# A GAME DEVELOPED USING SCRATCH AND ROBOTICS TO STIMULATE COMPUTATIONAL THINKING

Rodrigo Lira[a]

[a] Polytechnic School of Pernambuco, University of Pernambuco, Recife, Brazil, rcls@ecomp.poli.br

**Abstract.** Technology is increasingly becoming a part of our daily lives. Consequently, it is essential for current and future generations to know how to solve problems using these tools. These skills are often labelled as computational thinking, a term that captures the attention of many educators and governments. There are many ways to spark interest in this area, but teaching programming is one of the most common. To this end, some initiatives use robotics or drag-and-drop programming to attract children's attention. Thus, this work proposes a game utilizing robotics to stimulate computational thinking. The proposed game was developed both digitally using Scratch and physically using Arduino. To win, players must create the step-by-step (algorithm) movement of the robot on a game board. Additionally, an app was developed to enable Bluetooth communication with the robot. The entire scenario was designed playfully to engage the player.

**Keywords.** Computational Thinking, Arduino, Scratch.

## 1. Introduction

Human beings are driven by the need to transform and evolve. This drive leads to the creation of new technologies and the need to expand knowledge of existing ones. Computers were initially used in military applications, but later made their way into homes as a useful everyday tool. Programming, once a skill reserved for a select group, is now being introduced in schools as part of the curriculum. Initiatives like Code.org highlight how programming knowledge is essential for current and future generations. Consequently, the term 'Computational Thinking' is increasingly discussed among teachers and researchers.

According to Aho [1], Computational Thinking involves the use of techniques that encompass Computer Science to solve problems in any area. More and more initiatives have been encouraged by governments to stimulate this skill, such as the one initiated by former President Obama in the USA [2]. It is worth noting that in Brazil, the Brazilian Computer Society (SBC) proposed the inclusion of this knowledge in the National Common Curriculum Base (BNCC) [3].

These skills can be fostered in various ways and do not necessarily require a computer as a tool. Given the relevance of this topic and its many possible applications, it is essential to study which approaches are highlighted in the literature as means of stimulating computational thinking.

Among the well-known methods, educational robotics and drag-and-drop programming tools are examples used by educators to help children develop this skill.

According to Benitti [4], robotics is an excellent learning tool, even for subjects that are not directly related to this area. Based on this, this work aims to present a game designed to stimulate computational thinking in children.

This work is organized as follows: Section 2 presents the related works; Section 3 introduces the game proposal; Section 4 outlines the development methodology; results and discussions are presented in Section 5, and finally, conclusions are provided in Section 6.

## 2. Related Work

The related works predominantly feature publications from significant events in the field of computer science education in Brazil. Such as the Brazilian Symposium on Informatics in Education (SBIE), the Brazilian Congress on Informatics in Education (CBIE), and the Workshop on Informatics in School (WIE)). These forums serve as pivotal platforms for disseminating research findings and fostering academic discourse among educators, researchers, and practitioners dedicated to enhancing educational practices through the integration of technology. The focus of these events encompasses a wide range of topics, including pedagogical methodologies, technological innovations, and the development of curricular frameworks aimed at improving computational thinking and programming skills among students. By engaging with these established conferences, the

works contribute to the broader discourse on effective practices in computer science education within the Brazilian context.

Aono et al. [5] used Scratch as a tool to stimulate computational thinking in sixth-grade students at a private school. According to the authors, the purpose of the course was to create a game in the style of "FlipBird" on the platform. The authors compared the results obtained with those of other studies and also concluded that using Scratch prevents a traumatic first contact with programming, noting a high level of motivation among the students. As a result, there was no dropout, and all students completed their respective games. The authors concluded that the use of Scratch facilitated a pleasant development of learning in Computational Thinking and Programming.

Madeira [6] also employed Scratch as a tool to introduce the fundamentals of computational thinking. In the workshop he conducted, participants were tasked with developing a game in which the user had to catch a mouse in the environment while preventing it from reaching the cheese. While programming the movements of the animals, users engaged with basic programming structures such as loops, conditionals, variables, and functions. This hands-on approach not only facilitated the understanding of core programming concepts but also encouraged creative problem-solving and logical reasoning skills, which are essential components of computational thinking. By immersing users in a playful context, the activity aimed to make the learning process engaging and accessible, thereby fostering a positive attitude toward programming among participants.

Falcão et al. [7] used the game Lightbot as a case study. The game aims to introduce players to various programming logic challenges that increase in complexity with each level, requiring players to employ more sophisticated solutions involving functions, loops, and other programming concepts. The participants consisted of 11 children aged between 6 and 9 years, approached with their guardians in public spaces. The children, accompanied by their guardians, were invited to play Lightbot on one of the researchers' mobile phones, and information was collected to respond to the research questionnaires.

