robotics

MDPI

Article

Arduino-Based Mobile Robotics for Fostering Computational Thinking Development: An Empirical Study with Elementary School Students Using Problem-Based Learning Across Europe

Rolando Barradas [1,2,3], José Alberto Lencastre [4], Salviano Pinto Soares [1,2,3] and António Valente [1,5,*]

1 School of Sciences and Technology-Engineering Department (UTAD), 5000-801 Vila Real, Portugal; rolandobarradas@gmail.com (R.B.); salblues@utad.pt (S.P.S.)
2 Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, 3810-193 Aveiro, Portugal
3 Intelligent Systems Associate Laboratory (LASI), Escola de Engenharia—Universidade do Minho, Campus Azurém, 4800-058 Guimarães, Portugal
4 CIEd—Research Centre on Education, Institute of Education, University of Minho, Campus de Gualtar, 4704-553 Braga, Portugal; jlencastre@ie.uminho.pt
5 INESC TEC, Campus da Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
* Correspondence: avalente@utad.pt

Abstract: The present article explores the impact of educational robotics on fostering computational thinking and problem-solving skills in elementary school students through a problem-based learning approach. This study involved the creation of a framework which includes a robot and two eBooks designed for students and teachers. The eBooks serve as a guide to the construction and programming of a small Arduino-based robot. Through integration with gamification elements, the model features a narrative with three characters to boost a student's engagement and motivation. Through iteration of heuristic evaluations and practical tests, we refined the initial theoretical framework. An empirical study was conducted in two phases involving 350 students. The first empirical test involved a small group of 21 students, similar to end users, from five European schools. With a 100% completion rate for the tasks, 73.47% of these tasks were solved optimally. Later, we conducted a larger validation study which involved 329 students in a Portuguese school. This second phase of the study was conducted during the 2022–2023 and 2023–2024 school years with three study groups. The results led to a 91.13% success rate in problem-solving activities, and 56.99% of those students achieved optimal solutions. Advanced statistical techniques, including ANOVA, were applied to account for group differences and ensure the robustness of the findings. This study demonstrates that the proposed model which integrates educational robotics with problem-based learning effectively promotes computational thinking and problem-solving skills, which are essential for the 21st century. These findings support the inclusion of robotics into primary school curricula and provide a validated framework for educators.

Keywords: computational thinking; problem solving; educational robotics; problem-based learning; constructivism; STEAM education

Citation: Barradas, R.; Lencastre, J.A.; Soares, S. P.; Valente, A. Arduino-Based Mobile Robotics for Fostering Computational Thinking Development: An Empirical Study with Elementary School Students Using Problem-Based Learning Across Europe. *Robotics* **2024**, *13*, 159. https://doi.org/10.3390/robotics13110159

Academic Editor: Igor Verner

Received: 14 August 2024
Revised: 12 October 2024
Accepted: 21 October 2024
Published: 29 October 2024

1. Introduction

The development of computational thinking stimulates an increase in the analytical capacity of children and is considered to be a fundamental skill in the 21st century, due to its use in different areas of knowledge [1,2]. It is connected to the development of skills such as abstract, algorithmic, logical, and scalable thinking [2], which transpose to the daily lives of young people, allowing them to become more reflective and critical [3].

In the 1980s, Papert introduced its Turtle robot, controlled by the LOGO programming language, "to serve as a model for other objects, yet to be invented" [4]. Robotics became an important means to develop computational thinking and pushed further the penetration of programming and robotics in schools.

Robots have become an essential part of contemporary educational environments, encouraging both hands-on learning and social engagement [5]. They can be as basic as LEGO Mindstorms kits or as complex as humanoid robots like Nao. Robots help students develop their critical thinking and creativity skills and support both constructionism and constructivism.

Through the construction and programming of robots, students are exploring important computational concepts such as algorithms, sequences, loops, and conditionals. They also develop other essential skills like debugging and abstraction. With all of these benefits, we would expect that robotics would succeed in educational contexts. However, this is not the case. Some studies concluded that, mainly due to the lack of guidelines on how to integrate robotics into education, teachers and schools are not exploring the benefits which robotics could bring about [6]. Other studies referred to the need to create large common activities in order to make robotics in education stronger and sustainable [7].

1.1. Research Questions and Objectives

The main research question of this study was to find how the integration of educational robotics and problem-based learning environments fosters the development of computational thinking in elementary school students. Within this context, we investigated the relationship between the use of robotics in Science, Technology, Engineering, Arts, and Mathematics (STEAM) activities and computational thinking in both controlled and real classroom environments. The objective was to determine if robotics is an effective tool for fostering computational thinking skills.

The present work continues and reports on previous studies by Barradas et al. [8], aligning with Papert's constructionist theory [9] in which children construct knowledge more effectively when creating tangible artifacts, such as building and programming robots. This approach aligns with problem-based learning, where students learn while solving real-world problems.

Previous studies focused on the development of *Stemie*, an Arduino-based robot used for the the development of computational thinking and problem-solving skills. Using the development research methodology by Van den Akker [10], we created and refined *Stemie*, a supporting programming framework, and two interactive eBooks: one for students and one for teachers. These resources' objective is to provide a structured environment for guiding students through problem-solving tasks and, at the same time, support teachers in the implementation of these activities.

1.2. Study Limitations

It is also relevant to mention some possible limitations of this study. The school contexts in which this study was conducted may have had some influence on the results. In both phases of this study, each of the involved students had at their disposal one robotics kit for their personal use. This is something that is quite difficult to find in most schools. Also, the teachers involved had already been given special training on robotics and innovative teaching methodologies. This may have also contributed to more positive results, as educating with robotics requires teachers to guide students, troubleshoot issues, and link the activities to specific learning objectives [11]. This may give rise to some difficulties when trying to replicate this study in different schools which may have less robotics-motivated teachers. This study's results may also be limited by the absence of pretesting of computational thinking and problem-solving skills prior to the students' training.

2. Background

The need for frameworks which integrate theoretical approaches like constructivism and constructionism to make educational robots more accessible and impactful has been noticed for several years [12]. In our idea on how to respond to this need and emphasize active learning through experience, we also added computational thinking and educational robotics. In this way, we provided students with hands-on activities to develop critical problem-solving skills. Also, motivation and competition play a fundamental part in engagement and collaboration by creating a dynamic learning environment.

2.1. Construtivism

In 1971, Jean Piaget published a book where he outlined the different stages of children's cognitive development over time, introducing the concept of construtivism [13]. In this theory, children's thinking evolves over time, and each learner is responsible for the construction of their own knowledge through their interactions with the surrounding environment.

Since then, many authors have focused on this topic. Coll and Salé [14] stated that constructivism, in its broadest sense, is an articulated set of principles from which it is possible to diagnose, create opinions, and make informed decisions about teaching. Schunk [15] considered that constructivism is not a learning theory but rather an epistemology or philosophical explanation of the nature of learning. For the constructivist conception, the learning process happens when we are able to create a personal representation of an object or content which we want to learn [14]. Learning has an active characteristic, which is a result of the personal construction of knowledge, and can follow different paths or forms, depending on the learner's cultural involvement. In this way, personal constructions are valid for a given individual but not necessarily for others, since knowledge, being formed within each person, does not emanate from the outside [15]. Despite this individual construction, constructivism does not deny social interaction, since it not just is constructed but also taught and learned [14]. Constructivism emphasizes the integrated curriculum in which a given subject must be studied from different perspectives, which means that teachers are forced to give up the instructional method and are able to structure their classes so that students are actively involved in the pursuit of knowledge through the manipulation of content and social interaction, making use, for example, of collaborative work activities [15]. For constructivism, teaching is a shared process in which a student receives help from his or her teacher in order to become competent and autonomous in solving tasks and using concepts [14]. For this, it is necessary to teach students to take an active part in their learning, set goals, and monitor and evaluate their progress, extending beyond typical passive behavior to creating activities and experiences which challenge their thinking [15] and lead them to see the classroom differently. A constructivist school is one in which students learn and develop by building personal meanings around the contents which are part of the school curriculum [14].

2.2. Constructionism

Papert [9] refers to constructionism in his work, closely linking it to the notion of constructivism. For Papert, constructionism shares the idea of learning with constructivism. Papert's theory added the fact that learning happens in a context in which the learner is consciously committed to building something tangible, namely using a computer and creating knowledge through making and doing.

The constructionist way of teaching implies performing a task in a way where one can produce the most results with less effort for the teacher, namely by letting children find for themselves the knowledge they need to solve a certain problem [16]. Students find solutions for their problem-solving experiments by following their own research and ways of learning. Constructionism has its focus on the connected nature of knowledge with its personal and social dimensions [17]. Because "teaching without curriculum does not mean spontaneous, free-form classrooms or simply 'leaving the child alone"' [4], it is important that in constructionist experiences, the teacher supports students on the process of creating their intellectual structures.

Referring to the use of educational software, to explain his notion, Papert [4] stated that by using a programming language like Logo, children will be able to create their own educational software and learn much more while they think, design, and build than if they simply used it.

