Design and Evaluation of Computational Thinking Tasks in the <colette/> Project: Experiences Gained from Workshops with Secondary and Grammar School Students in Austria, the Netherlands, and Slovakia

Eva Schmidthaler[1], Sylvia van Borkulo[2], Martin Cápay[3], Bjarnheiður Kristinsdóttir[4], Rebecca S. Stäter[5], Tim Läufer[5], Matthias Ludwig[5], David Hornsby[1], Jakob Skogø[1] and Zsolt Lavicza[1]

[1]*School of Education/STEM Didactics, Johannes Kepler, University, Altenbergerstraße 69, Linz, Austria*
[2]*Freudenthal Institute, Utrecht University, Princetonplein 5, Utrecht, The Netherlands*
[3]*Department of Informatics, Constantine the Philosopher University in Nitra, Tr. A. Hlinku 1, Nitra, Slovakia*
[4]*School of Education, University of Iceland, Stakkahlíð 105, Reykjavík, Iceland*
[5]*Institute of Mathematics and Computer Science Education, Goethe University Frankfurt, Robert-Mayer-Str. 6-8, Frankfurt (Main), Germany*

Keywords: Computational Thinking, Augmented Reality, Block-Based Programming, Mobile Educational Application, mAR, STEM Education.

Abstract: In recent years, numerous applications (apps) for mobile devices have been developed for STEM education, but there is a lack of suitable educational apps that support teachers in promoting computational thinking (CT) in mathematics and computer science (CS) lessons. In this position paper, two types of CT tasks, *Building Cubes* and *Draw-o-Bot*, of the newly developed <colette/> app with augmented reality (AR) function, are described, and preliminary results from four workshops that were held in total with 76 10-18-year-old secondary and grammar school students in Austria (W1), the Netherlands (W2), and Slovakia (W3) are discussed. The tasks and the mobile app itself were created as part of the <colette/>-project, an Erasmus+ project, in which seven institutions from five European countries are involved. Each type of task includes a set of CT tasks related to the block-based programming (BBP) app. In the workshops, we set out to explore how the participating secondary school students solved the CT tasks, whilst using <colette/>. The experiences made in the workshops will be used to inform the further development of the application, and to prepare teacher training to support the successful implementation of <colette/> as an educational tool in schools. The first findings indicate that the participating students react positively to the app, can solve BBP tasks successfully, and create loops to shorten their code. In the future, further task types will be implemented in the app and researched.

1 INTRODUCTION

Recently, curricula in many European countries call for the integration of computational thinking (CT) skills into STEM subjects in compulsory education (Bocconi et al., 2022). STEM teachers and students can find a variety of mobile and web-based educational applications (learning apps) freely available on the Internet or in app stores, especially for mathematics education. However, they will not find many apps that combine mathematical topics and CT, and both the development of such apps and the research on them are still lacking in secondary school (Lv et al., 2022). In the context of this paper, six core CT skills can be identified: 'abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization' (Bocconi et al., 2016, p.7). Each of the introduced task types, currently mainly tasks that implement block-based programming, addresses core CT skills differently or directly (Csizmadia et al., 2015; Bocconi et al., 2016). As a first step, apps with visual block-based programming (BBP) languages are introduced to novice students by their Computer Science (CS) or STEM teachers. BBP tools, based on *Google Blockly* (Blockly, 2022; Blockly Games, 2022), such as *Scratch* (Scratch, 2022), or *Alice*

Schmidthaler, E., van Borkulo, S., Cápay, M., Kristinsdóttir, B., Stäter, R., Läufer, T., Ludwig, M., Hornsby, D., Skogø, J. and Lavicza, Z.
Design and Evaluation of Computational Thinking Tasks in the <colette/> Project: Experiences Gained from Workshops with Secondary and Grammar School Students in Austria, the Netherlands, and Slovakia.
DOI: 10.5220/0011974700003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1*, pages 297-304
ISBN: 978-989-758-641-5; ISSN: 2184-5026

