digital

Article

Analysis of the Learning Process of Computer Programming Logic in an 8-Year-Old Elementary School Student at Home through the Scratch Program

Victor García

Faculty of Computer Sciences, Multimedia and Telecommunication, eLearn Center, Universitat Oberta de Catalunya (UOC), Rambla del Poblenou, 156, 08018 Barcelona, Spain; vgarciahe@uoc.edu

Abstract: This paper presents a study guide and an analysis of its use in the computer programming learning process of an 8-year-old elementary school student through the Scratch program. The research's objective is to explore and understand how this individual student approaches learning programming skills and tackles challenges within the Scratch environment. An individual case study approach was adopted at home, combining qualitative and quantitative methods to gain a comprehensive insight into the student's learning process. The study was conducted without grant support, and the researcher actively participated as an educator and observer in the student's learning sessions. Performance was assessed, and a semi-structured interview was conducted to inquire about the student's experiences, motivations, and interests regarding programming in Scratch, as well as their feelings after the training. Additionally, the student's activities during programming sessions were meticulously recorded, and projects created in Scratch were analyzed to assess progress and understanding of concepts. The findings of this research have the potential to contribute to the field of programming education and provide valuable insights into how young elementary school-aged individuals can acquire computer and programming skills in an interactive environment such as Scratch. The results obtained demonstrate that using the proposed guide to introduce elementary school students to programming at home, with parents acting as educators, is feasible. Therefore, it helps facilitate access to this knowledge, which is currently limited for many individuals in an official educational setting.

Keywords: elementary student; computer programming learning; homeschooling; parents as educators

check for updates

Citation: García, V. Analysis of the Learning Process of Computer Programming Logic in an 8-Year-Old Elementary School Student at Home through the Scratch Program. *Digital* **2024**, *4*, 69–91. https://doi.org/10.3390/digital4010002

Academic Editor: Manuel Reis

Received: 29 November 2023
Revised: 14 December 2023
Accepted: 21 December 2023
Published: 25 December 2023

1. Introduction

Early education in new technologies such as robotics and programming is emerging as an option in many primary schools, and its popularity is on the rise [1]. However, currently, there is a significant disparity between areas, schools, and families in access to this type of education. In most cases, resorting to private extracurricular classes is necessary [2] if elementary school students are to acquire these skills. Therefore, the motivation of this research lies in the current need to comprehensively define and strategically address pervasive disparities in programming learning [3,4], especially in the unique context of home education. While traditional classroom activities have played a crucial role in mitigating educational gaps, substantial challenges persist, hindering widespread access in various regions, schools, and family environments [5]. Despite concerted efforts, the need to innovate in educational approaches beyond the limitations of conventional classrooms is evident, and this research aims to contribute to this need. The existing literature recognizes the importance of classroom interventions; however, it consistently highlights persistent disparities in programming education, emphasizing the need for multifaceted strategies [6–10]. This research positions itself as a bridge to overcome this educational gap by delving into the effectiveness of home learning. It strives to provide nuanced insights into the central role parents can play in developing their children's programming skills, an aspect that can be highly relevant in contemporary educational discourse [11,12]. In doing so, this study aims to move beyond the conventional paradigm of programming education, focusing on family and home learning environments as powerful contributors to fostering a more equitable distribution of programming skills among elementary school students. This study may gain further relevance due to demographic trends in certain areas, especially in Europe, where families increasingly consist of one or two children [13]. The significance of this demographic pattern lies in its implications for educational dynamics, as smaller family sizes can amplify the role of parental involvement in the learning process [14,15]. The subsequent focus on home teaching is both pragmatic and contextually significant.

