Computational thinking through unplugged activities in early years of Primary Education

Javier del Olmo-Muñoz, Ramón Cózar-Gutiérrez*, José Antonio González-Calero

University of Castilla-La Mancha, Faculty of Education of Albacete (Edificio Simón Abril), Plaza de la Universidad, 3, 02071, Albacete, Spain

ARTICLE INFO

Keywords:
Computational thinking
Elementary education
Evaluation methodologies
Gender studies
Teaching/learning strategies

ABSTRACT

The inclusion of computational thinking (CT) in the classroom is something that the present and future society is demanding. However, this domain remains largely unexplored, especially in the first years of Primary Education. The purpose of this study is to evaluate whether the inclusion of the so-called unplugged activities favours the development of CT in the students of the early years of Primary Education. To this end, a quasi-experimental study has been carried out to explore the eventual benefit of a mixed approach that combines both unplugged and plugged-in activities. In particular, 84 second-grade students took part in the experiment. Three questions were evaluated: the development of their CT skills, their motivation towards the proposed instruction, and the influence of students’ gender in the two previous areas. The intervention was designed on a selection of activities extracted from Code.org courses, and was divided into two phases, one in which one group worked with unplugged activities and the other with plugged-in activities, and a second where both groups worked with plugged-in activities. Analysing the three proposed questions through tests performed before, in between and after the instruction, it is concluded that the inclusion of unplugged activities in the instruction seems beneficial taking into account CT, motivation and gender.

1. Introduction

Considered a pioneer in the computational thinking (CT) field (Ilic & Haseski, 2018), Wing (2008) defined CT as “taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing” (p. 3717). A few years later she defined it as “the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out” (Wing, 2014, para. 5).

About this, Bundy asserted that a remarkable intellectual revolution is happening regarding CT, as it is influencing research in the majority of disciplines, either in sciences or in humanities (2007, p. 67). Accordingly, Wing inferred that today it can be stated that CT is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, CT should be added to every child’s abilities (2006). For Wing, this scenario establishes a new educational challenge for our society, especially for our children (2008, p. 3717).

Barr and Stephenson (2011) argued that all today’s students will live a life heavily influenced by computing, and many will work in fields that involve or are influenced by computing, so they must begin to work with CT in K-12. Specifically, Djurdjevic-Pahl, Pahl, Fronza, and El Ioini (2017) stated that CT can be started in early stages, since the first year in primary school.

* Corresponding author.
E-mail address: Ramon.Cozar@uclm.es (R. Cózar-Gutiérrez).

https://doi.org/10.1016/j.compedu.2020.103832
Received 6 September 2019; Received in revised form 24 January 2020; Accepted 2 February 2020
Available online 3 February 2020
0360-1315/© 2020 Elsevier Ltd. All rights reserved.
As for its research and implementation in classroom, since the cognitive capacity of the students vary depending on their age, the methods, contents and learning strategies for teaching CT should be adapted accordingly (Hsu, Chang, & Hung, 2018). In relation to that, Brackmann et al. (2017) pointed out that CT is nowadays being adopted and investigated by using two main approaches in schools: with computer programming exercises (plugged-in activities) and with unplugged activities. These two types of activities differ in that the latter are based on the approach of exposing children to CT without using computers (Bell, Grimley, Bell, Alexander, & Freeman, 2009).

Some research has been done regarding students’ motivation towards them and their results, as is the case of Brackmann et al. (2017) with students of 5th and 6th grade of Primary Education, who proved that the unplugged approach has a positive effect on motivation and may be effective for the development of CT, or those of Tsarava et al. (2017) and Tsarava, Moeller, and Ninaus (2018) with children of 3rd and 4th grade, who stated that playful unplugged activities should allow students getting a first grip on CT processes by actively engaging them. However, the little documentation about early years of Primary Education such as 1st and 2nd grade suggests that this is a recent issue whose research still have a long way to go.

In terms of students’ gender, Dagienė, Pelikis, and Stupuriene (2015) underlined the little research specifically exploring gender differences in young children’s problem-solving and programming abilities, as CT is so far a very little investigated domain. Since then, despite the materialization of several studies on the topic with students from secondary school and last years of elementary school (Atmatzidou & Demetriadis, 2016; Jenson & Droumeva, 2016; Owen et al., 2016; Torres-Torres, Román-González, & Pérez-González, 2019, pp. 209–215), there is still a research gap in the early years of Primary Education. In relation to the interest towards aspects related to learning programming and CT, Espino and González (2015a, 2015b) observed that there is greater homogeneity between boys and girls in the stages of Early Childhood Education and Primary Education, in comparison with the subsequent educational stages. However, Master, Cheryan, Moscatelli, and Meltzoff (2017) highlighted that girls in early elementary school report less interest and liking for computers compared with boys. These contradictory results indicate the need for further studies on the gender gap, and eventually for approaches addressing more motivating alternatives for girls in the early educational stages, since at that time the differences are still not too wide and can be more easily addressed.

With all this, the main aim of this work is to evaluate whether it is more appropriate to introduce CT in early years of Primary Education through unplugged activities before plugged-in activities rather than do it exclusively through plugged-in activities. Specifically, to address this aim we study the following research questions:

- RQ01.- What approach, unplugged-plugged-in activities or only plugged-in activities, promotes a greater acquisition of CT skills when introducing CT in the first years of Primary Education?
- RQ02.- What approach, unplugged-plugged-in activities or only plugged-in activities, produces better motivational outcomes when introducing CT in the first years of Primary Education?
- RQ03.- Are there significant differences in the effectiveness of the approaches in terms of CT skills or motivational outcomes related to the gender of the students?

