frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 25 October 2022
DOI 10.3389/fcomp.2022.1022778

OPEN ACCESS

EDITED BY
Simon K. S. Cheung,
Hong Kong Metropolitan University,
China

REVIEWED BY
Sandra Cano,
Pontificia Universidad Católica de
Valparaiso, Chile
Patricia Paderewski,
University of Granada, Spain

*CORRESPONDENCE
Matthias Funk
gunther.ma.funk@uac.p
José Cascalho
jose.mv.cascalho@uac.pt

SPECIALTY SECTION
This article was submitted to
Digital Education,
a section of the journal
Frontiers in Computer Science

RECEIVED 18 August 2022
ACCEPTED 28 September 2022
PUBLISHED 25 October 2022

CITATION
Funk M, Cascalho J, Santos AI, Pedro F,
Medeiros P, Amaral B, Domingos M,
Ramos A and Mendes A (2022) A
simple interactive robot to promote
computational thinking.
Front. Comput. Sci. 4:1022778.
doi: 10.3389/fcomp.2022.1022778

A simple interactive robot to promote computational thinking

Matthias Funk[1,2]*, José Cascalho[1,2]*, Ana Isabel Santos[3], Francisco Pedro[2,4], Paulo Medeiros[1,2], Bárbara Amaral[2], Manuel Domingos[2,4], Alberto Ramos[2,5] and Armando Mendes[1,2]

[1]Faculty of Sciences and Technology, University of the Azores, Ponta Delgada, Portugal, [2]GRIA and Artificial Intelligence and Computer Science Laboratory, Porto, Portugal, [3]Faculty of Social Sciences and Humanities, NICA-Interdisciplinary Centre for Childhood and Adolescence, University of the Azores, Ponta Delgada, Portugal, [4]Escola Básica Integrada de Rabo de Peixe, Ribeira Grande, Portugal, [5]Associação para a Promoção do Desenvolvimento dos Açores (APRODAZ), Ponta Delgada, Portugal

This paper described the functionalities of a simple robot adopted in the classroom to promote computational thinking (CT) in the context of the project PeCOT—computational thinking with tangible objects. This robot, similar to some commercial educational robots, was developed to be used in primary education classroom contexts (second, third, and fourth grades) by children between 8 and 10 years old. PeCOT is a pedagogical intervention project and research project that will run over the next 2 years. The aim of this paper is to present the robot, how it works, and some activities that can be implemented in the educational context to promote CT skills, including learning in different subject areas, such as mathematics or natural science. Thus, we begin by presenting a general description of the robot. Next, we identify activities and formalize the programming steps for each of the activities. Finally, we discuss the potential that the robot and the proposed activities may have to promote the development of CT skills.

KEYWORDS

educational robotics, computational thinking, tangible programming, playgrounds, PeCOT

1. Introduction

Computational thinking (CT), recognized by the European Union as fundamental and which should be promoted from the early years, refers to the articulated development of a set of practices that contribute to problem-solving, particularly those related to programming. As Bakala et al. (2021) pointed out in a systematic review on children, robots, and CT, this is a recent concept that is still under construction, thus there are several looks that converge toward its understanding, an idea that is shared by other researchers (Beecher, 2017; Li et al., 2021). However, Bers et al. (2022, p. 30) stated that "It is widely agreed that CT involves a broad set of analytic and problem-solving skills, dispositions, and habits, rooted in computer science but universally applicable," recognizing that it is essential to cross the boundaries of computing to leverage CT alongside other areas of knowledge, constituting another of the many literacies that should be built by the citizens of the twenty-first century (Wing, 2006). It is in this context that Ramos et al. (2022, p. 7) proposed, as part of a study on the training needs of educators and teachers for the introduction of programming and robotics in Portugal, that CT can be seen as a didactic "resource," seeking to "think about the content and form of the teaching and learning processes of programming and other computer science fields," given its relevance to the development and learning of children in aspects related to computing and digital literacy, but also to aspects of their social and school development (Funk et al., 2021), enhancing gender balance, equity, and inclusion (Bocconi et al., 2022). Bocconi et al. (2022) systematized a set of CT skills that are directly associated with programming and computing tasks and another set of skills associated with problem-solving in a more general way (Figure 1), concluding that there are six skills considered as the core of CT (abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization).

In Portugal, and included in the New Essential Mathematics Learning (Ministry of Education, 2021), the CT competencies that are intended to be integrated into the curriculum of the 1st cycle of basic education are specifically related to problem-solving, decomposition, debugging, pattern recognition, algorithms, programming, and robotics. These CT skills can be developed using different pedagogical strategies, which may or may not include technological resources, in particular robots, which can take on different characteristics or be used through different interfaces—tangible, graphical, and hybrid (Sapounidis et al., 2015; Strawhacker and Bers, 2015).

In the case of robotics, educational activities that include robots "offer opportunities for students to explore, create, and apply knowledge to solve real world problems" Ching et al. (2019, p. 591); to work on CT skills (Sáez-López et al., 2019; Žáček and Smolka, 2019; Uşengül and Bahçeci, 2020); and simultaneously allow leveraging learning objectives that lead children to learn concepts linked to different subjects, such as Mathematics, Physics, or Science (Eteokleous and Ktoridou, 2014; Ching et al., 2019; Uşengül and Bahçeci, 2020); as well as promoting various skills, linked to their personal and social formations, such as cooperation, creativity, attention, or motivation for learning (Somyürek, 2015; Sapounidis et al., 2019; Lopez-Caudana et al., 2020).

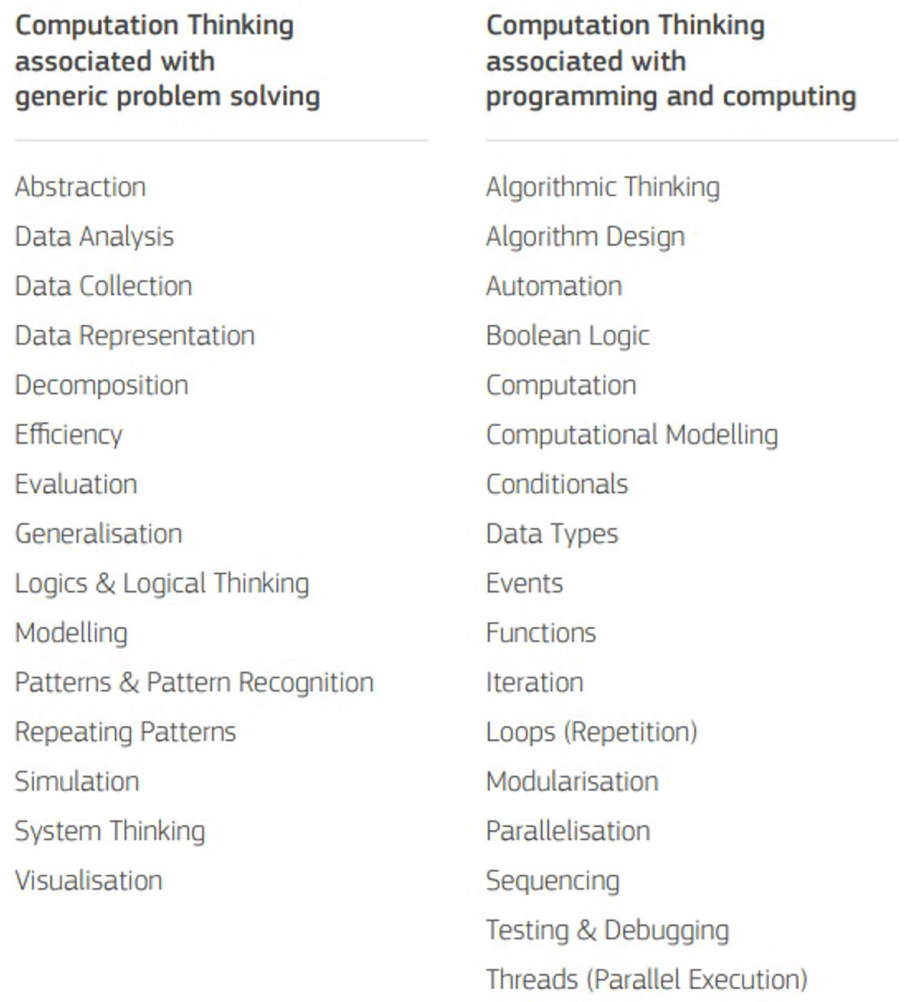
Thus, the use of educational robotics involves processes of motivation, collaboration, construction, and reconstruction on the part of its different actors, which is not always easy to manage since allied to technical requirements such as interactivity or low cost, there are other aspects that must be taken into account, such as the diversity of educational programs, the dependence on certain structures, or the necessary teacher training (Mondada et al., 2017). In any case, for its implementation, it is necessary to use concepts from different disciplines to build models, leading students to a rich interdisciplinary experience. The robot as a work tool allows the creation of new forms of interaction with the world and, as learning is fundamentally a social experience of interaction, through language and action, it is enhanced with its use in the classroom from a constructivist perspective, through physical and interactive learning experiences (Schneider et al., 2011; Somyürek, 2015), which lead children to seek to solve problems jointly and cooperatively, and autonomously, ensuring the centrality of the individual in the construction of knowledge and allowing for cognitive and effective results.

