

Article

The Effect of Robotics-Based Storytelling Activities on Primary School Students' Computational Thinking

Karin Tengler ^{1,*} , Oliver Kastner-Hauler ¹ , Barbara Sabitzer ² and Zsolt Lavicza ² 

¹ Department of Media Pedagogy, University of Teacher Education Lower Austria, 2500 Baden, Austria; oliver.kastner@ph-noe.ac.at

² Department of STEM Education, Johannes Kepler University Linz, 4040 Linz, Austria; barbara.sabitzer@jku.at (B.S.); zsolt.lavicza@jku.at (Z.L.)

* Correspondence: k.tengler@ph-noe.ac.at

Abstract: Robotics is needed as education keeps up with challenges students are facing in a technological environment. A long-term research project focuses on developing a feasible robotics-based learning environment that enables primary school teachers to introduce computer science education. This paper shows educational robotics combining storytelling to promote computational thinking through the method of Tell, Draw, and Code. The study was conducted via pre–post test, using the Beginners Computational Thinking test (BCTt), with third and fourth graders (N = 40) to determine if the integration of robotics-based storytelling activities enhances computational thinking skills. Results show that an increase in computational thinking is evident after the intervention is implemented. The approach of combining stories, texts, and literature with educational robotics seems to be a promising concept to equip students with the required skills.

Keywords: computational thinking; educational robotics; primary school



Citation: Tengler, K.; Kastner-Hauler, O.; Sabitzer, B.; Lavicza, Z. The Effect of Robotics-Based Storytelling Activities on Primary School Students' Computational Thinking. *Educ. Sci.* **2022**, *12*, 10. <https://doi.org/10.3390/educsci12010010>

Academic Editors:
Sapounidis Theodosios,
Michail Kalogiannakis,
Nikolaos Fachantidis
and Dimitrios Stamovlasis

Received: 22 November 2021

Accepted: 24 December 2021

Published: 28 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Technological progress is constantly advancing, and it is part of students' everyday lives. The education system faces the challenge of keeping up with it. In the meantime, computational thinking (CT) [1] has become an important term in the teaching of primary school students when it comes to required future competencies besides critical thinking, collaboration, and creativity [2]. The introduction of computer science education from the primary level onwards seems to be a step in the right direction. However, age-appropriate tools and suitable concepts and methods are needed to implement innovations in the classroom successfully [3]. There are different approaches to fostering computational thinking in K–12 education [4]. Some suggest making computer science education a compulsory subject [5], while others recommend integrating coding and informatics as interdisciplinary elements in the curricula [6]. One option to introduce computer science education and promote computational thinking in various subjects is educational robotics, as different CT components are used in each area when coding, robotics, and game design are applied to enhance CT skills [7]. Since educational robotics provides learners with hands-on experiences to promote problem-solving skills [8], it is particularly suitable for primary schools. The teaching and learning method of storytelling offers an adequate tool to combine narrative and programming language to develop problem-solving thinking [9]. A long-term research project is dedicated to this topic. It investigates the use of programmable robots by applying the method of storytelling in interdisciplinary lessons in primary education to develop a feasible learning environment to promote CT. This paper presents a sub-study of this research project. The investigation aims to explore how robotic storytelling activities influence students' computational thinking and what design principles can be derived to develop an appropriate learning environment. To be able to do it we will briefly outline the theories of computational thinking and educational robotics followed by a

description of the Tell, Draw, and Code method and the research methodology. Results, discussion, and an outlook of future work complete the paper.

2. Theoretical Background

2.1. Computational Thinking

In recent years, it has become apparent that one of the essential goals in introducing computer science education in primary school is to foster the development of computational thinking skills that can also be applied in other disciplines [10–12]. The term computational thinking (CT) is ubiquitous when referring to competencies that 21st-century students should possess to analyze, formulate, and solve complex problems. The focus is not only on computer science topics [13] but also on how to tackle everyday problems. Wing [1] considers CT a skill for everyone and emphasizes its applicability in many subjects. Therefore, CT should be implemented in educational systems to prepare students appropriately for their future lives [13]. Several reviews document educational interventions [14], teacher development [4], and assessments concerning this problem-solving competence [15–17]. Related concepts, practices, and frameworks, including CT components as data analysis, decomposition, pattern recognition, abstraction, algorithm, and evaluation can be found in Bitesize [18], Brennan and Resnick [5], and Grover and Pea [19]. Brennan and Resnick [5] identified in their framework the following computational concepts, which trained students' CT by programming: sequences, loops, events, parallelism, conditionals, operators, and data.

When assessing the development of computational thinking skills, one encounters several methods, each with advantages and limitations [20]. Nevertheless, to date, it has been a significant challenge to assess the CT skills of primary students [15,21] because the assessment of CT should also include the thinking process and not just the final product of a program or the problem solved [5]. Although research has clearly demonstrated the efficacy and necessity of promoting CT, only a few validated assessment instruments measure students' CT [15]. Román-González et al. [15] developed the Computational Thinking test (CTt), which can be used independently of a programming environment for assessment CT. Since this test was created for 10–16-year-olds, it was necessary to develop a specific one for primary school children. Zapata-Cáceres et al. [16], therefore, adapted the test, called the Beginners Computational Thinking test (BCTt), and extended it with innovations and improvements. The BCTt is considered a valid and reliable test in terms of content for assessing CT in primary school students [22], which “has proven to be an instrument aimed at the early stages of primary education (five to ten years old), [. . .] independent of any environment, it focuses on 3D framework computational concepts, partially on computational practices, and ignores computational perspectives” [23] (p.10). The contained concepts are sequences, (simple and nested) loops and if-then, if-then-else, and while conditionals because these are considered suitable concepts for beginners' level [22].

Since this test is tailored precisely to our target group of 8–10-year-olds, and the programming concepts of the robot chosen, we decided to use this assessment for our pre-post study.