2. Method

2.1. Design

Based on the main objectives of this study, and given the impossibility of subjecting the disposition of the groups under study to a previous sampling to randomize their structuring, a quasi-experimental design with control and experimental groups was proposed. Regarding the development of the experience, it was structured in two main training phases of instruction interspersed with pre-, mid- and post-tests, making a total of eight 45-min lessons developed during an eight-week period. Fig. 1 describes the process implemented in the development of the research.

In the first session the students performed the Pre-testCT for the evaluation of their previous level in the skills related to CT. After that, the first phase of instruction took place over three sessions in which the control group (hereinafter, unplugged group) worked with unplugged activities while the experimental group (hereinafter, plugged-in group) worked with complementary plugged-in activities. After the first phase of instruction, the Mid-testCT was performed by students to evaluate their CT skills, and the Mid-testMot to measure their motivation towards the activities previously implemented.
Once the Mid-tests were done, the second phase of instruction began. This phase consisted of two sessions in which the type of the activities was unified for both groups, being all of the type plugged-in. At the end of the second phase of instruction, the Post-tests (Post-test\textsubscript{CT} and Post-test\textsubscript{MoE}) were completed to evaluate again both the levels of CT and the motivation.

2.2. Participants

The sample of the research consisted of a total of 84 students of the 2\textsuperscript{nd} year of Primary Education divided into four groups of three different schools, all of them from the region of Castilla-La Mancha (Spain). Both the plugged-in group and the unplugged group were formed by 42 students each. Regarding the gender of the students, there were 18 boys and 24 girls in the unplugged group, and 23 boys and 19 girls in the plugged-in group. The demographic data of the groups summarized in Table 1 verifies the homogeneity of the groups in terms of gender.

2.3. Instruments

2.3.1. Computational thinking tests

Developing a valid and reliable instrument for the measurement of elementary students’ CT is challenging because of the lack of consensus in the field in terms of CT definition (Chen et al., 2017) or how to measure it (Román-González, Pérez-González, & Jiménez-Fernández, 2017). In this case, a test was specifically designed for this study in order to determine the performance of students in terms of CT. This test followed the Bebras learning model ([Dagien and Sentence, 2016]), whose measured skills directly correspond with what Selby and Woollard described as the core CT skills (2013), also adopted by Computing At School in the UK (Csizmadia et al., 2015), which are: abstraction, algorithmic thinking, decomposition, evaluation and generalisation. Therefore, the test was constructed from a selection of tasks-problems drawn from the ‘International Bebras Contest’, which can be used as measuring instruments (Jung & Lee, 2017; Román-González, Moreno-León, & Robles, 2019; Tang, Yin, Lin, Hadad, & Zhai, 2020), or, as in the case of Lockwood and Mooney (2018), to be exclusively used to develop a CT test. Furthermore, the characteristics of their tasks make them begin to be considered for future international large-scale assessments in the field of Computer Science as PISA (Programme for International Student Assessment) (Hubwieser & Mühling, 2015), and to be used in different studies (Chen et al., 2017; Chiazzese, Arrigo, Chifari, Lonati, & Tosto, 2019; Ruf, Mühling, & Hubwieser, 2014).

Although in an initial design some items of the Computational thinking Test proposed by Román-González (2015) were contemplated, the final selection was based exclusively on Bebras Tasks, since their general approach seek the solution of ‘real’ and significant problems for the students, through the transfer and projection of CT in the context of the problem (Román-González, 2016). This is also stated in Román-González et al. (2019) where is also proposed a classification of CT assessment tools according to their evaluative approach. Within this classification, Bebras Tasks are categorized as “Computational thinking skill transfer tools”, whose objective is to evaluate to what extent students can transfer their CT skills onto different types of problems, contexts, and situations.

Another reason to be taken into account when choosing the source is the aforementioned shortage of instruments to measure CT at early ages in Primary Education. In the case of the Bebras Challenges, they are organised in different age categories: Kits (age 6–8), Castors (age 8–10), Juniors (age 10–12), Intermediate (age 12–14), Seniors (age 14–16) and Elite (age 16–18). At the same time, within these categories, the proposed tasks are organised by their difficulty (A-easy, B-medium, C-difficult). Due to the age range that concerns us, the ‘A’ and ‘B’ difficulty tasks of the ‘Kits’ category were the source of our test.

In summary, the instrument consisted of an initial selection of 12 items from the category ‘Kits’ (difficulties ‘A’ and ‘B’, hereinafter ‘Easy’ and ‘Hard’) of the 2016 and 2017 Editions of the ‘International Bebras Contest’. Due to the need to transfer these tasks to paper format for their execution in this study, two of the questions were finally excluded due to the impossibility of performing them in text format. Therefore, the final test was composed of 10 items whose sequencing responds to an order of least to greatest difficulty, taking into account the elements previously presented.

It should also be mentioned that for each of the performed tests (Pre, Mid and Post) a different version with isomorphic items was designed, that is, they were variations of the same items in order to measure exactly the same objectives.

Below are two examples of the Pre-test\textsubscript{CT} (Fig. 2 and Fig. 3), one for each level of difficulty.

In each item, after the problem and the question, the possible answers were offered, which varied from four to six, having to choose the student only one of them (except for the last question of the test, where there was more than one possible answer).

