



Programming Teaching Tools Feature Assessment Associated with Brazilian Curriculum Base Obtained Through BPL Platforms Analysis

Aléssio Miranda Júnior^(✉) , Deisyamar Botega Tavares^(✉) ,
and Jordana Caires Carvalho^(✉) 

Centro Federal de Educação Tecnológica de Minas Gerais,
Belo Horizonte 35180-008, Brazil
{alessio, dbotegatavares}@cefetmg.br,
jordana.caires.carvalho@gmail.com

Abstract. Learning programming isn't only important for programmers. Studies show that learning computation is useful for logical reasoning development, which leads to an easier learning in mathematical fundamentals and language skills. Because of this, several computation platforms have emerged for children and teenagers. These tools can be used by teachers from several fields of knowledge to stimulate their student's learning. However, these platforms don't include topics from the Brazilian National Curricular Joint Base (BNCC), which make them difficult to use for this specific purpose. Looking to the relevance of creating or remodeling a platform for this goal, the first step is to survey for necessary features. The LORI method was used by students from elementary public school, using Code and Blockly platforms. This paper describes which and how features were studied during the exploration of two platforms, while analysing the audience. Computer education platforms directed to young people can highlight some important characteristics: daily use and cultural aspects of students in order to approach the platform, giving feedback with enough information for them to understand their error and the possibility for the teacher to visualize and measure in what aspect and how often the students make a mistake.

Keywords: Cultural issues in learning with collaboration technologies · Programming for children · E-learning · LORI · Education · Programming with education

1 Introduction

Logical reasoning is the basis of cognitive maturation and it supports the development of other skills. Several studies consider it one of the foundations for critical thinking skills development, as its absence can hamper linguistic learning abilities, like Portuguese and mathematics fundamentals [1].

Logical reasoning is also the basis of computation, as this area covers and reaches most other fields of knowledge and has steady growth in the professional market. However, unlike the others, this is neither predicted nor taught in the Brazilian National Curricular Joint Base (BNCC).

If a future professional doesn't need programming, it will still bring benefits by developing logical reasoning, critical thinking, and linear thinking in problem-solving. Furthermore, understanding how technologies are being developed and having the ability to develop oneself may grant opportunities [2].

The teaching of programming for children and teenagers aims to encourage not only logical reasoning development, but also other areas of knowledge so that solutions may be created to their own desires while checking if it is something they like [2].

The main objective is not to teach a specific programming language, but to show that logic is the same for all languages. It should be noted that not all students will become programmers, but with a differential, they will have a greater ability to think and be more creative, since learning programming logic develops various skills that are often hidden [3].

Although current programming teaching platforms are widely used and easily accessible, with playful methodologies and didactic resources aimed at children and teenagers, there is no perspective of targeting or customize based on cultures. Citing the example of Code platform, "there are some problems that are limited to being solved by requiring little study, but is not possible to fit them into a format to be used in a classroom, respecting the BNCC".

The proposal is to create or remodel a tool available to this target audience, involving both programming learning and the BNCC, jointly. That is, as long as the person has a problem in basic disciplines, its solution will be through programming and prior knowledge of the discipline itself. The intention isn't to replace classroom teaching in schools, but to be an instrument of support and fixation, while gaining the ability of programming.

The creation or remodeling of a tool is an extensive and complex work, for the initial steps are significant. One of them is the study of existing and non-existing features on the main platforms, that are essential to the proposed tool.

The Problem Based Learning (PBL) methodology emphasizes learning through problem solving. Through this method the student is the main actor of his learning, as when faced with a problem presented by the teacher, they should actively seek the solution, without theoretical classes or previous examples of resolution given by the teacher. The teacher's function becomes that of an instructor, indicating sources where students can research problem solutions and follow their reasoning, being able to redirect it if they escape the goal, except it is up to the student to research, develop and present a solution. In this method some concepts of collaborative work are also employed, since the students are separated in small groups of a maximum of six, where they discuss and analyze together the solutions to the problems.

