# A Collaborative Learning Framework for Computational Thinking Development through Game Programming

Ângelo Magno de JESUS[1], Ismar Frango SILVEIRA[2]

[1]*Federal Institute of Minas Gerais/Cruzeiro do Sul University*
 *Afonso Sardinha st., 90, Ouro Branco – MG, 36420-000, Brazil*
[2]*Cruzeiro do Sul University*
 *Galvão Bueno st., 868 – Liberdade, São Paulo – SP, 01506-000, Brazil*
*e-mail: angelo.jesus@ifmg.edu.br, ismar.silveira@cruzeirodosul.edu.br*

**Abstract.** The Computational Thinking (CT) teaching approach allows students to practice problem-solving in a way that they can use the Computer Science mindset. In this sense, Collaborative Learning has a lot to contribute to educational activities involving the CT. This article presents the design and evaluation of a Collaborative Learning framework for the development of CT skills in students. To design the proposed strategy, several fundamental features of the Collaborative Learning concept of the literature have been studied and sketched. The strategy was applied to middle school students through a digital games programming workshop. Data were collected by three means: (1) collecting artifacts produced during activities; (2) recording of game programming sessions; and (3) applying a structured interview to students. The data analysis showed evidence that the strategy was able to mobilize Computational Thinking skills in addition to mobilizing collaborative skills in learners.

**Keywords:** computational thinking, game development, collaborative learning.

## 1. Introduction

Computer Science (CS) and the use of computers have shown over the years as a promising means for empowering science and education. According to Papert (1980), computers may affect the way people think and learn. In this sense, computers can be vehicles of powerful ideas and the seeds of cultural change. These devices can help people to build new relationships with knowledge that breaks with the traditional conception which separates Humanities from Natural Sciences and self-knowledge from both of these. However, the use of computers in the classroom must go beyond just receiving

instructions from the machine which means making the computer "teach" the learner (Papert, 1980). Educators need to apply a constructivist vision to these learning activities. In this context, Computational Thinking (CT) has a lot to contribute. CT is seen as a problem-solving approach so that the solution can be implemented on a computer (Barr and Stephenson, 2011). In this way, teaching activities based on CT should use and develop in students a set of CS skills.

However, the CT teaching should not be limited to technical features and computing methods, it is necessary to take into account areas of values, motivations, feelings, stereotypes and attitudes which include having students work with others to achieve a common goal or solution (Barr and Stephenson, 2011). Therefore, Collaborative Learning has a lot to contribute to this aspect.

In summary, collaboration or collaborative work can be understood, in general, as working together to achieve a goal (Martinez-Moyano, 2006). According to Rubin (2002), collaboration has relationships between people as its core principle. To perform a Collaborative Work, one should cultivate skills of relationship building and management. In the educational context, Vygotsky's social interactionist theory considers social interactions as the basis for people's cognitive development (Monereo and Gisbert, 2005). In this sense, Collaborative Learning procedures provide a means for students to learn new subjects while developing social skills (Monereo and Gisbert, 2005). It is excellent for conceptual learning, creative problem solving and developing academic language proficiency. Socially, it increases trust and cordiality. Also, it teaches team skills that can be transferred to many situations that go beyond the school context (Monereo and Gisbert, 2005).

Therefore, this article presents the design of a Collaborative Learning conceptual framework for the development of CT in students through digital game programming activities. The proposed approach is part of a teaching strategy that involves two other frameworks: problem-solving and agile-software engineering. In de Jesus and Silveira (2020) we describe the agile software engineering framework. In de Jesus and Silveira (2019) an overview of the teaching strategy is presented, but without considering final results, especially concerning the collaboration aspect. In this article, we will describe in-depth the collaboration framework that is the main element that composes the teaching strategy. Besides, the results achieved in this dimension will be presented and discussed in depth. The framework development took into account the mapping of collaborative learning features described in the literature; elements of widely adopted cooperative learning methods; and the outcomes of a systematic review of the literature regarding collaborative learning and CT and the application of teaching practices. The approach was evaluated through a game development workshop applied to 12–15 years old students from 7th and 8th grades of middle school. In this sense, data were collected by recording the programming sessions, performing structured interviews and collecting minutes of meetings held between student teams. The analysis of the recordings was performed through a social graph built from the interactions between learners during the implementation of the solutions.

## 2. Theoretical Background

### 2.1. *Computational Thinking*

CSTA and ISTE (2011) describe Computational Thinking as a problem-solving approach that highlights the integration of digital technologies with human ideas. This approach focuses on logical reasoning related skills, creativity, and critical thinking, while emphasizing ways to organize a problem so that a computer can aid in its solution. Barr and Stephenson (2011) argue that Computational Thinking can be understood as a problem-solving methodology that can be automated, transferred and applied across all subjects. Students do not become just tool users, but builders. According to the researchers, students should use a set of concepts (such as abstraction, recursion, and iteration) to process and analyze data, and to create real or virtual artifacts.

### 2.2. *Collaborative Learning*

Collaborative Learning is a methodology that converts heterogeneity (which can be found in any group) into a positive element that facilitates learning. This strategy enhances psychosocial interaction skills. These abilities include points of view acceptance, communication, negotiation, self-esteem, etc. Also, these skills are based on values such as mutual help and solidarity. It's known that the potential of interactions is a motor for meaningful learning (Monereo and Gisbert, 2005). Gokhale (1995) explains that the term Collaborative Learning refers to an instruction method. This method requires students to work together in small groups toward a common goal, at various performance levels. In this sense, learners are responsible for one another's learning as well as their own.

### 2.2.1. *Collaborative Learning Features*

Collaborative Learning Methods must have a set of features that distinguish them from traditional teaching approaches. Table 1 shows the characteristics of a Collaborative Learning group in contrast to groups from "traditional" teaching approaches, according to (Johnson *et al.*, 1984).

In addition to the elements described in Table 1, it is necessary to consider other features introduced in the literature. Group members must respect and highlight individual group members' abilities and contributions (Panitz, 1999)(J). Collaborative approaches should enable metacognition skills development (Lee, 1997; Torres and Irala, 2014) (K). Peers with approximately the same level should perform the same actions, which is called symmetry (Dillenbourg, 1999)(L). In collaborative situations, students need to work together so there must be a division of labor (Dillenbourg, 1999)(M). Finally, learners must be intrinsically motivated (Lee, 1997)(N).

Table 1

Collaborative Features according to Johnson *et al.* (1984)

| Cooperative Learning Groups | Traditional Learning Groups |
| --- | --- |
| (A) Positive interdependence | No interdependence |
| (B) Individual accountability | No individual accountability |
| (C) Heterogeneous | Homogeneous |
| (D) Shared leadership | One appointed leader |
| (E) Shared responsibility for each other | Responsibility only for self |
| (F) Task and maintenance emphasized | Only task emphasized |
| (G) Social skills directly taught | Social skills assumed and ignored |
| (H) Teacher observes and intervenes | Teacher ignores group functioning |
| (I) Group processes their effectiveness | No group processing |

### 2.2.2. *Collaborative Learning Methods*

Collaborative learning methods are common ways of organizing procedures and strategies to achieve collective goals related to learning. As noted earlier, collaborative learning requires a different environment from traditional ones, as teachers and students play new roles (Castro *et al.*, 2011).

- **Peer Tutoring:** this method consists of forming pairs of students with asymmetric relationships (one tutor and another tutored) and with a common goal, known and shared.
- **Jigsaw:** The central idea of the Jigsaw strategy is to investigate a theme from individual explorations that bring to the group the "pieces" for the assembly of the "jigsaw." A division-and-conquer process is applied to the theme, and each part of the division is initially explored by an "experts group". The contribution of each specialist member is essential for the team to learn the lesson (solve the jigsaw).
- **Group Investigation:** this approach understands the class group as a social community that researches a particular topic. Thus, the class-group acts as the scientific community, in which a theme of mutual interest is divided by teams into subthemes.

### 2.3. *Coding Dojo*

Coding Dojo is a programming learning technique based on agile development values. The main Coding Dojo goal is to act as a safe and non-competitive environment, in which everyone is allowed to learn and make mistakes and share their knowledge so that everyone can improve and become better at their craft (Heinonen *et al.*, 2013). A Coding Dojo is defined as a periodic meeting (usually weekly) organized around a programming challenge where people are encouraged to participate and share their coding skills with the audience while solving the problem (Sato *et al.*, 2008). Collaboration, through knowledge sharing and discussion of ideas, is one of the central principles of the Cod-

ing Dojo. This way, Pair Programming is one of the methods that the Coding Dojo uses to promote learning. One of the benefits of Pair Programming is that it allows constant knowledge sharing with the programmer's team (Cockburn and Williams, 2000). In this approach, two persons use a single workstation and act together to pursue a goal. The person who possesses the keyboard is often called the driver (or pilot) while the other person is called the navigator (or co-pilot).