2.3. Computational Thinking

Computational thinking is defined as a set of procedures involved in articulating a problem and its solutions such that a computer—human or machine—can solve it successfully [18]. The development of computational thinking fosters skills such as **abstract thinking** (understanding and solving problems with different degrees of abstraction); **algorithmic thinking** (finding solutions in stages to determine the most efficient one); **logical thinking** (formulating and rejecting hypotheses); and **measurable thinking** (dissecting a big problem into smaller ones or combining small parts to formulate a big solution) [2].

Brennan and Resnick [3] developed a reference framework for studying and assessing the evolution of computational thinking. This framework includes three dimensions: (1) **computational concepts**, (2) **computational practices**, and (3) **computational perspectives**.

They identified seven **computational concepts**:

- **Sequences** are collections of instructions or steps in a specific order which can be used to finish a coding task;
- **Loops** are structures which allow one to run the same instructions repeatedly to solve problems which have patterns of repetition;
- **Events** are specific occurrences which result in particular actions;
- **Parallelism** occurs when multiple sequences might happen simultaneously in order to solve a problem;
- **Conditionals** enable a program to use certain structures to make decisions;
- **Operators** represent and resolve logical and mathematical operations;
- **Data** are values kept in variables which are stored, retrieved, and updated using specific structures.

Related to **computational practices**, shifting the emphasis from what is learned to how it is learned [3], four categories were identified:

- **Being iterative and incremental** is a practice which occurs when children assess whether a project is successful or not and come up with new ways to solve problems;
- **Testing and debugging** occur when children check what does not work and fix errors through trial-and-error procedures and the analysis of previously created situations;
- **Reusing and remixing** happen when one constructs something using previously completed projects or projects from others and gains knowledge;
- **Abstracting and modularizing** constitute the process of creating something large by connecting groups of smaller components.

Referring to **computational perspectives**, Brennan and Resnick classified children's perspectives on computing into three main categories:

- **Express**: Through the use of computers, students can express themselves creatively and begin to see themselves as builders rather than merely consumers;
- **Collaborate and Connect**: Using computers enables students to create with and for other people, inspiring them to take on new projects and fostering the growth of a critical spirit;
- **Questioning**: Technology is being questioned by technology. Investigating the methods used to solve some issues may cause one to doubt and question on how other real-world scenarios operate [19].

As a result, by using this framework, it will be possible to assess how young people's computational thinking develops by examining how their projects and experiences, created with the three specified dimensions, are carried out.

Learning how to use computers and thinking computationally can change the way children learn everything else [4].

2.4. Educational Robotics

Educational robotics is increasingly used in classrooms to implement activities which aim to develop students' computational thinking skills. Robotics provides a hands-on, interactive environment for students to apply skills such as problem solving, abstraction, decomposition, and pattern recognition along with algorithms.

Problem solving is closely associated with computational thinking [20], and by using robotics to solve real-world challenges, educators can foster both and spark their students' creativity. As Jonassen [20] mentioned, students can undertake the most relevant activities when they are developing their problem-solving skills because the knowledge they acquire during the process is easier to understand and retain. Students develop their domain of processes and "learn how to learn" [16] while solving problems on their own rather than waiting for a teacher to provide an answer [21]. The ones who use this strategy become more motivated and take the lead in the learning process.

For some years now, companies like LEGO, for example, have been creating educational robotic kits such as LEGO Mindstorms NXT or EV3 systems. Those kits mix robots and technology with the main ideas of constructivism. With building materials, sensors, actuators, and a main processor unit, these kits allow children to build and later program robot behaviors by using a programming interface on a computer or tablet.

However, although technology plays an essential role in constructivism, the central focus is not on the machine but on the mind [4].

2.5. Motivation and Competition

The role of motivation in the learning process is among the most important ones. Motivation makes a student define his or her goals and use cognitive (e.g., planning and monitoring) and behavioral (e.g., persistence and effort) processes to achieve them [15].

During the learning process, ideas regarding the contents are constructed as well as the teaching situation itself. Depending on the motivation, it can be either stimulating and challenging or boring and without any particular interest. Associated with these ideas are the representations which each person builds for themselves [22], which influence motivation. One of the ways to stimulate motivation is through competition [23], and competition-based learning has been used for some time to teach robotics as it allows applying math, physics, and other scientific subjects.

Competition is one of the key factors for motivation, and obtaining physical results actively contributes to the development of independence and leadership skills and also promotes a positive educational process [24].

Recent studies showed that students who are motivated in robotics activities perform better in practical challenges. Hands-on learning using robots allows students to better express their technical skills. These findings suggest that incorporating motivational tactics into robotics education could improve the learning experience, particularly by personalizing activities to students' goals and learning styles [25].

However, despite all of its benefits, using competition in educational scenarios is somehow difficult to implement. Robotics competitions are expensive to prepare and organize, and the number of participants is limited [26] compared with the students in a classroom or school.

These facts make it extremely important to find ways to include robotics in the curriculum, both as a learning objective in information and communications technology (ICT) or robotics classes and as a tool to facilitate learning in other subjects, such as mathematics and physics.

3. Method

The used methodology is one of the most important parts in a study. We focused on development research by Van den Akker [10], chosen to iteratively refine the educational tools we created by allowing continuous improvement through empirical testing and evaluation.

Development Research

Using the development research methodology, we were able to develop an intervention, starting from the analysis of practical problems to the development of solutions within a theoretical framework, followed by evaluation and testing of solutions in the field to ensure documentation and reflection were carried out, which could lead to future investigations.

Akker divided the process into four stages or cyclical research activities:

- Preliminary investigation is the first stage, in which the problem and its context are analyzed in order to know the target audience, with their expectations and motivations, but also their limitations and relate them to the state of the art in the literature;
- Theoretical embedding is a stage of the research process where solution designs are formed based on knowledge obtained from the state of the art, in conjunction with specific learning objectives;
- Empirical testing is the longest stage of the process in which the previously designed prototype is developed and evaluated, with the aim of confirming the choices made in the previous stage. The phase of evaluating the process is fundamental and must exist from the beginning, providing information to feed the entire cyclical process of intervention design and development and improving the intervention itself and the construction of a prototype;
- Documentation and analysis of and reflection upon a process and its outcomes is the stage in which much attention must be paid to systematic documentation and analysis of and reflection upon the design, development, evaluation, and implementation of a process and its results [10].

This choice was based on the fact that, as stated by Van den Akker [10], it is a methodology which provides more adequate information and allows the creation of a working basis for a designer's choices and timely feedback which can be used to improve a product or intervention.

4. Preliminary Investigation

Based on the previous works from Barradas et al. on the development of *Stemie* the robot [8], we started by forming a diagnosis of the good practices in using mobile robotics. By contacting several European entities, we created a partnership with five schools from Portugal, Croatia, Italy, Lithuania, and Turkey. This partnership, within the context of the Erasmus+ program, allowed us to further develop and test our previous works. We surveyed each partner to find the state of the art of each of their countries' teachers for mobile robotics. These states of art helped us to understand what they used and learn about mobile robotics and innovative pedagogical scenarios. After data analysis of the questionnaires, we verified that the economic factor was one of the main reasons for the low penetration of robots in schools. This factor limited activities and experiences to small groups of students. Also limiting robotics penetration in schools is the lack of training that the teachers have. Most of them, even with previous approaches to robotics in extracurricular activities, used rather different methodologies, with some having low productivity indexes.

Communication was a highly important part in this stage, as the teachers involved shared guidelines and knowledge regarding training plans for mobile robotics. Taking into consideration the local contexts of the schools, we designed a training plan which would put all of the involved teachers at the same level of knowledge on the subject, preparing them from the scientific and pedagogical points of view for using robotics in an educational context with their students.

The practical side of our work was then put to proof as the teachers tested the plan themselves and with other invited teachers from their own schools, with quite positive evaluations.

This first training plan led to the creation of a set of 21 STEM-related exercises published in two books which were aimed at students and teachers, providing an easier way for students to develop skills such as computational thinking and problem solving and a guided way for teachers to support them. The exercises were organized in the student's book [27] as a sequence of tasks which build upon the knowledge created by previous ones, aligning with key principles of computational thinking such as decomposition, pattern recognition, algorithm design, and debugging. Each exercise is framed around computational concepts and practices and explicitly connected to concepts like sequences, loops, conditionals, and data handling as well as practices like iterative refinement, testing, reusing, and modularizing. It is important to remember that students are not just following instructions. In certain exercises, like creating music with robots or navigating paths, students are expressing their creativity while modifying or creating new code. Also, collaboration and communication are being fostered, with many exercises including discussion with peers to compare solutions. Similarly, the exercises are also directly linked to computational practices. All exercises were designed with a correlation to real-world problems. The robotics challenges lead students to develop skills like understanding a problem, planning, implementation, testing, and debugging. Examples of exercises, or challenges as they are called in the eBook, include the following:

- Challenge 1 ("Learning to Drive"): This helps students understand the importance of precision while programming motor controls;
- Challenge 7 ("Creating Sequences and Loops"): This challenge focuses on the computational concepts of sequences and loops, which are essential for code complexity reduction;
- Challenge 10 ("Avoiding Obstacles"): Here, students need to combine conditionals and sensor readings to use real-time data in decision making while driving their robots.