Zanatta [4] shows some programming educational platforms where the author mention history, aspects, and purpose. The text called "*Programação de Computadores para Crianças: Metodologia do Code Club Brasil*" (Computer Programming for Children: Code Club Brazil Methodology) has the experience of building and applying the Code Club to students and observe its methodology. The difference in this work is

that Zanatta (2015) doesn't apply any method to validate the other platforms, (like Learning Object Review Instrument) [5], there is no intention of listing the characteristics of each of them and the purpose to join the Brazilian National Curricular Joint Base (BNCC).

This article describes what these characteristics are and how they were examined. Section 2 clarifies the methods used to scan the features and Sect. 3 presents the result obtained by the assessment.

2 Construction of the IPC and Application of the Methodology

An extension project was created at CEFET-MG (*Centro Federal Tecnológico de Minas Gerais*), called Introduction to Programming and Competitions (iPC), with the purpose of involving elementary and high school students that don't study in CEFET into logical learning, with oriented use in the current platforms.

The project started in 2017 and attended until 2018 a total of 84 students. The team involved 15 undergraduate students, three Computer Engineering teachers and two students from the Informatics technical course. Through this project, it was possible to accomplish actions that allowed the study of existing and/or non-existing features of the current programming platforms for children and teenagers. The actions were (i) choice of platforms and activities, (ii) Code and Blockly analysis with LORI method: results in the Sect. 3.3, (iii) invitation of the students, (iv) use and analysis of Code and Blockly with the students.

2.1 Choice of Platform(s) and Activities

The first step was the selection of existing platforms, as well as verification of which of them possessed the elements that would facilitate compliance with the BNCC. It was decided not to show code lines and advanced concepts to beginner students in programming. Bau et al. [6] demonstrated that the goal should be focused on logic and avoid complicated elements in the educational environment.

We adopted Blockly in the first group of students and it has been more explored by the iPC development team to see and test the possibilities. Thus, Code was applied in a primary plan for students, as to evaluate the aspects that are appropriate or not to the iPC purpose.

Noticing the experience of the iPC team and seeking for a platform that could be used as a test with its own method and what would the students think about it, Code and Blockly were analyzed by the LORI (Learning Object Review Instrument) [4]. The results are in the Sect. 3.2.

Visualizing the games in Code, by experience in computer teaching, the one that demonstrated to be completer and more suitable for young people aged between 14 and 15 years old was the "Accelerated Intro to CS Course".

2.2 Inviting Students

Over two years of project, 84 students were invited and distributed in four classes, as shown in Fig. 1. The first group was composed of ninth grade students from municipal schools personally invited in each school, accompanied by the Education Secretary of the municipality of Timóteo.

Teachers, principals, and students from the ninth-grade elementary school proposed were introduced to the purpose of the project and informed how it would work. The Education Secretary was requested to select 21 students, analyzing math punctuation, personality, talking with the teachers about the responsibility of each one and considering the equal distribution of representant by school.

After the first class finished the course with 21 students (Fig. 1), the next two classes were composed of first year students from municipal and state high schools. The students made a subscription by form and each group was composed between 18 and 25 people each (Fig. 1), having about two to three dropouts individually.

Classroom Sections

Add a new classroom section
Create a new classroom section to start assigning courses and seeing your student progress. Create a section

Section	Grade	Course	Students	Login Info	
IPC 2/2018	9	Accelerated Intro to CS Course	21	CWGRGC	▼
IPC 1/2018	Other	Accelerated Intro to CS Course	18	KQSZWC	▼
IPC - 1	9	Accelerated Intro to CS Course	25	NSZHDP	▼

Fig. 1. Three classes of the iPC project

2.3 Use and Analysis of Code and Blockly with Students

The recruited students were submitted to various activities from the accelerated course of Code (all classes) and some activities of Blockly (only the first class).

There was one weekly meeting of three hours each for the first group and two weekly meetings of two hours for the second and third classes.

During the meetings, the instructors (students from the Informatics technical course and the Computer Engineering course) coordinated the activities by teaching students how to use the platforms and asking questions. During this follow-up process, several interventions were needed since the students were sometimes euphoric or couldn't progress, or could not properly exploit the platform's resources. In addition, in some instances the resources of the platforms were not sufficient for the students' progress, and therefore intervening with more practical teaching strategies were necessary, mainly in the teaching of programming concepts, such as function for example.

As the instructors were monitoring the class, it was possible to perceive several characteristics on the platforms, and if they were sufficient or not. Section 3.3 presents these raised characteristics.