### 2.4. *Computational Thinking and Collaborative Learning*

Computational Thinking demonstrates a close relationship with Collaborative Learning. Barr and Stephenson (2011) described CT values such as setting aside differences to work with others to achieve a common goal or solution; and knowing one's strengths and weaknesses when working with others. The authors also report strategies to benefit the CT learning experience, which includes the accomplishment of teamwork among students, with explicit use of negotiation (groups within the team working together to merge parts of the solution into the whole) and consensus-building (working to build group solidarity behind one idea or solution) among other elements. These are essentially collaborative aspects. The standards for Computer Science teachers, proposed by CSTA (2011), report that effective Computer Science teachers should support student collaboration. In this sense, the teachers should provide structured opportunities (like collaborative learning methods, for instance) for students to collaborate in the Computer Science field. These practices are performed to develop students' ability to provide, receive, and respond to constructive feedback in the design, implementation, and review of computational artifacts. Dybdal and Bjorn (2017) argue that strategies of moving beyond the keyboard and screen, including considerations of the socially-embedded context, are relevant for programming at various levels of education and participation.

Besides, it should be noted that systems development is usually a collective effort (Teles, 2004). The success of a software project involves the quality of the constant interactions between the members of the development team. Teles (2004) also argues that, in Computer Science courses, it may be interesting to prepare professionals who are more in tune with the human factors that affect the software design process. Pham and Pham (2011) report that when team members do not understand each other and nothing is done about it, the project almost always ends with problems as they perceive. The authors argue that one of the rules of engagement in software development is that everyone must depend on everyone to work as a team.

## 3. Research Methodology

The employed research method attempted to follow the subsequent steps: (1) perform a literature review regarding the main features related to Collaborative Learning; (2) perform a literature review regarding characteristics of Collaborative Learning

methods; (3) employ a systematic literature review on CT and Collaborative Learning (section 4); (4) design of the proposed framework according to the data collected in the previous steps; (5) perform a game development workshop to assess the framework as reported in section 7; (6) employ the data collection and analysis methods described in section 6. It should be noted that steps 5 and 6 were carried out simultaneously.

## 4. Systematic Literature Review

The purpose of the systematic review was to understand how collaborative learning was being used in Computational Thinking activities. The procedure adopted to conduct the systematic review was based on the method described by Kitchenham (2004). The full results of the systematic literature review can be found in de Jesus *et al.* (2019). Thus, the purpose of this section is only to report an overview of the employed method and outcome in summary. To carry out this task, the following research question was first formulated "*How does Collaborative Learning manifest itself in activities that use and stimulate Computational Thinking?*". From this core question, the search String was defined which was structured in the following format: "*computational thinking*" *AND* ("*collab*"* *OR* "*cooper*"*).

It's noteworthy that the string considered the varieties that could arise from the term "*collaborative*" to reach a larger set of studies. As some authors do not yet differentiate the terms of Collaborative Learning and Cooperative Learning Panitz (1999), the word "*cooperative*" was also considered in the search string.

The research has been restricted to the last 5 years since it's a constantly emerging field and it's expected to obtain current studies that consider the knowledge already produced by the previous work. The search was performed in two important databases:

- CAPES Journal Portal (CAPES, 2017): This is a site that allows access to several international journals in multiple knowledge domains. The publication's impact factor, according to the Journal Citation Reports of the Institute for Scientific Information (ISI), is one of the main criteria adopted for the inclusion of journals on this data basis CAPES (2017).
- ACM (ACM, 2017): This base was chosen because it was the institution in which the term Computational Thinking gained popularity among researchers. It's a base that aggregates several articles in many research tracks in Computer Science.

Articles were first filtered by the abstract. In this stage, papers in which the concepts of Collaboration and Computational Thinking were approached very superficially. Studies that raised doubts about the depth of addressing key concepts or about their contribution to the research question were further refined by consulting the articles' full text.

Once the articles were selected, in order to identify trends, advances and limitations in this research track, a series of classifications was performed. The following are the defined classifications: (1) Empirical/Conceptual; (2) Target Audience; (3) Research Instruments; (4) Collaborative Model Performed; (5) Model of Activities Developed; (6) Developed Artifacts; (7) Technologies Used in Activities; and (8) Conceptual Work Model.

The procedure involved 523 articles retrieved from the databases. After filtering, 37 studies were defined as appropriate (according to the research question) and underwent the proposed analysis and classification. One of the main gaps found, among others, was the lack of studies that explore more structured collaborative learning approaches or employ a set of collaborative features from the Education literature. Therefore, the design of the proposed framework aimed to overcome this sort of limitation.

## 5. The Collaborative Learning Framework

The theoretical background section presented some of the Collaborative Learning methods. However, these procedures have been designed to be very general. The Computational Thinking field has its own characteristics that may not be considered in the best way by the direct employment of these methods. Therefore, it's necessary to develop an approach that takes into account the possibility of building and testing technological artifacts, proposing solutions to problems as algorithms, using abstraction among other elements that make up the Computational Thinking mindset. Despite this, these procedures have strong collaborative foundations that have already been validated in the literature. Thus, these principles were observed in the proposed framework:

- **Peer Tutoring:** the responsibility to teach each other, which in the proposed framework is represented by Pair Programming (as well as in the Coding Dojo).
- **Jigsaw:** activities that must be solved in the groups and each learner must contribute. Sense of belonging to a team that must achieve a goal.
- **Group Investigation:** the possibility of participating in the choice of the theme (choice of the game to be implemented) and having a goal to be faced. Reflection on teamwork.

Some values and elements were also based on the Coding Dojo approach, especially the practice of Pair Programming. In this way, the framework uses an adaptation of this approach also involving time-boxed rounds of a period of 5 minutes.

Also, the main collaborative features analyzed in the literature, as presented in the Theoretical Background section, were enumerated so that they could be taken into account in the framework design: (A) Positive interdependence; (B) Individual accountability; (C) Heterogeneity; (D) Shared leadership; (E) Shared responsibility for each other; (F) Task and maintenance emphasized; (G)Social skills directly taught; (H) Teacher observes and intervenes; (I) Group processes their effectiveness; (J) Respect and highlight the individuals' skills and contributions; (K) Metacognition's skills development; (L) Symmetry; (M) Division of labour-work together; and (N) Intrinsic motivation.

Fig. 1 illustrates the steps of the collaborative learning approach. Steps from 1 to 6 are performed iteratively. Steps 7, 8 and 9, although important, are optional and do not need to be performed exactly in the proposed order, it is up to the facilitator to define when the group will be mature enough to carry out these activities.

Table 2 shows the responsibilities of the facilitator and the responsibilities of the learners at each main stage of the proposed framework. The table also illustrates the
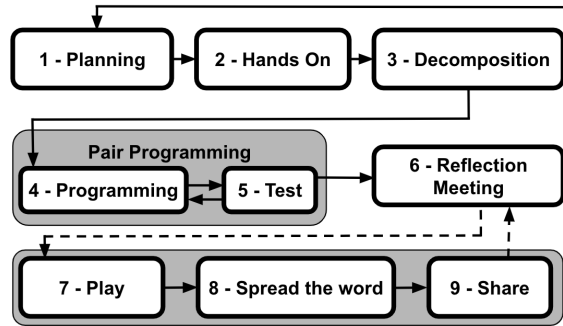
Fig. 1. Steps of the proposed collaborative learning strategy.

Table 2

Members' responsibilities in the main steps of the framework

| Step | Students' Role | Facilitator's Role | Collaborative Features |
|---|---|---|---|
| 1 | Students define the game to be implemented or a new scope for the game through brainstorming. | Ensures the project feasibility. Mediates conflicts. Ensures that all students are giving ideas and being heard. | (A), (D), (G), (H), (M) and (N) |
| 2 | Learners watch and replicate the teachings on computers. They must pay attention to the concepts presented so that they can contribute to the whole group when they start the implementation steps. | Explains the basic concepts of Programming and Game Development that will be involved in the next activities. | (B) |
| 3 | Students will decompose the game implementation into smaller problems. They will create stories that represented game features to be implemented. Each story will be written on a post-it and fixed on a Kanban board, which will be used to guide the progress of the group. | Calls for student participation. Ensures that everyone is being heard and is contributing. Questions students about the game to form new stories. | (A), (D), (F), (G) and (M) |
| 4 | Perform Pair Programming with time-boxed rounds. | Monitors the rounds and the exchange of pairs, ensuring that everyone has the opportunity to participate in the development. If necessary, give recommendations verbally or prepare and deliver a printed tips guide to assist students in solving problems. Assists in understanding the solution by code debugging. | (B), (E), (G), (J), (K) and (M) |
| 5 | Perform tests. This step must take place in parallel with the previous one. | Same as the previous step. | Same as the previous one |
| 6 | Students should hold a meeting to reflect on collaboration in the group during the activity and on the progress of the project. | Delivers a model of minutes with the items to be discussed at the meeting. Let students speak freely. Uses the information discussed at the meeting to help solve social conflicts and other group problems. | (A), (D), (F), (G), (I), (K) and (N) |

Table 3

Members' responsibilities in the additional steps of the framework

| Step | Students' Role | Facilitator's Role | Collaborative Features |
|------|---------------|-------------------|------------------------|
| 7 | Have fun interacting with the game. | Just makes sure everyone is interacting. | (N) |
| 8 | Present the game. Invite and teach people to interact with the game. | Provides and organizes the presentations. | (A), (D), (N) and (M). |
| 9 | Share the source code on an open repository. | Teaches students about open source culture. Supports students in sharing the project. | (N) |

collaborative features that were taken into account at each step. Table 3 shows the responsibilities of each member in the additional steps of the framework. As described in Gokhale (1995), in the proposed framework it is recommended to form small teams concerning between 2 to 4 members. All listed collaborative features were considered in the proposed processes, except for features (C) heterogeneity and (L) symmetry. As these are features that depend on the group's composition, the facilitator needs to try to observe these elements in a pre-execution step where the students are forming the teams.