2.2. Robotics-Based Learning

There are several approaches to promoting CT in the classroom. One of them, the implementation of educational robotics in education, is becoming an emerging trend, mainly because programmable robots tend to be inexpensive, have various functions, and are also well received by students [9]. Robotics-based learning offers a possible approach to introduce computer science education at the primary level. Programmable robots serve as a motivating, didactic tool to promote computational thinking in addition to traditional approaches [24]. Coding is considered an essential skill of computational thinking [15,25]. Arfé et al. [25] found in their study with primary school children that coding activities are effective in developing planning skills that are plastic in young learners. Grover and

Pea [13] (p.40) mentioned in their review of CT environments and tools for K–12 students that “curricular activities such as game design and robotics have typically served well as a means for the iterative exploration of CT”. Intuitive and problem-based activities [26] enhance children’s critical thinking and problem-solving thinking skills [14] and can change their perspectives towards informatics [27]. Atmatzidou and Demetriadis [28] showed that educational robotics is suitable for encouraging children to program and promote problem-solving and computational thinking with no differences in age or gender. “Robotics activities in education offer opportunities for students to explore, create and apply knowledge to solve real-world problems” [29] (p. 2). Programmable robots are particularly suited to storytelling because students see the robots as living actors of their stories [30]. Leoste et al. [31] described in their paper “that robotics has a great potential in merging the best sides of digital storytelling and gamification”. Hassenfeld et al. [32] examined in their study the relationship between students’ literacy skills and the successful introduction of programming KIBO robots. Hassenfeld et al. [32] explored “The Coding as Another Language (CAL)” curriculum by using KIBO robots to integrate programming and literacy concepts. To demonstrate how literacy and CT skills can be equally promoted, this study also aims to contribute by bringing stories to life in a special way, by moving from written language to visual language to spoken language, and by activating synergies in the learning process through motivational control of robots.

The robot used in this research is the Ozobot. An Ozobot is a little floor robot that moves on two wheels and uses color sensors to follow lines and recognize color codes. Its appearance [30] and simple handling [33] make it suitable, especially for beginning programming with young learners. To control the robot with color programming, you draw lines and stick color dots in red, green, and blue on them, which initiate certain commands. Basically, Ozobots can be used with paper and pen as well as on a tablet or desktop computer. Using the Ozobot in several games like mazes, skills such as creativity, autonomy, logic, and programming can be developed [34]. The advantage of this robot is that it can be used from simple programming to learning complex programming [35] by applying the visual programming language Ozoblockly. However, since we are working with younger students and beginners in programming, only the paper-based version of programming the Ozobot is used in the intervention and serves as a basis of the Tell, Draw, and Code method, which is described in the following paragraph. As a result of the chosen programming version, only a limited selection of computational concepts [5] can be implemented. In particular, this is about the use of sequences and conditionals. A detailed explanation of the concepts in the context of the Tell, Draw, and Code method is given in the description of the learning environment (see Section 3.1).

2.3. Storytelling Activities—Tell, Draw, and Code Method

In their previous study, which is part of an educational design research study carried out in iterative cycles, the authors of this paper showed that storytelling activities combined with programmable robots can be used successfully at the primary level [9]. The primary aim of this design-based research project is to provide primary school teachers who have limited experience of teaching computer science in their training with a simple concept for introducing computer science lessons and an approach to computational thinking without using computers in the first instance. Through the simple use of programmable robots, where simple computer science concepts such as sequences and conditions are applied in a playful and hands-on way, teachers lose their hesitancy about programming and desire to go deeper.

A first evaluation of the robotics-based learning environment showed that teachers consider it a viable and feasible way to implement computer science and promote computational thinking [36]. The design was based on preliminary design principles identified in an empirical study with expert interviews [37]. In the context of the long-term study, the Tell, Draw, and Code method emerged, which combines narrative, visual, and programming languages to initiate successful student learning. This approach promotes narrative,

reading, and coding skills and serves as a tool for weaker students because it supports structuring and more intensive engagement with the text.

Tell, Draw, and Code is a didactic design that combines storytelling activities with educational robotics. This approach combines the strengths of the analog with those of the digital and aims to promote computational thinking in connection with creative storytelling and writing processes. Table 1 shows the components of the CT in the context of the Tell, Draw, and Code method.

Table 1. CT components in the context of the Tell, Draw, and Code method.

Components of the CT-Process	Description	Stages of the Problem-Solving Process Using the Tell, Draw, and Code Method
Data collection and analysis	the process of gathering appropriate information, making sense of data, finding patterns, and drawing conclusions	repeating or reading the text or fairy tale, reading the task, identifying the problem, which details of the story are relevant, which lines and codes are needed
Decomposition	breaking down a complex problem or system into smaller parts that are more manageable and easier to understand	defining which sequences are essential to draw the plot of the story, which characters are relevant
Pattern recognition	finding the similarities or patterns among small, decomposed problems that can help to solve more complex problems more efficiently	considering and discussing how certain activities of the protagonists can be represented
Abstraction	reducing complexity to define the main idea	visualizing details of the fairy tale by drawing lines and relevant details, designing costumes for the characters
Algorithm	series of ordered steps taken to solve a problem or achieve some end	programming the storyline of the tale in the correct order, using sequences and conditionals by drawing or sticking the appropriate codes
Evaluation	a process that allows making sure the solution does the job it has been designed to do and to think about how it could be improved	filming and (re-)telling the story, checking that the codes fit the plot of the story, presentation to other groups

With the introduction of simple programming languages, literary texts become a toolkit for encoding and decoding language. The division into groups is particularly suitable for solving the problem-based task collaboratively and purposefully. The robot used in this design is the Ozobot.

The students' task is to read a text or invent a story and graphically represent the story's plot as part of creative action. Texts or stories are first structured by having students consider and decide which characters are needed, which activities are depicted, and in what order. Then the path of the Ozobot and necessary details are drawn. Then, codes corresponding to the sequences of actions are drawn or pasted so that the Ozobot can carry out the commands, e.g., speed up out of fear of the dragon, etc. This transformation requires problem-solving strategies that the children apply in dialogic negotiation processes. Meanwhile, a discussion takes place, in which the story can be interactively developed further because the drawing and the programming can be optimized at any time. When the graphic representation is done, the participating group has their story run, retells it, and presents it to the other students. It has proven particularly effective when the story is filmed using a tablet, for example, and the children have an analog and digital artifact at their disposal. The dialogic negotiation of informatics problems in combination with creative representations of the stories in visual form should sustainably promote the children's narrative language. At the same time, the structuring and coding of the text are also intended to give children with reading and spelling difficulties the opportunity to engage

more deeply with the individual sections of the story through programming. Furthermore, this method also contributes to promoting multilingualism since the story can also be told in different languages.