3 Results of the Platforms Analysis and Students

The process of arriving to a tool that can combine the learnings of Code and Blockly and show the user basic study elements in a motivational way is long, since a lot of detailed perception is needed and the students behavior especially counts. With the application of the LORI method, the chosen platforms and their use by the groups of students in a monitored way, it was possible to raise series of important features for a platform that has the final design to be an instrument of assistance in learning the contents of the BNCC through practice. The following sections highlight these raised characteristics.

3.1 Analyzing the Code and Blockly Tools Using LORI Method

The purpose of Code and Blockly normally attend very well, but it isn't enough for the iPC project's goals. In order to overcome those limitations, such as a little approach of math content and a cover of all disciplines in basic education according to the BNCC, while observing these platforms by the LORI method [4], some aspects were noticed (Table 1).

Table 1. Application of the LORI method.

Item	Code	Blockly
Content quality	Code presents a more detailed content with explanations in each game. There are also videos for more context	Blockly only have a statement of the problem
Learning objectives alignment	Both have the purpose to teach logic programming for children and teenagers. But, for the iPC's goal, it is not enough	
Feedback and adaptation	Code sends feedback to the user with different sounds; a message on top of a statement informs a mistake and offers a help	Blockly only have a message in the screen center when a mistake has occurred, there isn't a note to click for help or any box with more information
Motivation	Code conducts the user in all screen games and the characters are distributed in levels, as they come back to reinforce the knowledge as time goes	The games are presented with little information on screen and a message appears if the exercise is correct. The platform has motivation, but doesn't give enough feedback

(continued)

Table 1. (continued)

Item	Code	Blockly
Presentation design	Both have animations and sounds to encourage the student to continue the exercises	
Interactive usability	The screens are colorful and the explanation videos lead the user to have an easy experience with the platform	Blockly has a screen focusing on the game as a way not to compete with functionalities
Accessibility	Disapproved. Both don't have resource for low vision or blind people	
Compliance with the pattern	Doesn't apply in this case	

3.2 Lessons Adaptation

Through observation during the three months of classes with the selected students in CEFET, by the students questions, monitoring of reasoning improvement, the perception the teenagers resolution for the most difficult exercises, some aspects in terms of functional and non-functional requirements were regarded.

As cited in Sect. 2.3, some theoretical classes were taught. It occurred in view of the teenagers' anxiety, with regard to new concepts and situations, and also because Code doesn't analyze code quality. For example, the tool counts blocks (Code uses drag-and-drop code blocks), but not which of them was put by the student in the solution. In other words, if the student in a repeated loop lesson doesn't bring any repeat loop block, but instead puts less or the correct number of blocks to execute, and the execution solves the problem, Code accepts the student's solution.

The interventions in each classes had the purpose of alerting the students about it and reinforce computational concepts according to Brazilian culture. The instructors had the responsibility to create appropriate activities taking into account a close observation of harder exercises in classroom and conversations at the end of activities with the teenagers. Those exercises could be a presentation, word research, a game to write a story with algorithm words (if, do, while, etc.), an explanation of "functions" using cake shapes with equal shapes but with different sizes, and others of this type.

Therefore, the explanatory resources: video-lessons, since textual explanations on these platforms are not enough to solve the students' doubts, due to their generic nature and not contextualized with the national reality. In this way, a mechanism that allows the insertion of more video-lessons is necessary as well as textual explanations in these platforms are a relevant resource, based on the lived experience.

3.3 Learning and Teaching the Students

Through the follow-up of students during the execution of the activities and by the necessary interferences performed in this process, series of situations were perceived and make a difference for a better growth of the student's learning. Among them:

- Empathy - Understanding how the students felt about the new situation at the end of each class, asking and dialoguing about how were the exercises, the difficulties and changes for a better adaptation, and if they had enjoyed it. This conversation made possible to spot the empirical aspects in the Sect. 3.4. This empathy could be exposed with an AI resource that would talk to the user about their experience and enable the user to asks about their doubts, as the tool would analyse this data, like a chat boot. It would show and repeat the exercises that the person had difficulties, and over the exercise display, some interesting facts about the tool and how logical reasoning can be helpful day by day, in life in general, and applied at school.
- Simple thinking - Showing a student a farmer that digs a field doesn't approach the teenagers to the tool, because it's possible that someone has never been into a field. Simple things like the weather are general subjects that anyone knows, and it doesn't involve a cultural or regional point of view. When the scenarios of the activities are closer to the reality of the students, their understanding of the problems, as well as their solution, becomes simpler. Consequently, it is important to adapt the problems to the knowledge and competences foreseen in the BNCC, in order to make an expected use of the learning resource.
- Together is funnier than alone - Code and Blockly are tools to learn by oneself. But altogether, having healthy competitions and to be able to ask for help is a way to support other people. A specific screen (Fig. 2) was found harder by the students, and when they spoke to each other while the instructors were observing, they understood that the problem wasn't so difficult, which leads to questioning whether the tools available or to be created should have at least some group activity.

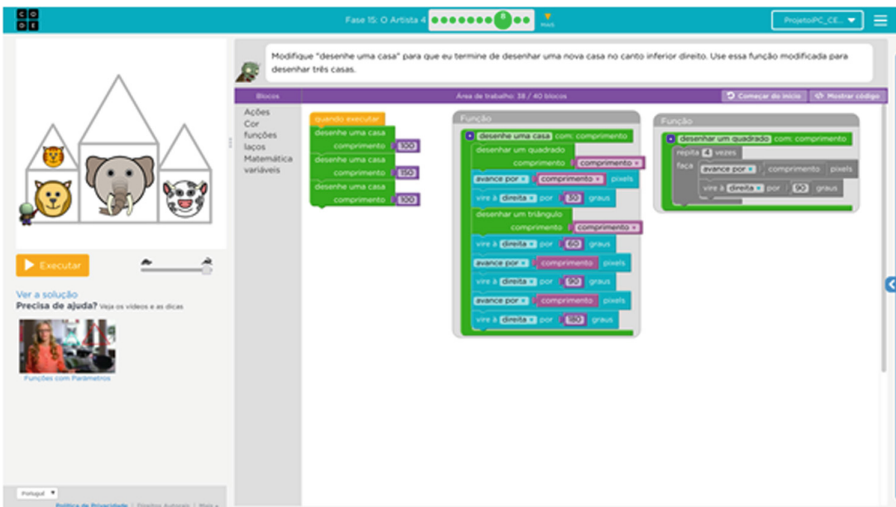


Fig. 2. A screenshot of an exercise

- The tool is important, but knowledge goes far beyond - The purpose of Code and Blockly is to teach logic to children and teenagers. However, like the screen shown

on Fig. 2, if the person doesn't know the basic concept, the computational matter stays on a second plan. Mathematical concepts had to be reviewed and even taught, as well as notions of right and left and text interpretation. Even though Code has explanatory videos and some interventions with offline exercises, it was perceived by practice that it is not enough, and there must be other means of explanation for the students. And some of these could be through the intervention of teachers in classroom, via platform, by video-lessons to deal with specific subjects of each activity when necessary and/or through a chatbot that the student could receive some other explanation, one that could be more comfortable, choosing a text, a video, a book, a game or other kinds of educational resources.

3.4 Analyzing the Code and Blockly Tools After an Application in a Classroom

With these experiences and observations, the final goal is to think and prototype a platform with the search and development teams. Beyond that, it is required that

Table 2. Analysis result of Code and Blockly platforms.

Platforms	Positive aspects	Negative aspects
Code	<ul style="list-style-type: none"> • It presents playful elements that are familiar to students • It is self-explanatory • It does not present student-ready resolution, forcing them to create their own reasoning • It has several modules that deal with different concepts and characters from movies or games • There is a quality evaluation of the student's code 	<ul style="list-style-type: none"> • There is no view of the student progress history (hits and misses) • There is no way to communicate with the student through this tool • The administrative part, mainly the possibility of creating questionnaires for the students shows to be inefficient and is a difficult understanding functionality, with little support for such • The simultaneous translation into Brazilian Portuguese go through deformations, hindering communication • It doesn't obligate the student to complete modules with perfection, giving the possibility to obtain a certificate of completion without this requirement
Blockly	<ul style="list-style-type: none"> • The exercises have a more complex level than code, such as the music game and the geometric game, that include time variables • The interface is visually clean 	<ul style="list-style-type: none"> • There is no option to log in by storing progress • Problems have insufficient instructions for the student's understanding • There is no return guidance information after some programming error is committed • There is no evaluation of the student's code

teachers and other professionals of education, computing and pedagogy create, analyze and approve each step according to the level of difficulty, the learning method, the content of disciplines simultaneously as making the code and the Software Engineering tests.

