge graph, in: 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), pp. 2158–2163. https://doi.org/10.1109/ICMCCE51767.2020.00469

Henrich, J., Heine, S. J., & Norenzayan, A. (2010a). Beyond weird: Towards a broad-based behavioral science. *Behavioral and Brain Sciences, 33*(2–3), 111–135. https://doi.org/10.1017/S0140525X10000725

Henrich, J., Heine, S. J., & Norenzayan, A. (2010b). Most people are not weird. *Nature, 466*, 29.

Hernán, I., Velázquez, J. A., Lázaro, C. A. (2006). Dos herramientas educativas para el aprendizaje de programación: Generación de comentarios y creación de objetos. VII Congreso Internacional de Interacción Persona-Ordenador, Centro de Estudios Universitarios de Puertollano, Ciudad Real, España, pp. 325–334.

Hsu, T. C., Chang, S. C., Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310. URL: https://www.sciencedirect.com/science/article/pii/S0360131518301799, https://doi.org/10.1016/j.compedu.2018.07.004

Jones, D. (2010). A weird view of human nature skews psychologists' studies. Science (New York, N.Y.), 328, pp. 1627.https://doi.org/10.1126/science.328.5986.1627

Kalelioğlu, F. (2015). A new way of teaching programming skills to k-12 students: Code. org. *Computers in Human Behavior, 52*, 200–210.

Keeler, L. W., & Bernstein, M. J. (2021). The future of aging in smart environments: Four scenarios of the united states in 2050. *Futures, 133*, 102830.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin, 37*, 14.

Lee, V., & Frank, K. (1990). Student characteristics that facilitate transfer from two-year to four-year colleges. *Sociology of Education, 63*, 178–193.

López-Escribano, C., & Sánchez-Montoya, R. (2012). Scratch y Necesidades Educativas Especiales: Programación para todos. *RED Revista De Educación a Distancia, 34*, 2–14.

Martínez, J. D. (2021). An overview of the end-of-life tires status in some latin american countries: Proposing pyrolysis for a circular economy. *Renewable and Sustainable Energy Reviews, 144*, 111032.

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education, 62*(2), 77–90. https://doi.org/10.1109/TE.2018.2864133

Medeiros R. P., Ramalhomand G. L., & Falcão T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. IEEE Transactions on Education, pp. 1–14.

Mesoudi, A., Magid, K., & Hussain, D. (2016). How do people become w.e.i.r.d.? migration reveals the cultural transmission mechanisms underlying variation in psychological processes. *PLoS ONE, 11*, 1–17.

Niess, M. (2005). Preparing teachers to teach science and mathematics with technology: Developing a technology pedagogical content knowledge. *Teaching and Teacher Education, 21*, 509–523.

Pope, S. M., Fagot, J., Meguerditchian, A., Washburn, D. A., & Hopkins, W. D. (2019). Enhanced cognitive flexibility in the seminomadic himba. *Journal of Cross-Cultural Psychology, 50*, 47–62.

Ramírez-Montoya, M. S., Andrade-Vargas, L., Rivera-Rogel, D., & Portuguez-Castro, M. (2021). Trends for the future of education programs for professional development. *Sustainability, 13*, 7244.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: Programming for all. *Communications of the ACM, 52*, 60–67.

Roberts, S. O., Bareket-Shavit, C., Dollins, F. A., Goldie, P. D., & Mortenson, E. (2020). Racial inequality in psychological research: Trends of the past and recommendations for the future. *Perspectives on Psychological Science, 15*, 1295–1309.

Rodríguez-Martínez, J. A., González-Calero, J. A., Sáez-López, J. M. (2020). Computational thinking and mathematics using scratch: an experiment with sixth-grade students. *Interactive Learning Environments, 28*, 316–327. URL: arXiv: https://doi.org/10.1080/10494820.2019.1612448

Rojas-Valdés, P., Vidal-Silva, C., de La Fuente, C. (2022). Successful development of problem-solving and computing programming competences in children using Arduino. International Symposium on Measurement and Control in Robotics (ISMCR), Houston, TX, pp. 1–6. https://doi.org/10.1109/ISMCR56534.2022.9950596

Román-González, M. (2016). Codigoalfabetización y pensamiento computacional en Educación Primaria y Secundaria: validación de un instrumento y evaluación de programas. Tesis. Universidad Nacional de Educación a Distancia (España). Escuela Internacional de Doctorado. Programa de Doctorado en Educación. https://e-spacio.uned.es/fez/eserv/tesisuned:Educacion-Mroman/ROMAN_GONZALEZ_Marcos_Tesis.pdf. Accessed 01 Apr 2023.

Ruggeri, K., Bojanic, L., Bokhorst, L., Jarke, H., Marev, A. S., OjinagaAlfageme, O., & Norton, S. (2019). Editorial: Advancing methods for psychological assessment across borders. *Frontiers in Psychology, 1*, 503.

Saito, T., & Watanobe, Y. (2020). Learning path recommendation system for programming education based on neural networks. *International Journal of Distance Education Technologies (IJDET), 18*, 36–64.

Sánchez, A., Font, V., & Breda, A. (2022). Significance of creativity and its development in mathematics classes for preservice teachers who are not trained to develop students' creativity. *Mathematics Education Research Journal, 34*, 863–885.

Sun, Y., & Liu, X. (2022). How design technology improves the sustainability of intangible cultural heritage products: A practical study on bamboo basketry craft. *Sustainability, 14*, 12058.

Thieman, G. (2008). Using technology as a tool for learning and developing 21st century skills: An examination of technology use by pre-service teachers with their k-12 students. *Contemporary Issues in Technology and Teacher Education, 8*, 342–366.

Tomczyk, L ., Eliseo, M. A., Costas, V., Sánchez, G., Silveira, I. F., Barros, M. J., Amado-Salvatierra, H. R., Oyelere, S. S. (2019). Digital divide in latin america and europe: Main characteristics in selected countries. 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1–6, https://doi.org/10.23919/CISTI.2019.8760821

Tupac-Yupanqui, M., Vidal, C. L., Sánchez-Ortiz, A., Pereira, F. (2021). Experiencias y beneficios del uso de Arduino en un primer curso de programación. Formación universitaria, Centro de Información Tecnológica, Universidad de la Serena, Chile, 14(6), 87–96. https://doi.org/10.4067/S0718-50062021000600087

van Der Vleuten, C. P., & Schuwirth, L. W. (2019). Assessment in the context of problem-based learning. *Advances in Health Sciences Education, 24*, 903–914.

Vidal Silva, C., & Pavesi Farriol, L. (2005). Desarrollo de un sistema de adquisición y tratamiento de señales electrocardiográficas. *Revista Facultad De Ingeniería-Universidad De Tarapacá, 13*, 39–46.

Vidal-Silva, C., Serrano-Malebran, J., Pereira, F., 2019. Scratch and arduino for effectively developing programming and computing-electronic competences in primary school children, in: 2019 38th International Conference of the Chilean Computer Science Society (SCCC), pp. 1–7. https://doi.org/10.1109/SCCC49216.2019.8966401

Wing, J. M. (2010). Computational thinking: What and Why? thelink - The Magazine of the Varnegie Mellon University School of Computer Science 49, 1–6. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why. Accessed 25 Jun 2023.