In this sense, in the construction of a robotic model, the collaborative process takes place when problems are analyzed and solved as a group, and autonomy is exercised to the extent that each element of the group has responsibility for a part of the solution, and with respect for the other individuals. Each one has the responsibility for his/her own knowledge and for the group. Everyone must participate in the solution. In this way, the doubt of one and the certainty of the other makes the group grow and develop.

On the one hand, thinking about the introduction of educational robotics in a classroom context implies simultaneous thinking about pedagogical strategies that lead children to become interested in its use, but to do so with pedagogical intentionality that leads them to learn. This implies thinking about the whole contexts of intervention beyond the mere use of the robot, it implies looking at the playgrounds, the scenarios where problems are raised and where thinking is materialized in the form of action to solve them. As Bers (2018) pointed out, looking at these learning experiences conducted through educational robotics implies understanding the type of experiences that are best suited to enhance CT, as well as all the other skills that, in parallel, are promoted and that we listed above, because "children have many options for things to do within the environments" (p. 1).

Chevalier et al. (2020) designated this more comprehensive look as Educational Robotics System (ERS), which results from the combination of three main components: one or more educational robots, an interaction interface allowing the user to communicate with the robot and one or more tasks to be solved on a playground, proposing a model that they called creative computational problem solving (CCPS). According to this model, different phases are described through which children should go through to develop CT skills—understanding the problem (USTD), generating ideas (IDEA), formulating the behavior (FORM), programming the behavior (PROG), and evaluating the solution (EVAL). This model, based on other existing models, has the particularity that, in addition to looking at the activity developed by children, it provides teachers with explicit guidance on the implementation of robotics activities in this context, also focusing on training teachers to better promote this type of pedagogical intervention, an aspect that emerges in many investigations as necessary to leverage digital and technological literacy skills along with CT (Campbell et al., 2018; Chalmers, 2018; Bevčič and Rugelj, 2020).

In this sense, the aim of this work is to analyze the Azbot-1C robot and its functionalities and, at the same time, to reflect on a set of pedagogical strategies related to the construction of



Computation Thinking associated with generic problem solving	Computation Thinking associated with programming and computing
Abstraction	Algorithmic Thinking
Data Analysis	Algorithm Design
Data Collection	Automation
Data Representation	Boolean Logic
Decomposition	Computation
Efficiency	Computational Modelling
Evaluation	Conditionals
Generalisation	Data Types
Logics & Logical Thinking	Events
Modelling	Functions
Patterns & Pattern Recognition	Iteration
Repeating Patterns	Loops (Repetition)
Simulation	Modularisation
System Thinking	Parallelisation
Visualisation	Sequencing
	Testing & Debugging
	Threads (Parallel Execution)

FIGURE 1
Concepts concerning computational thinking (CT) skills development, adapted from Bocconi et al. (2022).

learning scenarios that allow researchers and teachers to build better practices and more in line with the educational context where the PeCOT project will be implemented.

2. Educational robots and CT: Project PeCOT

2.1. CT with tangible objects

Computational thinking with tangible objects (PeCOT) is a project that seeks to implement the use of an educational robot in a classroom context in the 1st cycle of basic education, for children aged between 8 and 10 years old, in order to evaluate the mobilization of CT skills in children who use it. Simultaneously, this project aims to monitor and analyze the pedagogical strategies implemented by teachers, which can be identified as enhancing CT. In particular, the objectives of PeCOT are as follows:

a) to analyze the way children use the robot in the challenges that will be proposed, taking into account the different programming interfaces available;
b) to evaluate the role of the interface in the way learning activities are developed, that is, in the type of pedagogical strategies to be implemented;
c) to identify the strategies that best promote CT skills.

In methodological terms, this project will be implemented during one school year in a school located in a socially disadvantaged context. It is organized into two distinct moments—the use of a robot with a coupled tangible interface and a robot with a decoupled tangible interface. The two moments followed the same logic: presentation of the robot to teachers, exploring its potential and structuring the challenges to be implemented; preparation of a model for assessing CT skills, based on the CCPS model (Chevalier et al., 2020); experimentation of the activities in the classroom context and data collection; and analysis of the results. In terms of evaluation, the methodology followed was based on Chevalier et al. (2020) and Chevalier et al. (2022), seeking to encourage active participation from the teachers in the design of the proposed activities. Following the ERS (Giang, 2020), and in the evaluation of the activity to be performed in the classroom, in partnership with the researchers, adapting the strategies to be followed to ensure learning that enhances CT. Simultaneously, and during the process of implementing the activities, children were assessed on how they participate in the proposed tasks (through observation) and their perspective on these same tasks (through an interview). In particular, throughout the process, the following elements will be assessed, taking into account the proposal of Chevalier et al. (2022):

(a) **Utility**-how the use of the robot promotes transversal thinking skills, in particular CT, but also other skills such as communication, reasoning and problem solving, or interpersonal skills and technological literacy (Martins et al., 2017).
(b) **Usability**-the ease of use of the proposed system in close connection with the skills that the teacher needs to implement in the classroom.
(c) **Acceptance**-the degree of autonomy in performing the proposed tasks, coupled with motivation and persistence in the face of difficulties that may arise.

The robot to be used, its characteristics, and the way of functioning are described in the following sections.

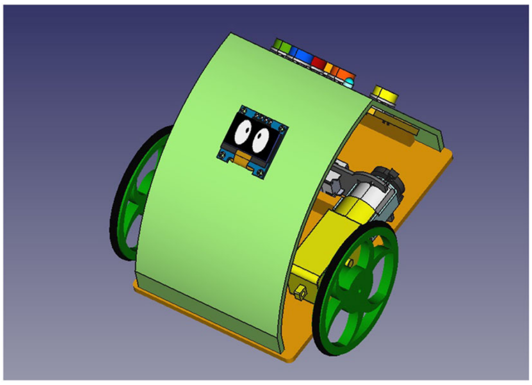


FIGURE 2
The final version of the robot Azbot-1C used in the PeCOT project.

TABLE 1 Hardware components of the robot Azbot-1C.