To facilitate effective home teaching, a comprehensive guide has been meticulously developed. This guide serves as a pedagogical tool, introducing elementary school students to the intricacies of video game creation using the Scratch program [16], a block-based programming language designed for educational purposes. The deliberate choice of Scratch is derived from its suitability for beginners and its alignment with educational objectives, fostering a practical and creative approach to programming learning [17,18]. Despite the prevailing trend towards tablets and mobile devices, this study recognizes the need to examine students' transition to alternative platforms, thus emphasizing a focus on PC-based programming—a critical nuance often overlooked in the literature [8,18]. This deliberate approach underscores the importance of preparing students for diverse technological landscapes, acknowledging the multidimensional nature of their future digital interactions.

1.1. PC Usage

The fact that most students today are more familiar with the use of tablets and mobile devices rather than desktop PCs [19] posed the first challenge of this study. Although there is a version of Scratch for mobile devices, it was decided that the introduction to programming learning would be carried out through a PC to analyze the student's transition to other platforms. The challenges encountered around PC usage during the teaching process are multifaceted:

Mouse and Keyboard Handling: Students accustomed to using tablets and touch devices may face difficulties in adapting to the use of a computer mouse and keyboard. It might take them some time to develop the fine motor coordination necessary to handle the mouse precisely and use the keyboard efficiently.

Familiarization with the PC Environment: A student who has been homeschooled may not be as familiar with the computer environment, including how to navigate the operating system, open and close programs, and manage files and folders.

Keyboard Shortcuts: Keyboard shortcuts are an efficient way to interact with a computer and perform quick actions. Introducing the student to the use of keyboard shortcuts may require time and practice for them to incorporate them into their workflow.

Internet Search: Although students are increasingly exposed to technology, it is important to teach them safe and efficient internet search skills. Identifying relevant and reliable information, as well as avoiding inappropriate sites, can be an initial challenge for the student.

The observed lack of computer skills among contemporary schoolchildren, despite their proficiency with smartphones and tablets, draws attention to the evolving landscape of technological familiarity. In the early days of introducing computer skills, the focus primarily revolved around familiarizing individuals with desktop PCs [20,21]. However, the present scenario presents a noteworthy shift, with students now predominantly engaging with touch-based devices. This shift poses distinctive challenges, as evidenced by our study. Unlike the era when computer skills were initially introduced, where desktop PCs were the primary interface, today's students grapple with adapting from touch interfaces to traditional mouse and keyboard interactions [22,23]. This transition highlights the evolving nature of the challenge, requiring a nuanced understanding of the multifaceted hurdles faced by students in developing the fine motor skills necessary for precise mouse handling and efficient keyboard usage. Furthermore, the need for familiarization with the broader computer environment, including operating systems and keyboard shortcuts, reflects the dynamic nature of the skills now essential for effective engagement in a digital learning landscape [24]. By delving into these nuances, this study aims to contribute valuable insights into the contemporary challenges associated with PC usage in programming education, setting it apart from the historical introduction of computer skills.

1.2. Concepts of Mathematics and Logic

To introduce programming concepts, it is necessary to introduce new mathematical concepts to an elementary school student, which poses another challenge. Specifically, the necessary mathematical concepts are:

- Concept of Negative Number: Necessary for moving 2D objects along a coordinate axis, so the student must interpret coordinates (0, 0) as the center of the Scratch editing screen and moving left or down will be done with negative numbers.
- Object Concept: Understanding how objects interact in Scratch, representing specific entities in the program.
- Creation of Custom Objects: Learning to create and customize objects within the programming environment.
- Variable Concept: Understanding how variables store information and can be used in programs.
- Sequential Task Concept: Grasping the importance of organizing instructions in a logical sequence to achieve specific results.
- Event and Action Concept: Learning how events trigger actions in the program.
- Conditional Concept: Understanding how conditional decisions affect the program's flow.
- Loop Concept: Grasping the repetition of actions through loops in programming.
- Function and Function Call Concept: Introduction to designing and using functions to organize and reuse code.
- Event Concept: Understanding how events can be used to interact with the user and control the program's flow.

