Computers & Education 180 (2022) 104431

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

The role of feedback and guidance as intervention methods to foster computational thinking in educational robotics learning activities for primary school

Morgane Chevalier [a,b,*], Christian Giang [a,c], Laila El-Hamamsy [a], Evgeniia Bonnet [a], Vaios Papaspyros [a], Jean-Philippe Pellet [b], Catherine Audrin [b], Margarida Romero [d], Bernard Baumberger [b], Francesco Mondada [a]

[a] École Polytechnique Fédérale de Lausanne (EPFL), ME B3 424 (Batiment ME), Station 9, CH, 1015, Lausanne, Switzerland
[b] Haute École Pédagogique du Canton de Vaud (HEP-VD), Avenue de Cour 33, 1014, Lausanne, Switzerland
[c] Department of Education and Learning (DFA), University of Applied Sciences and Arts of Southern Switzerland (SUPSI), Piazza S. Francesco 19, 6600, Locarno, Switzerland
[d] LINE Research Lab (LINE). Université Cote d'Azur, ESPE George V, 89 Avenue George V, 06046, Nice, France

ARTICLE INFO

Keywords:
Collaborative learning
Elementary education
Improving classroom teaching
Teaching/learning strategies
21st-century abilities

ABSTRACT

Computational thinking (CT) is considered an emerging competence domain linked to 21st-century competences, and educational robotics (ER) is increasingly recognised as a tool to develop CT competences. This is why researchers recommend developing intervention methods adapted to classroom practice and providing explicit guidelines to teachers on integrating ER activities.

The present study thus addresses this challenge. Guidance and feedback were considered as critical intervention methods to foster CT competences in ER settings. A between-subjects experiment was conducted with 66 students aged 8 to 9 in the context of a remote collaborative robot programming mission, with four experimental conditions. A two-step strategy was employed to report students' CT competence (their performance and learning process). Firstly, the students' CT learning gains were measured through a pre-post-test design. Secondly, video analysis was used to identify the creative computational problem-solving patterns involved in the experimental condition that had the most favourable impact on the students' CT scores.

Results show that delayed feedback is an effective intervention method for CT development in ER activities. Subject to delayed feedback, students are better at formulating the robot behaviour to be programmed, and, thus, such a strategy reinforces the anticipation process underlying the CT.

1. Introduction

There is a consensus nowadays that Computational Thinking (CT) is a multidimensional construct that consists of three widely

* Corresponding author. Morgane Chevalier ME B3 424, Station 9, CH, 1015, Lausanne, Switzerland.
E-mail addresses: morgane.chevalier@epfl.ch, morgane.chevalier@hepl.ch (M. Chevalier), christian.giang@epfl.ch, christian.giang@supsi.ch (C. Giang), laila.elhamamsy@epfl.ch (L. El-Hamamsy), evgeniia.bonnet@epfl.ch (E. Bonnet), vaios.papaspyros@epfl.ch (V. Papaspyros), jean-philippe.pellet@hepl.ch (J.-P. Pellet), catherine.audrin@hepl.ch (C. Audrin), margarida.romero@unice.fr (M. Romero), bernard.baumberger@hepl.ch (B. Baumberger), francesco.mondada@epfl.ch (F. Mondada).

https://doi.org/10.1016/j.compedu.2022.104431
Received 6 May 2021; Received in revised form 31 December 2021; Accepted 4 January 2022
Available online 7 January 2022

spread dimensions (Tikva & Tambouris, 2021; Guggemos, 2021; Zhang & Nouri, 2019): i) computational perspective (i.e., "the perspectives designers form about the world around them and about themselves", Brennan & Resnick, 2012, p.1), ii) computational concepts (i.e., "the concepts designers engage with as they program", ibid.), and iii) computational practices (i.e., "the practices designers develop as they engage with the concepts, such as debugging projects or remixing others' work", ibid.). Nevertheless, in their review, Lye and Koh (2014) deplore the lack of a specific intervention approach to foster both the "computational practices" and "computational perspective" dimensions, with most CT interventions focusing on computational concepts (Li et al., 2020).

Converging with problem-solving abilities (Yadav, 2016), the computational perspective dimension appears to be the most transversal dimension constituting CT. Indeed, this dimension is composed of elements such as identifying a situation, understanding a problem, and modelling a solution (Romero et al., 2017). All these elements are competences present in other disciplines. Therefore, intervention approaches favouring this CT dimension should be based on approaches already widely used in pedagogy. As guidance is considered the most common and efficient instructional approach to sustain the student learning process (Kirschner et al., 2006; Atmatzidou & Demetriadis, 2012; Sapounidis & Alimisis, 2020), this intervention method should also be relevant in the context of CT.

As far as the computational practice dimension is concerned, it is often addressed through programming (Brackmann et al., 2017), which represents the phase of reification of what has been thought upstream. Unfortunately, the programming act, or any form of experimentation in itself, is not sufficient to ensure learning (Mayer, 2004), with explicit guidance being required (Kirschner et al., 2006) to help students reflect on their practices. Moreover, during the human-machine interaction while programming, feedback is of first-order to guarantee the good development of the practice (Hattie & Timperley, 2007; Shute, 2008; Smith & Lipnevich, 2018; Wise & O'Neill, 2009) and depending whether it is immediate or delayed the effects may vary.

In response to the shortcomings put forward by Lye and Koh (2014), we propose to study the role of feedback and guidance as pedagogical interventions to foster computational thinking in educational robotics learning activities for primary school. Hence, we ask the following interrelated research questions:

1) Between guidance and feedback, which of these two intervention methods most favourably fosters CT competence for 8-9-year-old students?
2) How do(es) that most favourable intervention method(s) impact students' problem-solving strategies?

In the following sections, we detail the experiment we conducted to address these two questions. Section 2 first presents the state of the art of the types of guidance and feedback and their effect in ER problem-based learning activities to foster CT. Section 3 presents our experiment's details. Section 4 examines the data collected and shows, on the one hand, the positive impact of using delayed feedback on CT and, on the other hand, that cognitive processes and learning outcomes differ according to the type of feedback used (immediate or delayed). Finally, Section 5 presents a conclusion of this work.

2. Background

This section examines the literature concerning guidance (section 2.1) and feedback (section 2.2), which are considered effective intervention methods to foster CT competences in ER settings.

2.1. Guidance as an intervention method to foster CT in ER settings

2.1.1. The purpose of guidance in education

Guidance is the act of guiding students to construct knowledge (scaffolding process, Vygotsky, 1978). Through the tutorial interaction (Bruner, 1983), teachers try to get students to solve a problem that they cannot solve alone. This help provided by teachers is then gradually withdrawn (fading of the scaffolding process). Intending to make students actors of their learning, teachers can resort to various guidance acts such as reformulating the problem, decomposing, demonstrating … provided through multiple media (lectures, videos, computer-based presentations …, Clark et al., 2012). Hence, guidance varies according to an intensity cursor (minimal vs. strong) and a wide range of action types.

2.1.2. Variating guidance in problem-solving

The variation of guidance depends on whether students are novices or experts (Clark et al., 2012). Indeed, novices or students "dealing with novel information [so using their working memory] should be explicitly shown what to do and how to do it, and then have an opportunity to practice doing it while receiving corrective feedback" (Clark et al., 2012, p. 7). Contrariwise, as expert students benefit from "both their working memory and all the relevant knowledge and skill stored in long-term memory" (Ibid., p.9), they can still learn with minimal guidance.

2.1.3. The case of trial-and-error

The distinction between novices and experts highlights a specific case in computer problem-solving: the trial-and-error strategy. Indeed, as stated by Clark et al. (2012):

> "Solving a problem requires searching for a solution, which must occur using our limited working memory. If the learner has no relevant concepts or procedures in long-term memory, the only thing to do is blindly search for possible solution steps that bridge the gap between the problem and its solution" (p.10).

This blind search may refer to a blind trial-and-error strategy (i.e., a quick alternation between the programming and evaluation/test phases) in the specific context of ER. Chevalier and Giang et al. (2020) refer to this behaviour as "an over-investment in programming concerning other problem-solving tasks during ER activities" (p.2). They found that "a non-instructional approach for educational robotics activities (i.e., unlimited access to the programming interface) can promote trial-and-error behaviour" (p.16). Such an unfettered exploration leads to behavioural activity during learning which may not systematically imply cognitive activity (Mayer, 2004). Indeed, without guidance, students may not actively think while problem-solving, tempted by the natural strategy of trial-and-error rather than reflecting on action (Biesta & Burbules, 2003).

Thus, trial-and-error is a usual approach for novices, whereas experts are likely to use domain-specific strategies to solve problems (Mayer, 2004). Nevertheless, novices may go beyond such an approach and try more rational approaches (Alimisis, 2019). Despite being novice or expert, Merisio et al. (2021) state that "all robotic programming is trial-and-error and reasoned at the same time" (p.199). Namely, they recall that AI problem solving uses blind search methods, i.e., considering all the possible paths until a "right" one is found. However, such a strategy is time and memory consuming.

