ACTA DIDACTICA NAPOCENSIA

Volume 16, Number 1, 2023 - DOI: 10.24193/adn.16.1.3

DEVELOPMENT AND PRELIMINARY TESTING OF THE ALGOPAINT UNPLUGGED COMPUTATIONAL THINKING ASSESSMENT FOR PRESCHOOL EDUCATION

Iuliana ZSOLDOS-MARCHIȘ and Éva BÁLINT-SVELLA

Abstract. The concept, development and assessment of computational thinking have increasingly become the focus of research in recent years. Most of this type of research focuses on older children or adults. Preschool age is a sensitive period when many skills develop intensively, so the development of computational thinking skills can already begin at this age.

The increased interest in this field requires the development of appropriate assessments. Currently, there are only a limited number of computational thinking assessments for preschool children. Based on this shortcoming, an assessment tool, named AlgoPaint Unplugged Computational Thinking Assessment for Preschool, was created addressed for 4-7 years old children. It is a paper-pencil-based test, which examines the following computational thinking domains: algorithms and debugging. Regarding computational concepts, simple instructions, simple and nested loops, and conditionals are included in the test. For the preliminary testing, AlgoPaint test was applied by 11 preschool teachers with 56 preschool age children. The test was also evaluated by 6 experts in algorithmic thinking working at universities. Based on the feedback given by the teachers and the experts, and the results of the children, AlgoPaint Computational Thinking Test was revised and completed. The revised version of the test is included in the appendix of the paper.

Keywords: computational thinking, preschool, AlgoPaint Computational Thinking Test

Introduction

Today's children are born into a world in which smart devices and technology are part of their everyday lives. That is why it is no longer surprising that computational thinking is being investigated by more and more researchers. Although interest in this area has increased, there is not enough research on how to teach and assess computational thinking (Rich et al, 2018), especially for young children (Curumisu, Adams & Lu, 2019). Computational thinking (CT) is a complex competency, not only focusing on the use of a computer. CT requires "taking an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing" (Wing, 2008, 3717.). As observed in this definition, developing CT doesn't necessarily require the use of a smart device, computer, it is a competence which can be used in many areas of life and in many careers. For preschool age children Bers (2018, 70.) defined CT as the ability to abstract computational behaviors and identify bugs.

The increased interest in this field highlights the need to develop appropriate assessment tools for all age groups. This is a big challenge, but in recent years there have been more and more attempts to do so. In the last two decades, many assessment tools have been developed to measure computational thinking, but only a few of them focused on young (3-7 years old) children (El-Hamamsy et al., 2022; Marinus et al., 2018; Relkin, 2018; Relkin et al., 2020 Relkin & Bers, 2021; Zapata-Cáceres et al., 2020; Zhang & Wong, 2023). Two of these tests are using robots for assessment (Marinus et al., 2018; Relkin, 2018). The tasks with robots require some knowledge about handling and coding the specific robot. The other two tests are focusing on six domains of computational thinking (Bebras Challenge,

Received June 2023.

Cite as: Zsoldos-Marchiș, I. & Bálint-Svella, É. (2023). Development and preliminary testing of the AlgoPaint Unplugged Computational Thinking Assessment for Preschool Education. *Acta Didactica Napocensia*, 16(1), 32-50, https://doi.org/10.24193/adn.16.1.3

Relkin & Bers, 2021), from which algorithms are one domain. Regarding the algorithm domain, these tests are using simple algorithms, without if or loop. The other three tests are focusing on the algorithm domain, assessing different control structures, as if, repeat with a given number of times, and repeat while a condition is true (El-Hamamsy et al., 2022; Zapata-Cáceres et al., 2020; Zhang & Wong, 2023). These tests require some spatial abilities, as the child must follow the path on a square grid.

This paper presents the *AlgoPaint Unplugged Computational Thinking Assessment for Preschool* which is a paper-pencil based CT test for 4-7 years old children which emphasizes the algorithm domain of CT. The test was designed so that coding or special spatial abilities are not required.

The paper presents those ideas on which the AlgoPaint test design is based and the pilot testing of it. AlgoPaint was tested by 11 preschool teachers, by applying it to some of their pupils. In total 56 children solved the test. The test was also sent to experts in the field working at university level. Based on the feedback given by the experts and the preschool teachers, and the results of the children, AlgoPaint Computational Thinking Test was revised and completed.

2. Theoretical background

2.1. Computational thinking

The foundation of the computational thinking is algorithmic thinking, which was first defined by Papert (1980) as "the art of deliberately thinking like a computer, according, for example, to the stereotype of a computer program that proceeds in a step-by-step, literal, mechanical fashion' (p. 27). The concept of computational thinking was popularized by Wing (2006). According to Wing, computational thinking has two fundamental aspects: creation of abstractions and implementation of abstractions. She defined the computational thinking as problem solving, designing systems, and understanding behaviors by drawing upon the concepts of Computer Science (Wing, 2006)

Although there is no universally accepted definition of the concept, Grover and Pea (2013) concluded that researchers should accept that computational thinking is a thinking process that uses elements of generalization, abstraction, decomposition, algorithmic thinking and debugging (detection of errors and correction of errors).

Selby and Woollard (2014) followed and examined the development and change of Wing's definition of computational thinking in literature. According to their recommendation, computational thinking is a cognitive or thinking process that reflects the ability to think in abstractions, the ability to think in terms of decomposition, the ability to think algorithmically, the ability to think in terms of evaluations and the ability to think in generalizations. This proposed definition attempts to incorporate only those terms for which there is a consensus in the literature.