Qty	Description	Module
1	ESP32 DevKitC	Main board
1	DRV8833 H-Bridge	Main board
6	Push Button with color cap	Control board
3	LED 5 mm (Three LED version)	Control board
3	220 Ohm Resistor	Control board
1	NeoPixel Diffused 8 mm LED (RGB LED version)	Control Board
2	DFRobot 6 V micro DC motor with encoder	
2	MR18650 batteries	Battery pack/module
1	WH-2S80A BMS	Battery pack/module
1	NCP1117 3.3 V	Battery pack/module
1	NCP1117 5 V	Battery pack/module
4	10 uF Capacitor	Battery pack/module
1	Display LCD i2c SSD1306 0.96 inch 128 X 64	Display

2.2. Azbot-1C: A simple interactive robot

Azbot-1C is a simple interactive robot with two actuators with encoders and a straightforward interface with buttons for programming. It is specifically designed to be used by children between 8 and 10 years old. The robot is programmed using an interface dubbed control unit. The discussion concerning its design led us to consider an option with five buttons to control the movement of the robot and one additional button to command the controller state. The most recent prototype of the robot is depicted in Figure 2. The picture shows the front of the robot with a small screen that allows communication with users, providing feedback on the success of the robot's activity. It also shows the back of the robot with the control unit (see Figure 3). The battery pack is protected and has a connector to recharge the batteries.

2.2.1. Technical description

Azbot-1C (Pedro et al., 2021) is a robot that uses ESP32 system on a chip (SoC) micro-controller developed by Espressif (Espressif, 2022) as the processing device. It has an integrated wireless communication module which is a feature that is important in the context of the project PeCOT. Table 1 lists the hardware components used by the robot. It is equipped with two *DFRobot 6V micro DC motors* controlled by a *DRV8833 H-bridge* driver operating in a voltage range from 2.7 to 10.8 V. Each motor has a *120:1 gearbox* and an integrated quadrature encoder providing 16 pulse single per round. We also had built-in protection against reverse-voltage, under-voltage, over-current, and over-temperature because of the targeted ages.

Figure 3 shows the option adopted for the control unit. To program the robot, children must first select the programming mode by pressing the control button on the left, which will

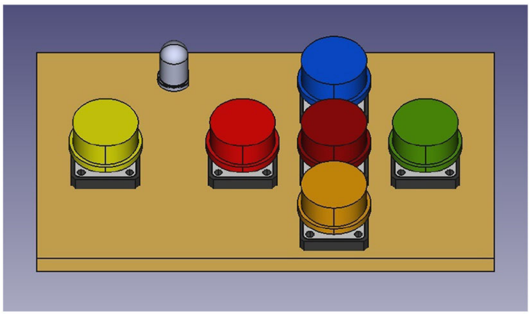


FIGURE 3
A final version of the control unit with a control button on the left and the five programming buttons on the right. The RGB-led, signaling the state of interaction, is on the top left. The left (L), right (R), forward (F), and backward (B) correspond to the left, right, top, and bottom positions on the board, respectively. In the middle, the stop or halt button (H).

set RGB-led to the red color. Then, can add a sequence of commands corresponding to pressing each one of the five buttons to make the robot move forward, backward, turn left or right, or stop. Finally, the robot becomes ready to execute the program by pressing the control button again, which switches the color RGB-led to green. The robot is calibrated to execute one step forward or backward at a certain distance and turn 90° to the left or right. This behavior can be easily modified if necessary.

Each button has its own color. Two colors were specifically chosen: red for the left turn and green for the right turn, corresponding to port and starboard colors in maritime and aerial navigation. Blue was chosen for the forward movement and orange for the backward movement. The halt button is a darker color (brown or black). A light yellow button was chosen to change the programming mode. In any case, the layout of each button in the control unit is nowadays easily understood through the most diverse devices used by children on a daily basis.

The center of each wheel forms the center of a coordinating system where the horizontal line is the *x-axis* and the vertical line is the *z-axis*. The associated coordinating systems of both wheels are parallel aligned by the *y-axis* passing through both centers.

The wheel rotation angle (WRA) is determined by the rotation's time and angular velocity. The x-movement length is calculated by the constant *PI*, the rotation angle parameter, and the diameter of the wheel. The number of robot encoders ticks is used to make this calculation. Pressing the button *F* or *B* produces an equal value of WRA, distinct only by its inverted sign. The movement results in the robot's position changing along the x-axis. The chosen distance corresponds to the robot's translation unit (TU) and has an implicit executing time (ET).

The robot turns as follows: one of the wheels rotates clockwise while the other rotates in an anti-clockwise direction. The robot rotates in the *x-y plain* in a clock-wise direction by pressing the *R* button and in anti-clock-wise direction by pressing the *L* button. The chosen rotation corresponds to the robot's rotation unit (RU). The *H* button means that both wheels for ET will stop. This halt command is used to highlight that a specific location was achieved through a certain trajectory i.e., it achieved a certain milestone. This button extends the semantics associated with the children's programming activity, as detailed in the following sections.

2.2.2. Options adopted to build the robot

The robot Azbot-1C follows a general strategy to build educational robots (Pérez-Vázquez et al., 2019) which includes a KIT, i.e., that can be assembled and disassembled, inviting children to understand its inner mechanisms. It is built to be adopted by public schools and so it must be accessible to school's budget, and it assumes the philosophy of do-it-yourself, with free access to the online repository (https://github.com/Robotics-and-AI-Group-of-UAc/Azobopi).

In the context of the commemoration of Children's Day activities, an initial experiment using the Azbot-1C took place. Four groups of students from Rabo de Peixe Elementary school with ages between 7 and 10 years old were invited to draw on a sheet of paper of places of their daily lives (e.g., home, school, shop, or football field) as depicted in Figure 4. The robot was then programmed with teachers' help to establish a route on the sheet according to the programmed instructions, mimicking the daily activity of each one of them. This first *free* scenario was a first interaction with the robot, with positive feedback from children and a confirmation of their enthusiasm in using and programming the robot along the activity. In this interaction, no data was collected in a formal and systematic way. The aim was to understand whether the children found the robot interesting and were enthusiastic about using it in the classroom. Furthermore, primary school teachers had initial contact with the robot, allowing the team to adjust some of the robot's features (e.g., adding the small screen in front of the robot). The PeCOT project contemplates the possibility of making the necessary adaptations to the robot's interface and design in case teachers consider them important for accomplishing the planned tasks.

Additionally, it is important to highlight that some commercial robots on the market have similar features, like Botley or Bee-Bot, for example, but the great advantage of the Azbot-1C is that it is an easy-to-assemble, low-cost kit, a key aspect, as Mondada et al. (2017) pointed out. It is also a robot that allows, beyond all the learning in its construction, a student's affection for the robot thus generating an extra motivation to solve the problems that will be proposed.

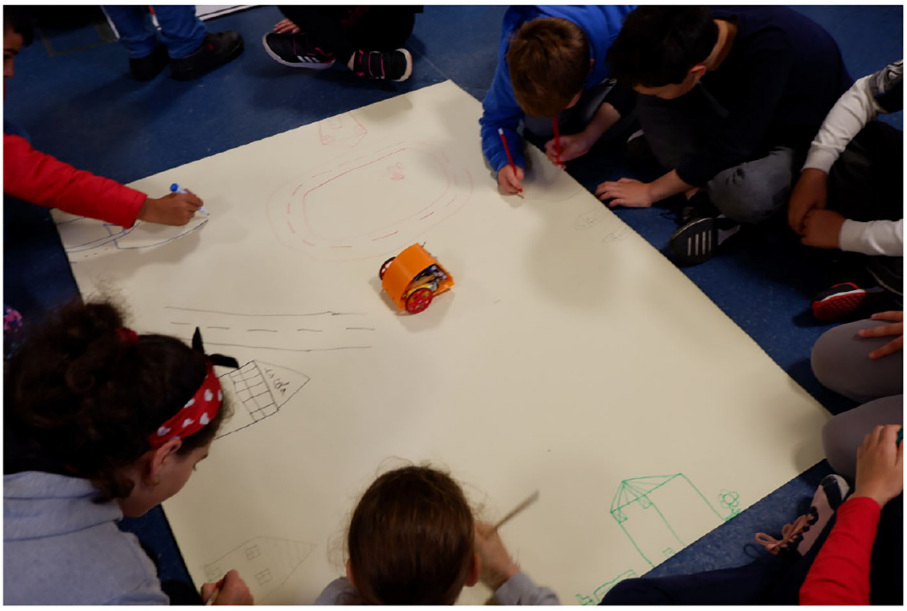


FIGURE 4
First testing of Azbot-1C in a free activity at the Rabo de Peixe elementary school.