In sum, explicit instructional guidance is more effective and efficient than partial guidance (Kirschner et al., 2006; Sweller et al., 2007). Meanwhile, minimal guidance techniques can reinforce or practice previously learned material (Clark et al., 2012). Whether it is minimal or not, the type of guidance must be described.

2.1.4. The type of guidance in ER

As CT is considered a problem-solving tool (CSTA, 2011), it is relevant to identify the effect of guidance in problem-solving. To date, Tikva and Tambouris (2021) have listed in the literature 12 studies out of 37 that refer to scaffolding strategies ("strategies that offer support to students as they learn, including instructional scaffolding, support/guidance, and adaptive, peer-, resource-scaffolding," ibid. p.10) as an intervention approach to enhance CT in the context of ER. Among these, Atmatzidou and Demetriadis (2016) used two different forms of guidance in their study: on the one hand, the teacher acting as a facilitator to scaffold students' actions while solving the programming tasks; on the other hand, worksheets guiding students in their investigation of increasingly complex programming tasks. The authors claim that such worksheets enable students "to start constructing understanding and developing the CT skills" (p. 664). In another study, Atmatzidou et al. (2018) compared the effect of two modes of guidance (minimal versus strong guidance) on the development of students' metacognitive (MC) and problem-solving (PS) skills in the context of ER activities. They found out that strong guidance in solving problems can positively impact students' MC and PS skills. This is consistent with the findings of Chevalier and Giang et al. (2020). They explored the effect of timed guidance in an ER learning activity and found that "a scheduled blocking of the programming interface helped foster cognitive processes related to problem understanding, idea generation, and solution formulation" (p.16), which are directly linked to the "computational perspective" dimension of CT. While timed guidance had a clear impact on learning, the intervention did not adapt to each group's progress as it was imposed on all groups without differentiation. In their review, Honomichl and Chen (2012) point out three elements that facilitate guided discovery learning: (1) strategic presentation of materials, (2) consequential feedback, and (3) probing questions and self-explanations. In this regard, probing questions helps direct students' attention to essential features in situations (Chen & Klahr, 1999).

2.1.5. Problematisation inductor

Despite these previous studies, guidance needed to promote CT in ER is not sufficiently explained in the literature, although it is in other STEM areas. Indeed, from the same worksheet perspective, Fabre and Musquer (2009) recommend that science teachers build what they call "problematisation inductors," aiming at "marking critical features" (Wood et al., 1976, p. 98), which is one of the six functions of scaffolding. Such a document provides metacognitive questions on the task to realise, the means available, and how to achieve the mission. By answering these questions, students reflect on the problem and improve their identification of the situation, understanding the problem, and solution modelling. As a result, they may enhance their computational perspective dimension of CT. Therefore, the problematisation-inductor worksheet seems relevant because it is an equivalent and adapted intervention for all groups in a class. Indeed, although evident, one should not forget that teachers do not have the gift of ubiquity.

Consequently, teachers' choice of intervention approach impacts the equitable distribution of their guidance among the students. This is what a recent study shows (Mehrotra et al., 2020) comparing the use of paper-based versus a screen-based interface in ER learning activities. The authors show that when using screen-based (instead of paper-based materials), the teachers spent over half their time with a single group of students. In this context, the use of the screen seems to take up the teacher's attention and thus be a physical barrier to equalising teacher interventions with students. Using a worksheet can help structure and sustain the teachers' scaffolding and become a tool for guidance.

2.2. Feedback as an intervention method to foster CT in ER settings

2.2.1. The purpose of feedback in education

According to Wisniewski, Zierer and Hattie's (2020) meta-analysis on the effects of feedback on student learning, "feedback not only refers to how successfully a skill was performed (knowledge on the result) but also to how a skill is performed (knowledge of performance)" (p.7). Thus, in a formative context, feedback aims to "enhance learning, performance, or both, engendering the formation of accurate, targeted conceptualisations and skills" (Shute, 2008) by conveying information about the following three questions (Hattie & Timperley, 2007): Where am I going? How am I going? Where to next? Such information should make sense to learners (Henderson et al., 2017) so that they can reduce the gap between their actual performance and their desired outcome (Bahula & Kay, 2020; Hattie & Timperley, 2007) according to their context and needs (Evans, 2013).

2.2.2. The robot as a source of feedback

As feedback is inherently socially constructed and contextually situated (Ajjawi & Boud, 2017), in a classroom, this traditionally involves two stakeholders: students and teachers. This feedback can take one of two "directions" (Wisniewski et al., 2020): feedback can be given by a teacher to one or many students (and vice versa) or given by a student to another student. However, let us consider feedback "as information provided by an agent (e.g., teacher, peer, book, parent, self, experience) regarding aspects of one's performance or understanding" (Hattie & Timperley, 2007, p. 81). This increases the number of stakeholders in the context of technology-enhanced learning environments, increasing the number of feedback directions. In ER activities, feedback can also be provided by the robot (behaviour of the robot that executes the programme), the programming interface (which sends back information concerning the programming), or by a third party (a classmate or the teacher).

When considering the robot itself as a source of feedback, one can consider two sources (Merian & Baumberger, 2007; Patchan & Puranik, 2016). On the one hand, intrinsic feedback comes from the subject's perceptual channels. It is what the performer feels as they execute a skill or performance (physical feel of the movement as it is being performed). On the other hand, extrinsic (or augmented) feedback is provided by external sources, during or after a performance (teacher, timer …). In the context of ER, intrinsic feedback refers to the natural sensory information resulting from the manipulation of the robot. This source is rare in ER (except with haptic solutions, Özgür et al., 2017). Conversely, extrinsic feedback is much more present in ER since it deals with the information provided by the robot or the programming interface as soon as an event triggers an action or an error is made. While extrinsic (or augmented) feedback is beneficial in motivating students, informing them about mistakes to be corrected, and directing their attention to their goal, this may create a dependency on feedback (Schmidt & Lee, 2013).

In sum, while the robot in itself is a source of mainly extrinsic feedback in ER learning activities, researchers nonetheless warn us that not all types of feedback are beneficial to student learning and that feedback must be adapted to the students' needs to promote a mindful reflective process.

2.2.3. Types and channels of feedback with robots

According to Hattie and Timperley (2007), "the type of feedback and the way it is given can be differentially effective" (p.81); it is, therefore, necessary to describe the type of feedback found in ER and the channels at our disposal to convey it. Bakala et al. (2021) identified three kinds of feedback provided by robots linked to the affordance of most educational robots' actuators which we link back to three channels: sound (auditory), light (visual), and movement (haptic). Information can be transmitted through the visual channel employing LEDs, the observable behaviour of the robot (directly or through video recordings when the robot is not directly accessible by the programmer as in our case), or even its programming interface. Videos are a source of extrinsic feedback which can give information about both the knowledge of result and the knowledge of performance (Merian & Baumberger, 2007, section 2.2.1). To the best of our knowledge, no studies have been conducted on video-based feedback during problem-solving in ER activities. The closest work is in physical education using video recordings as a type of delayed video feedback (combined with oral feedback) to reinforce students' learning. Indeed, this type of feedback helps amplify their auto-perception capacity (Aranha & Gonçalves, 2012), and thus "construct a more accurate mental representation of the performance" (p.117, Merian & Baumberger, 2007), permitting a better regulation of their action and accelerating performance improvement. Nevertheless, these authors recall that too much feedback can lead to a decrease in long-term performance (Magill, 1993; Swinnen, 1996) because "the learner who receives feedback after each trial no longer bothers to engage his or her perceptual system and, in the absence of an external source of information, is unable to regulate his or her movements" (p.109). As a consequence, the frequency and timing of feedback should be taken into consideration.

2.2.4. The timing of feedback

Hattie and Timperley (2007) noted the importance of the information provided during feedback and the appropriateness of the feedback's timing concerning the students' instructional cycle. That is why in ER learning activities, the timing of feedback also plays an important role. Observing the robots' behaviour provides immediate feedback (Papert, 1980) on the program's quality and concept reification. This feedback elicits the debugging process and triggers "an iterative cycle of observation, hypothesis generation, hypothesis testing and evaluation of the solution" (Sullivan, 2008, p. 389). Students, however, may enter into a trial-and-error strategy (Weintrop & Wilensky, 2015) which is dependent on the status granted to error as a fault or as something to overcome. Teachers may thus choose to give either immediate or delayed feedback by considering the impact of each.

Immediate feedback generates more rapid problem solving (Hattie & Timperley, 2007; Shute, 2008; Smith & Lipnevich, 2018; Wise & O'Neill, 2009) and appears adapted to short and procedural skills (such as programming and maths) by making an explicit association between outcomes and their causes but may generate a type of dependency towards the feedback and promote less careful and less mindful behaviour (Shute, 2008). Indeed, Siegfried et al. (2017) investigated the role of immediate feedback on students' learning when programming a maze navigation task for a robot. As a result of providing real-time feedback and hints on the code's performances in the maze, the students were able to quickly solve the task without the intervention of an expert, significantly improved their program writing (from 50% to 96%) and decreased the average time to write a correct program by 30%. However, no correlation was found between student learning and performance on the task.