The idea behind creating these specific exercises was also for students to be able to relate the content to real-world applications like autonomous navigation systems and obstacle avoidance while always applying other STEAM knowledge.

It is important to remember that each of the partner schools chose a set of 4–5 students to work with and serve as a control group of students similar to end users. Upon completion of the student's eBook, this control group was used to test all of the created exercises.

5. Theoretical Embedding

One of the stages which was executed several times in this investigation was theoretical embedding. In the first cycle, we drew the first activities and had them tested by experts in heuristic evaluations. Our main idea was to propose a model or framework to help teachers design, implement, and evaluate educational robotics activities aimed at developing computational thinking. This proposed model should include specific STEM-related activities which allow children to develop relevant computational thinking concepts, practices, and perspectives. To accomplish this, we developed two complementary eBooks: one for students and another for teachers.

5.1. Stemie & Stemia's Fantastic Journey Through Space

To make this model attractive to children, we decided to use some game elements. Gamification [28] is an effective way to keep users active, engaged, and motivated [29] as well as learn through new means and enjoy otherwise tedious tasks. Comparing the student to a hero in a game, we created three characters which would follow him or her in the learning process through a global context and narrative. For the teacher, it is necessary to move away from content-based approaches to students and use different ones in trying to seduce them in their search for knowledge.

We created a student's guidebook [27] with activities to help them build and program a small robot. In addition to building instructions, this learning toolkit (see Figure 1) includes an introduction to Scratch and mBlock as well as exercise cards to help students develop the necessary programming skills to operate their robots. All proposed activities have the development of problem solving and computational thinking skills in mind.

Figure 1. Student's eBook cover.

We chose the eBook format to allow every educator and student to use it almost immediately. It also allows some level of interaction, as it had internet links to relevant and related websites and included audio, video, and animations relevant to each specific content.

5.1.1. Main Characters

The eBook uses storytelling to create a narrative which keeps students engaged. Using the three characters depicted in Figure 2, this approach gives more meaning to the exercises as students can better understand the practical results of their programming. Two of the characters represent the student's robot (*Stemie* and *Stemia*) and another one (At), a helper throughout the book, represents the AtMega processor equipped by the robots. The process of creation was performed with the help of our control group, with the intention of making the book and challenges even more attractive to students.

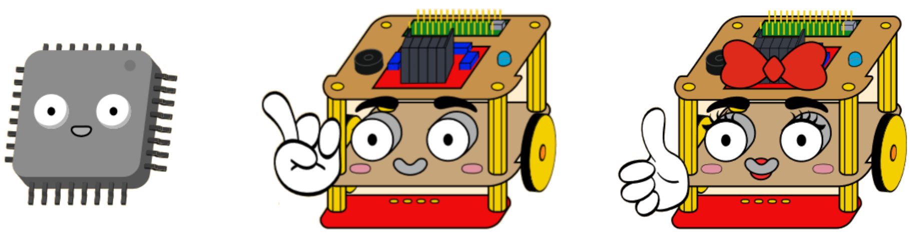


Figure 2. At, *Stemie*, and *Stemia*, the main characters of the eBook.

Our main idea was to promote emotional engagement as, when students identify with the characters, they become emotionally involved in the narrative [30,31]. Characters situate educational content within a story, making the contents easier to understand and retain as their actions, alongside with the narrative, can show or demonstrate some practical application of that knowledge in the real world. This emotional involvement would potentially increase motivation and improve learning outcomes.

5.1.2. Narrative

In a gamified educational activity, the narrative is one of the most important elements. The narrative needs to have the ability to control the user's experience in such a way that it can guide a student through the specific contents [32]. Well-constructed stories can capture students' attention and create an emotional connection with the content. A narrative provides context for challenges and activities, turning learning into something meaningful such that students can see the practical application of what they are learning. Through storytelling, it is possible to develop problem solving, computational and critical thinking and collaboration, as a narrative often presents challenges which require these skills to be solved.

For this eBook, we created a narrative on two robots which were traveling through our solar system (Figure 3) and had a problem in their spaceship while passing through the asteroid belt. Their ship failed, and they disintegrated entering Earth's atmosphere. The student's first mission is to help reassemble the robots. The remaining content was created to guide the student through all of the robot's components, helping them to understand their use and how to program them. In each mission, the story unfolds in such a way that students are led to learn about their robots and about other STEM subjects. They need to complete one mission before entering the next one.

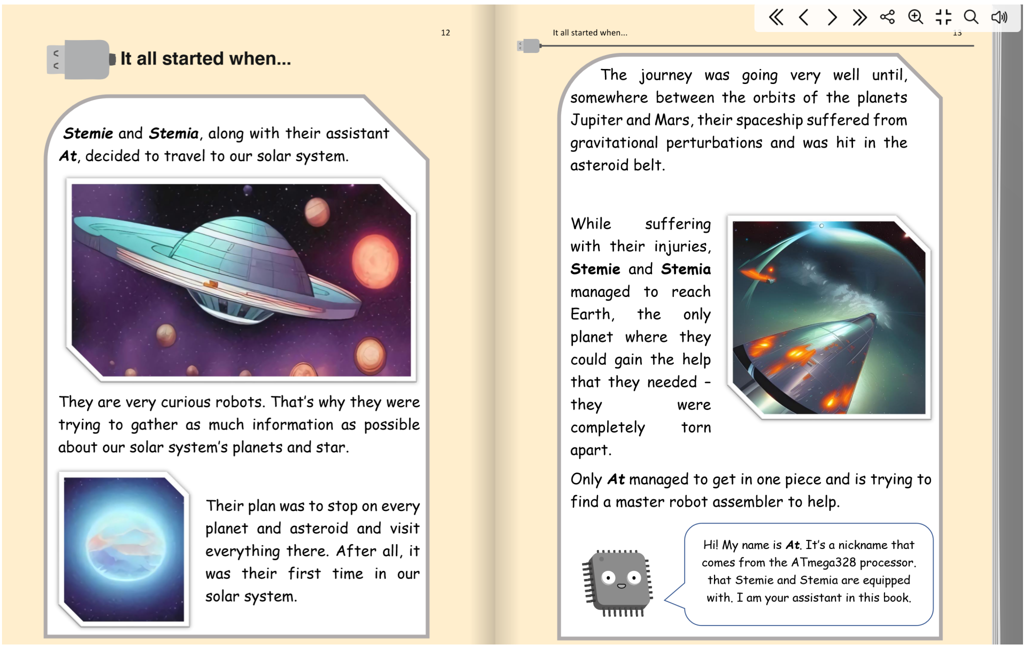


Figure 3. Narrative in the student's eBook.

5.1.3. eBook Organization

As previously mentioned, the problems or missions proposed to the students through the eBook were created to foster the development of computational thinking and problem-solving skills. Together with the narrative, students are led through separated chapters for each of the subjects. In Figure 4, we show the table of contents, and it is possible to have a glimpse at how the narrative flows through all of the chapters and how each chapter relates to a specific robot component.

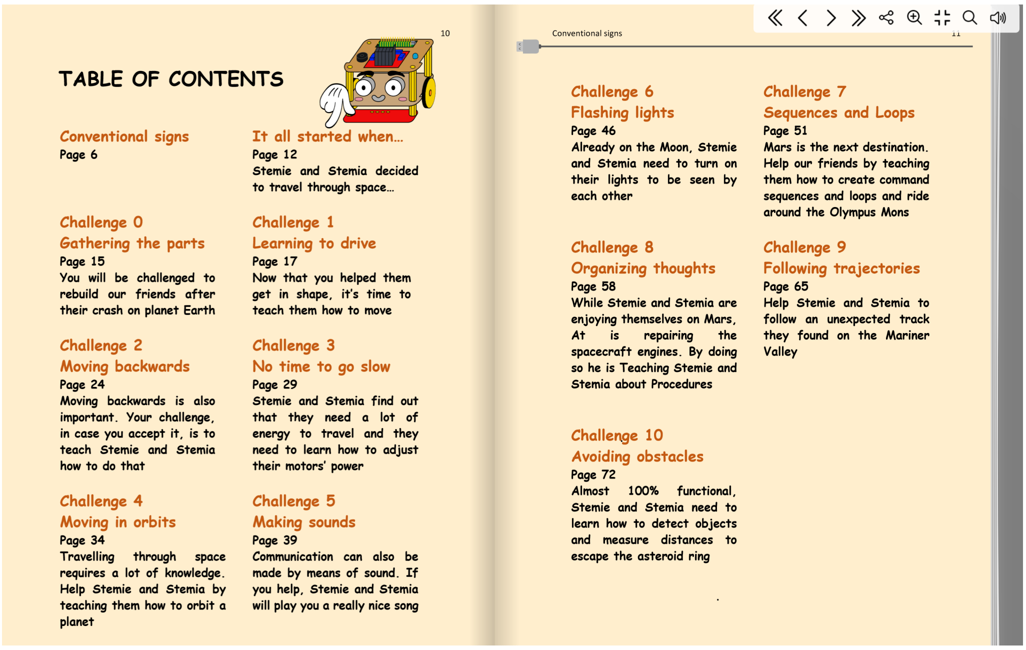


TABLE OF CONTENTS

Conventional signs
Page 6

It all started when…
Page 12
Stemie and Stemia decided to travel through space…

Challenge 0
Gathering the parts
Page 15
You will be challenged to rebuild our friends after their crash on planet Earth