Tell, Draw, and Code can be applied to the following settings:

1. Retelling stories or fairy tales
2. Creating stories—Using different text types
3. Extracting information of a text and presenting it
4. Becoming familiar with children’s literature

The Tell, Draw, and Code method finds its application in the intervention used as the didactic design for this study. In the following section discussing the method, a detailed description of the learning environment is presented.

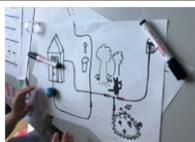
3. Method

Before outlining the method, participants, and instrument of this study, a description of the learning environment and its learning objectives and activities conducted during the study follows.

3.1. Learning Environment

The intervention consists of three units (see Table 2), an introduction to Ozobot programming, and a lesson with the fairy tale Little Red Riding Hood. In the third unit, the pupils retell the book “The Gruffalo” by drawing and coding it. A period of three weeks was available for the intervention.

Table 2. Lesson plans.

Units	Unit 1	Unit 2	Unit 3
Title	<i>Introduction—Ozobots</i>	<i>Fairy tale—Little Red Riding Hood</i>	<i>Retelling the book “The Gruffalo”</i>
Learning objectives	Getting to know the Ozobot robot Getting to know how it works Getting to know the programming	Telling the fairy tale Little Red Riding Hood Knowing the plot of the fairy tale Graphically representing the plot Programming the Ozobot according to the plot Retelling the fairy tale using the programmed lines and codes	Reading the book Graphically representing the plot of the book by drawing lines Including details of the plot in the plan (e.g., cave, forest) Programming the Ozobot according to the plot Retelling story Filming the story Giving feedback to the other students
Trained computational concepts	Introduction to programming Simple sequences	Advanced sequences If-then conditionals	Sequences If-then conditionals If-then-else conditionals While conditionals
Activities	Experimenting Drawing lines Using the programming codes	Telling the fairy tale or solving a riddle Drawing the path of Little Red Riding Hood Using appropriate codes, such as turn right, slow down, or speed up Retelling the fairy tale	Reviewing the functions of the Ozobot Reading the story Drawing the plot Sticking appropriate codes on the lines Retelling the story Filming Giving feedback
Artifacts			

The research design of the intervention chosen here refrains from using computers to program the Ozobot robot. Thus, there is a deliberate focus on the haptic experience in the design of the programming. Paper, pencil, and additional materials are used to give the story a physical form and to implement it in the spirit of constructionism. With lines and dots spiced with color codes, the programming of sequences and conditional branching takes place. According to the described type of use, the computational concepts of the assessment (BCTt) can be used except for loops, which can also be statistically proven based on the collected data.

The first unit starts with a description of the Ozobot and how it works. After that, the students have their first experiences with programming. Paper and pens are necessary for this. First, the lines are drawn, and the students observe how the Ozobot reacts. It is important to pay attention to the precise drawing of the lines and keep a certain width (about 4–5 mm). After that, students are given a table with the codes and apply them by gluing or drawing them. The children are introduced to the concept of sequences in a playful way. By sticking different codes, the Ozobot executes a sequence of commands, for instance by changing the speed. To encourage collaboration and communication, the students are divided into groups. Now, the learners can try out shapes of their choice and thus have their first experience with the Ozobot.

The second unit is about the fairy tale Little Red Riding Hood. The story of the fairy tale is repeated with the children to make sure that all students are familiar with the topic. Then they are given a worksheet with the problem-based task in which they are to graph the fairy tale plot. Students are instructed to draw lines for the Ozobot and the details of the story. Then the Ozobot is programmed to follow Little Red Riding Hood's path through the forest, past the wolf, to her grandmother. Various programming codes, such as turn left, speed up, etc., should be incorporated. In addition to using the sequences, this unit also introduces the if-then conditionals by programming details of the story. This can be seen in the following example. If Little Red Riding Hood wants to pick flowers, the robot has to slow down. The students must then add the appropriate code to slow down. If there is enough time, students can also design appropriate costumes for the characters. After that, the Ozobot runs the story, and the fairy tale is retold using lines and codes.

The third unit follows the same principle. First, the students are introduced to the book *The Gruffalo*. Then the goal is to represent the plot of the book graphically. The students can contribute their ideas in the creative process. This book is particularly suitable because the mouse follows a certain path and completes individual stations. The details of the story are easily identified and then implemented both narratively and programmatically. The third unit is expanded to include the if-then-else and while conditionals. As an example of the while conditional, students could program the following statement. While the Gruffalo walks through the forest, the Ozobot glows green. It is solved by drawing a green line in the section of the forest. In the end, the story can still be filmed with tablets, which also contributes to the evaluation. Afterward, each group can give feedback to the other groups on how they liked the lesson.

3.2. Design Research

The pre–post-test study is part of a long-term educational design research study conducted in iterative cycles [38]. The research has been conducted in selected primary schools since 2019/20. Previous cycles have included studies of preliminary design principles [37] and an evaluation of the design of the robotics-based learning environment and the perceptions of primary school teachers [36]. After developing the Tell, Draw, and Code method (see Section 2.3) during the designing process of the learning environment, this study aims to investigate whether this combination of storytelling activities and programmable robots is suitable to promote computational thinking skills and provide an applicable concept to teachers. Therefore, the following research questions emerged:

RQ1: To what extent do robotics-based storytelling activities affect students' computational thinking?

RQ2: How should robotics-based learning environments using storytelling activities be designed to promote primary school students' computational thinking skills?