According to Bers et al. (2019), computational thinking is the ability to use computer concepts to formulate and solve problems. It includes a wide range of abilities: abstraction, algorithm, resolution, error correction, generalization. It can be interpreted as being directly related to digital competences as its component. Computational thinking represents a type of analytical thinking that has many similarities with mathematical thinking (e.g., problem solving), scientific thinking (e.g., systematic analysis) and engineering thinking (e.g., designing processes).

More and more researchers argue that computational thinking is not only necessary for those interested in computer science and mathematics. These researchers believe that the nature of computational thinking is multi-theoretical and more generally corresponds to an example of thinking models (Li et al. 2020).

2.2. Assessing computational thinking of young children

In this section the available CT tests addressed to the age group targeted by AlgoPaint are presented.

The first two tests presented rely on the use of a robot. Marinus et al. (2018) developed the *Coding Development (CODE) test* for 3-6 years old children. The tasks in the test have to be solved using the

Cubetto robot. The items require building and debugging simple algorithms. The child has to plan a path on a square grid field and code the robot to follow this path. Thus, the solutions of the tasks require coding knowledge. Another test which is using a robot for solving the task, is *Tufts Assessment of Computational Thinking in Children - KIBO (TACTIC - KIBO)*, which was developed by Relkin (2018). This test is addressed to 5-7 years old children and uses the KIBO robot. The CT domains included in the test are control structure, hardware, software, representation, algorithm/modularity, debugging, and design process. The items require the competence of designing algorithms, and beside simple algorithms conditional structures and loops are included. Another test which partially relies on robot coding is the *Early Childhood Coding Skills Assessment Test* (Kalyenci et al., 2022) addressed to 5-7 years old children. In this test the computational thinking domains assessed are signs, sorting, debugging, loops, modularity, algorithm, program development. The test has a frame story, the child has to help a monkey to get to a given cell on a square grid. The test has two parts, the first part doesn't require coding a robot, only the second part.

In the above presented three tests no choices are given, in case of each task the satisfactory and unsatisfactory solutions are described.

The following tests do not require previous coding experience and the use of digital devices. In all tests presented below in case of each item choices are given from which the child selects the correct one.

The *Bebras Unplugged Computational Thinking Cards* are developed by the Bebras Challenge (www. bebras.org) international educational community for promoting Informatics and CT among school children. The tests are developed for 3-10 years old children, separate test sets for 3-4, 5-6, 7-8, and 9-10 age groups. The CT domains addressed by these tests are decomposition, pattern recognition, abstraction, modeling and simulation, algorithms, and evaluation. Bebras Cards are mainly used as an annual challenge, but there is also some validation on these cards as a CT test (Sung, 2022). Another test which addresses similar domains of CT is the *TechCheck Computational Thinking (CT) assessment* (Relkin et. al., 2020) for 5-9 years old children. This test assesses the following domains of CT: algorithm, modularity, debugging, control structures, hardware/software, representation. TechCheck has a simplified version for 5-6 years old children, called *TechCheck-K* (Relkin & Bers, 2021).

In the following tests the child has to work on a square grid (maze), she/he has to find a path in the maze in order to solve the problem. The items are multiple choice types: the child has to choose the right algorithm for solving the particular problem from four possibilities. The *Beginners Computational Thinking Test (BCTt)* was developed by Zapata-Cáceres et al. (2020) for 5-12 years old children. The computational concepts included in the test are sequences, loops (simple and nested loops), and conditionals (if-then, if-then-else, while). In the test the child has to lead a chicken to his mother, finding the right path on a square grid. The left/right instructions mean going to left/going to right in the grid, regardless of where the chicken is looking. Another test using similar ideas is the *Computational Thinking Test for Lower Primary (CTtLP)* developed by Zang and Wong (2023), which is addressed to children aged 6-10. The computational concepts included in the test are sequences, directions, loops, and conditionals (if-then). In this test there are two types of tasks: drawing with a pencil and leading a character on a square grid. In the case of drawing with a pencil, the left/right instruction means going to left/going to right on the paper. In case when the child has to lead a character on a square grid, the character was adapted from Román-González (2015) and the maze from Zapata-Cáceres et al. (2020). In this case the go to left/go to right instructions are replaced by turn left/turn right instructions, and the character goes forward in the direction where his open mouth is directed, so that turning is a separate instruction, and does not include progress on the path. Thus, these tasks on a square grid are different from those of BCTt test (Zapata-Cáceres et al., 2020), where going left/going right was regarding the maze and not the orientation of the character. In the task where the orientation of the character is taken into consideration, mental rotation skills are required for a successful solution, as the turn left/turn right depends on the orientation of the character.

3. Developing the AlgoPaint Unplugged Computational Thinking Assessment for Preschool

The AlgoPaint computational thinking test was developed in the frame of a wider research about experimenting the effect of activities with tangible robots on computational thinking of preschool children. The test design process was based on the following ideas:

- The test is suitable for preschool children (age 4-7).
- The test is screen-free. A screen-free test-design was chosen as preschool children's screen-time should be limited.
- The test is unplugged, it doesn't use a robot. In a test using robots for solving tasks children who already have prior experiences with robots do have an advantage. Also, if the test is used as a pre- and posttest for an intervention with educational robots, then the experimental group could increase its results on the posttest only due to the experience with programming robots.
- The tasks do not require prior programming knowledge.
- The tasks do not include directions. Handling the tasks with directions could require spatial thinking skills, especially mental rotation. Children with not so developed spatial thinking skills could have disadvantages when solving the test.

The test has a frame-story to raise children's interest and keep their motivation. In the story children are asked to help the painting elf to color different figures made from geometric shapes. The instructions are presented on cards. These cards are similar to those designed for creative drawing based on algorithmic thinking in the frame of the Erasmus+ project AlgoLittle.