According to the authors, despite the challenges faced—such as distractions, noise, and interventions from guardians—the children expressed enjoyment of the game and showed interest in playing it again. In some instances, it was necessary to ask for the release of the phone so that another child could participate in the proposed activity. The authors also noted that boys displayed a greater affinity for the game, hypothesizing that this might be due to the game's predominantly masculine interface. In one qualitative analysis, the researchers indicated that it was generally necessary to assist the children with hints during their interactions, as the exploratory nature of the game did not allow them to independently discover how it worked.

Through this analysis, the researchers highlighted several aspects that should be considered in educational games, as well as the development of an evaluation method for similar types of games. This study contributes to the understanding of how children engage with programming concepts through gaming and emphasizes the importance of thoughtful design in educational game development.

The findings from this experience underscored the importance of hands-on learning in understanding programming and robotics, highlighting how students engaged with the content through practical application. This approach not only fostered technical skills but also enhanced problem-solving abilities and teamwork among the students. The authors emphasized that while challenges were present, the overall positive response from the students indicated that such initiatives could effectively promote interest and competence in computational thinking and related fields.

## 3. Proposal

The proposal emerged as a tool to introduce children to the world of programming, thereby fostering skills related to computational thinking. Through challenges presented in a board game, the player needs to develop step-by-step instructions (algorithm) for moving a robot, thereby applying programming logic concepts. While the target audience of the project is elementary school children, its application is not limited to this age group.

The objective is to leave the starting point and reach the position indicated in the move. This position will depend on the map used in the game. In the example shown in Figure 1, there is an image created during the validation of the proposal. In this map, the focus is on letters of the alphabet, so the player will be required to know these letters to find the designated position as the final goal.

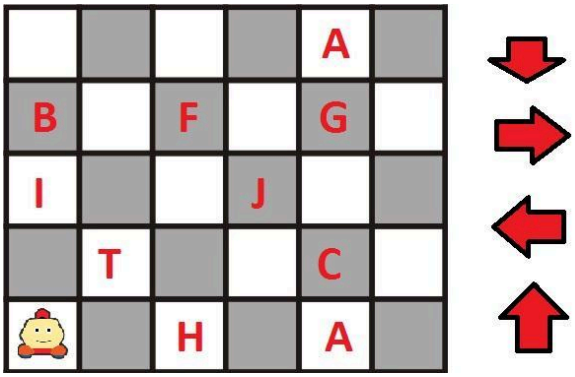



**Fig. 1 -** "Screen prototype that uses a map with information about the letters of the alphabet."

The game format was designed to be used across

different areas of knowledge. For example, in geography, the robot could be tasked with moving to the capital of Pernambuco. In English language teaching, the player might need to navigate to the translation of the word "House." Regardless of the board, the mission would be accomplished by creating a program for the robot's movement using the commands: right, forward, left, and down.

## 4. Methodology

The proposal consists of two games: one digital and the other developed in a physical format using robotics equipment. The first to be completed was the digital game, created using the Scratch platform from MIT. This platform was chosen for its ease of creating games and animations simply through drag-and-drop actions. Subsequently, the robotics game was developed using the Arduino platform. In addition to the robot, an Android application was created to control the actions taken by the robot.

Before acquiring the components and assembling the robot, the circuit, and part of the code were tested on the Tinkercad platform. This simulation allowed for the identification and correction of some existing errors in the formulation. The Fritzing tool was also used, enabling the creation of a replica of the circuits utilized in the project, thereby facilitating the identification of potential connection errors between the sensors and actuators employed.

Due to difficulties in accessing digital fabrication environments, it was decided to use ready-made robot chassis that are already available on the national market. Initially, the entire circuit was assembled on a rounded chassis (Figure 2a) purchased from a specialized website. However, during preliminary tests, it became evident that the robot's movement was affected by the two rotating wheels included in the chassis kit. This interference resulted in inaccuracies in the robot's movement on the map.

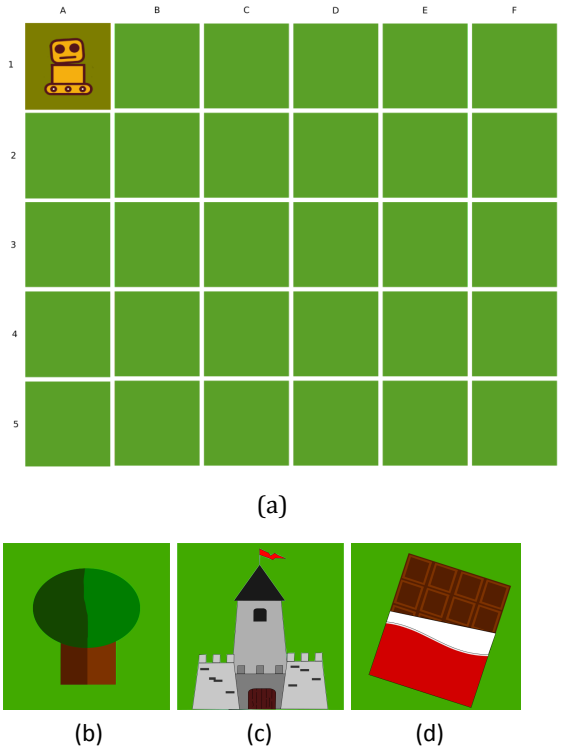
After several attempts to modify the chassis to correct the inaccuracies, it was decided to use a new chassis with different characteristics. The Falcon robotic platform was chosen (Figure 2b), which features a sliding sphere at its central axis. The use of this sphere was the key factor in selecting this chassis, as it does not interfere with the final movement.