3.3. Participants

Participants were recruited from an urban primary school, which is part of the university. The sample consisted of 45 third and fourth-grade students, 19 female and 26 male students, aged 9–10 years. The students already had prior experience using digital media but had no experience programming the Ozobot until the beginning of the study. Forty-five students participated in the three-week intervention. Finally, data from 40 students (Table 3) were available for analysis because these students completed both tests, the pre- and post-test. A total of 19 children of the third grade, 6 girls (31.6%) and 13 boys (68.4%), and 21 children of the fourth grade, 11 girls (52.4%) and 4 boys (47.6%) completed both assessments. To meet all the legal and ethical research requirements, we had obtained permission to conduct the research with these students. Their confidentiality was also assured, and their identity was not reported anywhere on the test. We used unique codes to compare pre- and post-test.

Table 3. Demographic data.

BCTt	Grade	Total	Female	Male	Age (M/SD)
pre/post	3.	19	6 (31.6%)	13 (68.4%)	9.45/0.36
pre/post	4.	21	11 (5.4%)	10 (47.6%)	10.27/0.66

3.4. Instruments

The Beginners Computational Thinking test (BCTt) version 2 developed by Zapata-Cáceres et al. [16] is used as the assessment tool for the CT measurement. This version of the BCTt is 25 items long, with an estimated time of 40 min. Four alternative responses are given to each item. The test contains the following computational concepts (Figure 1): sequences (6 items), loops (12 items), and conditionals (7 items). A chick and her mother (hen) are the main protagonists in the tasks that need to be solved [16]. The test was provided to the students as an online version. It was administered at the beginning and end of the intervention in the school's computer lab. Each student entered a unique identification code to match pre-test and post-test and maintain anonymity. Before the test, students were given an explanatory example of each of the computational concepts addressed in the test, as recommended by the BCTt's authors.

Items	Computational Concepts in BCTt					
	1. Sequences	Loops		Conditionals		
		2. Simple loops	3. Nested Loops	4. If-then conditionals	5. If-then-else conditionals	6. While conditionals
1 - 6						
7 - 11						
12 - 18						
19 - 20						
21 - 22						
23 - 25						

Figure 1. Computational concepts of the BCTt [16].

3.5. Data Analysis

To analyze data of the pre- and post-test, only the responses of the students, who completed both tests, were used. The socio-demographic data on age and gender and the results of the items were evaluated. The *t*-test, a post hoc paired sample test, was applied to determine a possible increase in CT skills, which is appropriate for comparing dependent samples. The items of the BCTt were coded binarily. That is if the answer was correct, 1 point was assigned, and if it was incorrect, 0 points. The data were analyzed with descriptive statistics using SPSS 27. Results of the BCTt are presented and discussed in the following paragraph.

4. Results and Discussion

4.1. Computational Thinking

The BCTt was administered in each class two equal times, before and after the intervention, so that everyone had the same conditions. As also noted and mentioned in a similar study [22], less time was required to administer the BCTt post-test than to perform the pre-test. In the second test, it was observed that students were well acquainted with explaining and performing the test and they could already explain individual computational concepts on their own. According to Zapata-Caceres et al. [22], this result is consistent with improved CT skills, especially automation skills.

The BCTt score is the sum of all correct answers in the 25 tasks of the test. An initial analysis of the BCTt was conducted after the pre-test. The sample population appeared to be normally distributed. After the three-week intervention, pre- and post-test were compared, and collected BCTt data were analyzed to determine to what extent robot-based storytelling activities affect students' computational thinking.

The evaluation found overall better mean scores in the third and fourth-grade post-test sample (Table 4), which indicates a positive impact of the intervention. In the pre-test, the full score of 25 points could only be achieved in the fourth grade (Table 4). In the post-test, students from both grades were able to answer all items correctly. The minimum score was 7 points on the pre-test and 9 points on the post-test. On average, many items of the pre-test and even more of the post-test were answered correctly. This ranges from 19.89 points to 22 points out of a total of 25 to be achieved.

Table 4. Results of the BCTt.

BCTt	Grade	N	Min	Max	Mean	SD
pre	3	19	14	24	19.89	3.035
post	3	19	17	25	22.21	2.974
pre	4	21	7	25	20.81	4.297
post	4	21	9	25	22.00	3.464

Table 5 shows the post hoc paired sample *t*-test. The *t*-test was applied between the BCTt pre-test and post-test results. Since $p < 0.05$, grade 3 ($p = 0.000$) and grade 4 ($p = 0.007$), there is statistical significance of the difference between the pre-test and post-test results. These results indicate that considering the overall results, an effective approach to computational thinking through robot-based storytelling activities was achieved.

Table 5. Post hoc paired sample *t*-test.

Grade	N	Mean	SD	<i>t</i>	df	Sig. (2-Tailed)
3	19	2.316	2.335	4.324	18	0.000
4	21	1.190	1.834	2.975	20	0.007

An interesting fact is documented in Table 4. If the average total number of points is considered, there is a higher increase in the third grade ($M = 2.316$, $SD = 2.335$) than

in the fourth grade ($M = 1.190$, $SD = 1.834$). However, the fourth-grade children already achieved a higher score on the pre-test ($M = 20.81$, $SD = 3.464$) than the third-grade students ($M = 19.89$, $SD = 3.035$), as can be seen in Table 3.

To be more detailed on partial aspects, the data of both grades were also analyzed about computational concepts and compared with the t -test. The scales evaluated were sequences (6 items), simple loops (5 items), nested loops (7 items), if-then-conditionals (2 items), if-then-else conditionals (2 items), and while conditionals (3 items). Table 6 shows the mean value of each item belonging to the respective computational concept of the third-grade students.

Table 6. Evaluation computational concepts, third grade.

Computational Concepts	BCTt	Grade	N	Mean	SD
Sequences	pre	3	19	0.877	0.199
	post			0.930	0.101
Simple loops	pre	3	19	0.979	0.063
	post			0.979	0.063
Nested loops	pre	3	19	0.759	0.243
	post			0.842	0.223
If-then conditionals	pre	3	19	0.711	0.303
	post			0.816	0.287
If-then-else conditionals	pre	3	19	0.579	0.344
	post			0.816	0.299
While conditionals	pre	3	19	0.737	0.285
	post			0.859	0.202
Total	pre	3	19	19.89	2.974
	post			22.21	3.035

Table 7 illustrates the evaluation of the computational concepts of the fourth-grade students. The results show the mean value of the items belonging to each computational concept.