Based on the frame-story, the Painter Elf draws the outline of a drawing consisting of geometric shapes, then colors these geometrical shapes with paint. When painting the geometrical forms from a figure, the Elf started from one edge, painting the shape on which he is standing on, and then painting a shape next to it that is in contact with it. He moves from one shape to another, painting each one in his way. He cannot go back to an already painted shape, otherwise he gets stuck in the paint. The rule of going from a geometrical form to one which is adjoined with that form ensures an order in painting, but not necessarily a unique order. In Figure 1 such a figure can be observed, in which the order in which the geometrical shapes are painted is not unique: after painting blue the circle any of the three triangles could be painted green. The selected order is influenced sometimes by the continuation of the algorithm. In the example from Figure 2 it is clear which triangle to color, as the algorithm continues and if any of the other two triangles are colored, the painting can't be continued, as the Elf can't step back to an already painted geometrical figure. The rule of not stepping back to an already painted geometrical form assures that each simple instruction refers to the current geometrical form where the Elf steps, so to each geometrical form a color is associated.

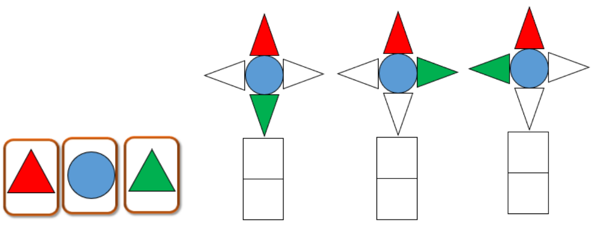


Figure 1. Example of figure in which a given sequence of simple instructions (in the left) could lead to three different possibilities of coloring (in the right)

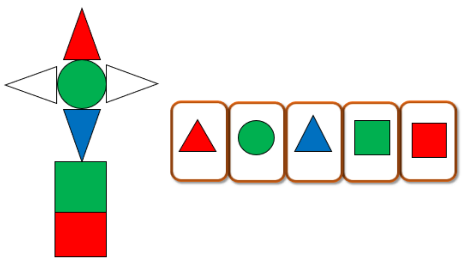


Figure 2. Example of painted figure and corresponding sequence of simple instructions

Some of the problems have more than one solution, see in the example given in Figure 3. Based on the sequence of cards given, the robot can be painted in two ways. Children are required to give only one solution, but this item could be a good example for checking if children think about multiple solutions when solving the problem.

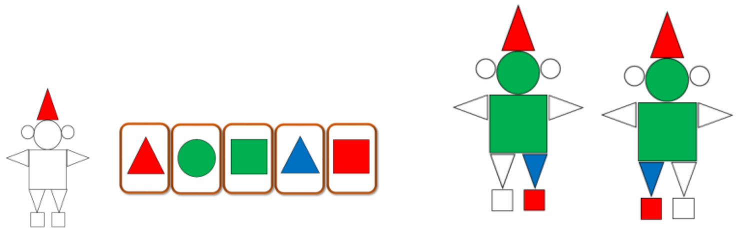
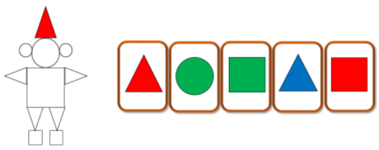
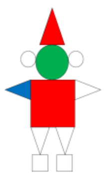
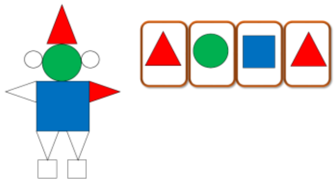


Figure 3. Example of item with multiple solutions

The test measures the **computational thinking abilities** described in Table 1.

Table 1. Computational thinking abilities measured by AlgoPaint Unplugged Computational Thinking Assessment

Computational thinking ability	Description	Example of item
Applying an algorithm	ability to identify the output of a given sequence of instructions	Item 1. Help Painter Elf, paint the robot. He laid out the instructions for you, follow them. I wonder if it is possible to paint the robot based on the cards. 
Designing an algorithm	ability to identify sequences of instructions to obtain a given output	Item 6. Help Painter Elf lay out the instructions! Paint Elf painted the robot and sent you the picture cards with the instructions. Your task is to lay out the cards in order, as Painter Elf painted. Next to it, arrange the cards one after the other in a row. I wonder if it is possible to lay out the cards according to the drawing. 
Debugging an algorithm	ability to compare a given sequence of instructions with the given output on it	Item 4. Painter Elf has already painted the robot. He posted the instructions; all you have to do is check whether he has painted the same as what is laid out on the cards. Did he lay out the cards correctly? 

The test used only simple instructions.

Analyzing all the items regarding the computational thinking ability addressed and which item contained error, the results are presented in Table 2. It is important to note that in the case of all the items pupils were asked to search for errors, so not only the debugging items required to identify errors.

Table 2. Analysis of the items in version 2 of the AlgoPaint Unplugged Computational Thinking Assessment

Item	Computational Thinking Abilities	Inclusion of errors in the task
1	Applying	no
2	Applying	yes
3	Applying	yes
4	Debugging	no
5	Debugging	yes
6	Designing	no
7	Designing	yes

The test is filled in individually by children. The testbook contains instructions for the teacher to ensure that the test is used according to its design. The testbook contains three solved examples: two with simple instructions and one with simple loops. The testbook also contains the correct solutions, a scoring grid, and the cards necessary for solving the items.

4. Methodology

The research was conducted in the 2021-2022 school year.

4.1. Objectives of the research

The research presented in this paper had four main objectives:

1. Developing version 1 of the AlgoPaint Unplugged Computational Thinking Assessment for 4-7 years old children.
2. Testing version 1 of the AlgoPaint Unplugged Computational Thinking Assessment with preschool children (age 4-6).
3. Mapping expert opinions on version 1 of the AlgoPaint Unplugged Computational Thinking Assessment.
4. Developing version 2 of the AlgoPaint Unplugged Computational Thinking Assessment.