2.3. Looking into scenarios to promote CT skills

The concept of ERS (Giang et al., 2022) identifies the artifacts associated with educational robotics, i.e., the Robot, the Interface, and the Playground. As mentioned in Chevalier et al. (2020), activities using educational robots are typically based "on three main components: one or more educational robots, an interaction interface allowing the user to communicate with the robot, and one or more tasks to be solved on a playground."

We identify different possible playgrounds with suggestive problem-solving activities inspired by activities used in classrooms for children aged between 8 and 10 years old. These playgrounds (or boards) measure 1.5 by 1.5 m fully open on the floor of the classroom. They are made of canvas with a smooth surface on which the robots can slide. Some of the selected playgrounds were previously tested in the context of the Robotics Festival—AZORESminiBOT, where it was possible to identify some clues on how to use them successfully in the classroom, as well as necessary adaptations to their format (Santos et al., 2022). Moreover, primary school teachers helped the team to adjust some of the contents on the boards. All these feedbacks were taken into account in the playgrounds now proposed. Then, we explored some of the robot's features by considering some of the steps to be given by the children when solving the problems. Finally, we explored possible connections to subjects discussed in a classroom to formalize the programming activity that must be considered to solve the problems in the proposed playgrounds and identified the CT skills each of those examples can foster.

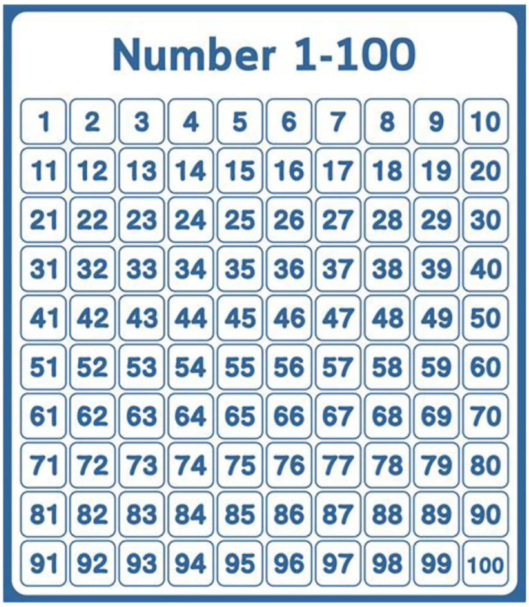
2.3.1. Educational number chart 1–100

The first playground is a table with squares, as depicted in Figure 5. This table is usually dubbed educational number chart $1 - 100$, and it is used as a scaffold to support addition and subtraction in the context of the decimal number system. The educational robot is placed in a random a position. Then, children are challenged to program the robot to, given a number b, move to the position corresponding to the sum $a + b$ with a result less than or equal to 100. As expected, the robot moves between squares when pressing the forward or backward button. Naturally, the robot movement pattern is interesting to discuss with children.

In this scenario, two possible situations can occur:

i. The result $a + b$ is within a simple forward move of b steps on the table, and the robot will only have to move this square forward. For example, $32 + 5 = 37$;
ii. The result $a + b$ is out of the board if the robot moves with b steps forward as in $32 + 9$.

While the first case is trivial, the second case arises the problem of identifying an algorithm such that the robot should end in the correct place in every case. Mathematically, it will be necessary to elaborate a function to keep the robot on the board while moving to the final square. The problem's solution is easy



Number 1-100

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

FIGURE 5
An example of an educational number chart 1–100.

to find and could be interesting to discuss in class. For example, if the term $a + b$ jumps outside the right side of the row, the equation $a+b = a-(10-b)+10$ indicates an alternative: first using the subtrahend $10 - b$ by stepping back this amount of squares then going one row down as equivalent of the addition with 10.

Analogously, children can be challenged to program the robot to solve a $a - b$ problem. Again, there are two cases. Either the result is in the white table, or the result is in the underflow area. In the latter case, they will have to discover a solution to the problem, e.g. $a - b = a + (10 - b) - 10$. They will only have to move $10 - b$ squares to the right and one square up (-10).

2.3.2. Isomorphic problems using the same path

As a second example, a set of playgrounds showed how a program can be interpreted differently using different board contexts, e.g., different semiotic signs associated to the squares.

Figure 6 presents the same path crossing through two different boards. In the first one, the squares are associated with numbers, and in the second, the squares are associated with letters. The green square is the starting point, and the red square indicates the final step, whereas the yellow squares are the path the robot should follow in between. The value indicated in the red square is the result of the task. In both examples, the result is a concatenation of the associated signs in the yellow squares of the path. The concatenation function in the case of the letters is the trivial one, and in the case of the numbers, we use a simple successive addition.

Children can solve different problems in different contexts using the same program. They can be challenged to create scenarios in mathematics and Portuguese, keeping the path structure. For example, by changing path numbers, but in such a way that the successive addition of these numbers is 40.

Finally, another kind of problem can be proposed to children: creating scenarios where the successive addition of path numbers corresponds to a specific value with several different solutions, i.e., where it is possible to obtain a final value through different paths. These paths could be translated into a scenario of letters where several words will be created through the concatenation of letters. There are numerous possibilities as they depend only on the imagination and creativity of teachers and students.

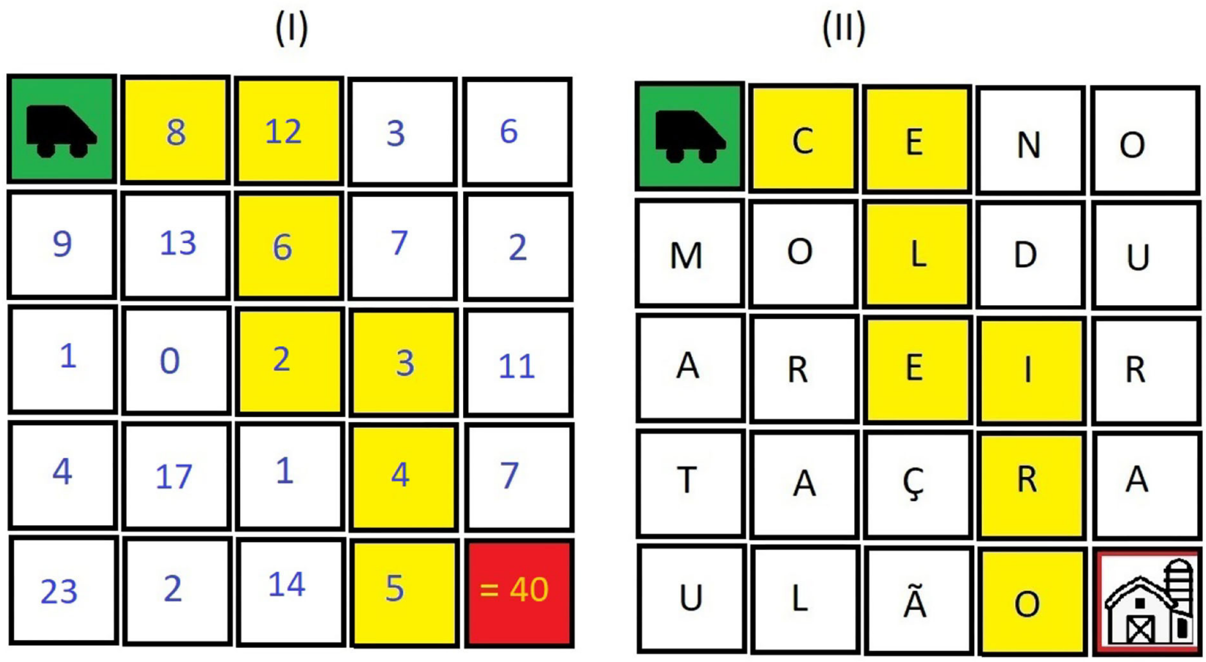
Examples in Figure 7 have an additional syntactical aspect compared to previous examples. In (III), the robot's movement sums up 139 at the end of the yellow path. However, in this new case, the process of concatenation is a symbolic concatenation. It has a specific syntax, i.e., the operator is applied between previous and actual integers. As the robot moves to *8*, *, and *6*, it is assumed that multiplication is executed, and the new value of 42 multiplied by three as the robot moves to the next position following the yellow path in the figure.