Table 7. Evaluation computational concepts, fourth grade.

Computational Concepts	BCTt	Grade	N	Mean	SD
Sequences	pre	4	21	0.944	0.199
	post			0.992	0.101
Simple loops	pre	4	21	0.914	0.063
	post			0.905	0.063
Nested loops	pre	4	21	0.726	0.243
	post			0.918	0.223
If-then conditionals	pre	4	21	0.786	0.303
	post			0.857	0.287
If-then-else conditionals	pre	4	21	0.595	0.436
	post			0.691	0.357
While conditionals	pre	4	21	0.603	0.202
	post			0.731	0.202
Total	pre	4	21	20.81	3.464
	post			22.00	4.297

Table 8 shows the post hoc paired sample t -tests of the third-grade students. The t -test is utilized to compare two dependent samples.

Table 8. Post hoc paired sample *t*-test, third grade.

Computational Concepts	BCTt	N	Mean	SD	Sig. (2-Tailed)
Sequences	pre/post	19	0.052	0.208	0.285
Simple loops	pre/post	19	−0.000	0.094	1.000
Nested loops	pre/post	19	0.827	0.279	0.213
If-then conditionals	pre/post	19	0.105	0.394	0.259
If-then-else conditionals	pre/post	19	0.368	0.367	0.000
While conditionals	pre/post	19	0.122	0.277	0.069

Although the comparison of the total results shows a significant result, in the values of the third graders (Table 7), one can see only significant differences ($p < 0.05$) in the concept of the conditions if-then-else conditionals ($p = 0.000$). This can be seen prominently in Figure 2.

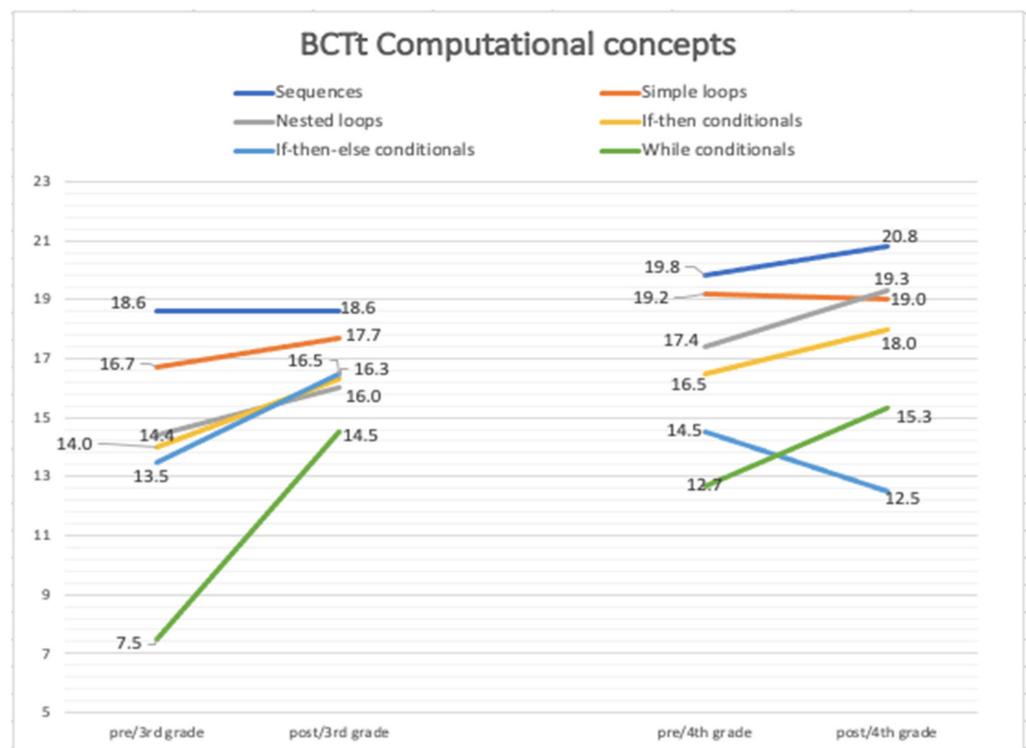


Figure 2. BCTt computational concepts.

Table 9 represents the post hoc paired sample *t*-tests of the fourth-grade students. While increases can also be noted in most concepts for fourth-grade students, only the concept of nested loops shows a significant difference ($p = 0.000$).

Table 9. Post hoc paired sample *t*-test, fourth grade.

Computational Concepts	BCTt	N	Mean	SD	Sig. (2-Tailed)
Sequences	pre/post	21	0.048	0.107	0.055
Simple loops	pre/post	21	−0.009	0.161	0.789
Nested loops	pre/post	21	0.192	0.155	0.000
If-then conditionals	pre/post	21	0.071	0.327	0.329
If-then-else conditionals	pre/post	21	−0.095	0.089	0.296
While conditionals	pre/post	21	0.127	0.071	0.088

To visualize the growth in problem-solving skills represented here in computational concepts skill increase, the following chart shows the average sum achieved for each

concept (scale), such as sequences, etc., on the pre-test compared to the third and fourth-grade post-test (Figure 2). The diagram demonstrates that, above all, an increase in the sequences and the conditions is recognizable. Growth of the competencies is visible for the if-then and the while-conditions for both classes. In addition, it is evident that the fourth-grade students got a higher score on most scales during the pre-test. The sequences scale was answered correctly by almost all fourth-grade students on the post-test ($M = 20.8$). The third-grade students could achieve a particularly enormous increase (from $M = 7.5$ to $M = 14.5$) in the scale if-then-else conditionals. In contrast, the fourth grade performed particularly weakly here ($M = 12.5$). The diagram points out that the single loop scale does not show any growth at all, but nested loops do.

In summary, it can be suggested that an increase in computational concepts is related to the intervention, especially for those that are promoted by the Ozobot programming.

4.2. Discussion

Belonging to the final cycle of educational design research, this study aimed to examine to what extent the Tell, Draw, and Code method is suitable to promote students' computational thinking skills. The instrument applied was the BCTt [16], which can be used to measure competencies of CT concepts. By using an easily administered, validated CT test, we were able to conduct this study with primary school children. Moreover, the study demonstrates the potential utility of the BCTt for assessing CT in primary school children.