4.2. Instruments

4.2.1. AlgoPaint Unplugged Computational Thinking Assessment and evaluation grid

The AlgoPaint Computational Thinking Test presented in Section 3 was used. The testbook, besides the solved problems and tasks required to be solved by the children, also contained an evaluation grid. For example, in case if Item 1 the following possibilities are given in the evaluation grid: colors correctly; colors in the right order but doesn’t take in consideration the rule how the Painting Elf choose the next shape; colors correctly, but don’t take into consideration, how many shapes have to be coloured (colors both legs of the robot). For each task there was the possibility for the tester preschool teacher to give her own answer, in this way collection of other errors was facilitated.

4.2.2. Questionnaire for experts

A questionnaire for collecting experts' opinions was developed. It was an online, anonymous questionnaire, which contained 9 questions, from which 5 formulated as closed items (multiple choice and scale) and 4 as open questions. The aim of this survey was to collect information about the appropriateness of the test items to the targeted age group and the appropriateness of the test to measure computational thinking. There were also questions regarding the clearness of the explanations in the testbook. See the questionnaire in Appendix 1.

4.3. Participants

11 kindergarten teachers from Romania used the test in their group, assessing a total of 56 children. The teachers were selected from the students of a master level course on STEM education. The participating children were 5-6-year-old preschoolers with no coding experience. In addition, the test was sent to university level teachers who are experts in this topic. These experts were selected from 6 different countries. They were asked to evaluate the test and fill in a questionnaire. 6 experts filled in the anonymous questionnaire.

5. Results

5.1. Pupils' results on the AlgoPaint Unplugged Computational Thinking Assessment

In the case of each task the number and the percentage of pupils solving it correctly is presented in Table 3.

Table 3. Pupils' results on the AlgoPaint Unplugged Computational Thinking Assessment

Item	Computational thinking ability	Inclusion of errors in the task	Number of pupils solved it correctly	Percentage of pupils solved it correctly	Most frequent error of the pupils	Number of pupils committed this error
1	Applying	no	22	39.29	Colors more shapes than given by the instructions	15
					Doesn't choose the right neighboring shape	12
2	Applying	yes	23	41.07	Doesn't observe the error	32
3	Applying	yes	25	44.64	Doesn't observe the error	28
4	Debugging	no	49	87.50		
5	Debugging	yes	33	58.93	Doesn't observe the error	23
6	Designing	no	44	78.57	Doesn't put the instructions in the right order	10
7	Designing	yes	30	53.57	Doesn't observe the error	22

To see the tasks from which computational thinking ability was the easier for pupils, the average percentage for each one was calculated (Table 4). The applying items were the most difficult for pupils (41.67% of the pupils solved these tasks correctly), then the designing items (66.07%). The easiest were the debugging items (73.22%). The success rate of the applying items could be influenced by the fact that these items were the first in the test, children needed time to be familiarized with the test. Another aspect to be taken into consideration is that some children can't concentrate on more requirements at the same time, thus they color the right geometrical form but don't care about the neighboring condition or the number of shapes to be colored.

Pupils' success rate was higher on tasks which do not contain errors (68.45%) than on tasks containing errors (49.55%). Many pupils didn't observe the error, they just tried to solve the tasks as being correct.

Table 4. Average results per computational thinking abilities respectively per items containing/not containing errors.

Computational thinking ability	Average percentage of pupils solved correctly	Inclusion of errors in the task	Average percentage of pupils solved correctly
Applying	41.67	yes	49.55
Debugging	73.22	no	68.45
Designing	66.07		

5.2. Preschool teachers' opinions about the AlgoPaint Unplugged Computational Thinking Assessment

Preschool teachers' opinions were asked in an open question. This led to the fact that different teachers pointed out different aspects of the test, thus quantitative analysis on the occurrences of different ideas is not indicated due to the small number of the sample (11 teachers). Thus during the analysis of the answers, the main ideas are highlighted and all the recommendations for improvement are discussed.

Most of the teachers consider the test interesting and enjoyable for children. One teacher mentioned that the test was enjoyable even for her. In teachers' opinion, the frame-story makes the test interesting for children, raising their interest in solving it.

> "The instrument is very creative and playful, the children from my group really enjoyed it. They color the shapes with pleasure, they didn't take the tasks as assignments. "

As regards the task difficulty, some of the teachers mentioned that they consider the test difficulty appropriate for 5-6 years old children, others considered it too difficult. One of the teachers considered the test too difficult even if some of the children performed well on it.

> "It is quite difficult for 5-6 years old children, as it is difficult for them to concentrate on more things at the same time, such as the guidance of the teacher and the shapes on the paper."

As regards the instructions given for the teachers conducting the test, they should be developed and described in more detail. For example, the testing instructions didn't explicitly specify taking the test individually with children, thus some of the teachers tried to complete the test with more children at the same time. They observed that it is difficult to explain for more children at the same time and to record children's performance according to the evaluation grid. Another aspect is related to the solved examples. A teacher mentioned that with more explanations, children solved the tasks more correctly. This comment highlighted the necessity to give the exact explanations in case of the solved examples in the testbook. This would be important to ensure the same testing conditions for each child, otherwise the amount or the quality of explanations influences the test results.