(Alice, 2020), allow novice students from an early age to build things, test, experiment, and tinker with CT topics. Furthermore, they assist in changing the way students are learning and problem-solving (Yamashita et al., 2017; Shih, 2017; Xu et al., 2019). In this paper, two types of tasks from the new visual programming app, <colette/> are introduced (*'**C**omputational Thinking **L**earning **E**nvironment for **T**eachers in **E**urope'*) (Colette-project, 2022; Milicic et. al, 2021). The <colette/> project consists of seven European partner institutions coming from five different countries: Austria, Germany, the Netherlands, Slovakia, and France. The main scope of the project is to implement the 'bring-your-own-device' approach to teach CT in pre-existing school subjects, such as mathematics and CS, and moreover, to train teachers to do so. The project outcomes range from an authoring tool for teachers (the web portal) to a mobile app with augmented reality (AR) function, intended for students to work on the CT tasks. In the mobile app, students can work on the given tasks, see hints, and get their solutions checked automatically; furthermore, they can view their self-coded structures (*Fig.1*). Throughout the employment of mobile devices (e.g., tablets, smartphones) and an AR-marker (*Fig.1*), students can see their coded figure embedded into reality. This feature gives the possibility to interact with the figure, i.e. the students can observe their objects from any perspective. By using BBP, the students can solve mathematical and CS tasks without any text-based codes. The proposed block-based programming language of <colette/> has many advantages, which have already been examined and discussed in several studies. Many factors contribute to making BBP easy, including the natural language description of blocks and drag-and-drop composition interactions (Weintrop & Wilensky, 2015). Furthermore, it is beneficial that the difficulty in understanding and memorizing a particular order of the BBP commands by novice students is decreased. Thus, further (syntax) errors in students' codes are reduced, and the learning curve gets more gradual. Another advantage is that teachers can save time when correcting students' errors (Yamashita et.al., 2017; Shih, 2017; Xu et. al., 2019).

With *Blockly*, students can drag-and-drop programming blocks from a predefined range onto the <colette/> app canvas (checkerboard), and further connect the visual programming blocks with each other. Moreover, they can be modified with input parameters (e.g., coordinates) to adjust the desired programming object (e.g., gate, pyramid, movement of a robot) (Xu et al., 2019; Blockly, 2022; Blockly Games, 2022). According to Lin and Weintrop (2021), many BBP environments have been developed, examined, and published but aren't yet publicly accessible. Within the <colette/> project, three different types of BBP CT tasks are already implemented in the app, and five are in preparation. In this paper workshops and findings with two of the implemented ones, *Building Cubes* and *Draw-o-Bot* are presented.

Figure 1: Student BBP solution of a task (left). To view the result, the student must point their camera toward the marker (center) for viewing the result in AR (right).

1.1 Building Cubes and Draw-o-Bot

Building Cubes encompasses a set of tasks that ask students to build a certain structure in a coordinate system. As previously mentioned, it makes use of BBP (Blockly, 2022; Blockly Games, 2022). This code is used to place unit cubes on a checkerboard using x, y, and z coordinates. When the code is executed, the resulting structure made of unit cubes is shown in AR using the device's camera and a given marker (*Fig.1*). The tasks invite students to work on algorithmic thinking (AT); debug, decompose problems, and think about coding principles, such as making code efficient, testing, and creating general solutions. Simple coding blocks to create one cube at a specific location are provided along with more advanced programming structures, such as loops and conditional statements. The concepts involved in the *Building Cubes* tasks were the coordinate system in three dimensions, and spatial orientation and visualization as part of spatial skills (McGee, 1979).

The second task type *Draw-o-Bot* includes a set of CT tasks that ask students to program a virtual robot to draw a certain pre-described pattern. Like *Building Cubes*, BBP is used to create commands for the robot, therefore the same already mentioned CT skills are targeted. The difference between these two task types is that no AR function is provided. Instead, when the code is executed, the resulting 'command' is shown on the screen of the user's device. The utilization of educational robots (ER) is becoming more common in schools nowadays because ER have the opportunity to encourage the usage of new technologies (Benitti, 2012; González et al., 2019; Pou et al., 2022). Within Draw-o-Bot, the students must program a virtual ER. The following commands can be set: *'set the color (of the pen) up/down', 'move x step(s) forward'*, and *'turn 90 degrees left/right'*. With these commands, the students can program the robot, for example, to draw a street sign pattern on a piece of paper (*Fig.2*). In the future, an addition is planned where an ER will draw the shape of the desired object virtually and drawing angles of the desired size (currently only 90°) will be possible.