2.3.2. Motivation tests

The second instrument corresponds to the objective of diagnosing whether the instruction proposed in the control group leads to a loss or a gain in the motivation of the students. Therefore, they were done after each training phase: the Mid-test\textsubscript{Mot} after session 3 (phase 1) of instruction, and the Post-test\textsubscript{Mot} after session 5 (phase 2) of instruction. To this end, we chose the four-dimensional model

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographic data of the groups under study.</td>
</tr>
<tr>
<td>Group</td>
</tr>
<tr>
<td>Unplugged</td>
</tr>
<tr>
<td>Plugged-in</td>
</tr>
</tbody>
</table>

Source: Own elaboration based on the collected data.
ARCS, which is named after an acronym referring to the categories that comprise it: attention, relevance, confidence and satisfaction. This model defined by Keller (1987) is based on the idea that there are personal and environmental characteristics that influence motivation, and therefore, performance against educational tasks. In this work we have used the validated reduction of Loorbach, Peters, Karreman, and Steehouder (2015).

As the participants were students in the 2nd year of Primary Education, their level of reading comprehension was basic. Therefore,

Table 2
Motivation test items.

<table>
<thead>
<tr>
<th>Item</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Attention</td>
<td>The quality of the activities helps me to maintain the attention.</td>
</tr>
<tr>
<td>2</td>
<td>Attention</td>
<td>The variety of activities helps keep my attention in the class.</td>
</tr>
<tr>
<td>3</td>
<td>Attention</td>
<td>The way in which information is organised helps me to maintain attention.</td>
</tr>
<tr>
<td>4</td>
<td>Relevance</td>
<td>For me it is clear how the lessons are related to things I already knew.</td>
</tr>
<tr>
<td>5</td>
<td>Relevance</td>
<td>The contents and activities convey the impression that it is worth knowing the contents of the lesson.</td>
</tr>
<tr>
<td>6</td>
<td>Confidence</td>
<td>While I work in class, I am sure that I will learn the contents.</td>
</tr>
<tr>
<td>7</td>
<td>Confidence</td>
<td>After working in class, I feel confident about passing an exam on the subject.</td>
</tr>
<tr>
<td>8</td>
<td>Confidence</td>
<td>The good organization of the lessons helps me to be sure that I am going to learn the contents.</td>
</tr>
<tr>
<td>9</td>
<td>Satisfaction</td>
<td>I enjoy so much in class, that I would like to know more about the topics.</td>
</tr>
<tr>
<td>10</td>
<td>Satisfaction</td>
<td>I like these lessons.</td>
</tr>
<tr>
<td>11</td>
<td>Satisfaction</td>
<td>It is a pleasure to work on such well-designed lessons.</td>
</tr>
</tbody>
</table>

Source: Own adaptation of Loorbach et al. (2015).
to carry out the motivation test, the choice of the answers was facilitated by a Likert scale that ranges from 1 (Strongly disagree) to 5 (Strongly agree) supported by emoticons. In Table 2 the items of the adapted motivation test can be consulted.

2.4. Procedure

As explained in the design section, there were five instructional sessions that allowed students to develop skills related to CT. The instruction was composed of two phases, being the first of three 45-min sessions, and the second of two 45-min sessions.

For the design of the instruction sessions we based on several coding and computer literacy development programmes from the website Code.org. This platform organises its courses in different programmes that, thanks to their characteristics, are a good option to support students to lay the foundations of computer science (Kalelioglu, 2015). It has become increasingly popular to promote and teach these abilities in primary school by initiatives such as Code.org (Tsarava et al., 2017).

To plan the activities and their sequencing throughout these sessions, we made a selection from the different Code.org online courses following the guidelines of Code.org (2018b) and Code.org (2018c), since they offer a good combination and alternation of activities, from which an equivalence can be established between the plugged-in and the unplugged activities. Specifically, we chose and adapted different activities from courses which are designed to a 2nd grade level, as indicated in Code.org (2018b), and that include simple computational concepts such as “directions”, “sequences” and “loops”, but not other more complex such as “conditionals” or “functions” (Román-González, 2016). The courses are the following:

- Course 1 (Code.org (2019a)), intended for early-readers who have little or no previous computer science experience.
- Course 2 (Code.org (2019b)), intended for readers who have little or no previous computer science experience.
- Course B (Code.org (2018a)), developed with first-graders in mind, and tailored to a novice reading level.

Regarding the materials that were used, we prepared a teaching guide for each lesson to facilitate the development of the sessions for each of the teachers, so that they only had to follow the proposed steps. Besides, the teacher had to support the explanation of the activities with the whiteboard or the blackboard. For their part, students used different manipulative materials (such as different templates, plastic cups or a reflection journal) for the unplugged activities, and tablets for the plugged-in activities, which allowed the students to work individually in the plugged-in activities.

Regarding groupings, students worked individually most of the time, except when some unplugged punctual activity required working in pairs or groups.

2.4.1. Phase 1, session 1, unplugged: Graph Paper Programming

In this first session, students begin to understand what programming is really about by “programming” each other to draw pictures. Basic concepts such as sequencing and algorithms are presented to the class in an unplugged activity. In this lesson, students learn how to develop an algorithm and code it in a program.

Objectives:

- Understand the difficulties of translating human language problems into machine language.
- Convert a sequence of steps into a coded program.
- Practice the communication of ideas through codes and symbols.