Another aspect mentioned by the teachers was about the children's work during the test. One of the teachers mentioned that in the case of the applying items (which required coloring based on the instruction cards) some children's aim was to color all of the robot, thus they didn't stop when all the instructions were followed, and continued coloring. This aspect can be seen in children's results and could be a reason why the applying items had the lowest success rate (Table 4). Teachers also mentioned that there are children who can't concentrate on more aspects at the same time, thus, for example, they follow the instructions, color the right geometrical shapes with the right color, but don't take into consideration the neighboring condition. Also, for some children it was difficult to understand how Painting Elf choose the next shape, so the notion of "neighboring shape" and the rule of "not stepping back to a colored shape" should be well explained in the solved examples. Teachers also suggested that the frame/story should contain the idea that the Painting Elf makes mistakes, so that the possibility of errors is included in each task.

5.3. Experts' opinions about the AlgoPaint Unplugged Computational Thinking Assessment

Most of the experts consider that the test is suitable for the 5-6 years old age group, the mean for this question is 4.67, measured on a scale from 1 (not at all) to 5 (totally). All the experts agree that the number of tasks is appropriate for testing the targeted abilities. Regarding the suitability for measuring the algorithmic thinking abilities for which the test was designed, respondents were asked to measure on a scale from 1 (not at all) to 5 (totally) to which extent the test measures the applying, debugging, and designing algorithms abilities. The averages obtained for these are 4.83 for applying algorithms, 4.33 for debugging algorithms, and 4 for designing algorithms. One of the experts mentioned that in his opinion there is no significant difference between the applying and designing tasks in terms of required abilities.

As regards the items, one of the experts recommended using other drawings besides the robot drawing used in all the 7 items. Another interesting idea is that not to give from which shape the coloring starts, this would be interesting especially for debugging tasks, i.e., the given algorithm is correct starting from one shape and incorrect starting from another one.

One of the experts mentioned that the solutions are not deterministic, some tasks have more solutions and there is a question if the test conducting teacher is aware of all the solutions. But as the solutions are given in the testbook it could not happen that the teacher marks as incorrect a correct solution given by the child.

Most of the experts think that the test is understandable for children. The only problem which can appear is about understanding the Painting Elf moves. As mentioned also by the preschool teachers, some of the experts think that it could be difficult to always correctly choose the next shape which has to be colored, the notion of neighbor should be better explained for children. The neighboring shape could be that shape which is in contact with the current shape. One of the experts thinks that the different orientations of the shapes from the drawings and the instruction card could raise difficulty for children. This problem wasn't mentioned by any preschool teacher and didn't reflect in the solutions given by the children.

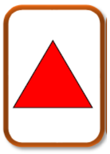
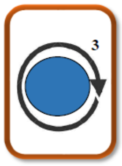
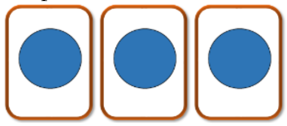
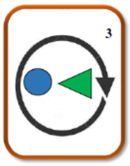
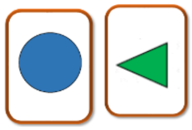
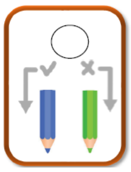
6. Revising the AlgoPaint Unplugged Computational Thinking Assessment

The test was revised based on the results obtained in the preliminary testing. The test was improved based on the following ideas highlighted from the preliminary testing results:

- Some more test items need to be designed to assess more computational concepts.
- In the new items other drawings should be included to have more types of drawings in the test.
- The instructions for the test conducting teacher should be more detailed.
- The frame-story addressed to the children should explain what neighboring shapes mean and also should contain the idea that the Painting Elf could make mistakes.
- The solved items should contain the exact words for explanations to ensure the same test conditions for each child.
- The evaluation grid for the children's work should be developed in more detail.

Five more items were added to include more computational thinking concepts in the test. In the first version of the test a solved problem with simple loops was presented but no task was given for the children with this computational concept. In the revised version of the test two tasks with simple loops and one task with nested loops were included. Another computational thinking concept included in the revised version of the test is the conditional. One solved item and two items with conditionals were included in the test. The **computational concepts** used in the test are presented in Table 5.

Table 5. Computational concepts measured by AlgoPaint Unplugged Computational Thinking Assessment

Computational concept	Description	Example of card	Explanation of the card
Simple instruction	painting a geometrical form with a given color		Paint the triangle with red color
Simple loop	a structure that allows a simple instruction to be repeated multiple times		It is the same as the following sequence of simple instructions: 
Nested loop	a structure that allows to a sequence of simple instruction to be repeated multiple times		In this case the following sequence of simple instructions are repeated three times: 
Conditional	a structure that allows to perform a simple instruction based on a condition		The condition from the top part of the card is tested for the : Is the shape a circle? If yes, then it is coloured blue, if not, it is coloured green.

As one of the experts recommended using other drawings beside the robot used in the 7 items from the preliminary testing, in the new items other drawings are used (see Items 8-12 in Appendix 2).

Analyzing all the items regarding the computational thinking ability addressed and the computation concept included, the results are presented in Table 6. It is important to note that in the case of all the items pupils were asked to search for errors, so not only the debugging items required to identify errors.

Table 6. Analysis of the items in version 2 of the AlgoPaint Unplugged Computational Thinking Assessment

Item	**Computational Thinking Abilities**	**Computational Concept**	**Inclusion of errors in the task**
1	Applying	Simple instruction	no
2	Applying	Simple instruction	yes
3	Applying	Simple instruction	yes
4	Debugging	Simple instruction	no

5	Debugging	Simple instruction	yes
6	Designing	Simple instruction	no
7	Designing	Simple instruction	yes
8	Applying	Simple loop	no
9	Applying	Nested loop	no
10	Debugging	Simple loop	yes
11	Applying	Conditional	no
12	Debugging	Conditional	no

The instructions for the teacher conducting the test were revised, more details are added to ensure same testing conditions for each child. Another solved example is added for the conditionals. Thus, the revised version of the testbook contains four solved examples: two with simple instructions, one with simple loops, and one with conditionals. The solved examples are described in detail as the teacher conducting the test only needs to read the text from the testbook. This ensures that the explanations are correctly given and the amount of explanations are the same for each child. The instruction part of the testbook highlights the fact that teachers need not give any additional help to the child and some examples of neutral responses are given for the child's possible questions. The testbook also contains the correct solutions and the cards necessary for solving the items. The scoring grid was revised, it is more detailed, and points are attributed to children's work. For example, the evaluation grid for Item 1 is presented in Figure 4.