Delayed feedback, on the other hand, generates better retention and transfer in the long term (Hattie & Timperley, 2007; Shute, 2008; Smith & Lipnevich, 2018; Wise & O'Neill, 2009) by engaging learners "in active cognitive and metacognitive processing, thus engendering a sense of autonomy" but may be inadequate for less motivated learners who find themselves frustrated and thus less likely to acquire the desired knowledge or skill set (Shute, 2008). Chevalier and Giang et al. (2020) studied the impact of delayed feedback in creative computational problem solving (CCPS). To break the unproductive trial-and-error loop, access to programming was blocked in certain project phases, thus breaking the access to the immediate feedback on the program's quality. This enabled

students to work iteratively on all other cognitive processes of the CCPS model (i.e., Understanding – Generating Ideas – Formulating behaviours). Nevertheless, a more natural access blocking to programming and evaluating through delayed feedback could make the trial-and-error strategy more productive and commensurate with the progress of each pair of students (Lye & Koh, 2014). To the best of our knowledge, there is no research employing delayed feedback in ER to promote CT.

3. Methods and instruments

To illustrate the implementation of our study, in section 3.1, we present the experimental setup of the collaborative remote ER programming mission, which integrates the proposed intervention methods (guidance vs. feedback). Then, in section 3.2, we explain the design of the experiment. Subsequently, in section 3.3, we describe the participants and, in section 3.4, the data collection we carried out to capture the students' CT competence, i.e., both their performance and their strategies in creative computational problem solving (CCPS).

3.1. Experimental setup

For this study, we have implemented a modified version of the collaborative remote programming mission R2T2[1] (Mondada et al., 2016) that uses Thymio robots (Papadakis, 2020, p.45; Mondada et al., 2017). Thymio is an open-source mobile robot with generic sensors and actuators. The adaptations limit the mission time to 45 min (compared to 3 h for the regular mission) to suit a real classroom situation.

The mission's goal was to repair an imaginary energy generator on planet Mars (see the upper part of Fig. 1, with orange background and zoom on sector C). The task consisted of taking the robots, aligned in front of the external wall of the station, to the access windows of the core generator (central area of the map, point 3 in Fig. 1) via the black track (point 2). In the description of the task, the access to the track (point 1) is neither explained nor even mentioned: the students thus had to study the setup, understand the starting situation of their robot, and find a solution how to reach and then follow the track. This mission has to be performed from Earth (see the bottom part of Fig. 1, with blue background), where each of the four classes has four computers, each one controlling one robot in the sector corresponding to the class. The Mars setup is situated physically in a distant room, connected by the Internet. This distant room is equipped with a camera allowing visual feedback to the students in their classrooms.

To understand and complete the mission, the students could benefit from a collaborative environment (in teams of 2 or 3 students) and different artefacts (a remote robot and a local one, paper tracks, the video feedback of the setup and their remote robot, and a programming interface). Students had to mobilise their knowledge about the robot's sensors and actuators and then develop an autonomous control strategy to achieve the mission. Their solution was then programmed using the visual programming language Thymio VPL (Shin et al., 2014) and sent to the station's robot (Fig. 8). Moreover, students could use local Thymio robots to test their solutions.

3.2. Design of experiment

We set up an experiment with four possible experimental conditions to identify which of the two selected methods (*with/without guidance* and *immediate/delayed feedback*) fosters the CT dimensions (Table 1). We thus created four groups (classes A, B, C, D) in which we randomly assigned students. For instance, in class A, students were subjected to the experimental condition "*with guidance* and *with delayed feedback*" (Table 1). *Guidance* refers to using a worksheet during the mission, guiding students in solving the mission through metacognitive questions (Appendix C). *Delayed feedback* refers to the video feedback from the energy generator on planet Mars with a delay of 30 s which is inherent to the R2T2 Mission. *Immediate feedback* refers to video feedback without delay between the moment students would push the "RUN" button (Fig. 8) to send their program to their robot and the moment they would get the video feedback of their robot executing the program on the energy generator on planet Mars. The teacher announced the delay of classes A and D at the beginning of the mission. Moreover, the mission's storytelling justified this delay since the transmission occurs between Earth and Mars (section 3.1).

Before the mission was carried out, we ensured that the students had the necessary prior knowledge regarding the robot's use and programming. The same person taught a science class 1 h a week for 12 weeks in each of the four classrooms (Fig. 2).

The data were collected using an individual written questionnaire (section 3.4.1) presented to the students at two stages: before the experiment (pre-test) and after the experiment (post-test). Video data (section 3.4.2) were collected during the experiment, i.e., the collaborative remote programming mission. The experiment was a collective process, as students solved the mission in groups of two or three people.

3.3. Participants

Four classes from two different schools in (anonymisation) participated in the mission, with 66 students (33 girls, 33 boys) aged 8 to 9. Authorisation for the participation of each student was granted by their legal guardians (parents). The school directors and the

[1] <https://r2t2-collaboration.com/mars-r2t2-mission/>

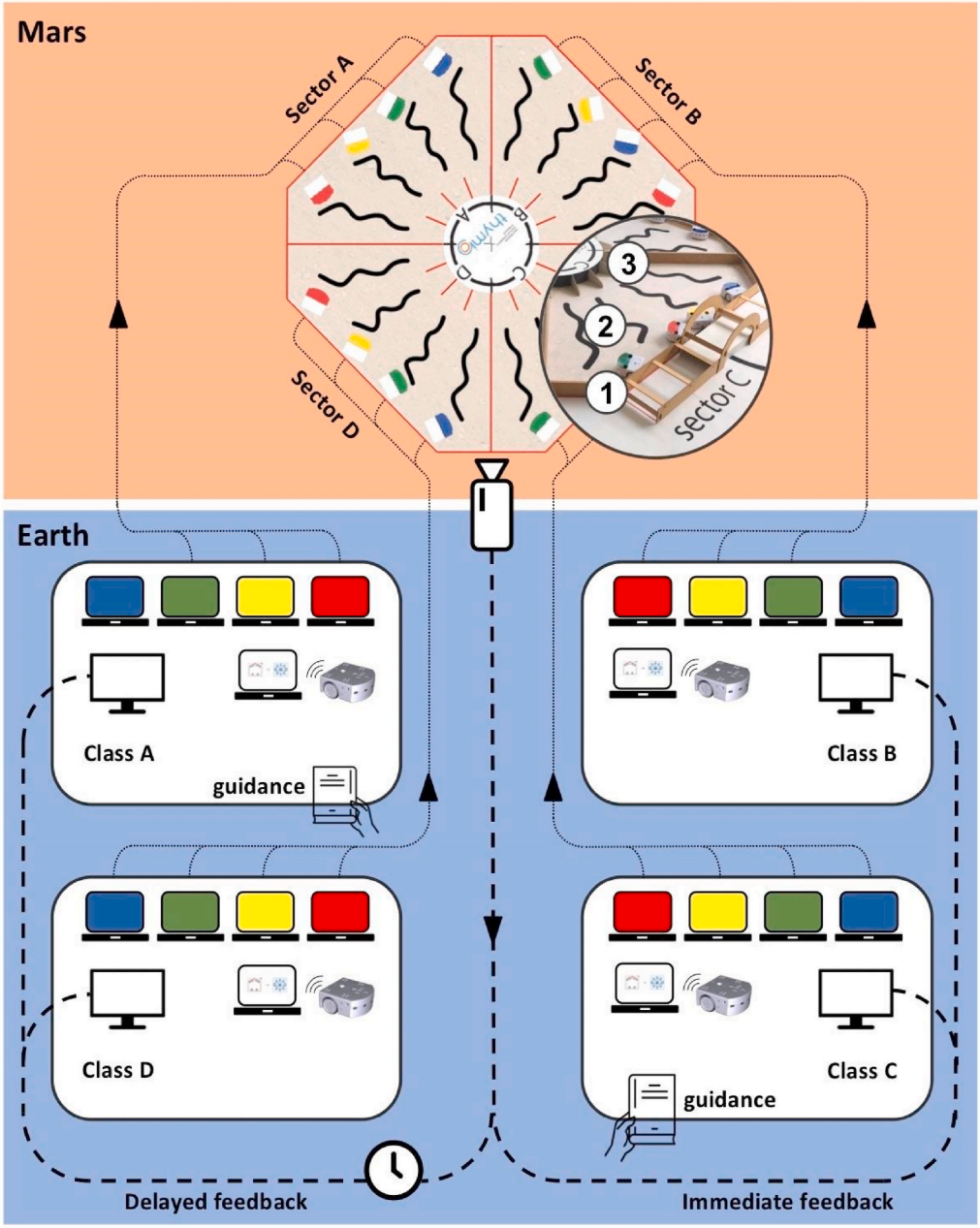

Fig. 1. Organisation of the experiment. In the orange background part, labelled "Mars", one physical setup located in a remote office space (standing for the energy generator on planet Mars) consists of 4 quarters (called sectors A, B, C, and D). In each of them, 4 distinct Thymio robots (blue, red, yellow, and green) can be remotely programmed. In the same room, a video camera streams the top view of the four sectors. In the blue part, labelled "Earth", the video streams (see the arrows in lines) are broadcast to 4 classrooms either with a 30-s delay, see classes A and D, or see classes B and C without delay. In each classroom, the participants are subject to 1 of the 4 different experimental conditions: class A is with *guidance* and with *delayed* feedback, class B is without *guidance* and with *immediate* feedback, class C is with *guidance* and with *immediate* feedback, class D is without *guidance* and with *delayed* feedback. In each classroom, each of the 4 pairs of students has the following at its disposal: a Thymio robot to carry out tests locally and a computer on which 3 applications are open (one to record the students' activity on the computer, Thymio Aseba VPL to locally program the robot, and Thymio VPL to remotely program the robot assigned to the team). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

teachers approved the research. A statement on ethics approval and consent was issued by (anonymisation).