2 METHODOLOGY

The <colette/> app is still in its development stage, and only three out of eight planned task types are implemented yet. Therefore, the already implemented task types and designed exercises must be tested, to be able to successfully introduce and implement the app as an educational tool in European schools. In the following, after the purpose of this study has been presented, the individual workshops, their procedure, and the data collection and processing are discussed. Afterward, the findings, broken down by country, are described.

2.1 Research Aim & Experimental Design

In an aim to test *Building Cubes* and *Draw-o-Bot*, and present the experiences gained from the test workshops, four lectures, based on discovery learning methodology (de Jong & van Joolingen, 1998) were held in Austria (W1), the Netherlands (2xW2), and Slovakia (W3). Stratified sampling, with 76 10-18-year-old secondary and grammar school students in total, was used. The students were observed based on the participant observation methodology (Musante & DeWalt, 2010). The instructors documented the students' progress and their final task solutions during and after the observation. At the end of each workshop, the students were asked to answer an app evaluation questionnaire (15min) to evaluate <colette/> regarding their perception of CT and the tasks in W1 (four open-ended questions), and its app design in W2-3 (ten Likert scale and three open-ended questions based on the Technology Acceptance Model) (Davis, 1985).

The final student codes were evaluated manually, quantitative data was analyzed using descriptive statistics, and qualitative data were evaluated using descriptive statistics (Vetter, 2017); Qualitative data (e.g., participants' perceptions of <colette/>) were processed and shown as a summary content analysis (Mayring, 2010).

2.2 Workshop Austria (W1)

In W1, held at the Johannes Kepler University (JKU) in Linz, Austria, three *Draw-o-Bot* exercises were tested. W1 (100min) began with a short introduction to CT. After this, students received a link to a <colette/> test environment for the tasks, a worksheet, and a short explanation of block-based programming. Afterward, the students had time to work on the tasks and experiment with the test environment. The tasks tested were designed by the authors and intended to gradually familiarise students with BBP. The students, paired into groups within all exercises, had to program a code for the commands, so the robot drew a square (Task 1), a traffic sign (Task 2), and the first letter of the student's name (Task 3). Once the code was done, one group member took the role of the 'Instructor', who read out loud the commands shown on the screen of the mobile device. Another group member took the role of the 'Robot', using a pen to draw according to the instructor's guidance on a piece of paper, checking if the desired object (e.g., a road sign) would appear. All participants had the opportunity to revise their codes at any time. After the first task, students swapped roles. Further, students were encouraged to find a way to shorten their code, e.g., using loops. After the coding exercise, the students had to answer the questionnaire (15min):

(1) *What is 'CT' to you?*
(2) *Did you like/dislike the tasks?*
(3) *Did you have any issues with the app during your tasks?*
(4) *Which CT aspects are included in the tasks?*

2.2.1 Sample & Data Processing (W1)

Nine students, aged 10–13 years, participated in W1 (female=1; male=8). The students were all part of a course for gifted students (COOL Lab Talents Club, 2022), meaning that they had all shown an increased aptitude for learning and understanding new concepts. Three male students that had been to previous *Scratch* workshops already had a reasonable understanding of BBP, as well as loops. The students split themselves independently into three groups of two, and one group of three. The group of three had two students with prior knowledge of coding, leaving the last of the three students with prior knowledge in a two-person group. The remaining groups had no members with prior knowledge of coding. For data collection, screenshots of final codes on the students' devices, and the drawn objects on the paper at the end of W1 were collected. The authors analyzed the pictures and codes after W1. Furthermore, the answers to the questionnaire were collected and processed using an Excel Sheet.