In (IV) and (V), the red square denotes the wished journey to the harbor *via* bakery, butcher store, and school. The chosen path satisfies this restriction in both cases, but only (IV) has an open pathway. Both solutions are semantically correct, but only (IV) is correct syntactically.

2.3.3. Horse jumping

Figure 8 depicts the horse jumping problem. In Figure 8A, we see the used playground where the green field indicates the robot's initial position and Figure 8B presents a solution to the problem. The problem involves moving the robot around the playground, simulating the knight's moves. In the end, it passes each square exactly once. In Figure 8C, we observe the eight knight moves allowed in this task.

This problem is different from the previous ones in that, until now, each physical step of the robot from one square to the next had an immediately semantic interpretation. However, it does not matter how the robot moves from an initial position to the final position within a knight's move to solve this new problem. Only the final position of each jump matters for the solution. In this way, it counts as a milestone in the program process. Therefore, we introduced a way to select these positions in the programming activity, i.e., by adding the halt command, corresponding to the "H" button.



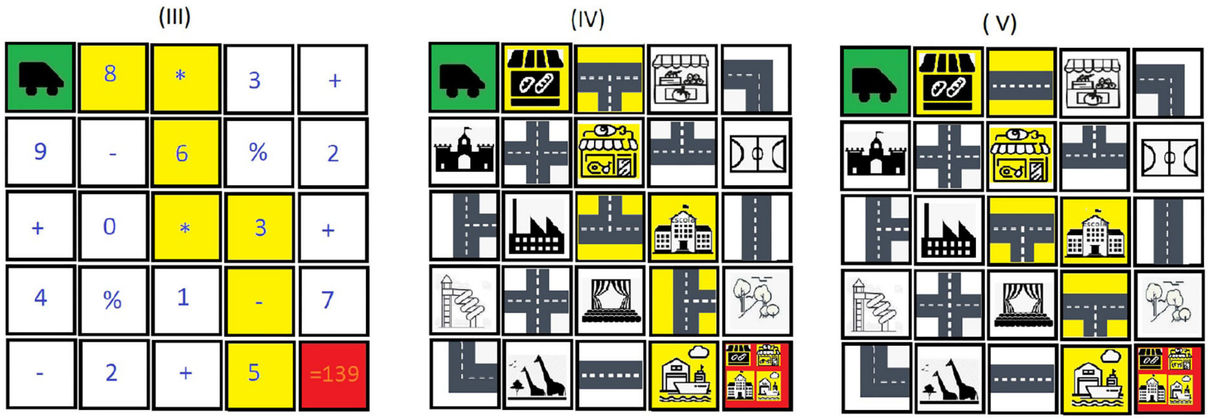
(I)

	8	12	3	6
9	13	6	7	2
1	0	2	3	11
4	17	1	4	7
23	2	14	5	= 40

(II)

	C	E	N	O
M	O	L	D	U
A	R	E	I	R
T	A	Ç	R	A
U	L	Ã	O	

FIGURE 6
In **(I)**, all path numbers sum up to 40, and in **(II)**, concatenated letters form the Portuguese word "celeiro," i.e., barn.



(III)

	8	*	3	+
9	-	6	%	2
+	0	*	3	+
4	%	1	-	7
-	2	+	5	=139

FIGURE 7
Different examples where a syntax related to operators is added in example **(III)**, and a more restricted syntax related to possible pathways is presented in examples **(IV)** and **(V)**.

2.3.4. Recycling games

The final example is presented in Figure 9, displaying a recycling game.

Now, the robot has to collect, using the selecting operator alternatively, first to indicate that it will grab a specific object to its left on the ground and then deposits the grabbed object in the proper recycling container. The mechanism to grab and deposit the litter will be implemented in the same way as the milestones in the "Horse Jumping" game.

In the following section, we have an approach to formalizing the programming activity based on the previous examples. We aim to find opportunities and constraints related to proposed playgrounds and, at the same time, identify some of the complexities related to the problems to be addressed.

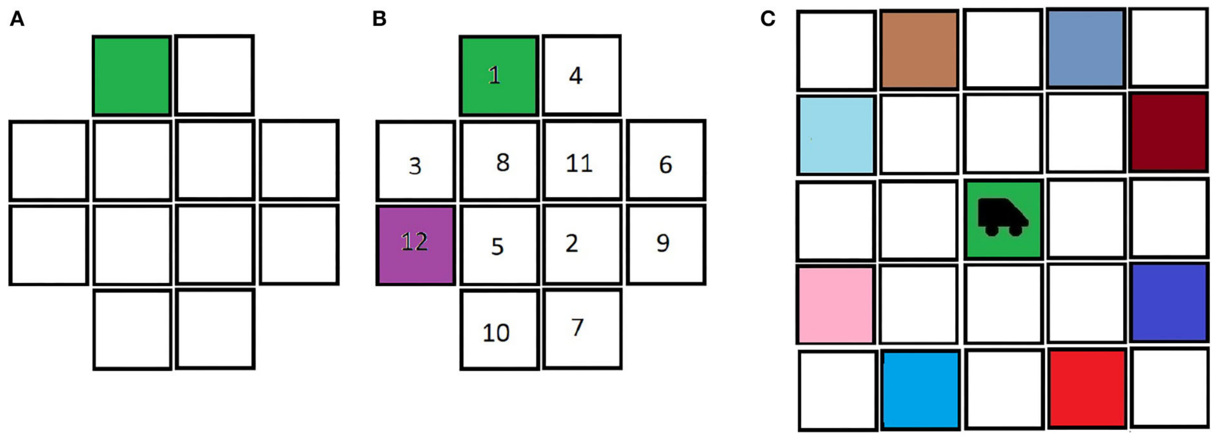


FIGURE 8
The eight knight moves in chess and their programs. **(A)** Ground with initial field. **(B)** Order to resolve the problem. **(C)** The allowed moves of the rebot (knight).

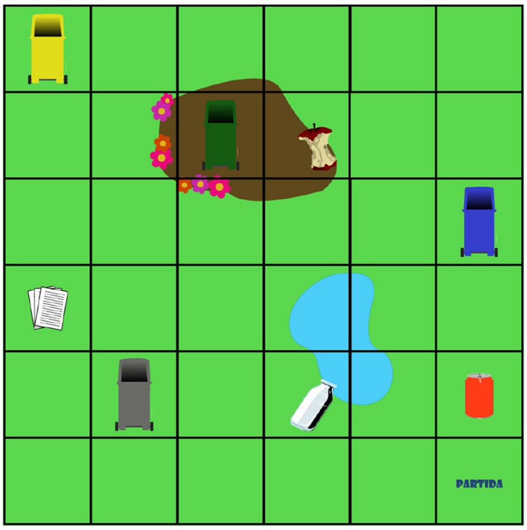


FIGURE 9
Collecting litter to put in recycling bins.