## 5.1. *Facilitator Actions to Promote Collaboration*

As previously described, the teacher must assume the activity facilitator role, this means that the approach must be learners centered. In this sense, the facilitator must act to empower collaboration between group members. Some students tend to dominate the activity by making all decisions, while others are overshadowed (de Walle, 2009; Cohen and Lotan, 2014). In this way, the facilitator's actions must also be directed to minimize this domain. The following are some attitudes that can assist in interactions between learners:

- **Steps 1 and 3:** The facilitator must ensure that all students are participating. He/She can do this by assuming a questioning posture, guiding students in exchanging ideas. It is important to note that in introductory classes, students are unlikely to know how to clearly define a digital game. Besides, some students may be repressing themselves for not knowing other students or even for shyness. In this sense, the facilitator must question the students, looking to compose a story narrative for the game. To encourage discussion, he/she may ask "*What happens if character X touches object Y?*", And seek the participation of others by asking targeted questions, for example, "*Lorene, what should happen? ..what do you think? ... and you Michael?*". When students feel that everyone has something to say and are listened to carefully, students with high status tend to less dominate the group (Cohen and Lotan, 2014). These steps require this kind of action by the teacher as the students need to be prepared for the collaboration (Cohen and Lotan, 2014) that should occur in the next steps. As new development process

iterations are carried out, the group will become more autonomous and the facilitator will be able to decrease his/her performance in these stages.

- **Steps 4 and 5:** in the steps that correspond to the pair programming, the facilitator should let the students work on the problem so that the group has autonomy and responsibility for the task (Johnson *et al.*, 1984). However, some students with a dominant profile manage not to want to pass control of the computer to another learner. In this sense, the facilitator must carefully monitor the exchange of peers according to the set time. Also, doubts may arise and students may be stuck. In this sense, the facilitator should assist the group without giving ready answers, but encouraging them to discuss ideas and try new strategies. Some tips and suggestions should be passed on to the learners, but they should never represent a complete solution or prevent the team from using strategies to reach a solution. The idea is to direct the group to its own resources (Cohen and Lotan, 2014). Finally, the facilitator must also observe the group behavior, the information collected will be useful in the next step.
- **Step 6:** In this step, the facilitator must hand over a model of minutes (which will be described later) and let the students freely discuss each item of this document. In this way, later, the facilitator will be able to evaluate the information described to identify social interaction problems between group members or progress gaps. He/she can also check for any inconsistency between what was reported and the team's behavior that was observed in the previous step. If relevant problems are identified, the facilitator can assist the group in solving them.
- **Step7:** The facilitator should encourage fun, ensuring that everyone can interact with the game. These moments of play can increase the friendship bonds between the members of the group or at least can ease tensions between them.
- **Step 8:** The facilitator can search for technology events, fairs and conferences to present the project and invite the audience to interact with the game. Presenting the game together can give a sense of purpose to the team.
- **Step 9:** In this step, the facilitator should explain about free software and the importance of making the project available to other people. The act of sharing the project openly can teach about the importance of collaboration.

## 5.2. *Employed Pair Programming Approach*

As groups can be composed of more than two students, in each turn, one student assumes the pilot role and the others assume the role of co-pilot. All students can give ideas freely regardless of who is in control of the computer. It is up to the pilot to decide whether or not to implement the ideas proposed by the peers. Turns are managed by employing a timer. As described, the facilitator can choose to deliver a printed guide with information that can assist with tasks. If this happens, there will be a definition of another role: the guide. The student who assumes the guide role will be responsible for studying the information described and passing it on to his/her peers. This new role can assist the participation of learners who feel less at ease to contribute as co-pilots.

## 5.3. *The Reflection Meeting*

The reflection meeting is an opportunity for the group to sincerely reflect on their performance. Thus, this meeting must be documented to assist in self-monitoring and future reflections. Therefore, in order to perform this task, a meeting minute must be produced. The template of the minutes should address the following issues to be discussed: "(1) *Was the goal achieved? If not, why?* (2) *What problems were faced? What needs to be improved in the group interaction?* (3) *Is the game (becoming) fun?* (4) *What can be done in the next session?*".

## 5.4. *Curriculum Planning*

The proposed collaborative learning framework employs flexible content planning. In this way, the content to be covered in a workshop can be extended as students' progress in their knowledge. Thus, it is up to the facilitator to carefully monitor the activities to understand how the students are evolving to propose the learning of new concepts. Despite this, the facilitator needs to plan the essential knowledge that the approach should address. In this sense, it is the teacher's role to define the minimum content to be addressed and to include each curricular item in the game development process at the best time and opportunity according to the students' performance. As game development can involve several concepts even in initial activities, the facilitator can plan activities, according to the tasks defined by the students in stage 3 (Decomposition), providing a guide with lots of information for resolution in the initial programming sessions. As the students evolve, the facilitator can simply increase the reasoning load, decreasing the available information in the next guides.

## 6. Data Collection and Analysis

Data collection to assess student collaboration was carried out by recording the pair programming sessions, collecting the meeting minutes and performing structured interviews. The recording data was used to build social graphs representing the interactions between students. The interview data were evaluated quantitatively and mainly qualitatively. Participant observation was also applied to support the assessment of the collected information.

To analyze the recordings, we have adopted the approach described by Goos and Galbraith (1996) and Goos *et al.* (2002) employed to investigate metacognitive strategies in a collaborative mathematical problem-solving. This approach intends to highlight major strategic decisions and assess the quality of the decisions per se. As described in Goos and Galbraith (1996), the students' verbal protocols were parsed into macroscopic episodes which represent a session (periods of time) where learners are engaged in distinctive kinds of problem-solving behavior. Each student's complete speech turn was

named Move. To analyze the interpersonal strategies and contributions of learners working collaboratively, the authors propose a kind of classification:

1. New information points were subdivided into two types:

   - Points where previously overlooked or unrecognized information came to light (abbreviated as NI).
   - Points where the possibility of using a new procedure was mentioned (abbreviated as NP).
   - The NI/NP's were classified further according to: (1) Who initiated the NI/NP; (2) How relevant the NI/NP was to the task; (3) The nature of the response to the NI/NP (ignore, reject, accept...).

2. Local Assessments (LA's) of a particular aspect of a solution were classified according to who made the assessment, and the function of the assessment: (1) knowledge (assessing what is known/not known); (2) task difficulty; (3) procedure (checking accuracy of execution, assessing relevance or usefulness); and (4) result (assessing accuracy or reasonableness).

3. Global Assessments (GA's) of the general state of the solution were also made.

Based on the identification of the collected episodes and metacognitive Moves, Goos and Galbraith (1996) present a table of metacognitive decisions. This table compiles the activity into a sequence of correct or incorrect actions taken to solve the problem and the response of group members to them. In the context of this research, this table was built for each pair programming session performed. From this, a social graph that represents the connections between the students' Moves was built. The resulting targeted graph allows the identification of patterns of collaboration and problem-solving. Therefore, we could verify by node connections whether students are participating in the solution by proposing new ideas and procedures.

Due to the nature of Computational Thinking activities having some peculiarities concerning generic mathematical problem-solving activities, it was necessary to make some adaptations in the procedure. The adaptations were based on the following principles:

- Not only verbal protocols were considered. The nature of the game development and programming activity must be performed through technological artifacts, the interaction with such artifacts can also be seen as a means of communication.
- The episodes related to the problem-solving framework adopted by Goos *et al.* (2002) since it is not in the interest of this study.