Challenge 1
Learning to drive
Page 17
Now that you helped them get in shape, it's time to teach them how to move

Challenge 2
Moving backwards
Page 24
Moving backwards is also important. Your challenge, in case you accept it, is to teach Stemie and Stemia how to do that

Challenge 3
No time to go slow
Page 29
Stemie and Stemia find out that they need a lot of energy to travel and they need to learn how to adjust their motors' power

Challenge 4
Moving in orbits
Page 34
Travelling through space requires a lot of knowledge. Help Stemie and Stemia by teaching them how to orbit a planet

Challenge 5
Making sounds
Page 39
Communication can also be made by means of sound. If you help, Stemie and Stemia will play you a really nice song

Conventional signs

Challenge 6
Flashing lights
Page 46
Already on the Moon, Stemie and Stemia need to turn on their lights to be seen by each other

Challenge 7
Sequences and Loops
Page 51
Mars is the next destination. Help our friends by teaching them how to create command sequences and loops and ride around the Olympus Mons

Challenge 8
Organizing thoughts
Page 58
While Stemie and Stemia are enjoying themselves on Mars, At is repairing the spacecraft engines. By doing so he is Teaching Stemie and Stemia about Procedures

Challenge 9
Following trajectories
Page 65
Help Stemie and Stemia to follow an unexpected track they found on the Mariner Valley

Challenge 10
Avoiding obstacles
Page 72
Almost 100% functional, Stemie and Stemia need to learn how to detect objects and measure distances to escape the asteroid ring

Figure 4. Student's eBook table of contents.

5.1.4. Self-Efficacy Control Tools

In each chapter, we wanted to include something that would allow students to know exactly what they were doing well and where they could improve. The objective was, as in this entire study, to make students more motivated to try, make mistakes, and learn. This trust is fundamental in an educational environment and contributes to the development of skills. For this, at the end of each mission, we created a small grid (Figure 5) for self-efficacy where students could track their own progress, making them feel responsible for and in control of their learning.

I can:	NOT YET	MORE OR LESS	YES, COMPLETELY
Name different measurements			
Program the robot to go straight			
Where can these ideas be applied in real word?			

Figure 5. Automatic evaluation in student's eBook.

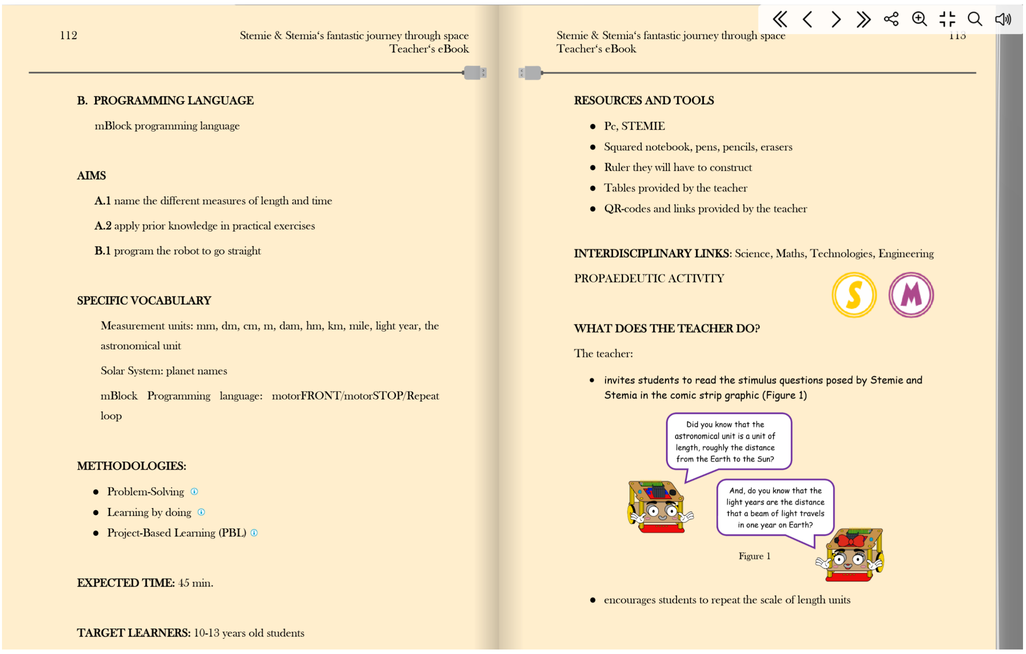
With the teacher's help, students completed the entries in the grid, indicating what knowledge they gained while solving the problem, how they tried out the experience, what they experienced, and what the outcome of the performed operations was. They were also asked to think about the meaning they attached to the experience by explaining or thinking about its value in real life.

5.1.5. Teacher's eBook

Recent studies emphasize the need for well-structured teacher guides which focus on translating complicated concepts into classroom activities [33]. Reinforcing teachers' vital role in promoting problem solving through robotics ensures that both teachers and students can receive the full benefit of educational robotics.

In this process, we also created a complementary manual [34] to help teachers implement innovative pedagogical practices when using mobile robotics in educational scenarios.

The teaching toolkit contains step-by-step directions and lesson plans (see Figure 6) so that every teacher will be able to teach mobile robotics.



112 Stemie & Stemia's fantastic journey through space Teacher's eBook

B. PROGRAMMING LANGUAGE

mBlock programming language

AIMS

A.1 name the different measures of length and time

A.2 apply prior knowledge in practical exercises

B.1 program the robot to go straight

SPECIFIC VOCABULARY

Measurement units: mm, dm, cm, m, dam, hm, km, mile, light year, the astronomical unit

Solar System: planet names

mBlock Programming language: motorFRONT/motorSTOP/Repeat loop

METHODOLOGIES:

- Problem-Solving
- Learning by doing
- Project-Based Learning (PBL)

EXPECTED TIME: 45 min.

TARGET LEARNERS: 10-13 years old students

Stemie & Stemia's fantastic journey through space Teacher's eBook 113

RESOURCES AND TOOLS

- Pc, STEMIE
- Squared notebook, pens, pencils, erasers
- Ruler they will have to construct
- Tables provided by the teacher
- QR-codes and links provided by the teacher

INTERDISCIPLINARY LINKS: Science, Maths, Technologies, Engineering

PROPAEDEUTIC ACTIVITY

WHAT DOES THE TEACHER DO?

The teacher:

- invites students to read the stimulus questions posed by Stemie and Stemia in the comic strip graphic (Figure 1)

Figure 1

- encourages students to repeat the scale of length units

Figure 6. Example of a lesson plan in the teaching toolkit.

The lesson plans, with a gradual increase in complexity, ensure students build a strong foundation before having to solve more complex tasks. In addition to programming tasks, this manual contains activities and scenarios intended for STEM fields. It allows the same level of interaction as the student's eBook as it has internet links to relevant and related websites. It also includes audio, video, and animations relevant to each specific piece of content. The teacher's eBook also has suggestions for what the teacher should do during and at the end of each one of the missions.

6. Empirical Testing

After complete development of the design we obtained in the previous stage, we ended up with 21 robotics exercises, along with other STEM-related exercises to use with *Stemie*. These covered all computational concepts, practices, and perspectives identified by Brennan and Resnick [3]. Together with the previously developed *Stemie*, these exercises could form a valid model to stimulate computational thinking by means of educational robotics.

6.1. Evaluation: Stage 1

To prove the validity of our work, we organized a first evaluation with 21 students from the 6th and 7th grade studying at the 5 European schools from our partnership into small tutored groups. From January 2022 to July 2022, each teacher worked with a small group of students (4 from each of the countries) similar to the end users. Each group spent approximately 1 hour per week with their teachers performing tasks related to robot assembly and testing each of the 21 previously created exercises. The small groups were selected from a larger group of students, and the ones which showed the most interest in robotics were selected in each country. Prior to this test, all of these students already had contact with robotics but not with the robot *Stemie* or the programming framework. During the time with their teachers, the students learned how to build and program their robots and use all of its sensors and actuators. It was also intended that they develop their programming techniques, problem solving, and computational thinking skills.

In March, May, and July 2022 all the groups met in person for 5 days (Figure 7).

Figure 7. Students training.