To address these challenges, an educational guide was developed, which can be seen in the section "Educational guide" in the Supporting Information. This guide outlines the steps taken through the creation of a Shmups-type video game inspired by the famous Star Wars fiction movie. This was chosen with the intention of motivating the student to create the project. Before that, this guide serves as a tool to help parents and educators introduce young students to basic mathematical concepts and acquire PC handling skills to tackle the challenge of learning block programming. After following the educational guide with the student, their performance was evaluated through a questionnaire, and their motivation and feelings after the learning process were assessed through a semi-structured interview.

2. State of the Art

It was considered relevant in this section to analyze the integration of classic works in the field of teaching computational thinking, especially considering the foundational contributions of LOGO programming researchers, among them the prominent Seymour Papert and his collaborators. Seymour Papert's groundbreaking publications, such as "Mindstorms: Children, computers, and powerful ideas" [25], established the intricate relationship between technology and education, highlighting the importance of individual student efforts to master mathematical concepts and create video games. This pivotal period marked a shift in focus towards children's engagement with computers, as documented in publications such as [26]. The "Logo Exchange", during the 1980s and 1990s, played a fundamental role as a central platform for exploring and discussing children's activities related to computers. This space was established as a meeting point for educators, researchers, and professionals interested in the LOGO educational approach developed by Seymour Papert and his colleagues. Its main objective was to exchange ideas, strategies, and experiences regarding the implementation of LOGO in educational settings. Participants shared resources, lessons learned, and pedagogical proposals focused on programming and computational thinking for children. The "Logo Exchange" provided a valuable network for those immersed in the integration of technology in education, offering an enriching insight into the practical application of LOGO and its benefits in student learning. Given the direct connection from LOGO to Scratch, a contemporary programming language used in the current study, referencing and incorporating ideas from these earlier works becomes essential to ground the research in a historical context [25,26].

However, in the last decade, we have witnessed a significant educational transformation with the introduction of robotics and programming in elementary school teaching. This shift reflects the growing awareness of the importance of preparing students from an early age for an increasingly digital world. Numerous studies have explored the multiple advantages of integrating these disciplines into the curriculum, highlighting their ability to develop cognitive skills, creativity, and problem solving [1,18,27–31].

In this context, Scratch has emerged as a pioneer in teaching programming at the elementary level. Its block-based approach and intuitive visual interface have proven to be effective tools for introducing coding concepts in an accessible and engaging way for children [18]. Educational projects worldwide have adopted this platform, highlighting its ability to foster logical thinking and creativity.

Despite these advances, we still face significant challenges in the widespread implementation of robotics and programming in primary schools. Disparities in the availability of educational resources in this field create gaps in students' exposure to technology [32]. The lack of equal access to these learning opportunities raises crucial questions about equity in STEM education from an early age.

To address this disparity, a specific guide has been developed for family members to teach programming with Scratch to elementary school students. This initiative not only aims to empower parents and guardians to provide additional programming education, but also addresses time and resource limitations by including student performance assessment tools [33–35]. This guide has the potential to close the gap by facilitating programming instruction at home, regardless of resource availability at school.

Homeschooling, in general, has experienced a significant increase in popularity in some countries such as USA. Research indicates that this approach can offer notable benefits by allowing parents to tailor education to their children's individual needs [36]. In the specific field of computer science, homeschooling computer science concepts has emerged as a valuable option. This personalized approach not only fills possible educational gaps but also allows for a deeper and more contextualized understanding of computer principles [37].

Homeschooling computer science is not just about filling educational gaps; it goes beyond by cultivating a solid understanding of the fundamentals and applications of computer science. This student-centered approach recognizes the diversity of learning styles and individual paces, thus addressing the limitations of the traditional education system.

Despite significant efforts to improve programming and robotics education in elementary school, challenges persist. Disparities in access and lack of time and resources on the part of parents are crucial concerns that must be addressed to ensure equitable and comprehensive education. Current solutions, such as guides for homeschooling and the promotion of homeschooling computer science, offer innovative responses to these challenges, but ongoing commitment is needed to achieve broader and more effective implementation.