Below is a detailed description of each of the parts of the lesson plan:

1) Vocabulary (2 min): Two new and important concepts for this lesson are explained for all the class: algorithm and program.
2) Introduce Graph Paper Programming (8 min): In this activity, the teacher explains to all the class how we guide each other to make drawings. For this exercise, we use sheets of 4 × 4 graph paper. Starting in the upper left corner, we guide our pairs with simple instructions. These instructions include: “Move a square to the right”, “Move a square to the left”, “Move a square up”, “Move a square down”, and “Fill the square with colour”. For example, Fig. 5 shows how we would write an algorithm to indicate our pair (who pretends to be a drawing machine) to colour their grid so that it looks like the image, by using the instructions in Fig. 4 (which the teacher projects onto the board).
3) Practice Together (5 min): At this point, the teacher fills the grid for the class on the board, square by square, and then asks them to help him/her describe what he/she has just done. First reciting the algorithm out loud, and then converting their verbal instructions into a program.

Fig. 4. Graph Paper Programming instructions. Source: Code.org (2018c)
4) Four-by-Fours (20 min): At this point students work in pairs, first choosing one of the proposed grids of a worksheet and writing the algorithm necessary to draw the chosen image, so that after the pair can interpret (as do the machines) and draw the grid of the other.

5) Flash Chat (3 min): What did we learn? A series of questions are proposed to the class to reflect on what has been done:
   - What we learned today?
   - What happens if we use the same arrows, but replacing “Fill-in square with colour” with “Place brick”? What could we be able to do?
   - What else could we program if we changed what the arrows meant?

6) Vocab Shmocab (2 min): Different definitions are proposed to the whole class to be associated with the new vocabulary learned, in order to settle it.

7) Graph Paper Programming Assessment (10 min): Students work individually with the same type of exercises in their ‘Assessment Worksheet’ to continue practicing what they have learned.

2.4.2. Phase 1, session 1, plugged-in: Jigsaw and Maze Sequence

In this first session, students become familiar with the basic use of drag and drop and the programming interface based on blocks that they will use in the rest of the activities. After that, students begin to work with sequences (one of the basic logical structures in programming).

Objectives:
- Use drag and drop to enter information on a tablet.
- Place the pieces of the puzzle in the correct order.
- Sort movement commands as sequential steps in a program.

Below is a detailed description of each of the parts of the lesson plan:
1) Jigsaw (15 min): Learn to drag and drop: Students start by simply dragging the images on the screen and then dropping the pieces of the puzzle in the correct order (see Fig. 6). To avoid any student being left behind, we all do it at the same time, waiting for everyone to finish each stage to move on to the next, and we will also do so in the future for all the plugged-in and unplugged sessions. The lesson is available in https://studio.code.org/s/course1/stage/3/puzzle/1.

2) Maze Sequence (20 min): In this puzzle series with Angry Birds game characters, students develop sequential algorithms to move a bird from one side of the maze to the pig on the other side (see Fig. 7). The lesson is available in https://studio.code.org/s/course1/stage/4/puzzle/1.

2.4.3. Phase 1, session 2, unplugged: My Robotic Friends

Using the set of symbols similar to the one learned in the first session, the students design algorithms to indicate a “robot” that stacks glasses with different patterns. Students take turns participating as the robot, acting only on the basis of the algorithm defined by their peers. This teaches students the connection between symbols and actions, the difference between an algorithm and a program, and the process of debugging.

Objectives:

- Pay attention to accuracy when creating instructions.
- Identify and solve bugs or errors in sequenced instructions.

Below is a detailed description of each of the parts of the lesson plan:

1) Talking to Robots (5 min): First we play a video to put students in context about the things robots can do (https://www.youtube.com/watch?v=kHBcVlqpvZ8). After that, we ask the students how they think the robot know what to do in the video and if a robot really understands what you say. The objective of this quick discussion is to point out that while robots seem to behave like people, in reality they only respond to their programming.

2) Introduction and Modelling (10 min): For this activity organised by groups, the teacher explains the students how to tell their “robot” friend to build a specific stack of plastic cups using only the four commands of Fig. 8 that the teacher shows on the board. The teacher shows a stack of plastic cups with a certain structure, asking the students what the algorithm (the step-by-step instructions) should be like for the robot to build it.

3) Programming Your Robots (20 min): The students work in groups of 4, dividing each group into two pairs. Each couple develops their own program to be “executed” by the other couple. Once the two couples in the group have completed their programs, they can take turns being “robots” by following the instructions that the other couple wrote. By way of example, Fig. 10 shows the stack of plastic cups resulting from the interpretation of the algorithm shown in Fig. 9.

4) Journaling (10 min): The students write in their diary of reflections on what they have learned, why it is useful and how they felt about it by drawing an emoticon. They also draw the stack of glasses they would like a robot to build, drawing the corresponding algorithm to build it. This can help to consolidate knowledge.

2.4.4. Phase 1, session 2, plugged-in: maze debug and bee & Artist Sequence

This session consists of three activities or lessons. In the first of them, students learn to debug, which is a process consisting of finding and solving errors (bugs) in the code. In the second activity we review the sequences seen in the previous session, adding some new element in addition to the movement arrows already known. Finally, another lesson is carried out to consolidate the knowledge about sequences, in which also some different elements are added to vary the type of exercises.

Objectives:
Below is a detailed description of each of the parts of the lesson plan:

- Predict where a program will fail and modify an existing program to solve its errors.
- Express the movement as a series of commands and arrange them as sequential steps in a program.
- Create a program to draw an image using sequential steps.

Fig. 8. Symbol key to build plastic cups stacks. Source: Code.org (2018c)

Fig. 9. Plastic cups stack algorithm example. Source: Code.org (2018c)

Fig. 10. Plastic cups stack example. Source: Code.org (2018c)

Fig. 11. Maze Debugging activity example.
1) Maze Debugging (15 min): In this activity, students find puzzles that have been solved incorrectly. They must go through existing code to identify errors, including missing blocks and unordered blocks, like in Fig. 11. The lesson is available in https://studio.code.org/s/course1/stage/5/puzzle/1.