2.3 Workshops Netherlands (W2)

Two workshops were organized by Utrecht University (UU) in the Netherlands in different settings, to test four *Building Cubes* tasks, developed and designed for <colette/> by the authors. The first setting (120min) was an online session with the theme *'Architect in the virtual world'*. The second setting (75min) was an on-site workshop at Utrecht University. In both workshops, the students were first introduced to the topic and BBP app and then worked either alone or in pairs (30–60min). The tasks gradually introduced the app and its components. For the BBP activity, the mobile app was used. The app provided both simple and straightforward programming blocks to create single-unit cubes at selected coordinates on a checkerboard and more advanced repeat blocks and variables. This way, students could create structures (e.g., buildings) made from unit cubes. To see and check their results, the students pointed their devices' cameras to an AR marker to view the cube building in ARand turn it around (*Fig.1*). After the programming exercise, the students filled in a questionnaire (15min) about their perception of the tool based on the Technology Acceptance Model (Davis, 1985):

(1) It was easy to understand the instructions.
(2) It took a long time to learn to use the app.
(3) The app is difficult to use.
(4) The app is clear.
(5) The app is fun to use.
(6) The app easily does what I want.
(7) I would like to use the app in school.
(8) I would like to use the app outside of school.
(9) The app has apparent faults. If so, please explain why.
(10) Did you have experience with programming before this workshop? If yes, describe your experience.
(11) Do you have tips/tops for us?

2.3.1 Sample & Data Processing (W2)

In the online workshop, a group of 27 girls, aged 13–14 years, participated. From this group, nine girls and their parents consented in participating in the research. In the on-site workshop a group of 26 girls, aged 14–15 years, participated, 15 of whom filled in the questionnaire. Both groups (W2) were part of a program for girls with a special interest in STEM topics in the Utrecht region in the Netherlands. Many of the students had prior experience with programming. The collected data were the responses to the questionnaire from 24 students about their perception of the app. For 33 students, the logged data of the app was used to analyze the successful tasks' completions.

2.4 Workshop Slovakia (W3)

W3 was held at Constantine the Philosopher University in Nitra, Slovakia, during the ordinary informatics lectures at grammar school Gymnázium Golianova 68 in Nitra. The students were supervised by one instructor in the same four tasks used in W2 (*Building Cubes*). During the lesson (Duration=45 min), students were divided into groups, given tablet PC and QR codes linking to the given task, and explained the AR environment and marker. A follow-up exercise involved students experimenting with the <colette/> environment.

2.4.1 Sample & Data Processing (W3)

In W3, 32 grammar school students, aged 14–18 years (female=27; male=5), participated. Most students worked in pairs; the rest worked alone. 88% had previous experience with coding mostly in text-based languages (e.g., *Python* and *Scratch*); and two students had experience with *C#* and *C++*. Three participants had no prior knowledge, even though programming is mandatory in Slovakia. The students completed the same questionnaire and tasks as in W2. Some students were required to fill out the questionnaires after the class due to lack of time. Therefore, only 25 answers (female=20; male=5) were collected.

3 RESULTS

3.1 Results Workshop Austria (W1)

While some groups used the 'cardinal' up, down, left, and right blocks to create the square in the first task, others immediately utilized the 'turn-and-walk' approach, with one group using the loop functions. This approach of using 90-degree turns was later spoken about by the students as being the better solution. According to the participants, this approach was helpful as it led them to use loops 'easier' or in a 'faster' way. It is worth noting that some discussions arose among the students using the 'go left' and 'go right'-blocks, as it was not clear to them if the robot was turning or walking sideways. In the second task

(*Fig. 2*) of drawing the pattern of a pharmacy road sign, all groups used the ‘turn-and-walk’-approach, and all groups tried to solve the task using loops from the beginning. Only after failing repeatedly and being told to try and do only one part of the drawing by the instructors, did a group try to solve the task without using loops. Instructors did not notice a major difference between students who had previous BBP knowledge and those who did not. Only one group had time to start working on the third task of drawing the first letter of their name. However, this group did not have time to refine their code and receive satisfying results. In their first attempts, they tested out using a loop within another loop. After a final group discussion, it was clear to all students that shortening the code was practical/helpful, and even though other solutions were possible, the students independently viewed the solutions that used loops as being more ‘correct’ than others, without any additional help from the instructors. All students successfully completed the first two tasks. Unfortunately, no app log data (e.g., number of trials per task) was available. In participants’ answers (n=9) to the questionnaire, the majority of the students explained that to them, CT is to *‘think like a robot and follow commands’*. During W1, it appeared to instructors that the participants had no major problems with the test environment of the <colette/> app and task design. In response to the question of whether there were any issues, and whether they liked the tasks, all students stated that they had *‘no issues’*, and *‘liked the tasks’*. One group stated that they thought that the ‘exercise was very interesting’. Some students noted that they found the questionnaire itself, and the mix of German and English languages in the app and on the worksheets ‘*a little bit annoying*’.