The current state of robotics and programming education in elementary school reflects both notable advances and persistent challenges. The promise of preparing students for a digital future remains a crucial goal, and current strategies seek to address gaps in access and educational equity. As we move forward, it is imperative to continue researching and developing innovative approaches to ensure that every student has the opportunity to acquire key programming and robotics skills, regardless of their educational environment or available resources.

3. Context

The participant in this study is an 8-year-old primary school student residing in southern Spain, whose parents are divorced. The student attends regular classes in the second year of primary education at a public school. The student comes from a family where both parents have university education.

It is important to note that, up to the point of this study, the student has had no experience in the field of programming. This includes the absence of programming-related courses in the school curriculum and extracurricular activities. Additionally, the student has never used the Scratch program, which was the main focus of this learning project.

Knowledge transmission in the field of programming took place in the student's home environment over a period of 2 months, averaging 1 to 2 h on Mondays, Wednesdays, and Fridays. Training occurred in a study room with ample lighting and no disturbances, providing a conducive space for concentration and learning. The primary facilitator of this process was the student's father, who played the role of educator and guide during interactions with the Scratch program.

In academic terms, the student maintains an above-average performance in the regular studies performed, although the student has had no previous exposure to subjects related to programming. Despite this lack of formal experience, the student exhibited notable enthusiasm and motivation toward the idea of creating their own video game. This initial positive and curious attitude may be a key factor influencing the learning process.

It is essential to consider that, given the student's age, the approach shown to learning programming is expected to be influenced by the student's natural disposition to explore new ideas and concepts in a playful manner. Leveraging this natural inclination toward play and exploration allows for fostering a dynamic and stimulating learning environment. As the home serves as the educational center, it is considered a comfortable and familiar space where the student can not only learn effectively, but also feel more relaxed and open to experimentation.

This pedagogical approach, based on experimentation and play, not only aims to facilitate the understanding of programming concepts, but also seeks to establish a solid foundation for the development of logical thinking and problem-solving skills in the student. The home environment offers flexibility to adapt learning sessions to the student's individual schedules and rhythms, allowing for more personalized attention.

Furthermore, this study not only aims to analyze the student's learning process, but also seeks to provide valuable guidance for other parents who may be interested in taking on the role of educators in teaching programming to their children. The challenges and successes documented in this research can serve as a reference for parents seeking effective strategies to introduce their children to the world of programming in an accessible and enjoyable manner. The shared experience may inspire other parents to embark on similar educational adventures, thus strengthening the connection between home learning and students' academic development.

With these contextual elements, the study aims to analyze in detail the programming learning process in an 8-year-old primary school student through the use of Scratch, highlighting both the challenges encountered and the achievements obtained throughout this unique educational experience.

4. Research Questions

In this study, the proposed educational guide was followed for the transmission of programming knowledge (see Supporting Information); thus, when following this guide, the questions for this study aim to determine the following:

How does the educational guide contribute to easing the adaptation of an elementary school student to desktop PC usage in the contemporary technological landscape?

In the context of current computational thinking education, how effective is the guide in facilitating an elementary school student to acquire foundational mathematical knowledge for initiating programming?

Can the educational guide effectively introduce and lead to proficient performance in computer programming learning for an elementary school student within the present educational paradigm?

What are the contemporary sentiments and experiences of an elementary school student upon following this educational guide, considering the evolving landscape of technology and education?

5. Theoretical Discussion

The guide designed for the introduction to programming through Scratch, aimed at elementary school students with the purpose of creating a Shmups-type game, is based on a pedagogical approach that amalgamates various educational methodologies of recognized value. This comprehensive approach seeks not only to cultivate programming skills but also to foster cognitive abilities, creativity, and autonomy in students.

Discovery Learning [38], inspired by Bruner's theory, is manifested through the assignment of open projects. These projects, designed to allow students to explore and discover programming concepts autonomously, encourage experimentation and independent problem solving.