The child solves the task correctly. 3 p

The child colors correctly, but colors both legs (follows the cards, chooses the shapes correctly, only at the end, he/she colors both legs) 2 p

The child observes the order of the cards, colors in the correct order, taking into account the adjacency clause, but does not pay attention to the fact that the Elf cannot go back to a shape that has already been painted. 2 p

Follows the order of the cards and finds a shape for it, colors them (the shapes do not follow the contiguity when selecting, jumps to the shapes, but understood that a card represents a shape) 1 p

The child doesn't follow the order of the cards, but she/he understood that a card represents a shape. 1 p

The child colors at random, she/he doesn't understand what she/he has to do. 0 p

Figure 4. Evaluation grid for Item 1

7. Conclusions

The preliminary testing of the AlgoPaint Unplugged Computational Thinking Assessment for preschool children shows that the test is suitable for the targeted age-group and appropriate to measure the algorithmic thinking abilities as applying, debugging, and designing algorithms. The results obtained by children also show that the test is suitable for their level of competences. The opinion of the preschool teachers conducting the testing and the university teaching staff expert in developing algorithmic thinking gave valuable ideas to improve the test. Based on the results an improved version of the AlgoPaint Unplugged Computational Thinking Assessment was developed.

Further work is in progress, such as applying the improved version of the test to children and analyzing test validity.

References

Algolittle. Algorithmic Thinking Skills Through Play-Based Learning for Future's Code Literates. Erasmus+ project, 2020-1-TR01-KA203-092333.

Bers, M. U. (2018). Coding as a playground: programming and computational thinking in the early childhood classroom. Routledge. https://doi.org/10.4324/9781315398945Cutumisu, M., Adams, C. & Lu, C. (2019). A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. Journal of Science Education and Technology 28(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

Bers M. U, González-González C., & Armas–Torres, M. B. (2019). Coding as a playground:Promoting positive learning experiences in childhood classrooms, *Computers & Education*, 138, 130–145, https://doi.org/10.1016/j.compedu.2019.04.013

El-Hamamsy, L., Zapata-Cáceres, M., Martin Barroso, E., Mondada, F., Dehler Zufferey J., & Bruno, B. (2022). The competent Computational Thinking test (cCTt): Development and validation of an unplugged Computational Thinking test for upper primary school, *Journal of Educational Computing Research*, 60 (7), https://doi.org/10.1177/07356331221081753

Grover, S. & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38–43.

Kalyenci, D., Metin, Ş. & Başaran, M. (2022). Test for assessing coding skills in early childhood. *International Journal of Technology and Design Education,* 27, 4685–4708. https://doi.org/10.1007/s10639-021-10803-w

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational Thinking Is More about Thinking than Computing. Journal for STEM Educ Res, 3(1):1–18.

Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). Unravelling the cognition of coding in 3-to-6-year olds: the development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER'18, 133–141.https://doi.org/10.1145/3230977.3230984.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.

Relkin, E. (2018). Assessing young children's computational thinking abilities (aster's thesis). Retrieved from ProQuest Dissertations and Theses database. (UMI No. 10813994).

Relkin, E., de Ruiter, L. & Bers, M.U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. Journal of Science Education and Technology, 29, 482–498. https://doi.org/10.1007/s10956-020-09831-x

Relkin, E., & Bers, M. (2021). Techcheck-k: A measure of computational thinking for kindergarten children. In 2021 IEEE Global Engineering Education Conference (EDUCON), pages 1696–1702.

Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2018). Coding in K-8: International trends in teaching elementary/primary computing. Washington: Springer Science & Business Media.

Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. Proceedings of EDULEARN15 Conference, 2436-2444.

Selby, C., & Woollard, J., (2014). Computational Thinking: The Developing Definition, SIGCSE.

Sung, J. (2022). Assessing young Korean children's computational thinking: A validation study of two measurements. *Education and Information Technologies,* 27, 12969–12997. https://doi.org/10.1007/s10639-022-11137-x

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational Thinking and Thinking About Computing. Philosophical Transactions of The Royal Society of London A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118.

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. In 2020 IEEE Global Engineering Education Conference (EDUCON), pages 1905–1914. IEEE

Zhang, S., & Wong, G.K.W. (2023). Development and validation of a computational thinking test for lower primary school students. *Educational technology research and development*. https://doi.org/10.1007/s11423-023-10231-2.

Authors

ZSOLDOS-MARCHIȘ Iuliana, Babeș-Bolyai University, Cluj-Napoca (Romania). E-mail: iuliana.marchis@ubbcluj.ro

BÁLINT-SVELLA Éva, PhD student, Babeș-Bolyai University, Cluj-Napoca (Romania). E-mail: eva.svella@ubbcluj.ro

Appendix 1. Questionnaire for experts regarding AlgoPaint Unplugged Computational Thinking Assessment for Preschool (version 1)

1. Is the test appropriate for 5-6 years old children?

1 not at all	2	3	4	5 totally
◯	◯	◯	◯	◯

2. Is the number of tasks adequate?

◯ too few tasks

◯ an appropriate number of tasks

◯ too many tasks

3. How clearly are the tasks formulated?

◯ Even the context of the test is difficult to understand (how Painter Elf paints).

◯ Some tasks are difficult to understand.