The experiment was carried out twice, once in each school. Each time, two classes were divided into 16 independent teams of 2–3 children, dispatched in 4 different groups, related to the 4 different sectors in the playground (Fig. 1). Based on an estimation of the

Table 1
Four experimental conditions corresponding to the two modalities with two levels chosen for the "pedagogical intervention" factor and their mapping to the four classes A to D.

	Guidance	No Guidance
Immediate Feedback	C	B
Delayed Feedback	A	D

Fig. 2. Research process and dedicated learning time.

students' level that the teacher provided, the students from the same level were randomly distributed to ensure a fair distribution of the "school performance level" across groups and thus get a homogeneous repartition between conditions (Table 2).

3.4. Data collection: a two-step strategy to answer the research questions

The first step was to determine whether there was a significant difference between the four experimental conditions (Table 1). If one (or more) of them were identified, then the second step would be to conduct detailed video analysis of what was going on under the one (or many) experimental conditions(s) that was (or were) favourable.

3.4.1. Measuring CT performance through pre-post tests

From state of the art and to have a CT definition with finer granularity compared to the 3 CT dimensions in Brennan & Resnick's model (2012), we selected the six components of the CT competence of Romero et al. (2017) framework (Table 3). We operationalised it into 18 questions (see Appendices A & B) to measure each CT dimension with six questions varied three times. The range was thus 0 to 18 points which we normalised to have a score between 0 (minimum) and 1 (maximum).

To avoid measurement reactivity (Van der Maren, 2014, p. 113), two different sets of 18 questions (MCQ A and B, Appendices A and B), including the same difficulty of questions, were developed. For the pre-test, the questionnaires MCQ A and B were randomly but equally distributed among the students. The two questionnaires for data collection (MCQ A ($\mu = 0.62 \pm 0.13$) and MCQ B ($\mu = 0.56 \pm 0.15$)) are assumed to be equivalent as there was no statistically significant difference between them ($t = 1.35$, $df = 66$, $p > 0.05$). Nevertheless, this non-significance could also be due to the small sample size, which is in itself a limitation of this study. For the post-test, each participant completed the complementary questionnaire. Subsequently, each answer was graded with either 0 (wrong), 0.5 (partially correct), or 1 point (entirely correct). We then summed the scores for each answer to compute the resulting score for each of the six components (COMP1 to COMP6) and the three corresponding dimensions ("Analysis", "Technological literacy", and "Making digital creation"). As the goal is to observe the learning gain in students' CT scores, we computed the normalised change (NC), a differential score, for each participant, representing the difference between the pre-test and the post-test. Our calculation was based on Coletta and Steinert (2020) 's formula:

$$NC = \begin{cases} \dfrac{Posttest - Pretest}{100\% - Pretest} & if\ Posttest \geq Pretest \\[2ex] \dfrac{Posttest - Pretest}{Pretest} & if\ Posttest < Pretest \end{cases}$$

Participants who obtained a score of 100% in both pre and post-test ($n = 2$) are not included in the analyses because it seems that our evaluation tool was not able to measure the effect of the experiment on their learning outcomes.

We first performed a multivariate analysis of variance (MANOVA) on the three dimensions ("Analysis", "Technological literacy", and "Making digital creation") to check for associations between our measures (CT dimensions), factors (*Feedback* and *Guidance*) and

Table 2
Number of students per sector having participated in the experiments according to an evaluation of their school performance as provided by the teachers.

Dimension	Sector A	Sector B	Sector C	Sector D
Level 1 (highest performers)	5	5	4	4
Level 2	4	5	4	5
Level 3	4	4	4	4
Level 4 (lowest performers)	3	3	4	4

Table 3
The 3 CT dimensions, according to Brennan and Resnick (2012), are put face to face with the 3 CT dimensions according to Romero et al. (2017). This face-to-face setting of the 2 models makes it possible to release a finer granularity of the CT (into 6 components) and thus to make it operational into 18 questions.

CT Dimensions of Brenan & Resnick's model (2012)	CT Dimensions of Romero et al.'s model (2017)	Component of Romero et al.'s model (2017)	Competency component	Questions in our 2 MCQs
Computational Perspective	Analysis	CT1	Problem identification	1, 2, 14
		CT2	Organising and modelling the situation	5, 15, 18
Computational Concepts	Technological literacy	CT3	Code literacy	4, 7, 8
		CT4	Technological system literacy	3, 16, 17
Computational Practices	Making digital creation	CT5	Create a computer program	11, 12, 13
		CT6	Evaluations and iterative improvement	6, 9, 10

their interactions. Then, three independent ANOVA were performed on each dimension.

3.4.2. Measuring CT strategies through video analysis

In a second step, we attempted to identify the students' strategies throughout the mission. The qualitative instrument used was videos and screencasts observations. During the collaborative remote programming mission, we recorded the students' activity thanks to a screen recording software, such as QuickTime Player (Apple, Cupertino, USA), which allows recording students' conversations and making a screencast of their VPL programming activities. Cameras located in each classroom allowed to record all groups of students in the classrooms.

To trace the ongoing cognitive processes from the student's activity, we used the existing phases of the CCPS model (Chevalier & Giang et al., 2020; Chevalier and El-Hamamsy et al., 2021, Fig. 3), which we evaluated according to the following phases (with indicators as an example): Understanding the problem (USTD, student questions what to do or debug), generating ideas (IDEA, students exclaims that he/she found or knows something; offers ideas), formulating the robot behaviour (FORM, the student expresses how the robot should act), programming (PROG, student codes the behaviour of the robot), evaluating (student presses the RUN button and observes the robot or video feedback screen), off-task behaviour (OFFT, student is no longer working on the task), and the direction of the gaze (at the VPL screen, the video feedback screen, robot, other groups). In addition to these indicators of the CCPS model, we deepened the "programming" dimension (PROG) to specify it more precisely: how many RUNs? How many STOPs? Programming locally or remotely? Are there any corrections to the program or even its deletion? (Appendix D). An example of how the judgment was made is given in section 4.2.

Following the same video analysis method explained in Chevalier and Giang et al. (2020, p. 10), one evaluator carried out the video analysis using software dedicated to creating activity chronicles (such as Actograph, developed by SymAlgo Technologies, Paris, France). Student verbatim was also collected and used in the discussion to support the annotations.

4. Results and discussion

To address our two research questions in a comprehensive way, results and discussion are jointly articulated. Firstly, to answer RQ1, we present the impact of the experimental conditions on CT performance (section 4.1). Subsequently, to answer RQ2, we present the results of the analysis of student strategies according to the experimental conditions found to be significant in RQ1 (section 4.2).

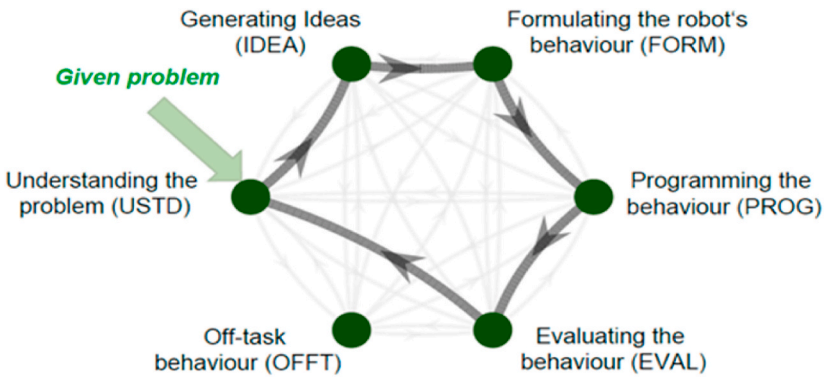

Fig. 3. The six phases of the CCPS model (Chevalier & Giang et al., 2020).