Comparing the pre- and post study shows a suggestive increase of CT skills in both the third graders and the fourth-grade students (see Table 5) regarding the total score of points. An improvement is particularly noticeable in the concepts of sequences and conditions (see Figure 2). These results can possibly be explained by the fact that programming Ozobots primarily teaches the concepts of sequences and conditions to the students. These results show that the Tell, Draw, and Code method helps to ensure that the individual plot steps are programmed in the correct sequence. Above all, conditions are rather important for interpreting the story, e.g., when Little Red Riding Hood sees the wolf, she stops. This condition must then be coded with the correct colors. When retelling the fairy tale, the text is decoded again. For teaching loops, the visual programming language Ozoblockly is more recommended. As also determined by observations in a previous study of the research project [9], this study provides empirical evidence that teaching coding with robotics-based storytelling activities to third and fourth-grade students (ages 9–10) can promote the development of CT skills. In particular, the good sub-results of the concepts sequences and conditions favor applying this method in primary education and offering a successful introduction to computational thinking already in young learners [39]. As some researchers have emphasized the importance of knowledge transfer and the application to gain problem-solving skills in the emergence of programming skills [6,19,40], this study also provides evidence to support this perspective. The results provide similar findings to those seen in the studies of Zapata-Caceres et al. [22] and Relkin et al. [41].

Combining stories, texts, and literature and educational robotics brings equal benefits to various subjects. The work provides curriculum designers and researchers with needed insights into the design and development of such application examples to introduce computer science education and increase students' knowledge. The great advantage of robots is that they immediately implement the newly programmed commands. Their actions allow direct feedback on whether a code has been programmed correctly or needs to be corrected [33]. Thus, also in this study, the plots of the stories were implemented as paths and codes. The students were given immediate feedback on whether the coding fit their story or not.

Thus, as a pilot study [42] investigated, in addition to creativity in programming their actions, the ability to recognize problems, abstract them, and solve them step by step is also promoted. By working in a team, the students always train communicative and cooperative skills at the same time. Great motivation, good collaboration, and constructive discussions, as also mentioned in previous studies by the authors [9,43] could be observed

during the intervention by both the researchers and the practitioners. Moreover, the results of the creative artifacts clearly show the successful implementation and achievement of the learning objectives, the coding of the stories. The combination of educational robotics and the storytelling method [29,32] offers educators a cross-curricular, feasible approach to teaching children problem-solving thinking. Through the results of this study, it can be assumed that this cross-curricular use is a good example of implementing the future curriculum, as described by Relkin et al. [41] in their investigation of programming and literacy concepts with the KIBO robot.

Furthermore, this study offers the opportunity for researchers to implement the Tell, Draw, and Code method in higher grades and evaluate it by administering the Computational Thinking test (CTt) [15].

4.3. Design Principles

As mentioned in Section 3.1, this study is part of a long-term project using educational design research [38]. Since the research goal is to develop an innovative intervention, the first step was to identify preliminary design principles through an empirical study [44]. The design principles represent the core element of an educational design study [38] to define the characteristics of an intervention, which are then concretized and evaluated in the instructional design [45]. Since the study presented is part of the last cycle of the research project, and it was found that CT skills can be promoted by using this robotic learning environment in combination with storytelling activities, the final design principles could be identified (see Table 10). These can be used as a guide for developing viable robotics-based approaches for implementing computer science instruction in primary schools.

Table 10. Design principles.

Six Design Principles for Designing a Robotics-Based Learning Environment	
If You Want to Design a Robotics-Based Learning Environment Supporting Computational Thinking Skills, You Are Advised to:	
Principle 1	<i>Choose familiar topics of the students</i>
Principle 2	<i>Use age-appropriate programmable robots</i>
Principle 3	<i>Provide tasks that encourage problem-solving thinking</i>
Principle 4	<i>Apply a cross-curricular approach</i>
Principle 5	<i>Consider using a playful approach</i>
Principle 6	<i>Encourage learners to create their own ideas</i>

The following design principles are recommended for designing an interdisciplinary robotics-based learning environment in detail.

Choose familiar topics of the students. When students are offered familiar issues, they are better able to focus on acquiring new knowledge, such as a new programming language in this case. This aspect also increases their motivation to solve the tasks and promotes their ability to think in a problem-solving manner [24]. The theme of fairy tales and children's literature used in the research included familiar themes for the children and had a very inspiring effect.

Use age-appropriate programmable robots. An important decision is which robots are selected for the learning environment [31]. In this context, it should be ensured that the application of robots appropriately depends on the students' age. Since haptic perception is crucial for primary school students, floor robots are particularly suitable for this purpose. The Ozobot robot used in this research project is well suited for the primary school sector and meets all requirements. Moreover, in addition to color-coding, it is possible to extend knowledge by programming it with the visual programming language.

Provide tasks that encourage problem-solving thinking. With problem-based and project-based learning [26], two methods for planning such learning environments should be considered, leading to an even more intensive approach to problem-solving strategies. Students should learn in a problem-oriented and exploratory way [28] by not receiving all

instructions but by discovering solutions themselves, whereby a gradual increase in the level of difficulty should be considered here.

Apply a cross-curricular approach. Especially in primary schools, it is advisable to introduce computer science education, in this case, programming robots, in an interdisciplinary way. As demonstrated in this study, applying the storytelling method results in a combination of narrative, image, and programming language, programming skills and literacy [32] are promoted simultaneously.

Consider using a playful approach. For young learners, the playful approach is especially important when introducing them to computer science education. Programmable robots are suitable for this purpose due to their appearance [30]. Since they are very appealing to children, they can be used to teach complex topics playfully. The playful aspect additionally enriches the combination of programming and storytelling [29] and promotes collaboration and communication skills [28].

Encourage learners to create their own ideas. It became apparent that working with the Ozobot and creating their stories inspired the students to develop their ideas. Creativity is perhaps the most important skill that students need to learn [2], and it is the beginning of many innovations [29].