The teachers created several goal-oriented approaches to further train problem-solving skills and put their newly found skills into practice in several robotic competitions. To evaluate students' problem-solving Skills, the group of teachers used rubrics for "problem identification", "planning", and "execution". To evaluate computational thinking, the responsible teachers created a set of specific objectives for each of the activities. All teachers' records were created using a 3 point Likert scale with "not yet", "more or less" (non-optimal solution), and "yes, completely" (optimal solution) for each of the rubrics and specific problem objectives. The teachers defined the optimal solutions primarily based on the accuracy of the results. The accuracy of the solution was determined by comparing the final result with the specific objectives of each activity. Depending on the activity, objectives could be, for example, navigating a certain path, creating a set of commands, avoiding objects in a path, or creating certain movement sequences. The efficiency of the solution was also considered to determine the optimal solutions. Solutions which accomplished the tasks using fewer lines of code could also be considered optimal, even if the accuracy was not the best. This criterion was taken into consideration because sometimes, errors occurred due to factors like hardware malfunctions, bad sensor readings, current, or actuator failure. These types of errors sometimes caused inaccurate solutions, even with a correct program. In these cases, a solution could be considered optimal even with low accuracy values. Every activity was recorded on video for later evaluation by the collective of teachers.

While carrying out these activities, the students were encouraged to use positive competition to achieve their goals and trained in working as a team with people they did not previously know, all while creating extra motivation to learn foreign languages and be able to communicate with their team members.

Competition was at its best with the students competing with but also helping their partners. At the end of each meeting, we awarded either a small cup or a medal to the best students in each of the activities (Figure 8).

Figure 8. Some of the competition winners.

6.2. Documentation, Analysis, and Reflection

The first test on our product and model to stimulate computational thinking by means of educational robotics led to the overall results shown in Table 1.

Table 1. Overall results.

	Different Problems/Total Analyzed	Optimal Solution	Non-Optimal Solution	Not Solved	Completion Rate	Completion Rate with Optimal Solution	Not Finished
Global results	21/441	324	117	0	100%	73.47%	0%

When analyzing these first results, it is possible to observe that we reached a completion rate of 100%, among which 73.47% of the results were what was considered an optimal solution. Globally speaking, it was a frankly positive result.

By using descriptive statistics analysis in Jamovi open statistical software [35], we were able to obtain some extra information on the students' performance across the 21 exercises. All of the exercises registered mean scores above 17 (with a maximum of 20). Exercises 12, 16, and 17 registered a standard deviation of zero due to all of the students achieving the maximum score. On the other hand, exercises 7, 8, 14, and 15 registered high standard deviations (2.91), which indicates that the students may have found these exercises more challenging, resulting in a wider range of final results. In exercises 13, 18, 19, and 20, the rounded value of the standard deviation was one, which indicates a consistent level of results.

Also using Jamovi, we conducted an analysis of variance, namely a one-way ANOVA using Fisher's method with a descriptives table, to determine if there were significant differences in performance between the different groups of students, categorized by country, for each of the exercises. This allowed us to identify whether these variations were statistically significant or not and determine which factors may have influenced the students' performance.

The one-way ANOVA results show the variance between groups (countries) and within each of the groups, with individual exercise scores within the same country. After analysis of the results, we found that for exercises 1–6, an F value of "Inf" and p values < 0.001 suggest a large separation of group means and zero variance within the groups. For exercices 12, 16, and 17, the F value of "NaN" occurred and was already expected because there was zero variance in the results for the within-group and between-group comparisons, as every student achieved the same result. However, differences occurred in the remaining exercises. For exercises 7–11 and 13–15, the F value results ranged from 5.33 to 13.46, with significant p values of <0.001 to 0.006, which indicates significant differences between the groups. Exercises 18–20 revealed F values of 6.74, 13.46, and 21.71 with p values of <0.001, 0.002, and <0.001, respectively, indicating notable differences in the group means. Exercise 21 had the highest F value (36.95), which indicates the existence of extremely strong group differences. This analysis, with high F values and significant p values for most of the exercises, suggests that there were strong group effects.

The group descriptive statistics showed that Croatia and Lithuania obtained the higher mean scores for most of the exercises. On the other hand, Turkey and Italy obtained the lower mean scores. Portugal's performance was not constant, with low mean scores of 16.4 in some exercises but with better performance in others.

For our data, as the sample sizes for each of the groups was equal or quite similar, Tukey's HSD test was chosen to perform the post hoc analysis for the four most significant variances. The results showed that Croatia and Lithuania obtained better results in the four exercises which we analyzed, with no significant variances between the two groups. Italy obtained lower scores in some of the exercises, with significant differences among both Croatia and Lithuania. Portugal and Turkey did not show significant differences from each other in any of the exercises but stayed below the two best groups.

When confronted with these conclusions, the responsible teachers debated possible explanations. Exercise 21 was one of the most demanding, as the students would need to have gathered knowledge of every other exercise to solve it. Thus, it is not surprising that the F value for it was so high. To these results, we may also contribute the fact that, although in the same number, the students from Portugal, Italy, and Turkey, by the schools' decision, were not the same in each of the training weeks. This fact caused the students from those countries to have less experience in both the robot and programming frameworks than the students from both Croatia and Lithuania, which remained the same throughout the study.

A different analysis allowed us to decompose the results into computational concepts, obtaining the results in Table 2 and Figure 9.

Table 2. Initial results, grouped by computational concepts.

Concept	No. of Different Problems/Total Analyzed	Completion Rate	Completion Rate with Optimal Solution	Not Finished
Sequences	21/441	100%	73.47%	0%
Loops	10/210	100%	78.57%	0%
Events	3/63	100%	93.65%	0%
Parallelism	3/63	100%	93.65%	0%
Conditionals	3/63	100%	93.65%	0%
Operators	3/63	100%	93.65%	0%
Data	5/105	100%	96.19%	0%

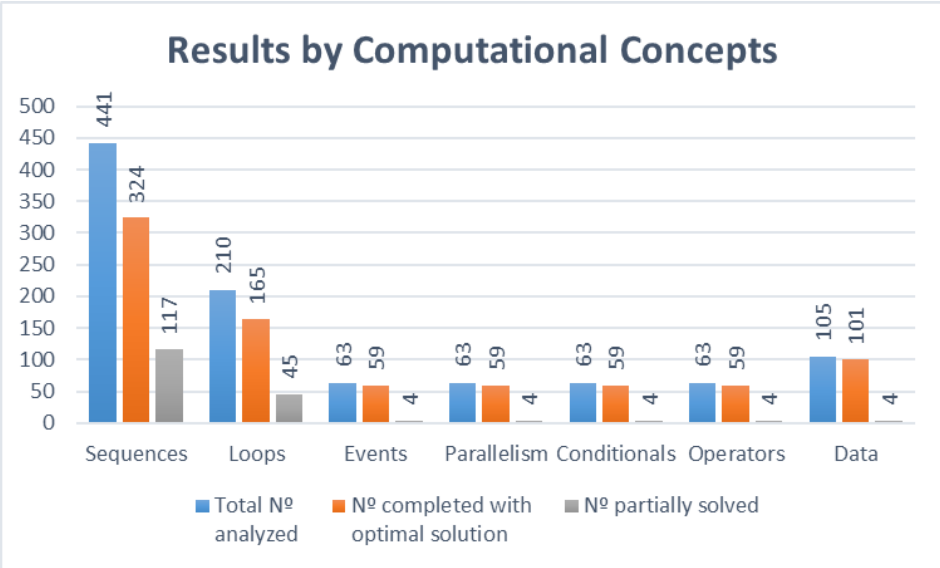


Figure 9. Summary of completion and optimal solution rates by concept.

The results, grouped by computational practices, can be observed in Table 3.

Table 3. Initial results, grouped by computational practices.

Practice	No. of Different Problems/Total Analyzed	Completion Rate	Completion Rate with Optimal Solution	Not Finished
Being iterative and incremental	21/441	100%	73.47%	0%
Testing and debugging	18/378	100%	73.28%	0%
Reusing and remixing	14/294	100%	77.21%	0%
Abstracting and modularizing	13/273	100%	78.75%	0%

It is important to observe that the number of problems indicated in the previous tables refers to the number of problems proposed to the students in which a specific concept or practice was approached. Something quite satisfying about this number is the fact that every student involved in the study was able to finish the proposed exercises withing the specified time. Also noticeable from the results is the fact that although a large majority was able to solve the problems with a solution considered optimal by the involved teachers, some of the students were unable to achieve this result, despite having found solutions to the problems. Nevertheless, the highly positive results foresee an equally positive continuation of this study.

7. Evaluation: Stage 2

In the development research methodology, evaluation is the aggregating stage in the process and should be performed several times during the empirical testing stage or whenever a sub-product is "concluded".

After the first tests with users similar to end users and refinement of the products, we were able to conduct a larger study with end users in a real classroom environment.

The final evaluation study was conducted during two different school years (2022–2023 and 2023–2024) using three different study groups from a Portuguese school in real ICT or robotics classes:

- Group 1 had 116 students from the 7th grade with previous knowledge on robotics and mBlock in the school year of 2022–2023;
- Group 2 had 105 students from the 6th grade with no previous robotics knowledge (2022–2023) and later in the 7th grade (2023–2024);
- Group 3 had 108 students from the 6th grade with no previous robotics knowledge (2023–2024).