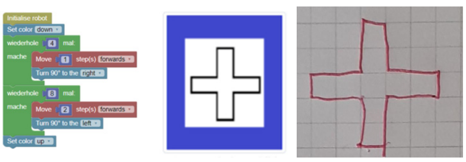

Figure 2: Example Solution of the Task *‘Draw a Road Sign’* from Austrian Students in W1 (n=9).

3.2 Results Workshops Netherlands (W2)

Overall, most of the girls managed to use BBP to create the target building and complete the exercises. Furthermore, from the app log data, it appeared that 19 of 33 students (57.6%) used the more advanced programming count block in one or more of their solutions and that 12 used it successfully (*Table 1*).

During the sessions, it appeared that the assignments worked differently for the different age groups. The concept of a variable seemed to be a difficult concept for the younger students. Although they succeeded in solving the tasks, they tended to avoid using the repeat block and variables, even if it would give a more efficient solution. The 14–15-year-olds picked up the concept of variables more easily. BBP concepts (e.g., repeat block and variables) were used more often and with greater success by the older ones. When learning and using BBP concepts and coordinate parameters, the students seemed to profit from the instant feedback given by the AR visualization, as it made the meaning of the code. For example, one student discovered how entering the coordinate parameters in the programming block led to placing the cube at the desired location. BBP with the parameters for each coordinate was at first an abstract concept with numbers and after seeing the result in AR, students made the link between the numbers in the programming block indicating the coordinates and the location of the cube in space. From the questionnaire data (n=24) and observations, it appeared that almost half of the students agreed or strongly agreed that the app is fun to use (46%), and the tasks’ instructions were simple to understand (46%). Further, the app was easy to learn (65%), not difficult to use (65%), clear (46%), and worked the way the students wanted it to (38%). 35% would like to continue using the application at school, and 27% would use it outside of school. It was not always clear to the participants how to utilize <colette/> to perform the task. According to the questionnaire, students were asked if the app had any clear faults and, if so, what faults: Students mentioned that users could easily lose their code, *‘if you reload your phone, the code is gone’*; it was unclear how the count block worked; problems with using variables *‘sometimes some variables didn’t work’*. In most cases, the technology worked well, but on a few phones, the AR view didn't work properly, so students used laptops with webcams or collaborated with another student on a working phone. The small size of a smartphone screen was sometimes experienced as too limiting. In W2 the time was a bit short for younger students but sufficient for the older ones.

Table 1: Successful Completion of Tasks (in%) and Number of Trials per Task of the Students W2 (n=33).

Tasks	Successful completion (%)	Participating students (f)	The average number of trials (SD)
Task 1	53.3%	30	2.33 (1.94)
Task 2	75%	20	3.33 (2.69)
Task 3	71.4%	14	5.36 (6.00)
Task 4	n.a. (free mode)	9	5.67 (2.55)

3.2 Results Workshop Slovakia (W3)

According to the instructor, 75% of the students were able to finish all three tasks, and six students or student pairs were able to continue their work to create their own structures (e.g., heart, tree, fish, sandwiches, buildings with elevators and a model of the *Slovak Radio Building*), as shown in *Fig.3*.

The biggest problem was a technical issue: the <colette/> test environment refused to run the AR mode. After changing the tablets to tablet PCs, the issue was resolved, and the students could easily employ the <colette/> environment. The most common problem that students encountered was putting the correct positioning of the variable as an argument in a loop (e.g., coordinates in the correct place of the BBP). Instead of using a loop, some students happily used a simple sequence of blocks. Their argument was that the output building was the same as the desired one. In some cases, students found it challenging to debug hidden argument mistakes when more overlapping blocks were placed in the same position. The instructor noted that longer codes should be cut into smaller parts and organised also horizontally, which should help students during their debugging process. As in W2, when learning and using the BBP concepts and coordinate parameters, the students seemed to profit from the instant feedback provided by the AR visualizations. The instructor observed that some students just tried the loop with coordinates, rather than experimenting with the app's BBP commands. According to students' answers (n=25) (*agreed* or *strongly agreed*), the tasks' instructions were simple to understand (76%), <colette/> was easy to learn (88%), not difficult (92%), clear (88%) and fun to use (76%). The app worked the way students wanted (72%), and they would like to continue using the application at school (76%), but only 40% wanted to use it in their leisure time. Some students stated that they would have preferred clearer task instructions, but liked the provided hints, and that they were able to be creative. The AR function surprised most of the students in a positive way: '*It also showed our progress even if we did something incorrectly*'.