2) Bee Sequence (15 min): In this activity the students help their bees to collect the nectar from the flowers and create honey in the honeycombs, as shown in Fig. 12. The students thus add blocks of action to the movement blocks that they already knew from the previous session. The lesson is available in https://studio.code.org/s/course1/stage/7/puzzle/1.

3) Artist Sequence (15 min): In this activity students take control of the Artist to make simple drawings on the screen, as Fig. 13 illustrates. The lesson is available in https://studio.code.org/s/course1/stage/8/puzzle/1.

2.4.5. Phase 1, session 3, unplugged: My loopy robotic friends

Having the acquired base of the previous activity “My robotic friends”, the students face more complicated designs. To program their “robots” and get to build these more complex designs, students must identify repeated patterns in their algorithms that could be replaced with a loop, thereby developing critical thinking skills.

Objectives:
- Identify repeated patterns in the code that could be replaced with a loop.
- Write algorithms that use loops to repeat patterns.

Below is a detailed description of each of the parts of the lesson plan:

1) My Robotic Friends Review (10 min): This review reminds the students about how quickly the “My Robotic Friends” activity programs can become complex. To do this, we show one of the easy structures from the previous session (see Fig. 14) and program it together as a reminder of the rules and terminology. Next, we show a more difficult structure, which requires the repetition of many symbols to program it (see Fig. 15). Once the students have come up with the idea of “repeating” the code, tell them the related vocabulary: “loop”.

2) Introduction and Modelling (10 min): As a demonstration for the whole class, the teacher shows a program for students to think about in which part they can find a repeating instruction pattern. Using one of the repetition patterns that the class identified, explain that this can be written in another way, as shown in Fig. 16.

3) Looping Your Robots (20 min): Similar to the activity of the “My Robotic Friends” session, groupings of four students are made and then each group is divided into two pairs. Each pair writes an algorithm that the “robot” can read later, and the teacher reminds the students to pay attention to the opportunities to replace a repeating pattern with a loop.

4) Journaling (5 min): The students write in their diary how they felt about the lesson by drawing an emoticon. They also write or draw something in their diary that reminds them of what loops are, such as describing what “repeat” means to them, or drawing a picture of themselves repeating something.

2.4.6. Phase 1, session 3, plugged-in: maze & Bee Loops

As a correspondence to the unplugged session on loops, this plugged-in session also focuses on them. It is divided into two activities or lessons. The first one introduces the concept of the loops with the characters and the workspace that the students already know, and the second strengthens it by adding some new elements in addition to the movement arrows already known.

Objectives:
- Identify the benefits of using loops instead of repetitions.

![Fig. 12. Maze Sequence activity example.](image-url)
Fig. 13. Artist Sequence activity example.

Fig. 14. Simple plastic cups stack example. Source: Code.org (2018b)

Fig. 15. Complex plastic cups stack example. Source: Code.org (2018b)

Fig. 16. Plastic cups stack simplified looped algorithm example. Source: Code.org (2018b)
Use a combination of sequential commands and loops to move and perform actions to reach a target within a maze.

Below is a detailed description of each of the parts of the lesson plan:

1) Maze Loops (22 min): Having worked in this workspace in the previous session, this stage makes students use loops to move around the maze more efficiently by needing fewer blocks of programming. Fig. 17 depicts a stage of the lesson, which is available in https://studio.code.org/s/course1/stage/13/puzzle/1.

2) Bee Loops (23 min): Like in the previous session, in this activity students help their bees to collect the nectar from the flowers and create honey in the honeycombs, but in this case, they use loops to help the bee collect more nectar and make more honey, as portrayed in Fig. 18. The lesson is available in https://studio.code.org/s/course1/stage/14/puzzle/1.

2.4.7. Phase 2, session 4: artist & Maze Loops (text blocks)

In this second phase the two groups perform the same activities, all of them of the plugged-in type. The only exception is that in this first session the unplugged group performs the initiation activity to the platform. From there, both groups continue to work with loops in two different activities.

Objectives:

- Identify the benefits of using loops instead of repetitions.
- Create a program that sketches complex shapes by the repetition of simple sequences.
- Use a combination of sequential commands and loops to move and perform actions to reach a target within a maze.
- Understand and know how to use other types of programming blocks such as text blocks, as well as their particularities.

Below is a detailed description of each of the parts of the lesson plan:

1) Jigsaw: Learn to drag and drop (10 min) (only unplugged group): This activity is explained in section 3.4.2.

2) Artist Loops (20 min): Returning to the Artist, students learn how to sketch more complex forms by looping simple sequences of instructions. Fig. 19 depicts a stage of the lesson in which students must draw the battlements of a castle. The lesson is available in https://studio.code.org/s/course1/stage/18/puzzle/4.

3) Maze Loops (Text blocks) (25 min): Students will use loops to move more efficiently through the maze they already know, but this time with text blocks, adding difficulty to the lesson. Fig. 20 shows an example of this lesson, which is available in https://studio.code.org/s/course2/stage/6/puzzle/1.

2.4.8. Phase 2, session 5: Maze Sequence & Bee Loops (text blocks) and Play Lab

This last session consists of three activities. The first of them serves to refresh the concept of sequence in which we worked during the first sessions, but this time with textual blocks. In the second lesson we continue working to consolidate the loops, also with textual blocks. Finally, in the last activity, students continue practicing with the environment in a relaxed way, creating an interactive story with new characters.