5. Conclusions and Future Work

The project's goal was to create a learning environment for the primary classroom that teachers could easily use in the regular classroom, and that meets the need to promote computational thinking in a cross-curricular context. Similarly, this study aims to provide a fun, effective, and practical approach using the Tell, Draw, and Code method to achieve widespread adoption in the education system. This study shows that the combination of programming language, visual language, and narrative language is appropriate to promote CT skills in primary school students. Thus, robotics education is a promising approach to promoting CT skills and introducing computer science education to today's K–12 learners. The Tell, Draw, and Code method can also be applied in their classroom using other robots, Bee-bots, for instance. In addition, it would be advisable to repeat the study with control groups at each grade level. Since the Tell, Draw, and Code method also shows a high benefit for students with special needs, further research on this aspect of the topic would also be desirable. Specifically, further investigations could determine the extent to which this method impacts students' narrative and literacy skills.

The design principles developed for robotics-based learning environments were developed in a specific context and cannot yet be generalized. For generalization, the proposed design principles need to be validated in different contexts in further research [46]. Nevertheless, we believe that the design principles presented are a valuable contribution to the existing knowledge on promoting computational thinking in primary education. The design principles and the developed method should complement each other. On the one hand, the storytelling method can help educators arrange both cross-curricular activities to implement computer science education. On the other hand, the design principles contribute to the theory and provide educators with more insights into designing the learning environment.

Future work is planned to introduce the Tell, Draw, and Code method to pre-service teachers and investigate their attitudes and willingness to use the method in their future educational work.

Author Contributions: Conceptualization, K.T.; methodology, K.T.; investigation, K.T. and O.K.-H.; writing—original draft preparation, K.T.; writing—review and editing, K.T.; visualization, K.T.; supervision, B.S. and Z.L.; project administration, K.T. All authors have read and agreed to the published version of the manuscript.

Funding: Open Access Funding by the University of Linz.

Institutional Review Board Statement: The study was conducted following the ethical guidelines of Johannes Kepler University, the University of Lower Austria, and international guidelines.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study. The study was conducted according to the university's ethical guidelines.

Data Availability Statement: For further information concerning the data please contact the author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33–35. [CrossRef]
2. Fadel, C.; Bialik, M.; Trilling, B. *Four-Dimensional Education*; Center for Curriculum Redesign: Boston, MA, USA, 2015; ISBN 1-5186-4256-X.
3. Rogers, E.M.; Singhal, A.; Quinlan, M.M. Diffusion of innovations 1. In *An Integrated Approach to Communication Theory and Research*; Routledge: New York, NY, USA, 2019; pp. 415–434; ISBN 0-203-71075-4.
4. Schina, D.; Esteve-González, V.; Usart, M. An overview of teacher training programs in educational robotics: Characteristics, best practices and recommendations. *Educ. Inf. Technol.* **2021**, *26*, 2831–2852. [CrossRef]
5. Brennan, K.; Resnick, M. New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, BC, Canada, 14–17 April 2012; Volume 1, p. 25.
6. Angeli, C.; Voogt, J.; Fluck, A.; Webb, M.; Cox, M.; Malyn-Smith, J.; Zagami, J. A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *J. Educ. Technol. Soc.* **2016**, *19*, 47–57.
7. Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying computational thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [CrossRef]
8. Chalmers, C. Robotics and computational thinking in primary school. *Int. J. Child-Comput. Interact.* **2018**, *17*, 93–100. [CrossRef]
9. Tengler, K.; Kastner-Hauler, O.; Sabitzer, B. Enhancing Computational Thinking Skills using Robots and Digital Storytelling. In Proceedings of the CSEDU 2021, online, 23–25 April 2021; pp. 157–164.
10. Barr, V.; Stephenson, C. Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community? *ACM Inroads* **2011**, *2*, 48–54. [CrossRef]
11. Chen, G.; Shen, J.; Barth-Cohen, L.; Jiang, S.; Huang, X.; Eltoukhy, M. Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Comput. Educ.* **2017**, *109*, 162–175. [CrossRef]
12. Lye, S.Y.; Koh, J.H.L. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Comput. Hum. Behav.* **2014**, *41*, 51–61. [CrossRef]
13. Grover, S.; Pea, R. Computational thinking in K–12: A review of the state of the field. *Educ. Res.* **2013**, *42*, 38–43. [CrossRef]
14. Tzagkaraki, E.; Papadakis, S.; Kalogiannakis, M. Exploring the Use of Educational Robotics in primary school and its possible place in the curricula. In Proceedings of the Educational Robotics International Conference, Siena, Italy, 25–26 February 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 216–229.
15. Román-González, M.; Moreno-León, J.; Robles, G. Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational Thinking Education*; Springer: Singapore, 2019; pp. 79–98.
16. Zapata-Cáceres, M.; Martín-Barroso, E.; Román-González, M. Computational thinking test for beginners: Design and content validation. In Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 27–30 April 2020; pp. 1905–1914.
17. Tang, X.; Yin, Y.; Lin, Q.; Hadad, R.; Zhai, X. Assessing computational thinking: A systematic review of empirical studies. *Comput. Educ.* **2020**, *148*, 103798. [CrossRef]
18. BBC. Bitesize. *Introduction to Computational Thinking*. 2017. Available online: <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1> (accessed on 20 December 2021).
19. Grover, S.; Pea, R. Computational Thinking: A competency whose time has come. In *Computer Science Education: Perspectives on Teaching and Learning in School*; Bloomsbury Publishing: London, UK, 2018; Volume 19. [CrossRef]
20. Govind, M.; Bers, M. Assessing Robotics Skills in Early Childhood: Development and Testing of a Tool for Evaluating Children's Projects. *J. Res. STEM Educ.* **2021**, *7*, 47–68. [CrossRef]
21. Lee, T.Y.; Mauriello, M.L.; Ahn, J.; Bederson, B.B. CTArcade: Computational thinking with games in school age children. *Int. J. Child-Comput. Interact.* **2014**, *2*, 26–33. [CrossRef]
22. Zapata-Caceres, M.; Martin, E.; Roman-Gonzalez, M. Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study. *IEEE Trans. Learn. Technol.* **2021**, *1*. [CrossRef]
23. Zapata Cáceres, M.; Martín-Barroso, E.; Román-González, M. BCTt: Beginners Computational Thinking Test. In *Understanding Computing Education, Proceedings of the Raspberry Pi Foundation Research Seminar Series, online, May–December 2020*; Raspberry Pi Foundation: Cambridge, UK, 2021; Volume 1, p. 1.
24. Angeli, C.; Valanides, N. Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Comput. Hum. Behav.* **2020**, *105*, 105954. [CrossRef]
25. Arfé, B.; Vardanega, T.; Montuori, C.; Lavanga, M. Coding in primary grades boosts children's executive functions. *Front. Psychol.* **2019**, *10*, 2713. [CrossRef]