The Experiential Learning or Practical Learning methodology [39], based on Kolb's theory of Experiential Learning, emphasizes the direct application of knowledge through practical sessions on the computer and the creation of projects. Practical experience in Scratch emerges as a fundamental pillar for the effective understanding and retention of programming concepts.

Although not explicitly mentioned, the guide promotes Collaborative Learning by encouraging students to present and explain their projects to other students. Collaboration in problem-solving and the joint creation of knowledge, grounded in Dillenbourg et al.'s research [40], becomes a key component of this educational proposal.

The Personalized Methodology, based on Hattie and Timperley's feedback theory [41], is reflected in individualized tutoring sessions. This approach involves adapting the educational strategy according to each student's learning style and identifying specific areas that require development.

Project-Based Learning, supported by Thomas's research [42], is manifested in the initiation and continuation phases of the ShuWars project. Through the gradual creation of a game, students apply and consolidate the acquired knowledge, promoting practical problem solving.

Reflective or Metacognitive Methodology, inspired by Flavell's theory [43], is incorporated through self-assessment. Students are encouraged to reflect on their progress and set goals for the future. This metacognitive approach promotes a deeper understanding and increased awareness of their learning processes.

Following the principles of Active Learning proposed by Bonwell and Eison [44], the guide encourages the direct participation of students in free exploration projects and challenges. This active approach ensures practical understanding and effective retention of programming concepts.

These methodologies, synergistically integrated, form a comprehensive educational approach that aspires to provide an enriching and effective experience in teaching programming to students through Scratch. The resources used to create the video game proposed can be found at [45].

6. Research Methodology

The methodology followed to carry out this research can be summarized in the following points:

- Guide Design

A structured guide was designed to teach programming to elementary school students through the creation of a Shmups-type game. The guide was divided into sections, each focusing on specific aspects, from the introduction to the PC to the creation and presentation of advanced projects. The guide's design follows a descriptive methodology approach [46] by detailing the activities and sections intended for programming teaching. The selection of a single student for the implementation of the programming introduction guide through Scratch is justified from an academic, research, and logistical perspective. This preliminary investigation can be considered a "pilot study" that explores the feasibility and effectiveness of the guide before more extensive research [47]. Furthermore, the choice of a single participant aligns with exploratory approaches, allowing for a thorough understanding before delving into broader research inquiries [48]. This study can be conceptualized as a "single-case study" providing detailed insights and specific context [49]. Additionally, this approach can be considered in line with personalized and adaptive strategies, especially considering tutoring and continuous adaptation [50]. Due to access to students and the logistics involved in conducting a more extensive study with other parent educators, it was decided to carry out this preliminary research. The limitations of this study can be seen in the limitations section of this work. In addition, the use of single-case studies in programming education can be justified by their ability to provide in-depth insights into the learning process of individual students and the effectiveness of specific instructional strategies. Plavnick and Ferreri (2013) emphasize the value of single-case experimental designs in educational research for conducting causal analyses in teaching and learning [51]. This approach allows educators to systematically evaluate the impact of different instructional methods on the programming proficiency of individual students. Additionally, the case study approach described by Clancy and Linn (University of California, Berkeley) provides a framework for presenting programming problems, expert problem-solving processes, and student engagement through the analysis of alternatives and reflection on problem-solving methods [52]. Using single-case studies, educators can tailor their instructional approaches to the specific needs of individual students, ultimately enhancing the effectiveness of programming education. Finally, the literature on single-case studies in educational research provides a methodological foundation for rigorously examining the impact of educational interventions on individual student learning outcomes [51]. This justifies the use of single-case studies as a valuable tool for informing evidence-based practices in programming education.

- Participant Selection

A single elementary school student was selected as a participant in the experiment, falling under intentional sampling [53]. Parental consent was obtained, and the objectives and nature of the activities were explained.

- Learning Sessions

Practical sessions were conducted according to the designed guide following an experimental methodology [54]. Each session focused on a section of the guide, addressing topics such as mouse handling, navigating the operating system, exploring Scratch, applying mathematical concepts, and project creation.