Objectives:

- Express the movement as a series of commands and arrange them as sequential steps in a program.
- Use a combination of sequential commands and loops to move and perform actions to reach a target within a maze.
- Understand and know how to use other types of programming blocks such as text blocks, as well as their particularities.

Fig. 17. Maze Loops activity example.
Fig. 18. Bee Loops activity example.

Fig. 19. Artist Loops activity example.

Fig. 20. Maze Loops (Text blocks) activity example.
Create an animated and interactive story using sequences, loops and event handlers, and share the creations with classmates.

Below is a detailed description of each of the parts of the lesson plan:

1) Maze Sequence (Text blocks) (15 min): Students review the sequences by programming with text blocks. Fig. 21 illustrates an example of this lesson, which is available in https://studio.code.org/s/course2/stage/3/puzzle/1.

2) Bee Loops (Text blocks) (15 min): In this activity students help again their bees to collect the nectar from the flowers and create honey in the honeycombs by using loops and text blocks, as shown in Fig. 22. The lesson is available in https://studio.code.org/s/course2/stage/8/puzzle/1.

3) Play Lab: Create a Story (15 min): In this culminating activity, students will have the opportunity to be creative and apply all the coding skills they have learned to create an animated story. Fig. 23 illustrates an example of this lesson, which is available in https://studio.code.org/s/course1/stage/16/puzzle/1.

3. Results

Considering the objectives of this work, this section shows the results obtained for both groups in each of the tests (Pre-, Mid- and Post-test) for each of the proposed areas to be compared in a grouped way. In the first place, the results related to CT are presented. Then, those corresponding to the motivation towards the instruction. Finally, the previous two taking into account the gender of the students.

3.1. Computational thinking

Regarding CT skills, Table 3 shows the average score of the three tests on a scale from 0 to 10, and the standard deviation of each of the measures is also included in parentheses. The descriptive data and a Mann-Whitney test indicated that there were already slight differences in favour of the plugged-in group (Pre-test\textsubscript{CT} = 3.87) in comparison to the unplugged group (Pre-test\textsubscript{CT} = 3.26) before the intervention that were statistically significant (U = 506.00, p = .009). Mann-Whitney tests were conducted since assumptions for parametric test were not fulfilled.

Table 4 presents the average difference between Pre-test\textsubscript{CT} and Mid-test\textsubscript{CT} (U = 369.00, p = .033), and between Pre-test\textsubscript{CT} and Post-
test\(_{CT}\) \((U = 373.00, p = .003)\), evidencing the greater improvement experienced by the unplugged group throughout the study.

Table 5 also shows the results of the Pre-test\(_{CT}\), Mid-test\(_{CT}\) and Post-test\(_{CT}\), but this time splitting every result by the difficulty level of the problem, depending if it is easy or hard. In this way, the plugged-in group was more successful both in easy \((U = 500.00, p = .003)\) and hard \((U = 645.00, p = .191)\) problems in the Pre-test\(_{CT}\), and in easy \((U = 586.00, p = .653)\) problems in the Mid-test\(_{CT}\). On the other hand, the unplugged group was more successful in hard \((U = 451.00, p = .004)\) and hard \((U = 594.50, p = .313)\) problems in the Post-test\(_{CT}\).

Next, Table 6 summarizes the differences between Pre-test\(_{CT}\) and the Mid-test\(_{CT}\) and between the Pre-test\(_{CT}\) and the Post-test\(_{CT}\) for both easy and hard problems. The comparison between Pre-test\(_{CT}\) and the Mid-test\(_{CT}\) showed higher gains for the unplugged group in the easy problems \((U = 466.50, p = .275)\) and, especially, in hard problems \((U = 300.00, p = .002)\). In the comparison between the Pre-test\(_{CT}\) and the Post-test\(_{CT}\), again larger improvements were experienced by the unplugged group in both easy \((U = 317.50, p < .001)\) and hard problems.
and hard \((U = 538.50, \ p = .285)\) problems.

### 3.2. Motivation

As shown in Table 7, no significant differences were identified between groups in the results of the Mid-test\(\text{Mot} \) \((U = 718.00, \ p = .814)\) and in the Post-test\(\text{Mot} \) \((U = 715.50, \ p = .413)\), although there is a slight decrease in the Post-test, more accentuated in the plugged-in group. The table also includes the results of both tests splitting them by dimensions (attention, relevance, confidence and satisfaction), in which no significant differences were found again.

### 3.3. Gender

In this section we organise the results of the computational and motivational areas from the prism of the gender of the students, to later analyse possible differences in this aspect.

Regarding CT, Table 8 shows the average score in all three tests, differentiating between boys and girls, giving us that the differences between both genders were not really significant in terms of CT. In the Pre-test, the girls from both groups clearly obtained better scores, but this was more balanced in the subsequent tests as the instruction developed.

Table 9 shows the average difference between Pre-test\(\text{CT} \) and Mid-test\(\text{CT} \) \((U = 86.00, \ p = .847)\), and between Pre-test\(\text{CT} \) and Post-test\(\text{CT} \) \((U = 127.50, \ p = .595)\), evidencing a greater improvement experienced by the boys throughout the study, although in this case the difference is not statistically significant.