The New Information was defined as follows:

- New Procedure (NP): straight procedures for inserting new instructions in the algorithm or re-arranging these instructions generating new algorithms. As informed, a student can propose an NP verbally, gesture or even just by adding an instruction or changing the solution steps (algorithm) through direct manipulation when he/she is playing the pilot role.
- New Idea (NI): it is related to students' higher abstract ideas for solving the problem and that does not represent a "concrete" implementation procedure, like the NP. In this sense, NI's are related to proposing new algorithms at a high level

of abstraction (in natural language); discover additional information that would assist in solving or understanding the problem; definitions of hypotheses about instructions; and identify elements of the source code that could assist with the solution.

It should be noted that New Information (NI/NP) was only registered once. This means that there can be no repetitions of the same NI/NP during the development sessions.

As in Goos *et al.* (2002), responses to new information (NI/NP's) were also registered. Besides, classification for each answer identified in the student's Moves table was carried out in order to analyze how students are responding to the new proposed information. In this sense, this makes it possible to analyze the Move graph that will be described later. This classification system is described following: (1) Agree (a) – as defined in Goos *et al.* (2002); (2) Disagrees (d) – as defined in Goos *et al.* (2002); (3) Ignores (ig) – as defined in Goos *et al.* (2002); (4) Implements (im) – in the context of this study, when a learner implements new information, it has the same meaning as agreeing.

Additional responses that reinforce proposals and implementations for New Information have also been noted: (1) Defends (de) – the student defends the proposed New Information; (2) Stress (s) – the learner insists on deciding/implementing the new information; (3) Explains (ex) – the student describes his/her idea better or explains how to program (code) it; (4) Help (h) – the learner asks for help to understand or implement the New Information.

As well as Goos *et al.* (2002), we also classify assessments of the solution status. Some of the evaluations were used by the mentioned authors, others were proposed in this study to fit the nature of the activity.

- **Local Assessment (LA):** an assessment of a particular aspect of the solution – (1) Usefulness of the procedure: assess whether a previously used instruction, algorithm or information is really contributing to the solution; (2) Sense of results: evaluate whether the solution is working while performing the tests; (3) Bug noticing: observe that something unexpected happened during the test. (4) Checking solution: assess whether something is missing or has something out of place in the algorithm. It also involves checking (trying to understand) the logic of the solution implemented up to that point; (5) Behavior analysis: analyze the result of the solution (the game in operation) to try to understand the problem and/or the solution; (6) Sub-solution missing: observe if punctual functionality (feature) is missing in the solution of the problem; and
- **Global Assessment (GA):** concerns the assessment of a development project as a whole. The GA's contain only one classification item: Missing Solution, which is concerned with observing whether complete solutions are missing to achieve the project objective.

In order to visualize the interactions, we set up a graph from this table. The vertices represent the Moves or NI/NPs. Therefore, each student's Move is represented by a vertex and labeled with an initial letter of the student name and a sequence number (as regarded in the Moves table). When a student proposes an NI/NP, an edge is directed from the vertex of its Move to the NI/NP vertex. The NI/NP vertex is labeled with a *V* (correct)
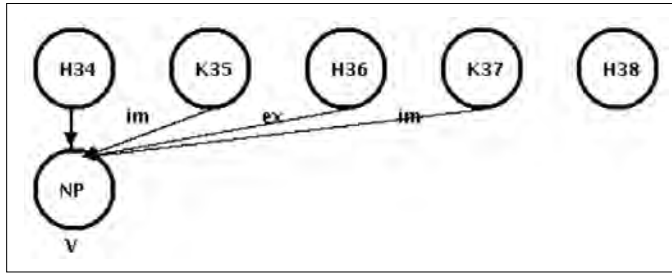
Fig. 2. Part of a social graph designed by the proposed approach.

or *X* (not correct) depending on its utility for the solution. Responses are represented by directed edges from the responses Moves for the NI/NP node. The responses were labeled according to the defined response classification system described previously. Fig. 2 shows an example of part of a social graph designed by the proposed approach.

To perform the analysis of the Move graph, social network analysis methods were adopted. Typically, social networks can be modeled in graph arrangement. This kind of approach was adopted since the students' interaction during the problem-solving activity can be modeled in the form of an interaction scheme of online social networks by the method proposed in Goos *et al.* (2002). Many studies explain that visual representation of data as a graph or sociogram allows researchers to uncover patterns that might go undetected. Also, a graph could be represented as matrices that are an alternative way to represent and summarize network data (Wasserman and Faust, 1994). In this kind of representation, actors (a social entity as a person, a person post, etc) are vertex/nodes and relations between them are edges.

According to Saltz *et al.* (2004), the simplest and most intuitive metric to a student-focused online class, is the student degree centrality, described by Wasserman and Faust (1994) and Karnstedt *et al.* (2010). The centrality measure for an actor (a student Move in this context) should be the degree of the node, $d(n_i)$. Formally, the centrality $C_D(n_i)$ is defined as:

$$C_D(n_i) = d(n_i) = x_{i+} = \sum_{i=0} x_{ij} = \sum_{j=0} x_{ji} \tag{1}$$

where $x_i$ is the $i_{th}$ edge.

An important aspect of this measure is that it depends on the size of the group. So, a proposed standardization measure is described by Wasserman and Faust (1994):

$$C'_D(n_i) = \frac{d(n_i)}{g - 1} \tag{2}$$

where $C'_D$ is independent of $g$ (group size), and thus can be compared across networks of different sizes.

Also, in the sense of degree centrality measure, two calculations could be performed: the $InDegree$ of a node $d_i(n_i)$ that is the number of nodes that are adjacent to $n_i$, and the $OutDegree$ of a node $d_o(n_i)$, that is the number of nodes adjacent from $n_i$. Saltz *et al.* (2004) apply the $OutDegree$ counting the number of messages sent by the student, and the $InDegree$ counting the number of replies to the messages posted by the student. Similar to the authors, we use these measures as follows:

- $InDegree$: number of Move responses received by a students' new information (NI/NP).
- $OutDegree$: number of responses given by a student's Move.

These measures help to observe the students' participation in the problem-solving process. For example, NI/NP student Moves with a high $InDegree$ can represent a strong iteration among learners around a student's New Idea. We also define High Centrality Degree ($HCD$) nodes, as follows:

$$HCD = \{x | x \in G \land sid(x) > 1\} \tag{3}$$

where $G$ is the general nodes set and $sid$ is a function that returns the standardized $InDegree$ of a given node.

In this way, a node is considered a High Centrality Degree ($HCD$) if it has the number of replies greater than the number of group members (except the author). This was done in order to check the existence of Moves that were able to generate greater interactions between students.

To assess the data of each student, it was necessary to define the *Breakeven Point*, which is the ideal proportion for all learners to contribute equally to the activity. This value can work as a common point of comparison to check the student's level of contribution for a given aspect. In this sense, the *Breakeven Point* ($BP$) for a given evaluated aspect is defined as:

$$BP = \frac{1}{g} \tag{4}$$

It should be noted that in the data analysis, replies labeled as *ig* (ignores) and replies made by the same author of the NI/NP Move were disregarded. This was done to achieve greater consistency in the results.

A fundamental element for CT development concerns the ability to solve problems. In this sense, teaching approaches should allow students to learn and apply different strategies and heuristics for problem-solving (de Walle *et al.*, 2016). Therefore, learners' Moves (speeches) were also analyzed in order to identify the use of heuristics classes and consequently assess their evolution in problem-solving skills. However, it is important to note that discovering strategies applied by learners is an extremely hard activity, since people may know how to solve problems, but they are not always able to verbalize or describe their actions. This is a typical aspect of all procedural knowledge (del Puy

Pérez Echeverría and Pozo, 1998). In the context of this study, learners can assist in solving the problem by proposing new relevant information, but without expressing which strategy he/she internalized to generate this information. Therefore, the demonstrations of the use of heuristics, that emerged during the activity, were reported. To perform the proposed analysis, classes of heuristics were defined to fit the development activities proposed by the framework:

- **Trial and error:** acts in which students suggest making attempts to check a procedure and go back if it demonstrates to be incorrect. Source: da Ponte *et al.* (2007).
- **Abstraction of a solution:** acts in which learners verbally describe an algorithm at a high level of abstraction (spoken language) trying to propose solutions. Sources:de Walle *et al.* (2016); CSTA and ISTE (2011); da Ponte *et al.* (2007).
- **Pattern recognition:** student suggests observing, comparing and/or reusing the source code of another solution similar to the algorithm to be developed. Source: CSTA and ISTE (2011).
- **Hypothesis generation:** acts in which students build a specific hypothesis about an instruction or information to develop the algorithm (solution). Source: de Walle *et al.* (2016).
- **Discovery through results:** acts in which the student analyzes the game's behavior during the test to try to discover new information for the solution.