2.4. Formalizing the programming steps

As previously detailed, the robot only uses encoders for controlling the robot's step. No other sensors are provided. Thus, controlling the robot implies to select a sequence of commands for moving (F)orward or (B)ackward, (L)eft, or (R)ight turn (of ninety degrees) and, finally, (H)alting the robot. Thus, a command can be associated to a letter of the set $\mathfrak{B} = \{B, F, H, L, R\}$.

As the robot acts in the *x-y-plain*, its orientation is part of the set $\mathfrak{O} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\}$. This is important because the direction of the translations movement in the plain is defined by the combination of the actual robots orientation and the used translation buttons F or B.

In terms of formalization, the relocation initiated by pressing the button B and F would be the same if the orientation is in 180° opposition as shown in Figure 10, i.e., pressing the button B on a robot with direction ↑ will result in the same x-y position as if we press the F button by its direction on ↓.

We differentiate the programming activity into spatial programming where children must delineate a path in a specified context (or playground) using the commands to move and to turn the robot and spatial-step programming activity, adding the "H" command when the robot is on a location that is considered as a milestone of the process.

2.4.1. Spatial programming

Spatial programming is simply a sequence of commands in the set $\mathfrak{B}/\{H\}$. The context sensitive spatial translation by the buttons F and B, shown in Figure 10A, change either the x value in the horizontal movements or the y value in the indicated vertical movements, while buttons L and R transform the orientation as given in Figure 10B.

The state of the robot is defined by the tuple (x, y, o), where x and y correspond to the coordinates in the plain and o, the orientation of the robot out of the set $\mathfrak{O}$. The y value is incremented by one value if the robot goes downward and decreased by one if it goes upward. The x value increases by one if it goes to the right side and reduces by one if it goes to the left. A general programming sequence is demonstrated in Figure 10C with the sequence $\pi(i)$ defined by $[L, L, F, R, B]$ for $0 \leq i \leq 4$ whose corresponding behavior is also shown in the same figure.

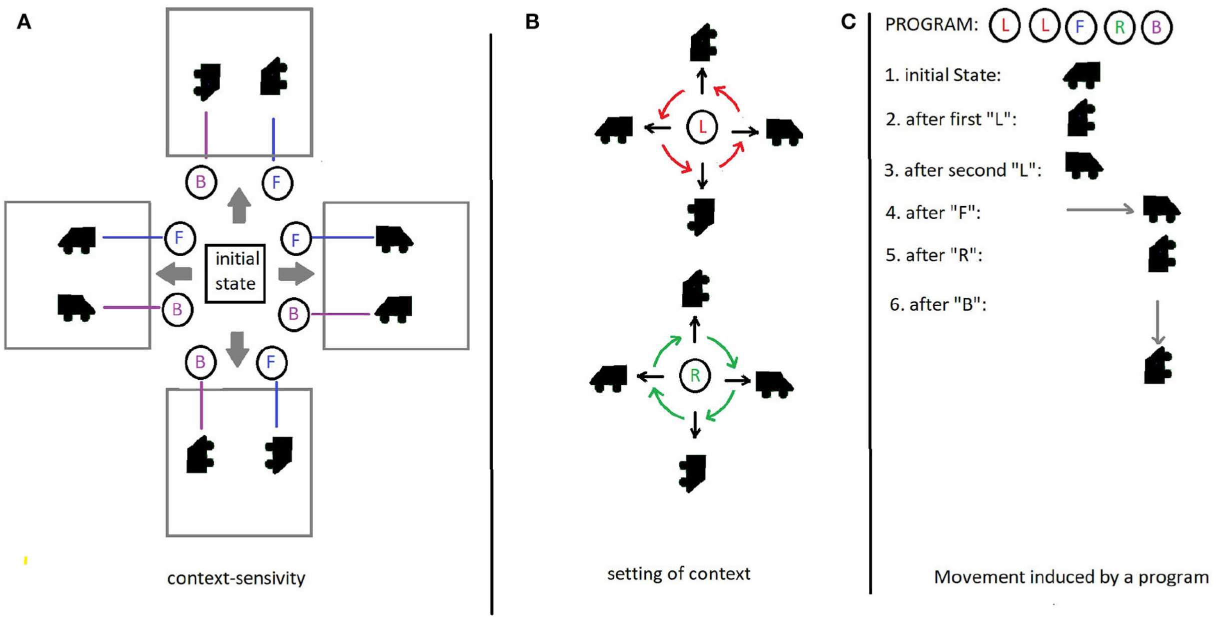


FIGURE 10
In **(A)**, we see how translation movements depend on the setting of the orientation state which can changed by the rotation movements in **(B)**. In **(C)**, we see how the program is executed by the robot.

Problems presented in Figures 6, 7 can be solved by the general definition, e.g., $\pi_a(i) = [F, F, R, F, F, L, F, R, F, F]$. If we assume that the robot is situated on the square with the number 32 and has the orientation → the problems $32 + 5$ and $32 + 9$ related to Figure 5 would be resolved by the programs $\pi_1(i) = [F, F, F, F, F]$ and $\pi_2(i) = [B, R, F, L]$.

The sensitivity to the robot's orientation transpires if we change the initial orientation to ← and the square to 39, the program $\pi_1(i)$ resolves $39 - 5$ and $\pi_2(2)$ resolves the problem $39 - 11$.

2.4.2. Milestone-step programming

In the milestone-step programing, we use a spatial program with the supplementary use of the command H. Therefore, we introduced a list of locations $\mathfrak{M}$ which was seen as the list of milestones the program achieved. Every application of the H command will insert to $\mathfrak{M}$ the correspondent location of the robot at the moment. We assumed that the last command is ever an H as this position would be utterly the objective.

Each milestone introduced in $\mathfrak{M}$ is naturally associated with the semiotic sign which is written on the respective square of location.

The "Horse Jumping" problem in Section 2.3.3 is a typical case where the H has a rule on defining the final positing of each involved knight movement as a part of the solution. Therefore, each of the eight knight moves defined in Figure 11A has a final H instruction to insert the final location of each executed move into the set of milestones $\mathfrak{M}$.

The strategy adopted in the first five instructions of each move was that the fifth command inverted the single rotation it has along the instructions before. It guaranteed that the robot showed its initial orientation after each movement, which is, in this case, "→."

Therefore, the last two instructions of each movement provided an essential requisite of modular programing as every single macro-step was complete and guaranteed that the next module could start with its proper initial condition.

This idea is expressed in Figure 11B where we adapted the algorithms defined in Figure 8B. The robot, orientated to "→," begin its journey at the green square and successfully applies the movement indicated at the actual square until reaching the final pink location. As each movement inserts its final position onto $\mathfrak{M}$, all milestones are achieved.

3. Programing activity and CT skills

In Chevalier et al. (2020), the phases of the CCPS model, based on the conceptual framework of ERS, are presented. They are an important referential to identify the development of CT skills and are used in the PeCOT project.