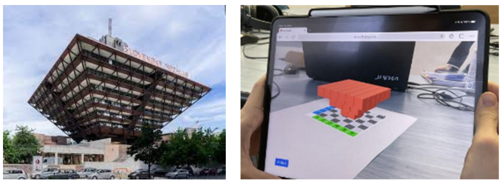

Figure 3: The *Slovak Radio Building* in Real Life (left; Ledl, 2017) and Student's Solution of its Model (W3).

4 DISCUSSION & LIMITATIONS

In this study, two BBP task types in the educational application <colette/> were presented based on individual student workshops (W1-3), to improve or acquire CS and CT skills, in the secondary area, as shown in similar research with BBP applications before (Saritepeci & Yildiz-Durak, 2017). Based on the students' final codes, approaches, loop utilization, and completed exercises, it can be assumed that <colette/> has the possibility to promote coding skills, CT (e.g., problem-solving, abstraction, AT), and can create situations that instructors can use to introduce CT concepts (e.g., variables and loops). It cannot be expected that the concept of variables is picked up and used automatically by students. However, by students 'hitting the wall' of getting into trouble with code when not using variables, a much more fertile ground for students to be willing to learn about variables is created. The concept of variables will always benefit from being introduced to students in connection with their existing app experiences. Thus, the introduction of variables becomes the answer to a problem that students already encountered. Limiting factors in this study are on the one hand its small and imbalanced sample, and on the other hand, the missing log data (e.g., the average number of trials) of W1 and W3, due to technical problems. Only screenshots of the final students' solutions and additional notes could be taken during W1 and W3. Hence, from the W1-3 findings, no generalization can be drawn, but a positive trend regarding <colette/>, student engagement, and enjoyment with CT tasks can be noted. Parts of this assumption were already explored in similar studies with digital technologies or apps, where a positive influence on participants' CT skills (Papadakis, 2022), engagement, and enjoyment during (STEM) lessons was researched (Attard & Holmes, 2020; Willacy, 2017; Drigas & Pappas, 2015). It remains to investigate why some students' uncertainties and issues (e.g., variables did not work) appeared during the workshops, and if it was due to the instructors, the students, the app, the course of the workshops, and/or the task design itself. Firstly, in the next workshops, the instructor should not evaluate the solutions in any way, and let the participants work completely independently without interfering and influencing them (e.g., example solutions, many explanations). Secondly, it is not necessary for students to know what the term 'computational thinking' means. After using the app, most of the students in W1 thought CT meant '*thinking like a robot'*. Therefore, the course and questionnaire of the *Draw-o-bot* were changed (adapted and aligned to the *Building Cubes* workshop). In the future, more time for the tasks and final group discussions will be provided. In W1 and W3, the time provided was too short (e.g., no time to start some tasks). In addition, the perception of the *Building Cubes* appeared different in W2-3: The questionnaire data showed that the younger students (W2) perceived the app as more difficult and less fun than the older users (W3). This might be explained by the fact that the app and the tasks were more challenging for the younger ones. Also, within the *Building Cubes* workshops (W2-3), some students had technical issues (e.g., test environment, AR function, variables), especially in W3. Therefore, some still existing bugs need to be fixed, and the AR view must be improved to work properly. Overlapping blocks formed another hurdle in W2-3. Perhaps the blocks should be colored differently (a feature that students suggested), marked more clearly with thicker outlines, or additional hints could be added for students to detect and fix their errors more quickly. This functionality might make it easier to create variable tasks. It should also be mentioned that the students interpreted some commands (in W1: *'go left'* and *'go right'-blocks*) differently from their intended meaning, thus creating different solutions (e.g., W1: 'turn and walk': unclear if ER was turning and/or walking). The approach of giving little or no assistance so that the participants create codes themselves and shorten them might be helpful, as the students rated the codes with loops as 'better' in W1. This might be explained by some biased reactions of the instructors (e.g., encouragement, a celebration of a specific approach). Basically, it was not mandatory to use loops in W1, as it was in W2-3. Nevertheless, in W2-3 some students were satisfied with using a simple sequence of blocks instead of using loops. In the future, it will be necessary to consider what types of tasks will include mandatory loops because it can only gradually become more convenient to use loops as the tasks gain more complexity.