4.1. Between guidance and feedback, which of these two intervention methods most favourably fosters CT competence for 8-9-year-old students? (RQ1)

The distribution of the students' pre and post-test scores is given in Table 4, and both follow a normal distribution according to Kernel density estimation. The results show that students from sectors A and D which both had *delayed feedback* significantly improved their scores ($t_A(15) = -2.628$, $p_A = .019$, Cohen's d = 0.68; $t_D(16) = -4.381$, $p_D = .000$, Cohen's d = 0.94 with a confidence interval to 95%), with the latter obtaining the best average scores in the post-test ($\mu_D = 0.68 \pm 0.08$). Only the group from sector B (*immediate feedback*, *no guidance*, as experimental conditions) remained on the same level between these two measurements (pre-test, $\mu = 0.67 \pm 0.10$; and post-test $\mu = 0.66 \pm 0.15$). This result is consistent with the low effectiveness of minimal *guidance* shown in the literature (Kirschner et al., 2006). However, sector D, subject to *delayed feedback without guidance*, has the highest post-test score. Thus, it seems that the combination of *guidance* and *feedback* methods might have affected student CT performance differently.

Based on the normalised change (NC, section 3.4.1) in Table 5, the multivariate analysis of variance (MANOVA) on the 3 CT dimensions allows us to identify an interaction effect between *feedback* and *guidance* ($F(3, 60) = 2.772$, $p = 0.049$). We then performed a one-way ANOVA on each CT dimension. As shown in Fig. 4, results for the "Analysis" dimension highlight a significant interaction between *feedback* and *guidance* ($F(1, 62) = 5.32$, $p = 0.024$), i.e., the effect of *feedback* seems to differ according to the *guidance*: when there is *no guidance*, then *delayed feedback* is relevant for learning. No other significant effects were found for the "Analysis" dimension, the "Technological literacy" dimension and the "Making digital creation" dimension.

At this stage, our results show that *delayed feedback* significantly fosters the "Analysis" dimension of CT, provided *no guidance* is given. We thus propose to consider *delayed feedback* as an effective intervention method for CT development in ER activities, thus addressing the lack of a specific intervention approach raised by Lye and Koh (2014). As only one guidance approach was tested, more research is required to investigate the impact of specific guidance methods.

4.2. How does feedback impact students' problem-solving strategies? (RQ2)

To further develop these results, we analysed the videos to investigate the relationship between the mobilised competences and learning among the students that were only subjected to *feedback* (immediate or delayed), i.e., in sectors B and D, which incidentally differ significantly (at 10% threshold) in terms of their normalised change ($t(32) = -2.392$, $p = 0.086$, Cohen's d = 0.82) for the "Analysis" dimension of CT.

The aim is to evaluate the students' learning strategies and thus to understand how they construct meaning in interaction with the playground (Chevalier & Giang et al., 2020) and in dialogue with each other (Denner et al., 2014) to investigate the reason behind these differences. Results from the analysis of 4 groups from sector D (9 students, 794 CCPS phase transitions in total) and 4 groups from sector B (10 students, 596 CCPS phase transitions in total) are presented in Fig. 5. As shown in Table 6, students in sector B completed the mission almost 3 times faster than those in sector D. On average, students in sector B performed almost twice as many transitions per minute than in sector D. Under immediate feedback, a "doing" based strategy (Lye & Koh, 2014) seems to be prevalent, as opposed to delayed feedback which seems to favour "thinking-doing" (ibid).

Moreover, students in sector B engaged faster in remote programming than those in sector D. As shown in Fig. 6, behaviours differ between the two experimental conditions in sectors B and D, which may explain the different NC scores between them. The students working with the *immediate feedback* (sector B) are more likely to transition into a nearly endless PROG-EVAL loop (see two-way arrows in Fig. 6): on average, under *immediate feedback* and from the PROG phase, students have a 70% chance of moving toward the EVAL phase (and 78% from EVAL to PROG). Comparably, the students subjected to the *delayed feedback* (sector D) do not get stuck into this same loop as they also promote other mental activities (see the one-way arrow between EVAL and USTD phases in Fig. 6). Moreover,

Table 4
Distribution (Mean μ, SD σ) of students' pre and post-test scores per sector. In bold are the significant changes in performance.

Sector	Experimental conditions	Time	Analysis	Technological Literacy	Making Digital Creation	Total Mean CT Score	Mean CT Score Improvement Significance
A	Guidance	Pre-test	0.61 ± 0.19	0.69 ± 0.15	0.40 ± 0.21	0.57 ± 0.15	**$t(15) = -2.628$, $p = 0.019$, Cohen's d = 0.68**
	Delayed Feedback	Post-test	0.63 ± 0.16	0.84 ± 0.18	0.50 ± 0.24	0.66 ± 0.17	
B	No Guidance	Pre-test	0.66 ± 0.18	0.76 ± 0.10	0.57 ± 0.25	0.67 ± 0.10	$t(16) = .118$, $p = 0.907$
	Immediate Feedback	Post-test	0.59 ± 0.13	0.84 ± 0.11	0.55 ± 0.23	0.66 ± 0.15	
C	Guidance	Pre-test	0.60 ± 0.16	0.65 ± 0.21	0.44 ± 0.30	0.56 ± 0.11	$t(15) = -1.115$, $p = 0.282$
	Immediate Feedback	Post-test	0.69 ± 0.17	0.67 ± 0.19	0.49 ± 0.20	0.62 ± 0.11	
D	No Guidance	Pre-test	0.63 ± 0.20	0.59 ± 0.21	0.44 ± 0.21	0.55 ± 0.10	**$t(16) = -4.381$, $p = 0.000$, Cohen's d = 0.94**
	Delayed Feedback	Post-test	0.74 ± 0.16	0.72 ± 0.16	0.59 ± 0.28	0.68 ± 0.08	

Table 5
Distribution of students' Normalised Change scores per dimension (Mean μ, SD σ, min and max values).

Dimension	Observations	Min./Max.	Mean +/- SD
Analysis	66	−0.55/1.00	0.13 ± 0.41
Technological Literacy	66	−0.60/1.00	0.32 ± 0.46
Making Digital Creation	66	−1.00/0.83	0.10 ± 0.46

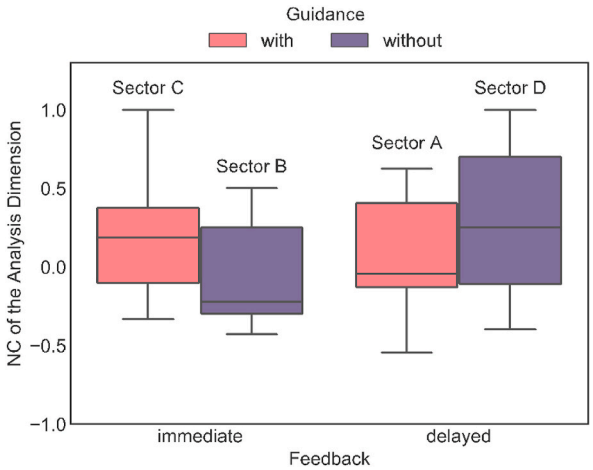

Fig. 4. Boxplots of CT Normalised Change (NC) on the "Analysis" dimension of CT (Romero et al., 2017) according to *Feedback* (delayed/immediate) and *Guidance* (with/without). Letters A, B, C, D indicate the sector in the experiment. The dark line indicates the NC median.

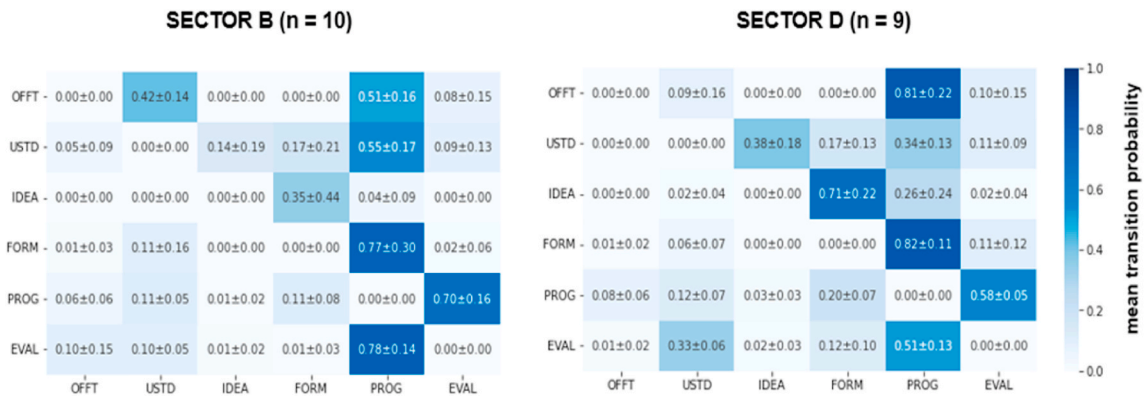

Fig. 5. Distribution (Mean μ, SD σ) of the normalised transition matrices for both sectors B and D. The colour of the boxes varies from lightest to darkest (from 0 to 1). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 6
Average (Mean μ, SD σ) of transitions and time during the mission in sectors B and D.