This study was conducted using the ICT and robotics classes for 50 minutes per week from the end of January until the end of May in both school years. Initially, the basic concepts and notations of the manual and robot were explained to all of the groups. At the beginning of each section of the manual, a theoretical explanation was given, and the expected results were detailed to make sure every student understood what they were asked to do. Group 1 was already familiar with the mBlock environment, and thus only the explanation of the newly developed framework was needed. Groups 2 and 3 felt the need to have extra introductory classes on how to work with mBlock and the newly developed framework for *Stemie*, as they had never programmed robots before. However, as they were already familiar with Scratch, these classes were quite simple for them because of mBlock's similarities to it. It is important to note that every student in this study had their own robot to take care of and take home after every class. After this, computational thinking was developed through hands-on problem-solving exercises [20] involving sequences, loops, parallelism, events, conditionals, operators, and data. The evaluations sent by the classes' teachers to each one of the students involved in the study were gathered. These data were sorted and classified by computational concepts and practices, and quantitative data analysis was performed to obtain comparable results. We also took into consideration the notes the teacher took about the students and their working methods during this period of time. Replicating what was performed in evaluation stage 1, to evaluate the students' problem-solving skills, the responsible teacher used rubrics for "problem identification", "planning", and "execution". The specific objectives for each of the activities were also used, and all of the teachers' records were documented using a three-point Likert scale: "not yet", "more or less", and "yes, completely". Video was the privileged way of keeping track of the activities. The classification of optimal and non-optimal solutions was performed using the same set of rules as in evaluation stage 1.

7.1. Results

The results of 20 different problems were evaluated for each of the 329 students involved in the study. In this phase of the study, we did not have data for one of the problems (Problem 19), and thus it was taken out of the analysis. As previously mentioned, not all of the 329 students tried to solve all 20 problems. Even with this fact in mind, its important to remember that for this analysis, a total of 3990 problems were taken in consideration. The overall results are summarized in Table 4.

Table 4. Overall study results.

	No. of Different Problems/Total Analyzed	Completion Rate	Completion Rate with Optimal Solution	Not Finished
Global results	20/3990	91.13%	56.99%	8.87%

In this global analysis, we obtained correctly solved problems at a rate of 91.13%, and 56.99% of those were solved using an optimal solution for that specific problem, which is a positive result. However, we also found that some of the problems (8.87%) were not solved by the students. According to the teachers' notes, this was mainly due to students not having the work material in all of the classes or, in some particular cases, due to hardware malfunctions.

Similar to what we did in the test with users similar to end users, it was important to determine if there were statistically significant differences in the scores across the three study groups. As the ANOVA assumed that the data were balanced, the missing values which we had due to not all groups having completed the same number of exercises would create an unbalanced design and impact the validity of the results. To overcome this problem, before performing the analysis, we decided to use multiple imputation, one of the most reliable techniques for handling missing data due to partial or incomplete responses from a portion of the sample [36]. Using the R programming language [37] and RGui editor, we applied multiple imputation with predictive mean matching (pmm) to our data. This procedure, executed with the *mice* package, allowed us to generate five different versions of plausible values for the missing entries. By integrating the imputed data into the original dataset, we obtained a new dataset but with most of the missing values replaced by statistically appropriate estimates. After the multiple imputation process, we were left with 17 exercises with complete data for the three different groups, which allowed us to perform further analysis.

Using Jamovi, we executed a one-way ANOVA with Fisher's method to determine if there were significant differences in performance between the different groups of students, categorized by different backgrounds in robotics, for each of the exercises. The one-way ANOVA results (see Table 5) showed that in most of the exercises, p values > 0.05 were registered, meaning that there were no significant differences between the groups. On the other hand, for exercises 8, 10, 14, and 17, the results table revealed p values < 0.05, which suggest significant group differences. The most significant differences occurred in exercises 8 ($F = 25.34$, $p < 0.001$), exercise 14 ($F = 13.11$, $p < 0.001$), and exercise 17 ($F = 26.02$, $p < 0.001$).

Table 5. One-Way ANOVA (Fisher's).

Exercise	F	df1	df2	*p*
Exercise 1	0.1494	2	326	0.861
Exercise 2	1.2234	2	326	0.296
Exercise 3	0.8933	2	326	0.410
Exercise 4	1.8608	2	326	0.157
Exercise 5	0.6769	2	326	0.509
Exercise 7	0.0249	2	326	0.975
Exercise 8	25.3355	2	326	<0.001
Exercise 9	0.1769	2	326	0.838
Exercise 10	3.9923	2	326	0.019
Exercise 12	1.0361	2	326	0.356
Exercise 13	1.8167	2	326	0.164
Exercise 14	13.1056	2	326	<0.001
Exercise 16	0.0249	2	326	0.975
Exercise 17	26.0207	2	326	<0.001
Exercise 18	0.7846	2	326	0.457
Exercise 20	0.6064	2	326	0.546
Exercise 21	1.0318	2	326	0.358

For the exercises with the most significant differences, we conducted a follow-up Tukey's post hoc test to identify which groups differred from one another. For exercise 8, the test showed significant differences between each pair of groups, with $p \leq 0.005$. For exercise 10, significant differences ($p = 0.017$) were only found between Group 1 and Group 2. Exercise 14 revealed significant differences between Group 1 and Group 2 as well as Group 2 and Group 3. However, it did not reveal a significant difference between Group 1 and Group 3. For exercise 17, a $p < 0.001$ in all group comparisons, showing that all groups differed significantly from each other. Generally speaking, Group 1 performed best in the most significant exercises (8, 14, and 17), and Group 2 was the worst in terms of the results for exercises 14 and 17.

The results we obtained were somehow expected. Group 1 was the one with older students and with previous knowledge on both robotics and the mBlock programming environment. It was not a surprise that they generally outperformed the other two groups. On the other hand, the worse results for Group 2 in exercises 14 and 17 were somehow surprising. Although both Group 2 and Group 3 started the study with no previous robotics experience, when those exercises were performed, Group 2's students were 1 year older and had one more school year of experience in robotics compared with Group 3. We did not find any specific reason for those results. It is possible that a slight lack of motivation may have affected the final results. Group 3 performed quite well, especially considering that the students were younger than the ones from the other two groups when they entered the study.

When we analyzed the results according to the three dimensions of Brennan and Resnick's framework [3] with computational concepts, practices, and perspectives, we also obtained quite interesting results.

7.1.1. Computational Concepts

Although all computational concepts were explored in this study, due to the different starting and ending points for each group, not every group was able to experience and test every one of them.

Table 6 indicates the number of tasks proposed to students in which a specific concept was approached. Furthermore, the same problem may have addressed more than one notion. There was also a disparity between the quantity of questions and concepts because complex topics such as events, parallelism, and data were only covered in a few of the book's final tasks. The concept of sequences was present in all problems and was evaluated with a completion rate of 91.13%.

Table 6. Study results grouped by computational concepts.

Concept	No. of Different Problems/Total Analyzed	Completion Rate	Completion Rate with Optimal Solution	Not Finished
Sequences	21/3990	91.13%	56.99%	8.87%
Loops	10/1739	89.36%	48.84%	10.64%
Events	3/232	96.55%	62.05%	3.45%
Parallelism	3/445	82.92%	47.15%	17.08%
Conditionals	3/232	96.55%	62.05%	3.45%
Operators	3/232	96.55%	62.05%	3.45%
Data	5/442	92.53%	59.41%	7.47%

Group 1, which had previous experience in robotics, was the one which tried to solve the larger number of problems involving events, conditionals, and operators. This may explain why, globally speaking, these were the skills students had less difficulty acquiring, with a completion rate of 96.55% and, among those, 62.05% finding the optimal solution for the problems, while only 3.45% of the problems were not solved.

As previously mentioned in the description of this study, each of the groups, due to the school year and previous experience, solved a different set of exercises. However, most of the concepts were included, making it of some importance to compare the results between groups.

It is possible to observe from Figure 10 that Group 1 included all of the concepts, although they only solved some of the more complex problems. Group 2 and Group 3 only worked with some of the concepts but, on the other hand, they solved more exercises. In every common concept, Group 1 had more expressive results. Despite this difference, its possible to observe that every concept which each group worked on was successfully developed, with the results being between 80% and 100%.

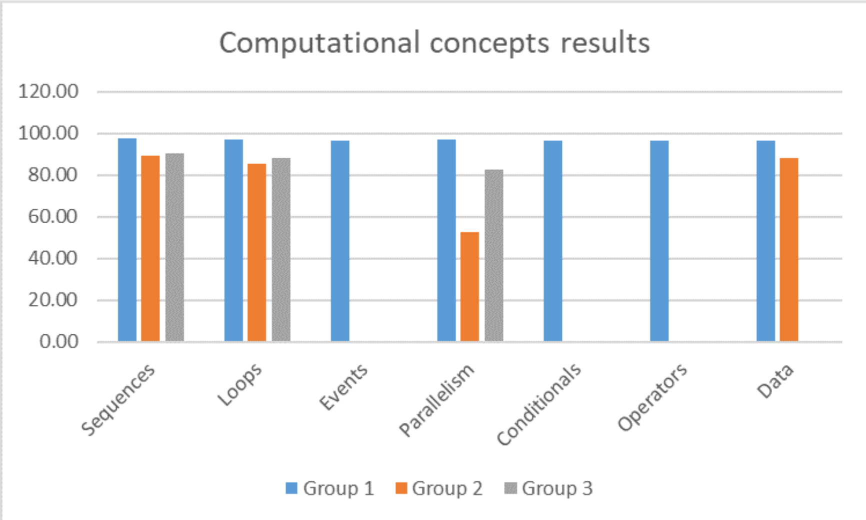


Figure 10. Group comparison of solved tasks. Percentages by computational concept.