5 CONCLUSIONS & OUTLOOK

The experiences made in four workshops will be used for further improvement of the application (e.g., issues with AR, time management), and to prepare teacher training courses for the implementation of <colette/> as an educational tool teaching CT in secondary education. Findings after testing and evaluating the app indicate that the participants reacted positively to <colette, the majority were able to solve BBP tasks successfully and create loops to shorten their code. Furthermore, it can be assumed that <colette/> increased the enjoyment of the participants in this study and has the possibility to promote CT skills (e.g., debugging, problem-solving, abstraction, AT). According to the participants, most of the students had no issues with the task instructions and the app was easy to use. Therefore, it can be assumed that <colette/> is a useful educational tool to teach CT, spatial representation, and BBP. Moreover, W1-3 and tasks were re-evaluated and adapted for future workshops, especially regarding time management and using loops. In addition to the currently implemented CT task types, more are planned to be incorporated. Future research, including a larger sampling of students and teachers, and an in-depth analysis of the user data, will focus on the individual types of tasks, and embedding <colette/> into European schools.

ACKNOWLEDGMENTS

The project is (partially) funded by the Erasmus+ grant program (2020-1-DE03-KA201-077363) of the European Union. Neither the PAD nor the European Commission, are responsible for the content nor liable for any losses or damage resulting from the utilization of these resources.

REFERENCES

Alice. (2020). Alice – Tell Stories. Build Games. Learn to Program. https://www.alice.org/ 16.11.2022

Aydeniz, M. (2018). Integrating Computational Thinking in School Curriculum. In: *Khine, M. (eds) Computational Thinking in the STEM Disciplines*. Springer. Cham. 10.1007/978-3-319-93566-9_13

Attard, C., Holmes, K. (2020). "It gives you that sense of hope": An exploration of technology use to mediate student engagement with mathematics, *Heliyon, 6*(1). https://doi.org/10.1016/j.heliyon.2019.e02945.

Benitti, F.B.V. (2012). Exploring the Educational Potential of Robotics in Schools: A Systematic Review. *Computer Education. 58*(1). 978–988. https://doi.org/10.1016/j.compedu.2011.10.006

Blockly, (2022). Introduction to Blockly. Google Developers.https://developers.google.com/blockly/guides/overview 12.11.2022

Blockly Games, 2022. About Blockly Games https://blockly-games.appspot.com 12.11.2022

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). *Developing computational thinking in compulsory education – Implications for policy and practice*. 10.2791/792158

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M.A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V. and Stupurienė, G. (2022). Reviewing Computational Thinking in Compulsory Education, Inamorato dos Santos, A., Cachia, R., Giannoutsou, N. and Punie, Y. editor(s), Publications Office of the European Union, Luxembourg. 10.2760/126955

Colette-Project. (2022). Computational Thinking Learning Environment for Teachers in Europe https://colette-project.eu 14.11.2022

COOL Lab Talents Club. (2022). https://www.cool-lab.net/clubs 13.12.2022

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. and Woollard, J. (2015). Computational thinking - a guide for teachers. Swindon. Computing at School. 1-8. https://eprints.soton.ac.uk/424545/

Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results* (Doctoral dissertation, Massachusetts Institute of Technology).

Drigas, A., Pappas, M. (2015). A Review of Mobile Learning Applications for Mathematics. *International Journal of Interactive Mobile Technologies (iJIM), 9*(3), 18–23. https://doi.org/10.3991/ijim.v9i3.4420

González, E.; De La Pena, A.; Cortés, F.; Molano, D.; Baron, B.; Gualteros, N.; Páez, J.; Parra, C. Robotic Theater: An Architecture for Competency Based Learning. *In Proc. Adv. Intell. Syst. Comput*. 126–137 [Google Scholar]

Jong, T. de, Joolingen, W.R. van. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research,* 68, 179-202.