Average number of transitions:	Sector B	Sector D
- during the entire mission	59.10 ± 19.85	88.22 ± 37.40
- per minute	4.97 ± 1.11	2.88 ± 0.40
Average time in minutes:		
- to complete the mission	11.91 ± 3.17	32.22 ± 15.01
- before the 1st remotely-RUN	5.67 ± 3.65	8.03 ± 6.85

time spent on PROG and especially on EVAL phases is more substantial under the *delayed feedback* than *immediate feedback* (see the diameter of the blue dots in Fig. 6). This point can be justified in part considering the 30 s *delayed feedback*.

Nevertheless, on average, only students in sector D complete a full CCPS cycle (USTD-IDEA-FORM-PROG-EVAL). They, therefore,

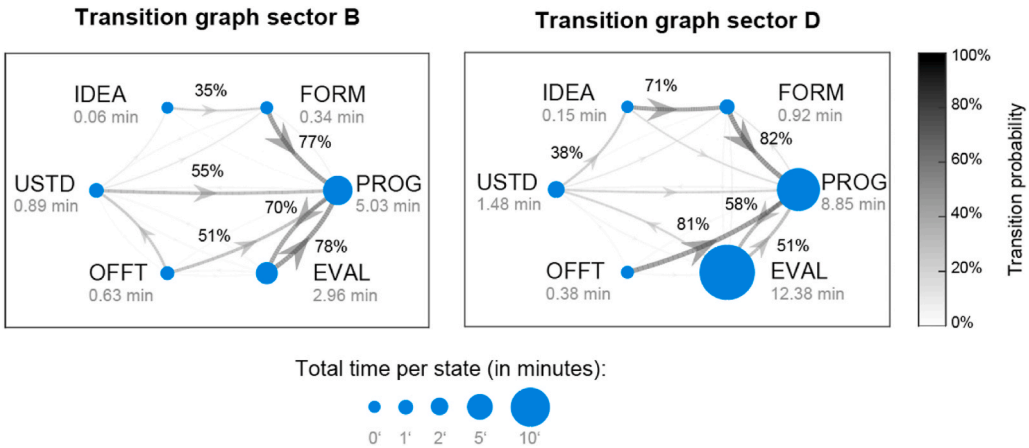

Fig. 6. CT strategies analysis between sectors B and D during the mission. The 6 CT phases (according to the CCPS model, Chevalier & Giang et al., 2020) are Understanding (UND), Generating idea (IDEA), Formulating the robot behaviour (FORM), Programming (PROG), Evaluating (EVAL), Off-task behaviour (OFFT). The thickness of the line shows the transition probability. The diameter is directly proportional to the duration. Note that students in sector D tended to spend more time doing the mission.

have a greater chance of developing the targeted CT competences. It is thus relevant to identify in the videos, thanks to students' verbalisations (Denner et al., 2014), what this type of *feedback* implies in the students' strategies.

As shown in Fig. 5, the means for groups B and D are different, but they seem consistent since the SD values are neither large nor variable. Among the mean differences, nine significant differences have been identified and highlighted in Fig. 7 (see the coloured boxes). We present them as follows and will discuss them in sections 4.2.1 and 4.2.2.

Compared to students in sector D, students in sector B (submitted to immediate feedback) were significantly more likely to move from the USTD to PROG phase ($p = 0.012$), i.e., once they have identified the problem or the bug, they program directly. In addition, they were significantly more likely to move backwards from the EVAL to PROG phase ($p = 0.002$), suggesting a more behavioural than cognitive process.

Compared to students in sector B, students in sector D (submitted to delayed feedback) were significantly more likely to move from the EVAL toward USTD phase ($p = 0.000$), suggesting a debugging strategy, and then from the USTD toward IDEA phase ($p = 0.018$). They were significantly more likely to move from the IDEA toward PROG phase ($p = 0.026$), skipping the FORM phase momentarily to better return to it from the PROG to FORM phase ($p = 0.041$), suggesting that the delay represents a cost of waiting students do not necessarily wish to pay (and thus they prefer to formulate the behaviour again to be programmed without executing the code). Furthermore, they were significantly more likely to move from the FORM toward EVAL phase ($p = 0.024$), representing what happens in the wait caused by the delay (after reformulating the behaviour to be programmed that they have just programmed, they execute the code to evaluate it). Finally, they were significantly more likely to move from the EVAL to FORM phase ($p = 0.006$), suggesting that they negotiated the behaviour of the robot to be programmed while evaluating the current programmed behaviour.

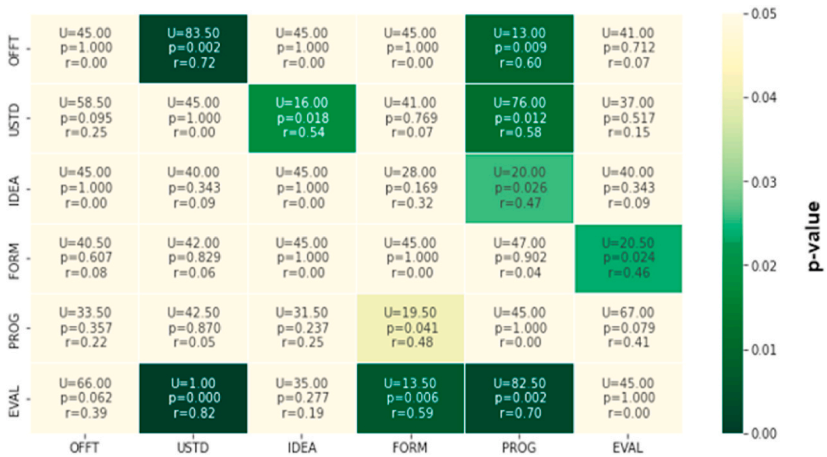

Fig. 7. Significant differences between the transition matrices of groups B and D. The boxes' colour varies from green to yellow, indicating a risk threshold from 1% to 5% (Mann-Whitney *U* test). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

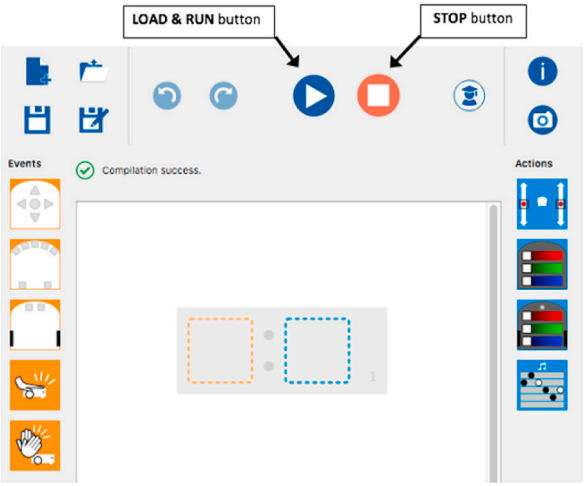

Fig. 8. The visual programming language Thymio VPL (Shin, Siegwart, & Magnenat, 2014) with the RUN button to upload the program on the robot and execute it, and the STOP button to stop the execution of the program.

4.2.1. The effect of immediate feedback on students' problem-solving strategies

The students in sector B (with *immediate feedback* and *no guidance* as an experimental condition) mainly adopted a programming and evaluating strategy (Figs. 5 and 6). This is a usual behaviour in ER (Chevalier & Giang et al., 2020) which refers to a "trial-and-error" strategy (Weintrop & Wilensky, 2015), i.e., a problem-solving method in which multiple attempts are made to reach a solution. They accomplished the task more quickly (Siegfried et al., 2017; Mayer, 2004; Biesta & Burbules, 2003) but to the detriment of developing all the cognitive processes necessary in CT (Chevalier & Giang et al., 2020). The *immediate feedback* makes it possible to act and react quickly (Smith & Lipnevich, 2018; Wise & O'Neill, 2009; Shute, 2008; Hattie & Timperley, 2007). Nevertheless, productive trial-and-error should bring "an iterative cycle of observation, hypothesis generation, hypothesis testing and evaluation of the solution" (Sullivan, 2008, p. 389). However, as highlighted in Fig. 6, there is little formulation (FORM) before programming, and therefore few hypotheses are generated. The trial-and-error strategy focuses on the action exerted directly on the robot rather than on evaluating the program's syntax that controls the robot's behaviour. This suggests little student anticipation whose PROG and EVAL phases are essentially dedicated to using the robot like a remote-controlled car. Thus, the students subjected to *immediate feedback* are in the immediacy of the action, which leads to a more reactive strategy, as opposed to the more deliberative strategy of those with *delayed feedback* (section 4.2.2). Moreover, immediate feedback generates a dependency on feedback (Schmidt & Lee, 2013). Students with *immediate feedback* start on average 5 min (Table 6) after the start of the mission to instrument both RUN and STOP buttons (Fig. 8) with minimalist programs to control their robot remotely. Consequently, the trial-and-error strategy (PROG-EVAL) seems to be due to this instrumentation to remotely control the robot (Rabardel, 1995) and a lack of agreement on formulating the robot behaviour before the execution. Indeed, in sector B, the students express very little agreement among themselves, and their verbalisations remain at the level of observation ("Oh no, he didn't turn …", "Yes, he made it!"). It seems that, in a remote collaborative robot programming mission, *immediate feedback* decreases group communication in favour of action: this leads to trial-and-error-based problem-solving strategies for 8-9-year-old students.