7.1.2. Computational Practices

Despite the different numbers and types of problems each group solved, every group was able to experiment with all computational concepts explored in this study, as can be seen in the group comparison chart in Figure 11. Similarly, Group 1 was the one which obtained the better results in app computational practices. Their previous experience and age may have been the differentiation factors. However, when comparing Groups 2 and 3, we can observe that Group 3 had better performance than Group 2, although they were younger and had less experience. Through the analysis of this chart, we can perceive that all of the groups successfully developed every computational practice.

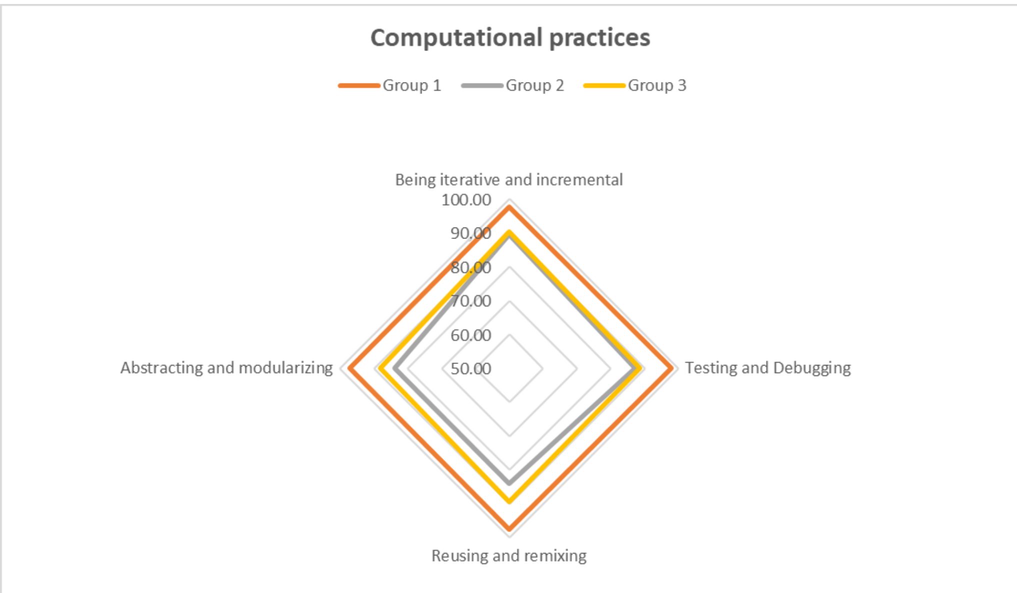


Figure 11. Comparison chart of tasks by computational practice.

As with computational concepts, it is important to observe that the number of problems indicated in Table 7 refers to the number of problems proposed to students in which a specific practice was approached, and the same problem often addressed more than one practice.

Table 7. Study results grouped by computational practices.

Practice	No. of Different Problems/Total Analyzed	Completion Rate	Completion Rate with Optimal Solution	Not Finished
Being iterative and incremental	21/3990	91.13%	56.99%	8.87%
Testing and debugging	18/3459	89.77%	56.04%	10.23%
Reusing and remixing	14/2599	88.80%	56.93%	11.20%
Abstracting and modularizing	13/2054	87.93%	52.21%	12.07%

Figure 12 gives a better understanding regarding the completion rate with optimal and non-optimal solutions to the problems, grouped by computational practices.

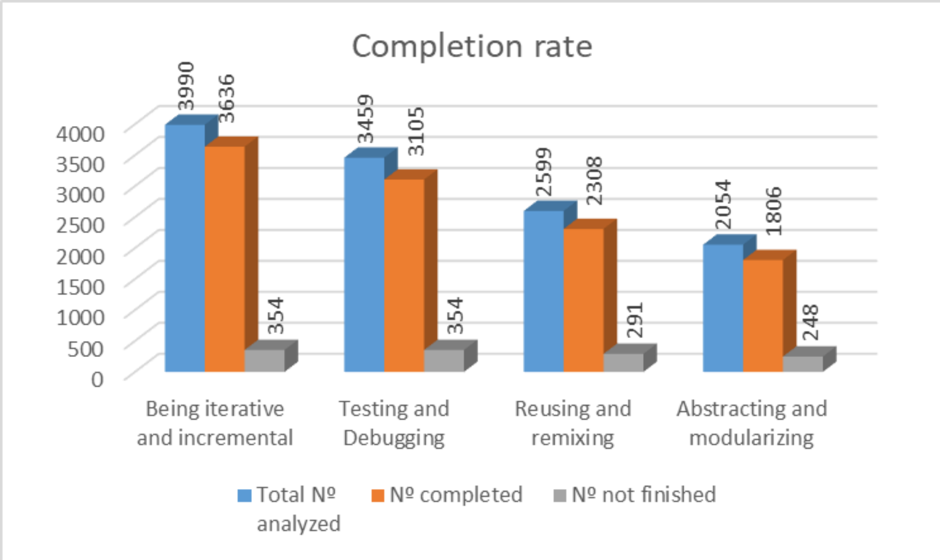


Figure 12. Completion rate by computational practice.

Through the analysis of the results grouped by computational practices, it is possible to find that being incremental and iterative was the most addressed practice throughout the proposed problems. It was also in this practice that the students showed less difficulty, with a completion rate of 91.13%. Reusing and remixing was also a practice which students were comfortable with, successfully solving 89.77% of the problems which involved this practice. Although achieving extremely positive results, the problems which involved abstracting and modularizing were those which more students were unable to solve, with 12.07% of them not finishing in time. From the teachers' notes, it was not possible to perceive if this was due to abstraction difficulties or if, given the slow pace of some students, there was simply no time to solve them.

7.1.3. Computational Perspectives

The three computational perspectives—express, collaborate, and question—were cross-sectional in all of the developed exercises, although they were not objectively measured. One's own **expression** was implicit, as the students solved the problem-solving tasks while following the guidelines but with the freedom to create something new and personalize the already existent elements through the inclusion of personal elements and preferences in the task scenarios. **Collaboration** was also a constant. Although the tasks were performed mostly individually as every student had their own robot, as soon as one ended, they asked to help the most delayed colleagues by performing peer work. Also, the curiosity about the processes, the similarities with some real-life situations, and the different problem-solving methods led them to **question** the technology. Some students even suggested new developments in the existing challenges.

8. Discussion and Conclusions

Computational thinking is the ability to formulate a problem and find a solution, whether executed by a computer or not [38]. Directly associated with this concept, we find the reference framework by Brennan and Resnick [3], which identifies concepts, practices, and computational perspectives. In all of the work we developed, students achieved highly positive results, training their problem-solving skills and building and retaining knowledge better [20]. With proper implementation, educational robotics in schools has been shown to considerably improve students' computational thinking and problem-solving skills, preparing them for the demands of the 21st century [33].

When comparing evaluation stage 1 with evaluation stage 2, there was a noticeable increase in the number of students which did not finish the problems within the specified time. We attribute this result to the fact that in evaluation stage 1, we worked with extremely small groups of students, and each of the groups had a specific teacher assigned to them to answer questions and direct them on the path to finding solutions to problems. Also, quite notable is the fact that at this stage of the study, the level of motivation generated by the competition made the students more committed to completing the tasks. Extrinsic motivation, due to the fact that they knew there would be a trophy at the end, perhaps also played an important role in these results. Winning an award at a competition not only gives students a sense of accomplishment but also gives visibility to schools [24] and, in the case of our study, countries.

Despite the differences between the two evaluation stages, the 91.13% rate of correct resolution of problems and 56.99% rate of optimal solutions found in the real test conducted in evaluation stage 2 are highly positive results.

Therefore, as concepts, practices and computational perspectives were present in all tasks, and we can state that **computational Thinking was successfully promoted, given the positive results obtained**.

The main conclusion from this study is that the model we developed (robot, programming framework, and STEM-related exercises) is a valid option for developing computational thinking and an interesting way for students to solve real-life problems. This type of pedagogical experience will provide children with essential skills for life in the 21st century.