Ledl, T. (2017). Upside down Pyramid, Bratislava. https://commons.wikimedia.org/wiki/File:Upside_down_Pyramid,_Bratislava_02.jpg 17.1.2023

Lv, L., Zhong, B., Liu, X. (2022). A literature review on the empirical studies of the integration of mathematics and computational thinking. *Educ Inf Technol*. https://doi.org/10.1007/s10639-022-11518-2

Lin, Y., David Weintrop, D. (2021). The landscape of Block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. *Journal of Comp. Languages.* 67(1) 10.1016/j.cola.2021.101075

Mayring, Ph. (2010). Qualitative Inhaltsanalyse. Grundformen und Techniken.11. Weinheim: Beltz.

McGee, M. G. (1979). Human spatial abilities: Psychometric studies and environmental, genetic, hormonal, and neurological influences. *Psychological Bulletin, 86*(5), 889–918. https://doi.org/10.1037/0033-2909.86.5.889

Milicic, G., van Borkulo, S.P., Medova, J., Wetzel, S., Ludwig, M. (2021). Design and Development of a Learning Environment for Computational Thinking: The Erasmus+ COLETTE Project. *Proceedings of the Conference EduLearn 2021*. 1(1). 7376-7383. DOI: 10.21125/edulearn.2021.1495

Musante, K., DeWalt, B. (2010). Participant Observation: A Guide for Fieldworkers. AltaMira Press .https://books.google.at/books?id=ymJJUkR7s3UC

Papadakis, S. (2022). Can Preschoolers Learn Computational Thinking and Coding Skills with ScratchJr? A Systematic Literature Review. *International Journal of Educational Reform*. https://doi.org/10.1177/10567879221076077

Pou, A.V., Canaleta, X., Fonseca, D. (2022) Computational Thinking and Educational Robotics Integrated into Project-Based Learning. *Sensor. 22.* 3746. https://doi.org/10.3390/s22103746

Saritepeci, M., Yildiz-Durak, H. (2017). Analyzing the Effect of Block and Robotic Coding Activities on Computational Thinking in Programming Education. *In book: Educational Research and Practice.* Chapter: 49. Publisher: St. Kliment Ohridski University Press. https://www.researchgate.net/publication/316890358_Analyzing_the_Effect_of_Block_and_Robotic_Coding_Activities_on_Computational_Thinking_in_Programming_Education 12.11.2022

Scratch. (2022). Scratch - Imagine, Program, Share. https://scratch.mit.edu/ 26.11.2022

Shih, W.C. (2017). Mining Learners' Behavioral Sequential Patterns in a Blockly Visual Programming Educational Game. In *International Conference on Industrial Engineering, Management Science and Application (ICIMSA).* 1-2. 10.1109/ICIMSA.2017.7985594

Vetter, T.R. (2017). Descriptive Statistics: Reporting the Answers to the 5 Basic Questions of Who, What, Why, When, Where, and a Sixth, So What? Anesth Analg. 125(5). 1797-180 10.1213/ANE.0000000000002471

Yamashita, S., Tsunoda, M., Yokogawa, T. (2017). Visual Programming Language for Model Checkers Based on Google Blockly. In Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D. (Eds.), *Product-Focused Software Process Improvement. PROFES. Lecture Notes in Computer Science*, 10611. 597–601. Springe. Cham. https://doi.org/10.1007/978-3-319-69926-4_49

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children*. 10.1145/2771839.2771860

Willacy, H., Calder, N. (2017). Making Mathematics Learning More Engaging for Students in Health Schools through the Use of Apps. *Education Sciences, 7*(2), 48. https://doi.org/10.3390/educsci7020048

Xu, Z., Ritzhaupt, A.D., Tian, F., Umapathy, K. (2019). Block-based versus text-based programming environments on novice student learning outcomes: a meta-analysis study, *Computer Science Education, 29*(2-3), 177-204. 10.1080/08993408.2019.1565233