26. Conde, M.Á.; Rodríguez-Sedano, F.J.; Fernández-Llamas, C.; Gonçalves, J.; Lima, J.; García-Peñalvo, F.J. Fostering STEAM through challenge-based learning, robotics, and physical devices: A systematic mapping literature review. *Comput. Appl. Eng. Educ.* **2021**, *29*, 46–65. [[CrossRef](#)]
27. Papavlasopoulou, S.; Giannakos, M.N.; Jaccheri, L. Exploring children's learning experience in constructionism-based coding activities through design-based research. *Comput. Hum. Behav.* **2019**, *99*, 415–427. [[CrossRef](#)]
28. Atmatzidou, S.; Demetriadis, S. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robot. Auton. Syst.* **2016**, *75*, 661–670. [[CrossRef](#)]
29. Stork, M.G. Supporting Twenty-First Century Competencies Using Robots and Digital Storytelling. *J. Form. Des. Learn.* **2020**, *4*, 43–50. [[CrossRef](#)]
30. Fanny, B.; Julie, H.; Anne-Sophie, C. Developing a critical robot literacy for young people from conceptual metaphors analysis. In Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 21–24 October 2020; pp. 1–7.
31. Leoste, J.; Pastor, L.; López, J.S.M.; Garre, C.; Seitlinger, P.; Martino, P.; Peribáñez, E. Using Robots for Digital Storytelling. A Game Design Framework for Teaching Human Rights to Primary School Students. In Proceedings of the International Conference on Robotics in Education (RiE), online, 30 September–2 October 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 26–37.
32. Hassenfeld, Z.R.; Govind, M.; De Ruiter, L.E.; Bers, M.U. If You Can Program You Can Write: Learning Introductory Programming across Literacy Levels. *J. Inf. Technol. Educ.* **2020**, *19*, 65–85. [[CrossRef](#)]
33. Picka, K.; Dosedla, M.; Stuchlikova, L. Robotic didactic aid Ozobot in Czech schools. In Proceedings of the 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), Košice, Slovenia, 12–13 November 2020; pp. 525–533.
34. Román-Graván, P.; Hervás-Gómez, C.; Martín-Padilla, A.H.; Fernández-Márquez, E. Perceptions about the use of educational robotics in the initial training of future teachers: A study on steam sustainability among female teachers. *Sustainability* **2020**, *12*, 4154. [[CrossRef](#)]
35. Fojtik, R. The Ozobot and education of programming. *New Trends Issues Proc. Humanit. Soc. Sci.* **2017**, *4*, 8–16. [[CrossRef](#)]
36. Tengler, K.; Kastner-Hauler, O.; Sabitzer, B. A Robotics-based Learning Environment Supporting Computational Thinking Skills—Design and Development. In Proceedings of the 2021 IEEE Frontiers in Education Conference (FIE), Lincoln, NE, USA, 13–16 October 2021; pp. 1–6. [[CrossRef](#)]
37. Tengler, K.; Kastner-Hauler, O.; Sabitzer, B. Identifying preliminary design principles. In Proceedings of the 4th International Conference on Computer Science & Education, Lancaster, UK, 17–21 August 2021; pp. 733–738.
38. McKenney, S.; Reeves, T.C. *Conducting Educational Design Research*; Routledge: London, UK, 2018; ISBN 1-351-60176-8.
39. Kong, S.-C.; Abelson, H.; Lai, M. Introduction to computational thinking education. In *Computational Thinking Education*; Springer: Singapore, 2019; pp. 1–10.
40. Repenning, A.; Escherle, N.A.; Lamprou, A. Die ersten 1000: Computational Thinking als obligatorische Ausbildung für Primarschullehrpersonen in der Schweiz. *Medien Z. Theor. Prax. Medien* **2020**, *17*, 595–616. [[CrossRef](#)]
41. Relkin, E.; de Ruiter, L.E.; Bers, M.U. Learning to code and the acquisition of computational thinking by young children. *Comput. Educ.* **2021**, *169*, 104222. [[CrossRef](#)]
42. Tengler, K.; Sabitzer, B.; Rottenhofer, M. "Fairy tale computer science"—Creative approaches for early computer science in primary schools. In Proceedings of the 12th Annual International Conference of Education, Research and Innovation, Sevilla, Spain, 11–13 November 2019.
43. Tengler, K.; Sabitzer, B.; Kastner-Hauler, O. First Programming with Ozobots—A Creative Approach to Early Computer Science Primary Education. In Proceedings of the INTED2020 Conference, Valencia, Spain, 2–4 March 2020; pp. 5156–5162. [[CrossRef](#)]
44. Plomp, T.; Nieveen, N. An introduction to educational design research. In Proceedings of the Seminar Conducted at the East China Normal University, Shanghai, China, 23–26 November 2007; Volume 23.
45. Hußmann, S.; Thiele, J.; Hinz, R.; Prediger, S.; Ralle, B. Gegenstandsorientierte Unterrichtsdesigns entwickeln und erforschen. In *M Komorek Predig. Hg Lange Weg zum Unterrichtsdesign zur Begründung Umsetzung Fachdidaktischer Forsch.—Entwicklungsprogramme Müntst*; Waxmann: New York, NY, USA, 2013; pp. 25–42.
46. Plomp, T.; Nieveen, N. Educational design research part A: An introduction. *Educ. Des. Res.* **2013**, 11–50.