4.2.2. The effect of delayed feedback on students' problem-solving strategies

The students in sector D (with *delayed feedback* as an experimental condition) mainly adopted a programming and evaluation strategy (Figs. 5 and 6) but with interim transitions towards other phases. Students cannot get into this PROG-EVAL loop as it costs them time (30 s) when they click on the RUN button Fig. 8). Indeed, while waiting for the visual *feedback* of the remote execution of their program, students verbalise their incomprehension ("but why isn't he moving?") or anxiety ("Let's hope it works!"). They even anticipate what should be done ("you should move the robot backwards because it might get into the other robot"). Thus, they are going into the path USTD-IDEA-PROG-FORM-EVAL ("it's going to move forward and then it's going to turn around", "Come on, Thymio, do what you've been told to do: move forward and turn before the wall!"). Such a strategy (trial-and-error and reasoned simultaneously) refers to Merisio et al. (2021), who stated it is time and memory consuming.

In sector D, students formulate more and pool their strategy: the delayed *feedback* seems to lead them to (re)negotiate the tasks before pressing the RUN button. As a result, we observe a return to PROG without even having had EVAL, hence the FORM-PROG loop (Fig. 5). This delay forces the students to give more thought to a larger program's syntax and agree on the program sent. For example, in a group of two students, the frustration of waiting to see the programming result leads the student who programmed to reflect on his program and identify aspects to be improved. This change calls out to the other student, who then demands explanations, and they complete their program together.

Moreover, in the long, students are also tempted to "remote control" the robot to succeed in the mission by instrumenting the RUN and STOP buttons (Rabardel, 1995). Nevertheless, the delay prevents them from doing so, hence they are forced to proceed to other

instruments (including testing with the locally available Thymio robots). Indeed, while the RUN button keeps its function to start the program and thus to evaluate it, the STOP button becomes random: it does not allow to stop the robot properly because of the *delayed feedback*. The decision to "stop" the program is then not taken for granted because it can amplify its potential error that cannot yet be evaluated. Thus, the students are more anticipatory as they are forced to think and verbalise to pool their thoughts. They are forced to anticipate and communicate (Denner et al., 2014) and make a shared decision. As a result, the robotics environment's affordances coupled with the *delayed feedback* may trigger a FORM-PROG loop in addition to the PROG-EVAL loop already identified, resulting in a more productive trial-and-error strategy.

4.2.3. The effect of guidance on problem-solving strategies

Despite a lack of significant results in terms of normalised change under the "*guidance*" experimental condition (sectors A and C), viewing the videos allows us to identify similar problem-solving strategies. Under *guidance* and *immediate feedback* (sector C), students could better decompose the problem and identify the robot's starting position, i.e., out of the track line. Nevertheless, the instrumentation of immediate video *feedback* coupled with the VPL's affordances resulted in an unproductive trial-and-error strategy. Concerning sector A, it was *a priori* the most favourable learning conditions as students were subjected to *guidance* and *delayed feedback*. Nevertheless, results do not indicate any particular performance under these experimental conditions. The video viewing allows identifying a strategy close to the one deployed in sector D, without allowing the students in Sector A to achieve equal or higher CT scores than the students in Sector D. Further research is needed to understand better what is at stake in the interaction between *guidance* and *delayed feedback* in sector A. In addition, the 12 h class before the mission may have made the students (in all sectors) experts on the type of tasks proposed during the mission. Thus, guidance may not have been necessary for the students in Sector A to complete the mission. Nevertheless, it remains that their CT score is neither equal nor superior to those in Sector D. It would seem that guidance then had a counteracting effect on the strategies subjected to delayed feedback. Moreover, as noted by Clark et al. (2012), expert students can still learn with minimal guidance. As a result, the role of guidance at the beginning of the course, when scaffolding is still needed, should be verified in a future study.

5. Conclusion

In this study, we were interested in intervention methods that foster CT competence in the context of ER activities for 8-9-year-old students. Based on the literature, we identified *guidance* (with/without) and *feedback* (immediate/delayed) as two promising intervention methods to foster CT competence in classroom settings, in accordance with teaching practices. After a 12-h ER course with the Thymio robot, 66 students accomplished a collaborative remote programming mission under four distinct experimental conditions: *with guidance* and *with delayed feedback*; *without guidance* and *with immediate feedback*; *with guidance* and *with immediate feedback*; *without guidance* and *with delayed feedback*. Before and after this mission, pre- and post-tests were carried out, and the results showed the significant positive impact of using *delayed feedback* on the "Analysis" dimension of CT.

Moreover, based on the videos recorded during this mission, we observed and evaluated the different strategies implemented under experimental conditions showing a significant difference: the two types of *feedback* (immediate and delayed) do not lead to the same cognitive processes and learning outcomes: each brings a specificity but also has its limits, as it has already been shown in other studies on *feedback* (Hattie & Timperley, 2007; Shute, 2008; Smith & Lipnevich, 2018; Wise & O'Neill, 2009). Indeed, if one seeks to facilitate accomplishing the task or mission, then *immediate feedback* is favourable. However, this does not allow the students to develop more elaborate CT processes such as those described in the computational perspectives and practices dimensions (Brennan & Resnick, 2012). On the other hand, if one seeks to foster deeper CT processes, *delayed feedback* is more favourable. However, care must be taken not to fatigue and therefore disengage students throughout the mission with *delayed feedback*. In our case, the delay made sense because it was naturally included in the challenge of having a robot on Mars.

This study addresses the four recommendations of Lye and Koh (2014) by exploring a classroom-based intervention, focusing on computational practices and computational perspectives, examining the programming process, and analysing qualitative data. In particular, our results on *delayed feedback* help address their recommendations regarding fostering the CT's computational perspectives dimension. Consequently, we also address the recommendations of Tikva and Tambouris (2021, p.26) for clarity about factors that may affect CT acquisition.

However, this research has some limitations such as the small size of our sample and the type of guidance selected. The results about guidance with a worksheet are not significant, although guidance is described in the literature as a factor promoting learning (Kirschner et al., 2006, pp. 80-81). On the one hand, the medium chosen to convey guidance (via metacognitive questions) does not seem relevant to 8-9-year-old students. Perhaps a certain level of self-direction (i.e., taking sole responsibility for one's learning) is necessary to make good use of it. On the other hand, due to our design of experiment, an interaction effect between the effect of guidance and feedback (immediate and delayed) may have occurred. It seems that the guidance modality chosen here (the worksheet) does not offer sufficient incentive to break the "doing" and "think-doing" patterns brought on, respectively, by the trial-and-error appeal offered by immediate feedback and the cognitive load imposed by delayed feedback.

What type of guidance should the teacher provide to encourage student development of CT? The solutions to be explored may be *a priori* counterintuitive, as shown in this study with delayed feedback (due to the live streaming latency). This leads us to believe that any technical constraint can be perceived as a pedagogical opportunity. Future work should focus on the guidance factor to find an appropriate balance between scaffolding methods and consider the "Help me do it alone" duality raised by Montessori (1973). In addition, future studies should make it possible, on the one hand, to identify the types of errors during a creative computational problem-solving situation and, on the other hand, to propose the kind of adequate feedback for the student to overcome his errors.

Credit author statements

Conceptualisation, M.C., C.G., V.P., M.R., B.B., F.M.; Methodology, M.C., C.G., V.P., L.E− H., J.-P.P., C.A., M.R., B.B., F.M.; Formal analysis, M.C., C.G., L.E-H., V.P., C.A., B.B.; Investigation, M.C., C.G., E.B.; Resources, E.B. M.C.; Writing—original draft preparation, M.C.; Writing—review and editing, M.C., C.G., L.E-H., E.B., V.P., J.-P.P., C.A., M.R., B.B., F.M.; Data curation, M.C.; Project administration, M.C.; Supervision, B.B. and F.M.; Funding acquisition, F.M.; All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the Swiss National Science Foundation through the National Centres of Competence in Research Robotics (NCCR).

Declaration of competing interest

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the Coordination Committee for Educational Research of the Canton of Vaud (Switzerland) as well as the school administration and all teachers, students, and their parents for the participation in the study. Moreover, the authors would like to thank the people at LINE research lab for participating in preliminary research for this study.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compedu.2022.104431.