In a nutshell, the phases are the following:

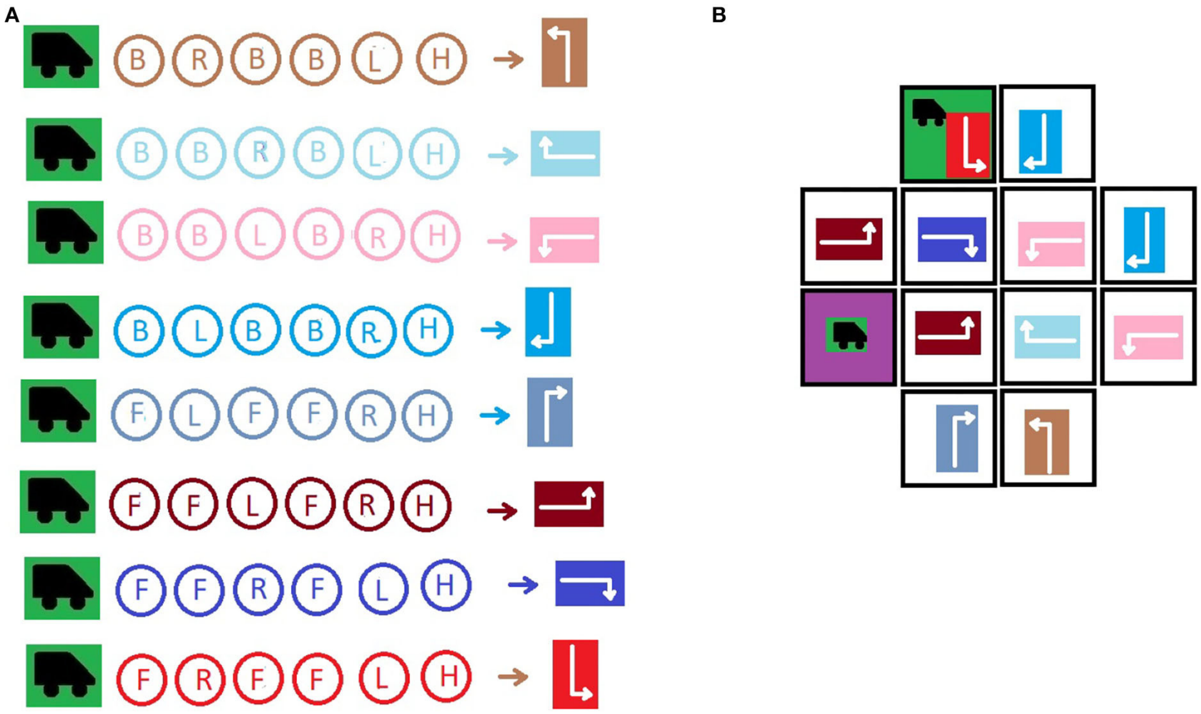


FIGURE 11
Steps to solve the problem with 11 functions of knight moves. **(A)** Definition of each knight move. **(B)** Solution of the knight problem.

- **Understanding the problem**-In this phase, children have to identify the problem through abstraction and decomposition. The output of this phase is the required transformation of the playground.

  As an example of this phase, children solving the problem of "Educational Number Chart 1–100" will have to, first, identify the different situations that occur in the context of the board, one that makes the robot leave the board, and others where the robot stay on the board. What is expected is that some general rule could be found for the robot to find the way to the target and that decisions should be made based on the numbers involved in the addition. Naturally, as a first step, in this case, the output is the robot in the final square with the number corresponding to the parcel addition.
- **Generating the idea**-This phase involves using the robot to satisfy the conditions given in the problem. This implies using creativity. As a result, children are expected to come up with sketches of different behaviors of the robots.

  Taking the example of the "Horse" problem, we expect children to identify all the possible movements of the robot as a chess knight, which is part of the starting idea to solve the proposed problem. Identifying the possible positions will help children to identify which movements the robot should execute in each step.
- **Formulation of the behavior**-In this phase, the characteristics of the robot are used, and the behavior idea from the previous step should be transformed into concrete robotic movements. The output of this phase is an algorithm in the problem solver's language.

  The scenario presented in the "isomorphic problem" could be explored by following different paths and conducting several experiments to achieve the goal, i.e., to achieve the target with the right result of a word. The algorithm is expected to be found as the path is defined. Children can mark down the steps using color, as in Figure 6.
- **Programing the behavior**-It is in this phase that the problem solver creates the program. This phase also serves as debugging, allowing the revision of previous implementations. The output of this phase is the behavior of the robot executing the program and presenting its results in the playground.

  The robot is programed simply by pressing a sequence of buttons. We expect that the children will have to record the programing sequence on a piece of paper or discuss the solution in a dialogue when solving the problem in groups.
- **Executing the solution**-In this phase, the problem solver acts as a judge. The problem solver "has to decide if the programmed behavior can be considered an appropriate

solution of the problem, or if it has to be refined, corrected, or completely redefined" (Chevalier et al., 2020).

Problems can have more than one solution. One particular case is the scenarios I and II in the "Isomorphic Problem" where more than one path could be found. In these cases, an additional feature could be considered to find the best solution. Testing the robot executing its program will provide the solution discovered.

With the examples presented, we hope to be able to characterize the situations that can provide clues about the CT skills used by children when exploring each of the problems proposed. The strategy adopted by the PeCOT research team is to find motivation among teachers to provide problems that can be aligned with the subjects that should be addressed in the classroom. At the same time, it is hoped that the use of the Azbot-1C robot will foster curiosity and enthusiasm for solving problems such as those described.

We made the following additional considerations:

- Programming using an Azbot-1C robot implies previewing a sequence of steps and introducing them, one by one, by selecting a specific button.
- There is no scaffolding supporting the programming activity. However, researchers will give the children paper and pencils for them to register ideas and, possibly, the sequence of executing steps of the program.
- The scenarios for playgrounds are intended to be prepared with teachers where the experiments with the robots will take place. It is expected, through interaction, to adjust the proposed examples and create new ones.

4. Conclusions

In this paper, we present the functionalities of a simple robot to be used in a primary education classroom context. The robot that had been developed for the PeCOT project aimed at implementing tangible programing and measuring children's knowledge in terms of CT. We make a description of the robot and then present a set of scenarios where the robot can be used. These scenarios propose different tasks to be executed by the robot. The scenarios were prepared to introduce mathematical and natural science themes.

The observation of the robots and the scenarios allows us to better adjust future work, trying to adapt it to the context in which it will be introduced. This reflection, made through different stakeholders—the researchers, the teachers who collaborated on the project and some children who had the opportunity to explore the robot—has enriched this preparatory phase, allowing us to identify aspects to be improved.

This text described all phases from how the Azbot-1C robot was developed to how the scenarios and tasks were planned to promote CT skills. This work aimed to improve the educational context where the different learning situations take place in the context of the project.

At the same time, it is hoped that the methodologies adopted can contribute to teachers considering alternative strategies in promoting CT in the classroom. Teachers are expected to analyze the nature of the pedagogical tasks to be implemented, adjusted the designed scenarios and resources, and monitored their implementation. Furthermore, this paper contributes to making their pedagogical work more problem-oriented in an active, constructive, and meaningful way.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: https://github.com/Robotics-and-AI-Group-of-UAc/Azobopi.

Author contributions

MF, JC, and AS conception and organization of the article. FP, MD, PM, and MF conceptual ideas for robotic activities. All authors contributed to the writing of the article, read, and approved the final version of the manuscript.

Funding

The project Pensamento Computacional com Objetivos Tangíveis (PeCOT) is supported by Direção Regional da Ciência e Tecnologia (DRCT) by the reference M1.1.C/C.S./058/2021/01.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bakala, E., Gerosa, A., Hourcade, J. P., and Tejera, G. (2021). Preschool children, robots, and computational thinking: a systematic review. *Int. J. Child Comp. Interact*. 29, 100337. doi: 10.1016/j.ijcci.2021.100337

Beecher, K. (2017). *Computational Thinking. A Beginner's Guide to Problem Solving and Programming*. London: BCS Learning & Development Ltd.

Bers, M. U. (2018). "Coding, playgrounds and literacy in early childhood education: the development of kibo robotics and scratchjr," in *2018 IEEE Global Engineering Education Conference (EDUCON)* (Santa Cruz de Tenerife: IEEE), 2094–2102.

Bers, M. U., Govind, M., and Relkin, E. (2022). *Coding as Another Language: Computational Thinking, Robotics and Literacy in First and Second Grade*. New York, NY: Association for Computing Machinery.