Author Contributions: Conceptualization, A.V. and R.B.; Formal analysis, J.A.L. and S.P.S.; Investigation, R.B.; Methodology, R.B., J.A.L. and A.V.; Resources, J.A.L. and S.P.S.; Writing—original draft, R.B.; Writing—review & editing, J.A.L., S.P.S. and A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT) I.P. through national funds within the scope of the UIDB/00127/2020 project (IEETA/UA, http://www.ieeta.pt/ (accessed on 2 March 2024); https://doi.org/10.54499/UIDB/00127/2020; https://doi.org/10.54499/UIDP/00127/2020). This work was partially funded by the Research Centre on Education (CIEd) of the Institute of Education at the University of Minho, projects UIDB/01661/2020 and UIDP/01661/2020, through national funds from the FCT/MCTES-PT. This work was partially funded by the Erasmus+ Programme of the European Union (2020-1-PT01-KA201-078680).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We want to thank Colégio Paulo VI (Gondomar, Portugal) and the students of the fifth, sixth, and seventh school years for their collaboration and the authorization to perform this research on their premises.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wing, J. Computational Thinking. *Commun. ACM* **2006**, *49*, 33–35. [CrossRef]
2. Resnick, M. Reviving Papert's dream. *Educ. Technol.* **2012**, *52*, 42–46.
3. Brennan, K.; Resnick, M. New frameworks for studying and assessing the development of Computational Thinking. In Proceedings of the Annual American Educational Research Association Meeting, Vancouver, BC, Canada, 13–17 April 2012; pp. 1–25. Available online: http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf (accessed on 15 October 2024).
4. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books: New York, NY, USA, 1980.
5. Pastra, K. La Robotica nell' Educazione: Entusiasmante o Essenziale? In *Robot Sociali e Educazione*; Raffaello Cortina Editore: Milan, Italy, 2023; Chapter 9, pp. 139–154, ISBN 9788832855579. [CrossRef]
6. Screpanti, L.; Miotti, B.; Monteriù, A. Robotics in Education: A Smart and Innovative Approach to the Challenges of the 21st Century. In *Makers at School, Educational Robotics and Innovative Learning Environments*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 17–26.
7. Bredenfeld, A.; Hofmann, A.; Steinbauer, G. Robotics in education initiatives in Europe-status, shortcomings and open questions. In Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2010) Workshops, Darmstadt, Germany, 15–16 November 2010; pp. 568–574.
8. Barradas, R.; Lencastre, J.; Soares, S.; Valente, A. Designing Stemie, the Evolution of the Kid Grígora Educational Robot. In Proceedings of the 16th International Conference on Computer Supported Education, ESEO Angers—Grande École D'ingénieurs Généralistes, Angers, France, 2–4 May 2024; Volume 1, pp. 159–169. [CrossRef]
9. Papert, S.; Harel, I. Situating Constructionism. Available online: http://www.papert.org/articles/SituatingConstructionism.html (accessed on 15 April 2024).
10. Van den Akker, J. Principles and methods of development research. In *Design Approaches and Tools in Education and Training*; van den Akker, J., Branch, R.M., Gustafson, K., Nieveen, N., Plomp, T., Eds.; Kluwer Academic Publishers: Alphen aan den Rijn, The Netherlands, 1999; pp. 1–14.
11. Ouyang, F.; Xu, W. The effects of educational robotics in STEM education: A multilevel meta-analysis. *Int. J. STEM Educ.* **2024**, *11*, 7. . [CrossRef]
12. Hong, L. The impact of educational robots on students' computational thinking: A meta-analysis of K-12. *Educ. Inf. Technol.* **2024**, *29*, 13813–13838. [CrossRef]
13. Piaget, J. *The Theory of Stages in Cognitive Development*; McGraw-Hill: New York, NY, USA, 1971.
14. Coll, C.; Salé, I. Los professores y la concepcion constructivista. *Constr. Aula* **1997**, 7–24. Available online: http://www.terras.edu.ar/biblioteca/3/3Los-profesores-y-la-concepcion.pdf (accessed on 15 April 2024).
15. Schunk, D.H. *Learning Theories—An Educational Perspective*, 6th ed.; Pearson: Boston, MA, USA, 2012.
16. Papert, S. *The Children's Machine*; Basic Books: New York, NY, USA, 1992.
17. Sawyer, R. *The Cambridge Handbook of the Learning Sciences*; Cambridge University Press: Cambridge, UK, 2005.
18. Wing, J. Computational Thinking's Influence on Research and Education for All. *Ital. J. Educ. Technol.* **2017**, *25*, 7–14.. [CrossRef]
19. Brennan, K.; Chung, M.; Hawson, J. Scratch Curriculum Guide Draft. *Nature* **2011**, *341*, 73.
20. Jonassen, D. *Learning to Solve Problems. A Handbook for Designing Problem-Solving Learning Environments*; Routledge: New York, NY, USA, 2011.
21. Echeverría, M.; Pozo, J. Aprender a resolver problemas e resolver problemas para aprender. In *A Solução de Problemas: Aprender a Resolver, Resolver para Aprender*; Pozo, J., Ed.; Artmed: Porto Alegre, Brazil, 1998.
22. Salé, I. Disponibilidad para el aprendizaje y sentido del aprendizaje. In *El Constructivismo en el Aula*; Coll, C., Martín, E., Mauri, T., Miras, M., Onrubia, J., Salé, I., Eds.; Graó: Barcelona, Spain, 1997; pp. 25–46. Available online: https://avdiaz.wordpress.com/wp-content/uploads/2012/07/disponibilidad_aprendizaje.pdf (accessed on 2 April 2024).
23. Shindler, J. Chapter 18: Competition in the Classroom. In *Transformative Classroom Management*; John Wiley & Sons: Hoboken, NJ, USA, 2009. Available online: http://web.calstatela.edu/faculty/jshindl/cm/Chapter18competition-final.htm (accessed on 1 August 2024).
24. Bazylev, D.; Margun, A.; Zimenko, K.; Kremlev, A.; Rukujzha, E. Participation in Robotics Competition as Motivation for Learning. *Procedia Soc. Behav. Sci.* **2014**, *152*, 835–840. [CrossRef]
25. Selcuk, N.A.; Kucuk, S.; Sisman, B. Does educational robotics improve secondary school students' course motivation, achievement, and attitude? *Educ. Inf. Technol.* **2024**, *29*, 1063–1080. [CrossRef]
26. Altin, H.; Pedaste, M. Learning approaches to applying robotics in science education. *J. Balt. Sci. Educ.* **2013**, *12*, 365–377. [CrossRef]
27. Barradas, R.; Lencastre, J.A.; Radžvilienė, A.; Jačunskienė, K.; Šiurnienė, J. *Stemie & Stemia's Fantastic Journey Through Space*; Student's Ebook [EBOOK]; Research Centre on Education, Institute of Education, University of Minho: Braga, Portugal, 2023. Available online: https://Robotsforstem.Eu/Io5 (accessed on 1 August 2024).
28. Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. From game design elements to gamefulness: Defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, Tampere, Finland, 28–30 September 2011; ACM: New York, NY, USA, 2011; pp. 9–15. [CrossRef]
29. Fuchs, M.; Fizek, S.; Ruffino, P.; Schrape, N. *Rethinking Gamification*; Meson Press, Hybrid Publishing Lab, Leuphana University of Lüneburg: Lüneburg, Germany, 2014.

30. Ryokai, K.; Vaucelle, C.; Cassell, J. Virtual Peers as Partners in Storytelling and Literacy Learning. *J. Comput. Assist. Learn.* **2003**, *19*, 195–208. [CrossRef]
31. Schell, J. *The Art of Game Design: A Book of Lenses*; CRC Press: Boca Raton, FL, USA, 2008.
32. Palomino, P.T.; Toda, A.M.; Oliveira, W.; Cristea, A.I.; Isotani, S. Narrative for gamification in education: Why should you care? In Proceedings of the 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 15–18 July 2019; Volume 2161, pp. 97–99.
33. Jordan, S.D. *Educational Robotics and Computational Thinking in Elementary School Students*; Paper 725; Digital Commons @ ACU, Electronic Theses and Dissertations: Auckland, New Zealand, 2023. Available online: https://digitalcommons.acu.edu/etd/725 (accessed on 5 August 2024).
34. Barradas, R.; Lencastre, J.A.; Rispoli, L.; Faruolo, G.; Santangelo, I.; Pirrone, L.; Pomponio, R. *Stemie & Stemia's Fantastic Journey Through Space*; Teacher's ebook [EBOOK]; Research Centre on Education, Institute of Education, University of Minho: Braga, Portugal, 2023. Available online: https://Robotsforstem.Eu/Io4 (accessed on 1 August 2024).
35. The Jamovi Project. *Jamovi*, Version 2.3; Jamovi: Sydney, Australia, 2022. Available online: https://www.jamovi.org (accessed on 10 October 2024).
36. Campion, M.E.M. Multiple Imputation: An Investigation of the Missing Data Techniques Effectiveness. *Kwantlen Psychol. Stud. J.* **2024**, 19–26. Available online: https://journals.kpu.ca/index.php/KPSJ/article/download/1947/1539/ (accessed on 10 October 2024).
37. R Core Team. *R: A Language and Environment for Statistical Computing*, Version 4.1; R Core Team: Vienna, Austria, 2021. Available online: https://cran.r-project.org (accessed on 10 October 2024).
38. Cuny, J.; Snyder, L.; Wing, J.M. Demystifying Computational Thinking for Non-Computer Scientists. 2010. Available online: http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf (accessed on 15 April 2024).