◯ The context of the test and the tasks are also understandable.

4. If the text of some tasks is difficult to understand, copy the text part that is difficult to understand here.

5. The extent to which the test questions assess the following abilities:

Ability	1 not at all	2	3	4	5 totally
following algorithm steps	◯	◯	◯	◯	◯
debugging	◯	◯	◯	◯	◯
creating an algorithm	◯	◯	◯	◯	◯

6. If there is a task that you do not find suitable, please enter here which task it is and why it is not suitable for assessing the algorithmic thinking of the target group.

7. How complete is the description of how to complete the test?

◯ is incomplete, based on this description, the children will not complete the test under the same conditions

◯ it is somewhat incomplete, but it ensures that the children take the test under the same conditions

◯ appropriate

8. If you find the description of how to complete the test incomplete, what information is missing?

9. If you have any other comments or suggestions for corrections/additions regarding the test, please write them here.

Appendix 2. The AlgoPaint Unplugged Computational Thinking Assessment for Preschool (version 2, with corrections and completions after the preliminary testing)

Instructions for the test conducting teacher:

1. The preschool teacher prepares the tools, separately for each task with the appropriate picture cards, three colored pencils (red, green, blue) and a writing instrument for filling out the observation sheet.

2. The children complete the test individually, preferably in a quiet place, under calm conditions.

3. We present the first two tasks to the child, slowly, step by step, checking that the child has understood the tasks.

4. After presenting the two tasks, we ask the child if she/he would like to help the Painter Elf with the big job. If the child says yes, then we ask her/him to be very attentive, because Painter Elf sometimes mixes up the colors, shapes or even the order in his haste. The Painter Elf can be wrong.

5. Before starting the tasks, make sure that the observation sheet is prepared, where you record the observations about each child.

6. Let the child work alone while solving the tasks.

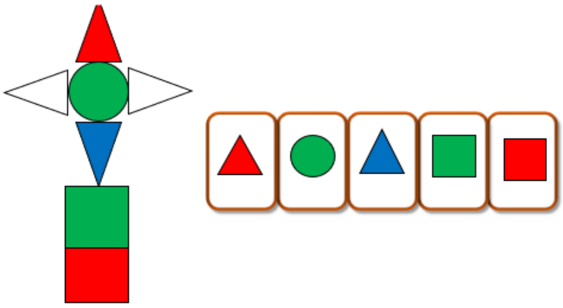
7. Do not help the child to solve the tasks, if she/he does not succeed, you write it down on the observation sheet as a comment.

8. You should store each child's work and observation sheet in a separate foil.

9. If the child gets stuck and asks, you try to encourage him by giving a neutral answer. It is important not to discourage further work. We should rather ask back if she/he asks: "What do you think?", "How do you think you could solve it?", "Why do you want it that way?". We can give feedback with the following sentences: "You will surely figure out how to solve it.", "The way you want it," "You are working very skillfully, you are making good progress," "You are already done, that's right!"

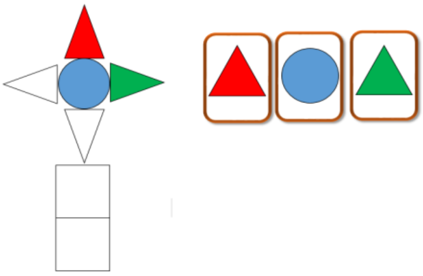
AlgoPaint frame-story, *to be read to the child*: “Once upon a time, there lived a very hard-working Elf whose favorite hobby was painting. His name was Painter Elf. He made everything colorful in his empire. He made the outline of drawings consisting of geometric shapes on the ground, then colored them. When painting the geometrical forms from a figure, the Elf started from one edge, painting the shape on which he is standing on, and then painting a shape next to it that is in contact with it. He moves from one shape to another, painting each one in his way. Unfortunately, in his haste, he bought paint that does not dry immediately, so he cannot go back to an already painted shape, otherwise he gets stuck in the paint.”

Solved item 1.



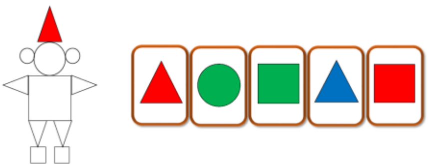
“Look, this is how his first robot turned out. Let's see and follow his work together. To make it easier for us to follow, Painter Elf laid out the instructions on small cards. Each card shows the coloring of a shape. He painted his hat first. You see, this red triangle is the first card, he started with it *(while pointing with our finger, we match it),* then he moved to the adjacent shape, the one that is in contact with the red triangle: the circle. This he painted green. Here the card shows that a green circle is coming *(pointing with our finger to the card with the green circle).* Then he painted the neck, which is in an adjacent shape to the green circle. Why do you think he didn't paint his ears? *(Because you wouldn't be able to step back onto the head afterwards, it would get stuck in the paint).* He then moved on to the next shape, painting the square green and using red last. Do you see? Here the last card shows that the last square is painted red.”

Solved item 2.

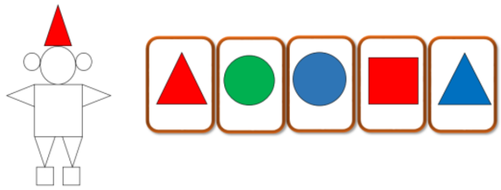
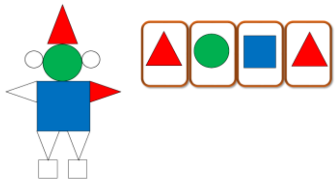
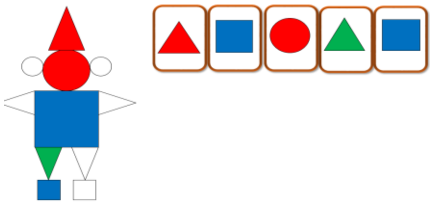
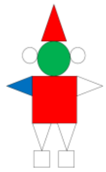
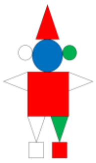


"This is how the second robot succeeded. The Painting Elf started painting from his hat, as always do . Do you see? This is shown by the first card which is the red triangle. Moving to the second card, which is a blue circle, we can check that he next painted the head, the circle into blue, that comes in contact with the red triangle. Going to the next card, which is a green triangle, we can check on the robot that the elf painted green one of the ears of the robot, which is a triangle that comes in contact with the blue circle. But then he realized that he can’t paint any more, as he can’t go back to the head, the blue circle, because the paint is still wet, and he would get stuck in the paint. So, he couldn't paint any more parts of the robot."