Regarding motivation, Table 10 shows that in the Mid-test\(\text{Mot} \) the girls of the plugged-in group obtained lower results in general in comparison with the boys, especially in the dimensions of relevance and confidence, being the difference in this case statistically significant \((U = 94.50, \ p = .015)\). The difference is even greater if we talk about the Post-test\(\text{Mot} \), where the girls scored even less than in the Mid-test\(\text{Mot} \) in comparison with the boys, being that again statistically significant \((U = 116.50, \ p = .030)\). However, in the unplugged group there were hardly any differences between both genders, there was even a slightly higher motivation in the case of girls in both the Mid-test\(\text{Mot} \), although without no significant statistically differences, \((U = 182.50, \ p = .900)\) and the Post-test\(\text{Mot} \) \((U = 155.00, \ p = .277)\).

### 4. Analysis of results

Following the same structure of the previous section, the results in each of the three proposed areas are now analysed.

If we focus on the computational area, the averages of the results obtained in the Pre-test\(\text{CT} \) for both groups are less than 4, which indicates a deficient initial level in the field of CT. On the other hand, this result cannot be considered surprising since these types of problem-solving processes are not currently taught at schools. Despite the low initial scores, both groups experienced an improvement in the other two tests. The plugged-in group slightly improved in the Mid-test\(\text{CT} \), and more considerably in the Post-test\(\text{CT} \), with a two-point improvement compared to the Mid-test\(\text{CT} \). But even more so did the unplugged group, which already in the Mid-test\(\text{CT} \) improved in two points, and another two more in the Post-test\(\text{CT} \). Observing Table 4 it can be deduced more clearly that the unplugged group experienced this greater improvement, especially in the Mid-test\(\text{CT} \), which was performed after the first phase of unplugged instruction,

### Table 7

<table>
<thead>
<tr>
<th>Mid-test(\text{Mot} )</th>
<th>Plugged-in</th>
<th>Unplugged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>12.82 (2.34)</td>
<td>12.44 (1.68)</td>
</tr>
<tr>
<td>Relevance</td>
<td>12.00 (2.65)</td>
<td>12.64 (1.69)</td>
</tr>
<tr>
<td>Confidence</td>
<td>13.42 (1.72)</td>
<td>14.08 (1.26)</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>14.26 (1.35)</td>
<td>14.18 (1.48)</td>
</tr>
<tr>
<td>Post-test(\text{Mot} )</td>
<td>13.60 (7.95)</td>
<td>13.20 (7.00)</td>
</tr>
<tr>
<td>Attention</td>
<td>12.40 (2.83)</td>
<td>13.18 (1.96)</td>
</tr>
<tr>
<td>Relevance</td>
<td>11.33 (3.27)</td>
<td>12.10 (2.93)</td>
</tr>
<tr>
<td>Confidence</td>
<td>12.53 (3.12)</td>
<td>13.13 (2.15)</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>13.85 (2.19)</td>
<td>13.90 (1.84)</td>
</tr>
</tbody>
</table>

Source: Own elaboration based on the collected data.

### Table 8

<table>
<thead>
<tr>
<th>Group</th>
<th>Gender</th>
<th>Pre-test(\text{CT} )</th>
<th>Mid-test(\text{CT} )</th>
<th>Post-test(\text{CT} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugged-in</td>
<td>Girls</td>
<td>4.06 (1.25)</td>
<td>4.09 (1.97)</td>
<td>6.35 (2.06)</td>
</tr>
<tr>
<td></td>
<td>Boys</td>
<td>3.71 (1.15)</td>
<td>3.95 (2.46)</td>
<td>6.25 (1.25)</td>
</tr>
<tr>
<td>Unplugged</td>
<td>Girls</td>
<td>3.35 (1.30)</td>
<td>5.04 (1.56)</td>
<td>7.00 (1.54)</td>
</tr>
<tr>
<td></td>
<td>Boys</td>
<td>3.18 (1.23)</td>
<td>4.95 (2.70)</td>
<td>7.20 (1.37)</td>
</tr>
</tbody>
</table>

Source: Own elaboration based on the collected data.
something that speaks well of this type of activities.

Tables 5 and 6 evidence the improvement that happened in both groups throughout the study, especially in hard problems, although a greater improvement was experienced by the unplugged group. With a short instruction like the one the students did, it could be expected beforehand that the improvement would come in the easy category problems. However, the obvious greater improvement in hard problems may be due to the low score obtained in the Pre-test CT for this category of problems (less than one in both groups), something easily to improve after even a short instruction.

Regarding the analysis of the second objective (the motivation towards instruction), it is worth emphasizing that although the results were slightly better in the unplugged group, they were very similar in general, according to what Table 7 shows us. Below are detailed each of the dimensions separately.

The attention in the Mid-test Mot was slightly higher in the plugged-in group, coinciding its score with the Post-test Mot of the unplugged group, which makes sense, since both occurred after carrying out activities of the type plugged-in. However, in the Post-test Mot of the plugged-in group there was a decrease in attention, which may indicate that the activities became repetitive for the students. The previous score also coincides with the Mid-test Mot of the unplugged group. This may confirm what Tsarava et al. (2017) outlined about how a combined instruction of unplugged and plugged-in activities allows an integrated constructivist approach to transmit the respective content. In terms of relevance, confidence and satisfaction, we again obtained very similar scores. In the first two, the unplugged group scored slightly higher in both tests, while in terms of satisfaction the scores were somewhat better after performing plugged-in activities. But as stated above, the differences are minimal in all three cases.

Finally, we analyse the last of the objectives of this work regarding the gender of the students and how this could influence the first two objectives.

In the first place, Table 8 reveals that the girls started from a higher initial level in terms of CT, although later they were more balanced as the instruction was developed, even in the unplugged group boys surpassed girls in the Post-test CT. Something that this study does not cover is whether these differences in improvement between boys and girls also shown in Table 9 would follow that trend if the instruction were extended over time. What is certain is that this can lead us to infer that, despite the possible initial differences, with adequate instruction all children can develop skills related to CT regardless of their gender.