(a) (b)

**Fig. 2 -** Chassis used in the project, with (a) the 2WD chassis and (b) the Falcon platform.

The first map developed in the project is presented in Figure 3a. It features a playful scenario with trees representing obstacles (Figure 3b), and the castle or chocolate, which are the objectives to be achieved (Figures 3c and 3d). These elements were printed on A4 paper and laminated to enhance durability. All of them are movable, allowing for a change of scenery with a simple rearrangement of the elements.

To reduce costs and allow for greater mobility, the game map was printed on banner material. The entire map was designed using an open-source vector drawing software called Inkscape. The dimensions defined in the software were 1.4m x 1.2m. These dimensions were determined based on the movement of the robot.




(a)



(b) (c) (d)

**Fig. 3 -** "(a) Map of the project's scenario developed using the Inkscape tool. Images of the scenario objects are (b) tree, (c) castle, and (d) chocolate."

## 5. Results and Discussions

Initially, the game was developed using Scratch. When starting the game, the player is required to choose the type of map for the match, as shown in (Figure 4). This screen presents three maps, each featuring a different scenario.

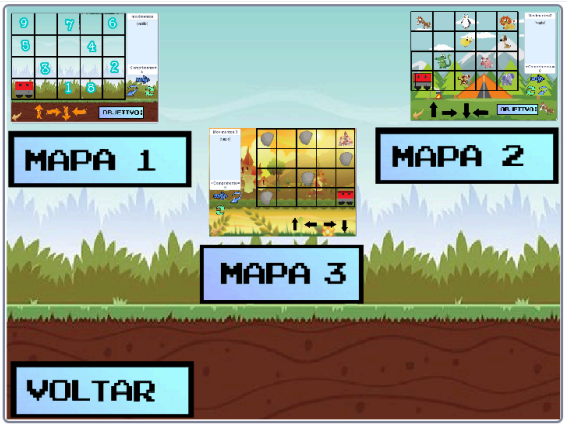
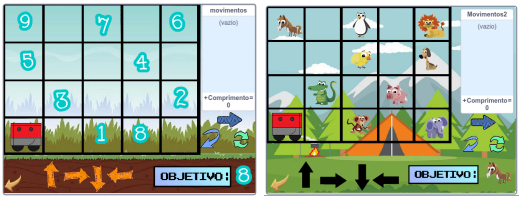



**Fig. 4 -** Game start screen.

Maps 1 and 2 follow the same gameplay mechanics. In each of them, a different element is randomly drawn on the screen for each round, with Map 1 focusing on numbers (Figure 5a) and Map 2 on animals (Figure 5b). After the draw, the player must create the algorithm for the robot's movement to reach the drawn number or animal.



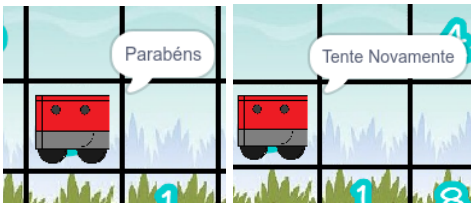

(a) (b)

**Fig. 5 -** Maps developed for the educational game

Map 3 (Figure 6) differs from the previous ones in that the objective remains the same: to reach the castle located on the map. However, the position of the castle changes randomly for each round.

**Fig. 6** - Map 3.

In all maps, upon reaching the goal, the character displays a message congratulating the player (Figure 7a). If the player fails to achieve the proposed objective, the displayed message advises them to try again (Figure 7b).




(a) (b)

**Fig. 7** - End-of-game information.

Next to the movement arrows used to create the movement algorithm, auxiliary commands are displayed (Figure 8). These include starting the movement, erasing the last movement, and restarting the round. Small figures should fit within the size of one column, while larger figures can span the width of two columns (plus the central white space).

**Fig. 8** - Actions available to the player

The physical fabrication required an investment of approximately US$ 60.00 in equipment. The physical version of the proposal consists of an Android application, a robot built using an Arduino Uno, a map printed on banner material, as well as elements that represent the objectives and obstacles on the map.