References

Ajjawi, R., & Boud, D. (2017). Researching feedback dialogue: An interactional analysis approach. *Assessment & Evaluation in Higher Education, 42*(2), 252–265.
Alimisis, D. (2019). Teacher training in educational robotics: The ROBOESL Project paradigm. *Technology, Knowledge and Learning, 24*(2), 279–290.
Aranha, A., & Gonçalves, F. (2012). The importance of video-feedback and instruction. *Journal of Physical Education and Sports Management, 3*(1). https://doi.org/10.5897/JPESM11.010
Atmatzidou, S., & Demetriadis, S. N. (2012). Evaluating the role of collaboration scripts as group guiding tools in activities of educational robotics: Conclusions from three case studies. July. In *2012 IEEE 12th international conference on advanced learning technologies* (pp. 298–302). IEEE.
Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems, 75*, 661–670.
Atmatzidou, S., Demetriadis, S., & Nika, P. (2018). How does the degree of guidance support students' metacognitive and problem-solving skills in educational robotics? *Journal of Science Education and Technology, 27*(1), 70–85.
Bahula, T., & Kay, R. (2020). Exploring student perceptions of video feedback: A review of the literature. November. In *Proceedings of ICERI2020 conference* (Vol. 9, p. 10).
Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, Article 100337.
Biesta, G. J., & Burbules, N. C. (2003). *Pragmatism and educational research*. Rowman & Littlefield.
Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. November. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65–72).
Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. April. In *Proceedings of the 2012 annual meeting of the American educational research association* (Vol. 1, p. 25). Vancouver, Canada.
Bruner, J. (1983). Play, thought, and language. *Peabody Journal of Education, 60*(3), 60–69.
Chen, Z., & Klahr, D. (1999). All other things being equal: Acquisition and transfer of the control of variables strategy. *Child Development, 70*(5), 1098–1120.
Chevalier, M., El-Hamamsy, L., Giang, C., Bruno, B., & Mondada, F. (2021). *Teachers' perspective on fostering computational thinking through educational robotics*. arXiv preprint arXiv:2105.04980.
Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem-solving. *International Journal of STEM Education, 7*, 41.
Clark, R. E., Kirschner, P. A., & Sweller, J. (2012). Putting students on the path to learning: The case for fully guided instruction. *American Educator, 36*(1), 6–11.
Coletta, V. P., & Steinert, J. J. (2020). Why normalised gain should continue to be used in analysing preinstruction and postinstruction scores on concept inventories. *Physical Review Physics Education Research, 16*(1), Article 010108.
Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education, 46*(3), 277–296.
Evans, C. (2013). Making sense of assessment feedback in higher education. *Review of Educational Research, 83*(1), 70–120.
Fabre, M., & Musquer, A. (2009). Les inducteurs de problématisation. *Les Sciences de l'Éducation pour l'Ère Nouvelle, 42*(3), 111–129. https://doi.org/10.3917/lsdle.423.0111
Guggemos, J. (2021). On the predictors of computational thinking and its growth at the high school level. *Computers & Education, 161*, Article 104060.
Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research, 77*(1), 81–112.
Henderson, M., Selwyn, N., & Aston, R. (2017). What works and why? Student perceptions of 'useful'digital technology in university teaching and learning. *Studies in Higher Education, 42*(8), 1567–1579.
Honomichl, R. D., & Chen, Z. (2012). The role of guidance in children's discovery learning. *Wiley Interdisciplinary Reviews: Cognitive Science, 3*(6), 615–622.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75–86.

Li, Y., Schoenfeld, A. H., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). On computational thinking and STEM education. *Journal for STEM Educ Res, 3*, 147–166. https://doi.org/10.1007/s41979-020-00044-w

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61.

Magill, R. (1993). Augmented feedback in skill acquisition. In R. N. Singer, M. Murphy, & L. K. Lemant (Eds.), *Handbook of research on sport psychology* (pp. 193–212). New-York: Macmillian.

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist, 59*(1), 14.

Mehrotra, A., Giang, C., Duruz, N., Dedelley, J., Mussati, A., Skweres, M., & Mondada, F. (2020). Introducing a paper-based programming language for computing education in classrooms. June. In *Proceedings of the 2020 ACM conference on innovation and technology in computer science education* (pp. 180–186).

Merian, T., & Baumberger, B. (2007). Le feedback vidéo en éducation physique scolaire. *Staps, 76*(2), 107–120.

Merisio, C., Bozzi, G., & Datteri, E. (2021). There is No such thing as a "trial and error strategy. February. In *Educational robotics international conference* (pp. 190–201). Cham: Springer.

Mondada, F., Bonani, M., Riedo, F., Briod, M., Pereyre, L., Rétornaz, P., & Magnenat, S. (2017). Bringing robotics to formal education: The Thymio open-source hardware robot. *IEEE Robotics and Automation Magazine, 24*(1), 77–85. https://doi.org/10.1109/MRA.2016.2636372

Mondada, F., Bonnet, E., Davrajh, S., Johal, W., & Stopforth, R. (2016). R2T2: Robotics to integrate educational efforts in South Africa and Europe. *International Journal of Advanced Robotic Systems, 13*(5). https://doi.org/10.1177/1729881416658165. London: SAGE.

Montessori, M. (1973). *From childhood to adolescence*. New York: Schocken books.

Özgür, A., Lemaignan, S., Johal, W., Beltran, M., Briod, M., Pereyre, L., & Dillenbourg, P. (2017). Cellulo: Versatile handheld robots for education. March. In *2017 12th ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 119–127). IEEE.

Papadakis, S. (2020). Robots and robotics kits for early childhood and first school age. *International Journal of Interactive Mobile Technologies (iJIM), 14*(18), 34–56. https://doi.org/10.3991/ijim.v14i18.16631 [S.l.] nov. 2020. ISSN 1865-7923.

Papert, S. (1980). *Mindstorms: Computers, children, and powerful ideas*. New York: Basic Books.

Patchan, M. M., & Puranik, C. S. (2016). Using tablet computers to teach preschool children to write letters: Exploring the impact of extrinsic and intrinsic feedback. *Computers & Education, 102*, 128–137.

Rabardel, P. (1995). *Les hommes et les technologies; approche cognitive des instruments contemporains* (p. 239). Armand Colin.

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education, 14*(1), 42.

Sapounidis, T., & Alimisis, D. (2020). Educational robotics for stem: A review of Technologies and some educational considerations. In *Science and mathematics education for 21st century citizens: Challenges and ways forward* (pp. 167–190). Hauppauge, NY, USA: Nova Science Publishers.

Schmidt, R., & Lee, T. (2013). Motor learning and performance, 5E with web study guide: From principles to application. *Human Kinetics*.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research, 78*(1), 153–189.

Siegfried, R., Klinger, S., Gross, M., Sumner, R. W., Mondada, F., & Magnenat, S. (2017). Improved mobile robot programming performance through real-time program assessment. June. In *Proceedings of the 2017 ACM conference on innovation and technology in computer science education* (pp. 341–346).

Smith, J., & Lipnevich, A. A. (2018). Instructional feedback: Analysis, synthesis, and extrapolation. In *The Cambridge handbook of instructional feedback* (pp. 591–603). Cambridge University Press.

Sullivan, F. R. (2008). Robotics and science literacy: Thinking skills, science process skills and systems understanding. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, 45*(3), 373–394.

Sweller, J., Kirschner, P. A., & Clark, R. E. (2007). Why minimally guided teaching techniques do not work: A reply to commentaries. *Educational Psychologist, 42*(2), 115–121.

Swinnen, S. P. (1996). Information feedback for motor skill learning: A review. In H. N. Zelaznik (Ed.), *Advances in motor learning and control* (pp. 37–65). Champaign, IL: Human kinetics.

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. *Computers & Education*, Article 104083. https://www.sciencedirect.com/science/article/abs/pii/S0360131520302815.

Van der Maren, J. M. (2014). *La recherche appliquée pour les professionnels: Éducation, (para) médical, travail social.* De Boeck Supérieur.

Vygotsky, L. S. (1978). In E. Rice (Ed.), *Mind in society: The development of higher mental processes*. Trans.

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. June. In *Proceedings of the 14th international conference on interaction design and children* (pp. 199–208).

Wise, A. F., & O'Neill, K. (2009). Beyond more versus less: A reframing of the debate on instructional guidance. In S. Tobias, & T. M. Duffy (Eds.), *Constructivist instruction: Success or failure?* (pp. 82–105). Routledge/Taylor & Francis Group.

Wisniewski, B., Zierer, K., & Hattie, J. (2020). The power of feedback revisited: A meta-analysis of educational feedback research. *Frontiers in Psychology, 10*, 3087.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem-solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89–100.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st-century problem-solving in K-12 classrooms. *TechTrends, 60*(6), 565–568.

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education, 141*, Article 103607.