The documents resulting from the reflection meetings (step 6 of the proposed strategy) were collected. Thus, through these artifacts, we analyzed and verified whether the group was managing itself in order to reach its goals and its relation to success/failure in the proposed activities. The employed analysis evaluated the dimensions of Progression and Collaboration of the strategy. For each report of the minutes, it was verified: (1) In the collaboration dimension, were the students able to identify and understand the problems of social interaction that occurred in the group?; (2) In the progress dimension, were the students able to identify and understand the problems in the proposed solutions that could hinder the group's progress?. The reports were validated with the reality experienced by the group by the participant observation and with the support of recorded students' Moves.

The interviews were applied with the support of a rubric based on Randall and Rusk (2009) and ReadWriteThink (2003). In this way, the rubric addresses four aspects of the learning strategy as a whole: Collaboration, Process, Programming and Project Design. Each aspect addressed in the rubric was an interview question that asks the participant to give their opinion. Learners were also invited to make a general assessment of the activity as a whole. This article address only the aspects of collaboration and process because they are directly related to the purpose of the evaluated framework. Table 4 shows the rubric for the two aspects evaluated. It is important to highlight that, for each dimension, the student was also invited to make extra considerations in order to collect qualitative data.

It should be noted that the activities were performed as extra-class practices and won't assign any value to the school grades. Students participated as volunteers, which may have corroborated fair answers.

Table 4

Rubric for guided interviews (Collaboration and Process aspects)

| Category Extra Considerations | Beginning | Developing | Proficient | Exceptional |
|---|---|---|---|---|
| Collaboration | - Contributes information to the group only when prompted.<br>- Rarely respectfully listens, interacts, discusses, and contributes to the group, helping the group to achieve a consensus. | - Contributes information to the group with occasional prompting or reminding.<br>- Sometimes respectfully listens, interacts, discusses, and contributes to the group, helping the group to achieve a consensus. | - Contributes knowledge, opinions, and skills without prompting or reminding<br>- Usually respectfully listens, interacts, discusses, and contributes to the group, helping the group to achieve a consensus. | - Consistently and actively contributes knowledge, opinions, and skills without prompting or reminding.<br>- Consistently and respectfully listens, interacts, discusses, and contributes to the group, helping the group to achieve a consensus. |
| Process | - Student did not get involved in design process.<br>- Did not use project time well and did not meet deadlines. | - Student tried out the design process.<br>- Used project time well sometimes and met some deadlines. | - Student used design process (stated problem, came up with ideas, chose solution, built and tested, presented results).<br>- Used project time constructively, met dead-lines. | - Student made significant use of the design process.<br>- Used project time constructively, finished early or added additional elements. |

## 7. Empirical Study

The experiments were conducted with students from two public schools in Ouro Branco, Brazil. In this way, students (aged 12–15 years old) from 7th and 8th grades of middle school from both institutions were able to register to participate in the game development workshops offered. Participation in the workshop was completely voluntary. The learners who signed up should present the Informed Consent Form signed by their legal sponsor.

The basic game development workshop's curriculum was inspired by the curriculum proposed by Code.org (2019) in Unit 3 – Interactive Animations and Games of the Computer Science Discoveries for 6–10th grade students course. According to Code.org (2019) the course was mapped to CSTA (Computer Science Teachers Association) standards. However, some arrangements were made, due to the differences in teaching approaches, workload (class/hour), students' background and tools adopted. All groups opted to develop a platform-type action game because this sort of game is in fashion among them. In this sense, the following summary schedule was adopted:

(1) Introduction to CS concepts.
(2) Sequence of commands and user input.
(3) Decision structures.
(4) Loops.
(5) Variables and operations.
(6) Procedures and additional topics.

The concepts were studied according to the evolution of the development of the proposed game. The Scratch environment was chosen as a base tool for carrying out the workshops. This technology was chosen due to its popularity and possibilities for creating interactive animations and digital games in an easy way (MIT Media Lab, 2019).

The workshop took place by weekly meetings from May to October 2019, with a break in June. A total of 41 meetings were held, 15 of which involved Pair Programming sessions. In some meetings, only one student from the group attended, which delayed the progress of activities.

At the schools, 15 students signed up to participate in the game development workshop, and 11 of them went to the end. The learners were between 12 and 14 years old. To perform the collected data analysis, fictitious names were assigned to the students to maintain their anonymity and privacy. The groups were created according to the time restrictions of the students themselves. Besides, the affinity between students was also considered, as it is a motivational circumstance for student participation. This may have limited the preservation of the symmetry collaborative feature in the groups, as defined by Dillenbourg (1999). The composition of the groups is listed as follow (only students who participated in any Pair Programming session were taken into account):

- **Group 1:** 1st formation – José Pedro, Eduardo and Darlene; 2nd formation – José Pedro, José Vinicius and Paulo;
- **Group 2:** 1st Formation – Michele, Kamila and Humberto; 2nd Formation – Michele and Kamila.
- **Group 3:** 1st Formation: Anália, Bruna, Naira and Iago; 2nd Formation: Anália, Bruna and Naira.
- **Group 4:** Humberto and William (Iago participated in defining the game and the tasks).
- **Group 5:** Naira and Kamila.

## 8. Results

### 8.1. *Participant Observation*

In this section, we will describe the main information raised in the participant observation. These reports are supported by the session's recordings and, consequently, by the students' Moves description made to draw the social graphs. The information described in the participant observation was compared with the data collected and evaluated by the other research instruments in order to reduce the subjectivity of the analysis. In this way, we describe the profile raised of each student according to the workshop facilitator's understanding.

According to Cohen and Lotan (2014), a way to evaluate group work activities is through observing the performance of learners with a low academic background (status). In this way, the participant observation sought to identify the students' background and analyze the participation of these students throughout the process. It is

important to mention that the background is not related to the capacity to learn, but rather to the educational basis that the student previously had access to. Therefore, the following low academic background learners were identified concerning their group colleagues: Anália, Paulo, Darlene, Kamila (concerning Humberto) and William (concerning Humberto).

As explained earlier, when working in groups there is always the possibility that a dominating student will lead the others (de Walle, 2009; Cohen and Lotan, 2014). This sort of dominant attitude can limit the participation of other students, especially those with low academic status. In this sense, the researchers also identified students who have the profile of activity dominators. Therefore, the following students were classified with this profile: Eduardo, Naira and José Vinícius.

In a general context, the researchers observed that:

- In the first stage – planning to define the game – the students showed that they wanted to implement games with elements that were "in fashion" between them. There was also a tendency for learners to like the same characters to be included in the games.
- Students, even with less elevated backgrounds, demonstrated to participate more in the problem-solving activities when they assumed the pilot role or also sometimes as a guide.
- Students with dominant profiles tended to be more individualistic and assumed the role of pilot, but when they were not playing this role, they showed to be more collaborative.
- In general, students showed enthusiasm to participate in the process of developing the game as a whole. Even students with a lower background were actively engaged in the development stages. Despite this, some students missed many workshop meetings.

## 8.2. *Results of Social Graph Analysis*

Table 5 shows the average proportions of Moves proposed by each student concerning the pairs. The learners' fictional names were abbreviated to their initials. The BE field denotes the average break-even point calculated from each Pair Programming session in which the learner participated. Special attention from the analysis was directed to students with lower academic status than the others. These learners could not actively participate in the project and generate collaboration gaps, especially concerning students who have a dominant profile (Cohen and Lotan, 2014). A gap situation is exemplified. Humberto demonstrated to have a much higher background than his colleagues. This fact can be noticed by his high proportion of NI/NP Moves. Thus, during session 5 with the G2 group, the student came to propose 100% of new information and new procedures. This session represented the greatest lack of collaboration. In this situation, when Michele assumed the pilot role, there was no interaction between the student and Humberto through the generation of new ideas (Michele reached an *OutDegree* equal to zero in this session).