The main components that make up the developed robot are: an HC-05 module for Bluetooth communication, a 18650 battery as the power source for the DC motors of the wheels, an L298N H-bridge, and an Arduino Uno board powered by a 9V battery. The circuit of the robot, displaying these components, is shown in Figure 9.

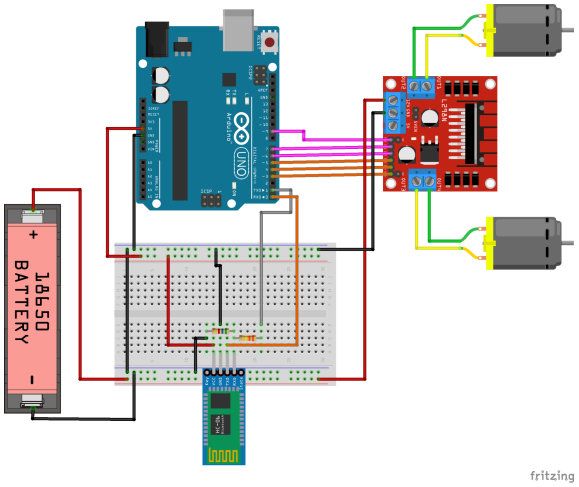
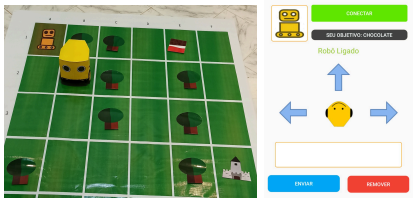


**Fig. 9** - Robot circuit created in Fritzing. Key components such as the Arduino Uno, Bluetooth module, H-bridge, and DC motors are visible.

The map, the robot, and the elements that make up the scenario are shown in Figure 10a. All interactions with the robot were carried out through an application (Figure 10b) developed using the Android SDK. In this application, the objective to be achieved in the robot's movement is randomly drawn. To accomplish this, the user needs to connect to the robot via Bluetooth communication and then develop the movement algorithm using the arrows located in the centre of the application.




(a) (b)

**Fig. 10** - (a) Image of the robot, map, and scenario objects; (b) application developed using Android SDK.

## 6. Conclusion

Computational thinking is a skill being incorporated into basic school curricula because it imparts important technological knowledge relevant to today's society. One way to encourage this knowledge is by introducing programming concepts to children.

Many projects use robotics and drag-and-drop tools as a way to engage children in programming, with Arduino and Scratch being two commonly used platforms for this purpose. This work utilized both tools to propose a game that incorporates basic programming concepts to stimulate computational thinking.

The proposal involves a robot that moves freely across a game board. The player must create the robot's movement (algorithm) using the available commands: up, down, right, and left. At the end of the movement, the robot must be in the designated target position.

Two versions were developed: one digital and the other physical. The digital version used Scratch and allowed for the validation of some construction ideas. It features three maps, each of which randomly assigns a target objective. The physical version utilized Arduino and components commonly used in robotics.

For the physical version, an application was created that communicates with the robot and allows the player to control it by sending movement commands. In all cases, the proposed maps are designed to be playful to engage the player.

As future work, it is expected to implement it in the classroom with the return of in-person classes in the municipal education system, as well as to create other map models for the developed physical game.

## 7. References

[1] Aho A. Computation and computational thinking. The Computer Journal. 2012; 55(7):832-835.

[2] Obama White House. Computer Science for All. Available at: <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>.

[3] Fundação Telefônica Vivo, Fundação Lemann. Programaê: Um guia para construção do pensamento computacional. 1st ed. 2018.

[4] Benitti F.B.V. Exploring the educational potential of robotics in schools: A systematic review. Computers & Education. 2012; 58(3):978-988.

[5] Aono A.H., Rody H.V.S., Musa D.L., Pereira V.A., Almeida J. A utilização do Scratch como ferramenta no ensino de pensamento computacional para crianças. In: WEI - Workshop sobre Educação em Computação. 2020; p. 2169-2178.

[6] Madeira, Charles. Introdução ao Pensamento Computacional com Scratch. CEUR Workshop Proceedings, vol. 1877, pp. 725–730, 2017.

[7] Falcão, T. P., Barbosa, R. 'Aperta o Play!' Análise da Interação Exploratória em um Jogo Baseado em Pensamento Computacional. Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015). 2015; vol. 1, p. 419.

[8] Da Silva, D. P., Sidnei, S., Jesus, Â., Silva, C. E. P. Aplicação de Robótica na Educação de Forma Gradual para o Estímulo do Pensamento Computacional. Anais dos Workshop do V Congresso Brasileiro de Informática na Educação (CBIE 2016). 2016; vol. 1, p. 1188.