Bevčič, M., and Rugelj, J. (2020). "Game design based learning of programming for girls," in *43rd International Convention on Information, Communication and Electronic Technology* (Opatija: MIPRO), 576–580.

Bocconi, S., Chioccariello, A., Kampylis, P., Dagiené, V., Wastiau, P., Engelhardt, K., et al. (2022). "Reviewing computational thinking in compulsory education," in *Scientific Analysis or Review KJ-06-22-069-EN-N (Online)* (Luxembourg).

Campbell, C., Speldewinde, C., Howitt, C., and Macdonald, A. (2018). Stem practice in the early years. *Creat. Educ*. 09, 11–25. doi: 10.4236/ce.2018.91002

Chalmers, C. (2018). Robotics and computational thinking in primary school. *Int. J. Child Comput. Interact*. 17, 93–100. doi: 10.1016/j.ijcci.2018.06.005

Chevalier, M., El-Hamamsy, L., Giang, C., Bruno, B., and Mondada, F. (2022). "Teachers' perspective on fostering computational thinking through educational robotics," in *Robotics in Education*, eds M. Merdan, W. Lepuschitz, G. Koppensteiner, R. Balogh, and D. Obdržálek (Cham: Springer International Publishing), 177–185.

Chevalier, M., Giang, C., Piatti, A., and Mondada, F. (2020). Fostering computational thinking through educational robotics: a model for creative computational problem solving. *Int. J. STEM Educ*. 7, 2196–7822. doi: 10.1186/s40594-020-00238-z

Ching, Y.-H., Yang, D., Wang, S., Baek, Y., Swanson, S., and Chittoori, B. (2019). Elementary school student development of stem attitudes and perceived learning in a stem integrated robotics curriculum. *TechTrends* 63, 590–601. doi: 10.1007/s11528-019-00388-0

Espressif (2022). *ESP32 series datasheet*. Available online at: https://www.espressif.com/en/products/socs/esp32-s2

Eteokleous, N., and Ktoridou, D. (2014). "Educational robotics as learning tools within the teaching and learning practice," in *2014 IEEE Global Engineering Education Conference (EDUCON)* (Istanbul: IEEE), 1055–1058.

Funk, M. G., Cascalho, J. M., Santos, A. I., and Mendes, A. B. (2021). Educational robotics and tangible devices for promoting computational thinking. *Front. Rob. AI* 8, 713416. doi: 10.3389/frobt.2021.713416

Giang, C. (2020). *Towards the alignment of educational robotics learning systems with classroom activities* (Ph.D. thesis). EPFL.

Giang, C., Piatti, A., and Mondada, F. (2022). "Aligning the design of educational robotics tools with classroom activities," in *Designing, Constructing, and Programming Robots for Learning, Volume 4 of 5, Chapter 1*, eds N. Eteokleous and E. Nisiforou (IGI Global), 1–21.

Li, Y., Xu, S., and Liu, J. (2021). Development and validation of computational thinking assessment of chinese elementary school students. *J. Pacific Rim Psychol*. 15, 18344909211010240. doi: 10.1177/18344909211010240

Lopez-Caudana, E., Ramírez-Montoya, M.-S., Pérez, S., and Rodríguez-Abitia, G. (2020). Using robotics to enhance active learning in mathematics: a multi-scenario study. *Mathematics* 8, 163. doi: 10.3390/math8122163

Martins, O., Gomes, C., Brocardo, J., Pedroso, J., Silva, L.;, Encarnação, M., Horta, M., et al. (2017). *Perfil dos Alunos à saída da Escolaridade Obrigatória [Profile of Students Leaving Compulsory School*]. Lisbon: Ministério de Educação e Ciência.

Ministry of Education. (2021). *Aprendizagens Essenciais da componente de currículo/disciplina de Matemática inscrita na matriz curricular base dos 1.°, 2.° e 3.° ciclos do ensino básico geral [Essential Learnings of the Mathematics Curriculum Component/Discipline Included in the Basic Curricular Matrix of the 1st, 2nd and 3rd Cycles of General Basic Education]*. Lisbon: Ministério da Educação.

Mondada, F., Bonani, M., Riedo, F., Briod, M., Pereyre, L., Retornaz, P., et al. (2017). Bringing robotics to formal education: the thymio open-source hardware robot. *IEEE Rob. Autom. Mag*. 24, 77–85. doi: 10.1109/MRA.2016.2636372

Pedro, F., Cascalho, J., Medeiros, P., Novo, P., Funk, M., Ramos, A., et al. (2021). "Azbot-1c: an educational robot prototype for learning mathematical concepts," in *Optimization, Learning Algorithms and Applications*, eds A. I. Pereira, F. P. Fernandes, J. P. Coelho, J. P. Teixeira, M. F. Pacheco, P. Alves, and R. P. Lopes (Cham: Springer International Publishing), 657–669.

Pérez-Vázquez, E., Lorenzo, G., Lledó Carreres, A., Lorenzo-Lledó, A., and Gilabert Cerdá, A. (2019). El uso del robot bee-bot como herramienta de aprendizaje de emociones en la enseñanza del alumnado con trastorno del espectro autista. *Int. J. Dev. Educ. Psychol*. 3, 577–584. doi: 10.17060/ijodaep.2019.n1.v3.1643

Ramos, J., Espadeiro, R., and Monginho, R. (2022). *Introdução à programação, robótica e ao pensamento computacional na educação pré-escolar e 1.°ciclo do ensino básico*. Necessidades de formação de educadores e professores. Available online at: https://digital.dge.mec.pt/sites/default/files/documents/2022/177-14551b4676f7711407147b60af2f9a3f.pdf

Sáez-López, J., Sevillano-García, M., and Vazquez-Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: educational use of mbot. *Educ. Technol. Res. Dev*. 67, 1405–1425. doi: 10.1007/s11423-019-09648-5

Santos, A. I., Pereira, D., Botelho, N., Medeiros, P., and Cascalho, J. (2022). Robotics for young people in azoresminibot. *Int. J. Inform. Educ. Technol.* Available online at: http://www.ijiet.org/show-157-2286-1.html

Sapounidis, T., Demetriadis, S., Papadopoulos, P. M., and Stamovlasis, D. (2019). Tangible and graphical programming with experienced children: a mixed methods analysis. *Int. J. Child Comput. Interact*. 19, 67–78. doi: 10.1016/j.ijcci.2018.12.001

Sapounidis, T., Demetriadis, S., and Stamelos, I. (2015). Evaluating children performance with graphical and tangible robot programming tools. *Pers. Ubiquit. Comput*. 19, 225–237. doi: 10.1007/s00779-014-0774-3

Schneider, B., Jermann, P., Zufferey, G., and Dillenbourg, P. (2011). Benefits of a tangible interface for collaborative learning and interaction. *IEEE Trans. Learn. Technol*. 4, 222–232. doi: 10.1109/TLT.2010.36

Somyürek, S. (2015). An effective educational tool: construction kits for fun and meaningful learning. *Int. J. Technol. Design Educ*. 25, 35–41. doi: 10.1007/s10798-014-9272-1

Strawhacker, A., and Bers, M. U. (2015). "I want my robot to look for food": comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *Int. J. Technol. Design Educ*. 25, 293–319. doi: 10.1007/s10798-014-9287-7

Uşengül, L., and Bahçeci, F. (2020). The effect of lego wedo 2.0 education on academic achievement and attitudes and computational thinking skills of learners toward science. *World J. Educ*. 10:83–93. doi: 10.5430/wje.v10n4p83

Wing, J. M. (2006). Computational thinking. *Commun. ACM* 49, 33–35. doi: 10.1145/1118178.1118215

Žáček, M., and Smolka, P. (2019). "Development of computational thinking: student motivation using ozobot," in *Proceedings of the 2019 3rd International Conference on Education and E-Learning, ICEEL 2019* (New York, NY: Association for Computing Machinery), 36–40.