Table 5

Result of Social Graph Analysis for each student – the values are computed averages

| Student | # Sessions | NI/NP | OutDegree | InDegree | GA/LA | Regular | BE |
|---|---|---|---|---|---|---|---|
| H | 6 | 0.78 | 0.28 | 0.72 | 0.57 | 0.54 | 0.47 |
| W | 4 | 0.27 | 0.64 | 0.35 | 0.40 | 0.44 | 0.50 |
| K | 4 | 0.27 | 0.69 | 0.31 | 0.51 | 0.43 | 0.46 |
| M | 4 | 0.34 | 0.40 | 0.34 | 0.36 | 0.43 | 0.46 |
| A | 3 | 0.09 | 0.32 | 0.13 | 0.14 | 0.20 | 0.28 |
| N | 4 | 0.47 | 0.30 | 0.48 | 0.35 | 0.35 | 0.33 |
| B | 3 | 0.42 | 0.30 | 0.33 | 0.38 | 0.36 | 0.28 |
| I | 2 | 0.18 | 0.11 | 0.18 | 0.28 | 0.26 | 0.25 |
| P | 2 | 0.03 | 0.34 | 0.01 | 0.15 | 0.29 | 0.33 |
| JV | 2 | 0.68 | 0.47 | 0.24 | 0.40 | 0.29 | 0.33 |
| JP | 3 | 0.33 | 0.19 | 0.77 | 0.47 | 0.44 | 0.33 |
| E | 1 | 0.55 | 0.40 | 0.20 | 0.50 | 0.35 | 0.33 |
| D | 1 | 0.00 | 0.40 | 0.00 | 0.00 | 0.20 | 0.33 |

The students who were considered in the participant observation with a lower background to their peers were also the ones who reached the lower performance in the proposal of new information and procedures (NI/NP's) for solving the problems: Anália, Paulo, Kamila, William and Darlene. Despite this, it is possible to observe that in all these cases these students had an *OutDegree* value higher than Break Even (*BE*) Point. Except for Paulo, all of these learners presented *OutDegree* higher than their group colleagues. This indicates that students, despite not being able to come up with many ideas for solving programming problems, were able to interact and react in some way to the proposals for new information and procedures from their colleagues. Similar reasoning can be applied to students who have proposed a greater number of NI/NP's. In general, these students achieved a high *InDegree* score (usually above *BE*), which indicates that these students received constant feedback from their colleagues.

Table 6 shows the compiled data for groups G1 to G5 respectively. The sign *V* indicates that NI/NP helped to solve the problem, and *X* indicates that NI/NP was an incorrect idea that could lead the group away from the solution. The proportion of Moves

Table 6

Result of the General Analysis of the Social Graph for each group – the values are computed averages

| Group | # Sessions | Total Moves | NI/NP | NI/NP (V) | NI/NP (X) | GA/LA | Standard. InDegree | HCD Nodes |
|---|---|---|---|---|---|---|---|---|
| G1 | 3 | 98.67 | 0.22 | 0.19 | 0.03 | 0.09 | 0.34 | 1.33 |
| G2 | 4 | 87 | 0.15 | 0.13 | 0.01 | 0.08 | 0.70 | 0.75 |
| G3 | 3 | 233 | 0.16 | 0.09 | 0.07 | 0.17 | 0.51 | 3 |
| G4 | 4 | 133.25 | 0.16 | 0.11 | 0.05 | 0.16 | 0.70 | 2.75 |
| G5 | 1 | 69 | 0.26 | 0.20 | 0.06 | 0.12 | 0.72 | 2 |

classified as *New Information* ranged from 15% to 26%. In all cases, the correct NI/NPs average was higher than the incorrect ones. This may mean that students were not proposing new ideas and procedures in a random way and were engaged to solve the problems. The *Assessments* (GA/LA) average fluctuated from 8% to 17%. The average of the *Standardized InDegrees* value ranged from 0.34 to 0.72. It should be noted that, except for the G1 group, all teams had an average higher than 0.5, which may indicate in these cases that at least half of the group was replying to the new ideas and procedures proposal for generating the solutions. The table also shows that the *HCD* nodes average ranged from 0.75 to 2.75. On the general average, the groups were able to generate approximately two (1.97) *HCD* nodes per session. This may indicate that approximately two ideas proposals per session included the interaction of the entire group. The G2 group had the lowest performance in this context. It is noticed that it was in this team that the worst collaboration gap happened, as previously described. The team reached the following sequence of *HCD* scores in its four Pair Programming sessions: {1, 0, 0, 2}. In this sense, the two sessions in which the values are zero were precisely those that involved Humberto's participation. With the advancement of activities and changes in the groups, it was possible to observe an improvement in this aspect in Humberto's participation. This can be evidenced in the *HCD* score achieved by the G4 group.

### 8.3. *Result of Employed Heuristics Classes*

The results achieved in the identification of heuristics classes applied to each session are illustrated in Fig. 3. The low values received in the last sessions 13 and 15 may have
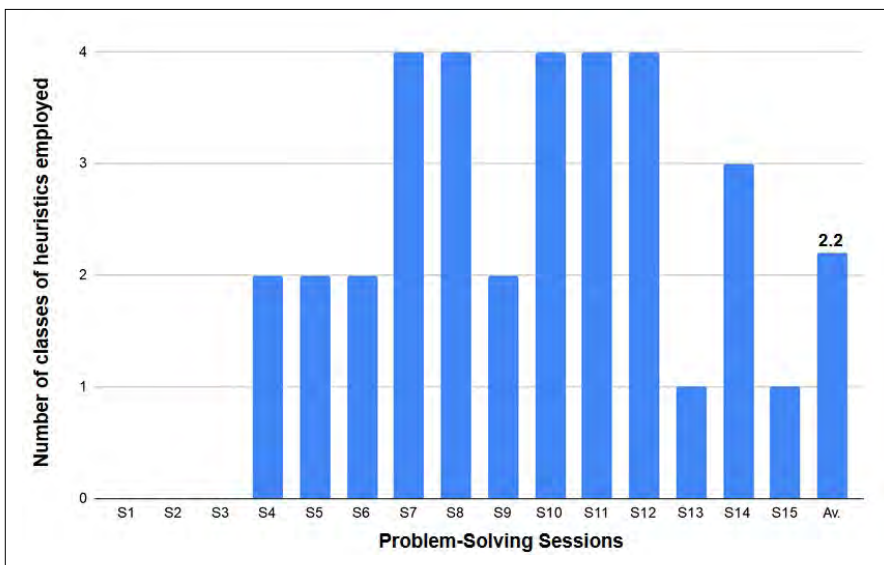


Fig. 3. Number of identified heuristics per session.

been due to the definition of less complex challenges, since the students planned to finish the games to have fun playing them. Therefore, the data show possible evidence that student teams have evolved over time in the problem-solving skills demanded in CT activities.

### 8.4. *Results of the Meeting Minutes Analysis*

In this subsection, we describe the results achieved in the meeting minutes analysis. In this sense, we highlight below the main collaborative and social issues identified at each meeting.

Group 1:

- **Report**: The students reported that they needed to improve their coexistence. **Context**: This information was proposed by Eduardo – who sometimes took over the computer, when it was his colleague's turn, in order to solve all problems by himself. This information in the minutes was proposed by this same learner, which may signal that he recognized his incorrect behavior during the activity.

Group 2:

- **Report**: The group members reported that they should give more ideas. **Context**: According to the participant observation, Michele did not provide any information and Humberto did not communicate his ideas directly to her in a situation where they were solving a problem. Also, when the students finished solving the problems, they wanted to implement new features in the game but they would have no ideas to propose a solution.
- **Report**: The group members report that they should improve the "*division*". **Context**: In session 5, Humberto was responsible for all the *New Information* from the team to solve the problems. Besides, it was possible to observe that Michele was ignoring Humberto's ideas. In this sense, this information may have been reported due to this collaboration gap in the group.

Group 3:

- **Report**: The students reported that the girls in the group were kidding a lot during the activities – what they called "*laughter conflict*". **Context**: The girls kidded a lot with each other during the activity. This information in the minutes was proposed by these learners, which showed the group's empathy for the student Iago. The learners thought that the jokes could disturb the other members of the group. Also, in the next meeting, the girls reported that the problem had not occurred again, which demonstrates group commitment.
- **Report**: The students reported that they should pay attention to the co-pilot's commands. **Context**: According to the participant observation, a student was having problems to follow the group in the implementations. However, team members need to understand that each student has his own learning pace. Therefore, this item represented an opportunity for the facilitator to talk to learners about this subject. However, the conversation did not happen due to the absence of group members.

Group 5:

- **Report**: The students reported that more people needed to participate. **Context**: This was the last session, and the learners reported this because only they attended some sessions of their respective groups, which harmed the progress of their projects.

Virtually all of the collaboration problems observed by the facilitator were identified in the minutes produced by the students. Therefore, according to the achieved results, the meetings were able to shed light on collaboration problems and make students reflect on them.

The students' perceptions of the project's progress are described following. Thus, for each session, we report the main difficulties identified by the students and relate them to the implementation of the tasks that were performed by them.

Group 1:

- **Report**: In the item of the minutes that asks if they had problems, they inform: "*Yes, the big boss jumps*". **Context**: Students recognized that they needed to improve the algorithm responsible for making the boss appear.

Group 2:

- **Report**: Students reported that they accomplished the objective but that they need to finish the game. **Context**: The team gave the objective accomplished even lacking functionality – they did not implement an important task. Despite this, they reported that they need to "fi*nish the game*".
- **Report**: Learners described that goal was achieved but reported an implementation problem. **Context**: The group addressed the objective as achieved despite the errors in two solutions. However, they admitted the problem that they were unable to solve concerning another task when describing "*the ninja star did not appear*"(a bug in an object that must move). They also described that they needed to improve "*a lot*" in the next session.