In the second place, Table 10 shows us that, although the differences in CT between boys and girls were not very marked, in relation to motivation it was found that in the plugged-in group there was a gap in which boys were less motivated than girls in the Post-test Mot. Something that this study does not cover is whether these differences in improvement between boys and girls also shown in Table 9 would follow that trend if the instruction were extended over time. What is certain is that this can lead us to infer that, despite the possible initial differences, with adequate instruction all children can develop skills related to CT regardless of their gender.

In the second place, Table 10 shows us that, although the differences in CT between boys and girls were not very marked, in relation to motivation it was found that in the plugged-in group there was a gap in which boys were less motivated than girls in the Post-test Mot. Something that this study does not cover is whether these differences in improvement between boys and girls also shown in Table 9 would follow that trend if the instruction were extended over time. What is certain is that this can lead us to infer that, despite the possible initial differences, with adequate instruction all children can develop skills related to CT regardless of their gender.

5. Conclusions

In order to establish the objectives of this work the following question was posed: Is it more appropriate to introduce CT in early
years of Primary Education through unplugged activities before plugged-in activities rather than do it exclusively through plugged-in activities? To answer this question, three research questions were set, consisting of analysing the development of CT skills, diagnosing the possible loss or gain in terms of motivation, and identifying possible differences related to the gender of the students regarding the two previous objectives. To do this, a comparison was made between two groups of the 2nd year of Primary Education, designating one as a control group and the other as an experimental group. To carry out this comparison, an instruction was designed for both groups, divided into two phases. The first phase was different for each group, and the second was the same for both groups.

In the case of the first research question it was found that the students had a low initial level in terms of CT, something that was expected considering the little or no instruction previously received on the subject. The proposed instruction emphasized on CT concepts that were suitable to the educational level of students, such as “directions”, “sequences” and “loops”. From there, it increased the level of CT skills, being that increase more remarkable in the experimental group, which attended unplugged activities in the first phase of instruction, as Brackmann et al. (2017) already revealed. This finding also underpins what Gitta et al. (2019) concluded, which is the connection between sensorimotor factors and high-level cognitive processes, which reveals the impact of an unplugged approach on the acquisition of computational thinking skills.

In reference to the second research question, related to motivation, the results obtained by the experimental group in the tests show a significant advantage in the use of an instruction with unplugged activities, even if these are followed by plugged-in activities. As in Tsarava et al. (2017) and Tsarava et al. (2018), this leads us to the conclusion that an instruction in CT based on unplugged activities or in unplugged activities followed by plugged-in activities is not only beneficial in terms of skills acquisition, but also in terms of the motivation of the students. This generates an obvious double advantage.

In the case of the third research question, related to the gender of the students and how it has been able to influence the development of CT and students’ motivation, the results confirm what was previously asserted in Espino and González (2015a, 2015b), allowing us to infer that it influences both, being more evident in the case of motivation. After this study it has been possible to verify that when working CT with students in early years of Primary Education (2nd year in this case), both respond better to an instruction composed of unplugged or mixed activities (unplugged and plugged-in), both in the development of CT skills and in terms of the motivation towards them. Speaking of motivation of girls in particular, this work responds to what (Kong, Chiu, & Lai 2018) suggested in attracting girls to this field of computer science. To the different interventions mentioned in that study, from here the promotion of unplugged activities is also proposed.

Analysing the achievement of the objectives, we can answer the initial question by concluding that it is more appropriate to work on CT in early years of Primary Education through a mixed approach that combines unplugged and plugged-in activities rather to do it only through plugged-in activities.

In future studies the size of the population could be increased, since although the one chosen for this study allows to draw some conclusions, it limits the generalization of the results. Also, the length of the instruction could be increased, since a deeper instruction from twelve to eighteen sessions as the ones collected in Code.org courses should provide more enlightening results to be measured with each of the proposed instruments. Regarding those instruments, there is a manifest shortage of validated tools that allow evaluating the development of skills related to CT that focus on the solution of ‘real’ problems, and not so much on others that are exclusive to programming. In this sense, we find appropriate the proposal for further investigation of the Bebras Tasks that several authors have already made (e.g. Román-González et al., 2019; Wiebe et al., 2019; or Araujo, Andrade, Guerrero, & Melo, 2019). In the other instrument used lies one of the limitations of this work, which is in the measure of motivation. Even though the wording of the questions was adapted to facilitate their understanding and the selection of answers was also facilitated by a Likert scale, it is fair to recognize that the complexity of the questions is perhaps too high to be used in early years of Primary Education. A future improvement could be the development of a measurement instrument adapted to these ages. Besides, the configuration of the instruction model may have affected the results on motivation. Future studies could assess different settings and analyse their impact on motivation.

Although the study developed in this work allows to draw some conclusions such as those mentioned above, there is still a long way to go. The inclusion of CT in the classroom is something inevitable, since present and future society yearns for people prepared to face its needs. This work aims to contribute its bit in this field of education.

CRediT authorship contribution statement

Javier del Olmo-Muñoz: Conceptualization, Methodology, Investigation, Data curation, Formal analysis, Writing - original draft, Writing - review & editing. Ramón Córza-Gutiérrez: Conceptualization, Methodology, Resources, Visualization, Writing - review & editing. José Antonio González-Calero: Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision.

Acknowledgments

This work was supported by the University of Castilla-La Mancha under Grant number 2019-GRIN-27150.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compedu.2020.103832.