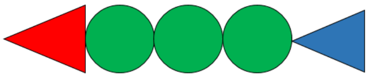
Item 1. Help Painter Elf, paint the robot. He laid out the instructions for you, follow them. I wonder if it is possible to paint the robot based on the cards.



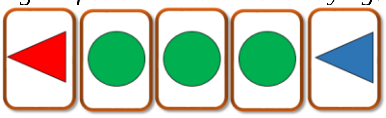
Item 2. Help Painter Elf, paint the robot. He laid out the instructions for you, follow them. I wonder if it is possible to paint the robot based on the cards.

Item 3. Help Painter Elf, paint the robot. He laid out the instructions for you, follow them. I wonder if it is possible to paint the robot based on the cards. 	**Item 4.** Painter Elf has already painted the robot. He posted the instructions; all you have to do is check whether he has painted the same as what is laid out on the cards. Did he lay out the cards correctly? 
Item 5. Painter Elf has already painted the robot. He posted the instructions; all you have to do is check whether he has painted the same as what is laid out on the cards. Did he lay out the cards correctly? 	**Item 6.** Help Painter Elf lay out the instructions! Paint Elf painted the robot and sent you the picture cards with the instructions. Your task is to lay out the cards in order, as Painter Elf painted. Next to it, arrange the cards one after the other in a row. I wonder if it is possible to lay out the cards according to the drawing. 
Item 7. Help Painter Elf lay out the instructions! Paint Elf painted the robot and sent you the picture cards with the instructions. Your task is to lay out the cards in order, as Painter Elf painted. Next to it, arrange the cards one after the other in a row. I wonder if it is possible to lay out the cards according to the drawing. 	

Solved item 3. *Painter Elf really got into painting, he got a little bored with the robots, he decided to paint something else on the ground: he likes colorful worms. He doesn't want to spoil it, so he goes through the shapes one by one, painting the neighbors one after the other, i.e., the shapes that touch each other.*

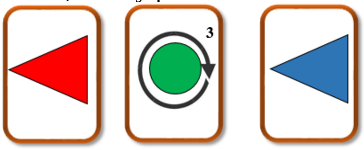


This is how we can lay out the steps with cards (*while pointing with our fingers to the cards and their corresponding shapes on the worm and saying: the first one is the red*

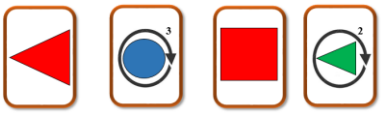


triangle...).

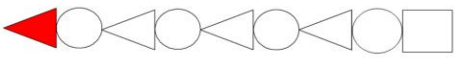
As he laid out the cards, he realized that many of the same cards were placed next to each other, he does not like to use so many of the same. So, he came up with an interesting card (you see, it's the middle one) with an arrow on it: it means to repeat what's in it, and the little number up here shows how many times it has to be repeated. On this card, it is shown to repeat the green circle three times (while pointing with our finger: red triangle card, red head painted, then green circle 1x, 2x, 3x, our finger goes through the three circles, finally paints the blue triangle).

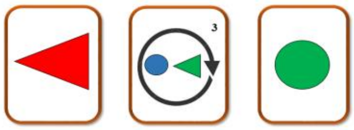


Item 8. Color the worm according to the instructions

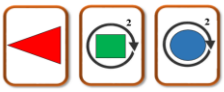


Item 9. Color the worm according to the instructions

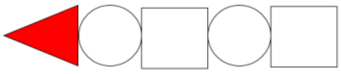




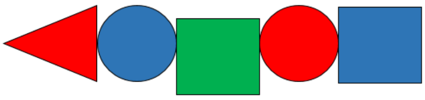
Item 10. Painter Elf planned in advance what color she wanted to paint the worm and laid out the instructions. Check that he is painted according to the instructions.



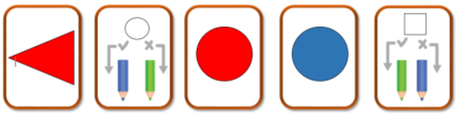
Solved item 4. *Painter Elf was very tired from the big job, so much so that he fell asleep while painting. When he woke up, he found a drawing and a strange card on the floor. At first, he didn't know how to paint. Then he began:*



This is how he solved it: based on the first card, he painted the triangle red, with the help of the next card he had to decide what color to paint the next shape, which is a circle. The card says: if a circle follows, it must be colored blue, because it is checked, if not a circle follows, then it must be colored green. Then he painted a green square and then a red circle. He has to decide the color of the last shape based on the last card. The card says: if a circle shape follows, it must be colored green, it is checked, if it is not a circle, then blue. Since the last shape is not a circle, he painted the last shape of the worm blue *(we also show the full explanation with our fingers, matching the shapes and cards).*



Item 11. Color the worm according to the instructions, decide which color to use.



Item 12. Painter Elf planned what color she wanted to paint the worm and laid out the instructions. Check whether Paint Elf painted the worm based on the cards.