Group 3:

- **Report**: The team reported that they fail in a task and that they were evolving. **Context**: The students admitted that they were unable to solve a task: "*we were unable to stop the characters from moving when they met*". A small bug in a solution was not reported. When describing whether the objective was achieved, the team declared "*we are evolving*".
- **Report**: Learners described that they were unable to complete a task. **Context**: In the item of the minutes that refers to whether students had problems, they said: "*yes the fainting*". "*Fainting*" was about making the enemy character disappear when hit by the player.
- **Report**: Students described that they needed to fix bugs. **Context**: The team reported that they could "*Fix the bugs*" in the next session, which may concern the bug in solving a task.

Group 4:

- **Report**: The team reported that they had a problem to implement a task that they did not finish in the respective session. **Context**: The group acknowledged the

difficulties they had in a task which involved making the Boss character disappear when he lost their lives. In the item of the minutes in which the learners could report if they had problems, they say "*Yes. Make the big boss die*".

As can be understood from the information described, the teams were able to identify most of the implementation problems that occurred during the development process. In some situations, such as group 2, the students demonstrated that they did not identify some problems, but they seemed to understand that they needed to continue improving the game.

### 8.5. *Result of the Interviews Analysis*

#### 8.5.1. *Collaboration Aspect*

Regarding the collaboration category, learners reported some positive aspects. Some of the students' reports will be presented following.

Paulo, for example, explains that everyone collaborated: "*(...) I think that everyone who took part in this game contributed a little to the game. The game is not just of one person, the game is of a set*". The student also highlighted the collaboration, when asked about extra considerations: "*(...) I think everyone collaborated, everyone helped, everyone went to program the game and I think that the game as I said before is together that we did it was not just separate*". José Vinicius explained that the group shared the tasks and that each one did their part. According to the student, they talked and everyone helped: "*I and they also contributed (...) just like... we shared tasks, right?! Each one did a part, I think this is a good thing, which is group work (...) we talked everyone helped.*" Anália, despite the possible social problem with Naira, mentions Bruna as a support for her participation in activities. This collaboration between the students may have helped Anália to feel more included in the project: "*Sometimes I failed a little, but Bruna helped*".

#### 8.5.2. *Process Aspect*

In general, the learners stated that they actively participated in the development process. Some examples are shown following.

Regarding the participation and use of his time in the process, Humberto highlighted his great interest in the subject addressed. In this context, it is possible to notice that the student referred to the incremental iterative development process employed by the learning strategy: "*I think so, every class we were trying to improve the project even with new ideas. I think that in this field I was very participative (...) I think it was great (...) I think so, because look... like I was going to talk about a thing but I couldn't fit in the subject (...) I .. I always liked video games, then having an opportunity to see how they, at least a little, how they developed, how they do it to develop a game, it was very, very good. So I kind of just focused on that because it was something I liked to do.*" When the interviewer asked if he did what he expected or beyond, Humberto replied: "*It was more inside than I expected to do*". Regarding the use of time in the process, Bruna replied that she thought she used her time well and was concerned

with improving the game during the process: "*I think so, but we kind of tried to do... do it right and not worry too much about time, to make the game better. We tried to do that.*"

### 8.5.3. *Rubric Scores*

To assess the rubric collected data, each classification level was associated with a corresponding score: Beginning (−2), Developing (−1), Proficient (+1) and Exceptional (+2). Table 7 shows the scores achieved for the evaluated dimensions. It is important to report that, each table column is associated with the student's initials (JP – José Pedro, JV – José Vinicius, P – Paulo, etc.) who assigned the score. The results show that the dimensions of collaboration and process were positively evaluated with an average above +1.

### 8.6. *Results Triangulation*

The participant observation described that even students with a lower background were able to collaborate in the production of the games, actively participating in the process. This information was also observed in the rubric's result in which all learners, including those with lower background, considered themselves proficient or exceptional in collaboration and participation in the process. It is important to mention that the qualitative data collected in the interviews showed some specific issues that corroborated the results. An example is the case of Anália who found support in Bruna to participate in the activities. The data point to a significant role of the pair programming approach so that students could participate. The centrality analysis of the social graph showed that students with lower backgrounds, in general, reached a high *Out-Degree* because they were participating by constantly reacting to their colleagues' ideas. Similar reasoning can be applied to students with a higher background who have proposed a greater number of NI/NPs. In general, these learners received a high *InDegree* score (above the breakeven point), which indicates that these students received constant feedback from their colleagues. This was also noted in the participant observation. Besides, the meeting minutes analysis showed that the students were concerned with solving, or at least reporting, the major social problems that emerged during the activities. Therefore, this study pointed out evidence that the proposed learning strategy is capable to support collaboration between students in the CT development activities.

Table 7

Archived rubric scores

| Student: | JP | JV | P | K | M | A | N | B | H | W | Av. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Process | +1 | +1 | +2 | +1 | +1 | +1 | +2 | +1 | +1 | +1 | +1.2 |
| Collaboration: | +2 | +2 | +2 | +1 | +2 | +2 | +1 | +1 | +1 | +2 | +1.6 |

In the context of CT problem solving, the analysis of the social graphs indicated that the number of proposals for correct new information and procedures was greater than the incorrect ones in all groups. Also, it was possible to observe in the employed heuristics evaluation that there was an increasing number of heuristics being applied per session. The evaluation of the progress dimension of the minutes of the meetings also showed that the students, most of the time, were conscious of the problems in the proposed solutions. This points out that students sought to strategically introduce solutions. Therefore, the proportion of correct proposed ideas and, especially, the progressive number of applied heuristics can be evidence of an evolution of the students' CT problem-solving skills.

## 9. Conclusion

This study proposed a Collaborative Learning framework for the development of CT through digital game programming activities. The design of the proposed approach was based on features and methods of Collaborative Learning, which were mapped from the literature, as well as a systematic review of the literature regarding CT and Collaborative Learning. The framework was evaluated through the application of rubric-guided interviews; the analysis of a social graph (generated from the interaction between students during the programming sessions); analysis of problem-solving heuristics employed by the learners; and the review of the minutes of the meetings conducted by the students. The results showed evidence that the proposed framework, enhanced interaction between students. The analysis of social graphs showed that when required to assume different responsibilities (in pair programming with role turns), students should explain, implement and help others to achieve solutions. The analysis of the meeting minutes and the data collected in the rubric-guided interviews, also showed that the students were able to collaborate and reflect on the aspects of the group's social interaction. In the CT problem-solving development skills context, it was possible to observe an evolution in the problem-solving heuristics proposed by the learners. This may be evidence that the framework has been able to empower students in the CT problem solving skills over time. Besides, the social graph analysis showed that students proposed more correct than incorrect ideas to achieve the solution. In this sense, it is not possible to directly observe the impact of collaboration in problem-solving, but the social graph also indicated that new proposals for ideas also received constant reactions from colleagues – this may have helped in the development of these skills. Also, the analysis of the minutes showed that the students were able to, in most cases, jointly identify gaps of progress in the resolution. There is a possibility that this kind of group reflection may have motivated students to seek new solution strategies.

The findings of this research can point out possible ways for CS teachers and instructors to apply activities that effectively develop CT-related skills in their learners. In this sense, the results show the importance of CT teaching approaches that: (1) allow students to effectively engage in defining what products they could develop and what

problems they must face, (2) allow the creation of a challenging environment that enables students to apply diverse problem-solving heuristics and (3) keep learners working collaboratively with appropriate teacher support.

As future work, the proposed strategy can be adapted to be applied in Computational Thinking activities concerning Educational Robotics. Besides, apply the strategy in a hybrid way with another collaborative learning approach, such as jigsaw, could bring interesting results.

## Acknowledgments

## References

ACM (2017). *ACM Digital Library*. https://dl.acm.org/dl.cfm/

Barr, V., Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, 2(1), 48–54. [Online]. Accessed on: Dec. 12, 2018. Available:
https://id.iste.org/docs/nets-refresh-toolkit/bringing-ct-to-k-12.pdf?sfvrsn=2%2018

CAPES (2017). *Portal Periódicos CAPES*. http://www.periodicos.capes.gov.br

Castro, A., Menezes, C., Prikladnicki, R. (2011). Aprendizagem colaborativa com suporte computacional. In: Pimental, M., Fuks, H. (Eds.), *Sistemas Colaborativos*. Elsevier, Rio de Janeiro, pp. 135–185.

Cockburn, A., Williams, L. (2000). The costs and benefits of pair programming. In: *In eXtreme Programming and Flexible Processes in Software Engineering XP2000*. Addison-Wesley, Boston, pp. 223–247.