After analyzing the selected platforms, it was possible to list a set of positive and negative aspects (Table 2). Such aspects should be considered in order to create or remodel future platforms to integrate the contents of the disciplines governed by BNCC in programming.

3.5 Empirical Result by the Teenagers

The monitoring of the use of the Code and Blockly platforms by the students has led teachers and monitors to perceive a series of aspects that aren't fully aligned with the purpose of this article, but deserve attention considering that they are experiences that motivate more and more the improvement of the tool for the use of programming for children and teenagers, as well as the advancement of pedagogical practice's that include computation as an instrument of teaching and learning. These aspects were:

- Improvement of logical and mathematical reasoning.
- They made new friends.
- They met and were immersed into CEFET-MG teaching institution, demonstrating the interest in entering of computer courses.
- Personal progress.
- Discovery of a possible new career.
- Greater interest in mathematics and programming.
- Reflection of the project as learning content in the school environment.
- Development of concentration, attention, reasoning and patience.
- The perception that reading is important and makes a difference.

Attentive to this opinion, the search team surveyed the third class with two questionnaires application (one at the beginning of the project and one at the end) to measure the transformations in reasoning. The method and results will be shown in another paper.

4 Conclusion

In this report about the first two years of the project, we have done evaluations of programming teaching tools for children and teenagers, looking for features that adapt to the reality of the BNCC.

Using the LORI method, we can make a comparative analysis of the two tools within the requirements placed. These results serve as guides for the analysis of the empirically collected features.

We indicate that the obtained results are preliminary but important to demonstrate the need and guidance for the improvement of the educational development of the tools. Mainly due to the fact that no tools are already in the desired state, and that when fitting, the results become detachable.

The teaching of programming logic focusing on pre-existing problems in the students' disciplines facilitated the learning of both. Complementing cognitive and interpersonal development of students in working in groups to solve problems was also presented in the reports.

We recognize that more formalism is needed in the next stages of validation. A step of municipal analysis of each raised feature and an amplification of the tests bases must reinforce the purpose of this work.

In addition, there is a consistent space for a new proposition that really adapts these educational environments to the Brazilian educational model.

References

1. García, J.N.: Manual de dificuldades de aprendizagem: linguagem, leitura, escrita e matemática, 1st edn, 274 p. McGraw Hill – Artmed, Brasil (1998). ISBN 8573073128
2. Fernandes, C.S.: Ciência da computação para crianças. Dissertação (Mestrado em Computação) - Instituto de Informática. Universidade Federal do Rio Grande do Sul., Rio Grande do Sul., 156 p. (2002). <https://lume.ufrgs.br/handle/10183/6843>. Accessed 01 Jan 2019
3. Garlet, D., Bigolin, N.M., Silveira, S.R.: Uma Proposta para o Ensino de Programação de Computadores na Educação Básica. Universidade Federal de Santa Maria (UFSM) – Campus de Frederico Westphalen, Brasil, pp. 1–25 (2016)
4. Zanatta, A.C.: Programação de Computadores para Crianças: Metodologia do Code Club Brasil. Universidade Federal de Santa Catarina Campus Araranguá, Brasil, 102 p. (2015). <https://repositorio.ufsc.br/xmlui/handle/123456789/158762>. Accessed 01 Jan 2019
5. Nesbit, J., Belfer, K., Leacock, T. (ed.). Instrumento para a Avaliação de Objectos de Aprendizagem (LORI): Manual do Usuário, 2nd edn. [S. l.: s. n.], 14 p. (2009). http://www.avu.org/avuorg/images/Documents/ODeLPD/lori_pt.pdf. Accessed 01 Jan 2019
6. Bau, D., et al.: Learnable Programming: Blocks and Beyond. arXiv, USA, ano 2017, pp. 1–8 (2017)