Code.org (2019). http://code.org

Cohen, E.G., Lotan, R.A. (2014). *Designing Groupwork: Strategies for the Heterogeneous Classroom*. Teachers College Press, New York.

CSTA (2011). *CSTA Standards for Computer Science Teachers*.
https://csteachers.org/documents/en-us/7a89a973-92df-4c32-bef5-d875e0d69144/1

CSTA, ISTE (2011). *Computational Thinking: Leadership Toolkit*.
https://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf

da Ponte *et al.*, J.P. (2007). *Programa de Matemática do Ensino Básico*. Discussion document, Ministério da Educação: Direcção Geral de Inovação e Desenvolvimento Curricular, Lisbon.

de Jesus, A.M., Silveira, I.F. (2019). A cooperative learning strategy to computational thinking development. *Revista de Ensino de Ciências e Matemática (REnCiMa)*, 10(4), 192–211.
https://doi.org/10.26843/rencima.v10i4.2387
https://revistapos.cruzeirodosul.edu.br/index.php/rencima/article/view/2387

de Jesus, A.M., Silveira, I.F. (2020). Applying agile software engineering to develop computational thinking in children. In: Álvaro Rocha *et al.* (Ed.), *Trends and Innovations in Information Systems and Technologies*. Springer, Gewerbestrasse, Switzerland, pp. 756–765. Álvaro Rocha *et al.*.

de Jesus, A.M., Silveira, I.F., de Lima Palanch, W.B. (2019). Desenvolvimento do pensamento computacional por meio da colaboração: Uma revisão sistemática da literatura. *Brazilian Journal of Computers in Education*, 27(2), 69–90.
https://doi.org/10.5753/RBIE.2019.27.02.69
https://www.brie.org/pub/index.php/rbie/article/view/v27n026990

de Walle, J.A.V. (2009). Ensinando pela resolução de problemas. In: de Walle, J.A.V. (Ed.), *Matemática no Ensino Fundamental*. Artmed, Porto Alegre, pp. 57–81. John A. Van de Walle.

de Walle, J.A.V., Karp, K.S., Bay-Williams, J.M. (2016). *Elementary and Middle School Mathematics: Teaching Developmentally*. (9th ed.). Pearson, London.

del Puy Pérez Echeverría, M., Pozo, J.I. (1998). Aprender a Resolver Problemas e Resolver Problemas para Aprender. In: POZO, J.I. (Ed.), *Solução de Problemas: Aprender a resolver, Resolver para Aprender*. ArtMed, Porto Alegre, pp. 13–42. Chap. 1.

Dillenbourg, P. (1999). What do you mean by collaborative learning? In: Dillenbourg, P. (Ed.), *Collaborative-learning: Cognitive and Computational Approaches*. Elsevier, Oxford, England, pp. 1–19. Chap. 1. [Online]. Accessed on: Dec. 10, 2018. Available:
https://telearn.archives-ouvertes.fr/hal-00190240/document

Dybdal, O.T.J.J.M., Bjorn, P. (2017). Coding as a social and tangible activity. *Interactions*, 24(6), 70–73.
https://doi.org/10.1145/3137099

Gokhale, A.A. (1995). Collaborative learning enhances critical thinking. *Journal of Technology Education*, 7(1), 22–30. https://doi.org/10.21061/jte.v7i1.a.2

Goos, M., Galbraith, P. (1996). Do it this way! Metacognitive strategies in collaborative mathematical problem solving. *Educational Studies in Mathematics*, 30, 229–260. [Online]. Accessed on: Nov. 05, 2018. Available: https://link.springer.com/article/10.1007/BF00304567

Goos, M., Galbraith, P., Renshaw, P.D. (2002). Socially mediated metacognition: Creating collaborative zones of proximal development in small group problem solving. *Educational Studies in Mathematics*, 49(2), 193–223. [Online]. Accessed on: Nov. 05, 2018. Available:
https://link.springer.com/article/10.1023/A:1016209010120

Heinonen, K., Hirvikoski, K., Luukkainen, M., Vihavainen, A. (2013). Learning Agile Software Engineering Practices Using Coding Dojo. In: *Proceedings of the 14th Annual ACM SIGITE Conference on Information Technology Education*. SIGITE '13. Association for Computing Machinery, New York, NY, USA, pp. 97–102. 9781450322393. https://doi.org/10.1145/2512276.2512306

Johnson, D.W., Johnson, R.T., Holubec, E.J., Roy, P. (1984). Why cooperative learning is important. In: D.W. Johnson, R.T. Johnson, E.J. Holubec and P. Roy (Eds.), *Circles of Learning: Cooperation in the Classroom*. Eric, USA, pp. 7–17.

Karnstedt, M., Hennessy, T., Chan, J., Basuchowdhuri, P., Hayes, C., Strufe, T. (2010). Churn in Social Networks. In: Furht, B. (Ed.), *Handbook of Social Network Technologies and Applications*. Springer Science Business Media, New York, pp. 185–222. Borko Furht.

Kitchenham, B. (2004). Procedures for performing systematic reviews. Technical Report TR/SE-0401, Keele University, Staffordshire.

Lee, G.S. (1997). *Internet Communication. Technical Report*. Institute for Distance Education University Pertanian Malaysia, Malaysia.

Martinez-Moyano, I.J. (2006). Exploring the dynamics of collaboration in interorganizational settings. In: Schuman (Ed.), *Creating a Culture of Collaboration*. Jossey-bass, San Francisco, pp. 69–85.

MIT Media Lab, L.K.G. (2019). Scratch: create, imagine, share. https://scratch.mit.edu/

Monereo, C., Gisbert, D.D. (2005). *Tramas: Procedimentos para Aprendizagem Cooperativa*. Artmed, Porto Alegre.

Panitz, T. (1999). Collaborative Versus Cooperative Learning: A comparison of the two concepts which help us understand the underlying nature of interactive learning. *Educational Resources Information Center ERIC*, 3–13. [Online]. Accessed on: Apr. 15, 2019. Available:
https://files.eric.ed.gov/fulltext/ED448443.pdf

Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, New York, NY.

Pham, A., Pham, P.-V. (2011). *Scrum in Action: Agile Software Project Management and Development*. Cengage Learning, Boston.

Randall, K., Rusk, N. (2009). Rubric for assessing Scratch projects-draft. ScratchEd. http://scratched.gse.harvard.edu/resources/rubric-assessing-scratch-projects-draft-0.html

ReadWriteThink (2003). *Cooperative Learning Rubric*.
http://www.readwritethink.org/files/resources/lesson_images/lesson95/

Rubin, H. (2002). *Collaborative Leadership: developing effective partnerships for communities and schools* (2nd ed.). Corwin Oaks, Thousand Oaks.

Saltz, J.S., Hiltz, S.R., Turoff, M. (2004). Student social graphs: visualizing a student's online social network. In: *Proceedings of the 10th ACM Conference on Computer Supported Cooperative Work*. ACM, New York, pp. 596–599. 1-58113-810-5. https://doi.org/10.1145/1031607.1031709

Sato, D.T., Corbucci, H., Bravo, M.V. (2008). Coding Dojo: An environment for learning and sharing agile practices. In: Melnik, G., Kruchten, P., Poppendieck, M. (Eds.), *Agile Development Conference, AGILE*

*2008, Toronto, Canada, 4–8 August 2008*. IEEE Computer Society, Washington, pp. 459–464. `https://doi.org/10.1109/Agile.2008.11`

Teles, V.M. (2004). *Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade*. Novatec, São Paulo, SP.

Torres, P.L., Irala, E.A.F. (2014). Aprendizagem colaborativa: Teoria e prática. In: Torres, P.L. (Ed.), *Complexidade: Redes e Conexões na Produção do Conhecimento*. SENARPR, Curitiba, Brazil, pp. 61–93. [Online]. Accessed on: Dec. 10, 2018. Available: `https://www.agrinho.com.br/site/wp-content/uploads/2014/09/2_03_Aprendizagem-colaborativa.pdf`

Wasserman, S., Faust, K. (1994). *Social Network Analysis: Methods and Applications* (Vol. 8). Cambridge university press, Cambridge.

**Â.M. de Jesus** obtained his Ph.D. in Mathematics and Science Teaching from Cruzeiro do Sul University. He is a research professor at the Federal Institute of Minas Gerais. He is interested in Informatics in Education, Problem-Based Learning and Computational Thinking.

**I.F. Silveira** is a research professor at Cruzeiro do Sul University and Mackenzie University. He obtained his Ph.D. in Electrical Engineering from the University of São Paulo. He was president of the CEIE (Special Commission on Informatics in Education) of the Brazilian Computing Society from 2012 to 2014. He is a member of the LA-CLO community and the Asociación LATIn. He is interested in Informatics in Education, Digital Games, Open Educational Resources, Computational Thinking, Software Engineering and Graphic Processing.