- Quantitative and Qualitative Assessment

A mixed methodology was followed [55] and was used as research instrument by using: a quantitative questionnaire, as mentioned earlier, which was designed to assess the participant's performance in following the provided guide based on the educational items and scales outlined in the Supporting Information. The questionnaire aimed to evaluate aspects such as mouse handling, Scratch comprehension, and the application of mathematical concepts. Additionally, a semi-structured interview was conducted to assess the student's feelings and perceptions toward learning and programming. The semi-structured interview served as a key instrument to gather qualitative data regarding the student's perception, experiences, and reflections throughout the learning process.

Its purpose was to explore the student's thoughts, feelings, and insights related to the learning journey with Scratch programming, providing an opportunity for the student to express their motivation, challenges faced, and overall perception of the educational approach. The interview was conducted in a semi-structured format, allowing flexibility for open-ended questions while maintaining a predefined set of topics related to the learning process. Questions covered aspects such as the student's enjoyment of the learning process, challenges faced, and future aspirations in programming. In relevance to the study, the semi-structured interview offered qualitative data that enriched the understanding of the student's subjective experience and provided valuable context to complement the quantitative performance evaluations. The interview was recorded, transcribed, and analyzed using a thematic coding approach with the following categories: (1) Love for Learning, (2) Ease with Mouse and Keyboard, (3) Fun in Creativity, (4) Sense of Achievement, (5) Sharing and Teaching, (6) Collaborative Problem-Solving, (7) Independent Exploration, (8) Positive Learning Outcome, and (9) Excitement for Future Learning.

- Tutoring Sessions

Regular tutoring sessions were scheduled following a longitudinal methodology [56] to review progress, address questions, and provide guidance. Adaptation of the educational approach according to the student's learning style was carried out during these sessions.

- ShuWars Project Creation and Continuation

The ShuWars project was initiated following a project-oriented methodology [57], where the student gradually applied the concepts learned to create a game. Creativity was encouraged by assigning new functionalities, and the student was encouraged to present and explain their projects.

- Ongoing Evaluation and Feedback

Periodic reviews of the student's work were conducted following a formative methodology [58], using quantitative and qualitative assessment. Detailed feedback was provided to highlight positive aspects and areas for improvement.

- Documentation and Portfolio Creation

The student was encouraged to document their learning process through anonymous blog communication following an ethnographic methodology [59]. An anonymous digital portfolio highlighting projects, skills, and reflections was created.

- Conclusions and Analysis

Results from quantitative and qualitative assessments were collected [60]. Conclusions were derived by analyzing the student's progress, emotional reactions to learning, and the quality of projects completed.

- Iteration and Improvement

Results were used to adjust the guide and educational strategies following an action research methodology [61]. Areas needing more focus were identified, and changes were implemented to enhance the effectiveness of the teaching-learning process.

7. Results

The implementation of activities designed to introduce a primary school student to the use of a desktop PC yielded some interesting results. The methodology structured in three phases—Mouse and Keyboard Handling, Operating System Navigation, and Keyboard Shortcuts—complemented with Visual and Multimedia Resources, has proven to be effective. In mouse and keyboard handling, practical exercises were crucial in improving the required fine motor skills and coordination. Although there was a brief learning curve at the beginning, these exercises laid the foundation for acquiring essential motor skills. The introduction of interactive games that required precise mouse movements and efficient keyboard usage proved to be a motivating approach. Despite a small initial learning curve, the playful nature of the games contributed to a quick adaptation.

In the operating system navigation phase, the introduction to the basics of using the Windows operating system provided a fundamental understanding. Practical exercises to open and close programs and manage files and folders allowed for practical application, facilitating the transition to a broader operating environment. Regarding keyboard shortcuts, the early presentation of shortcuts and their application in the programming environment offered an efficient perspective. Regular practice to incorporate these shortcuts into the student's workflow contributed to a quick and effective adaptation. In Figure 1, the student can be seen operating the PC mouse within the Scratch program.

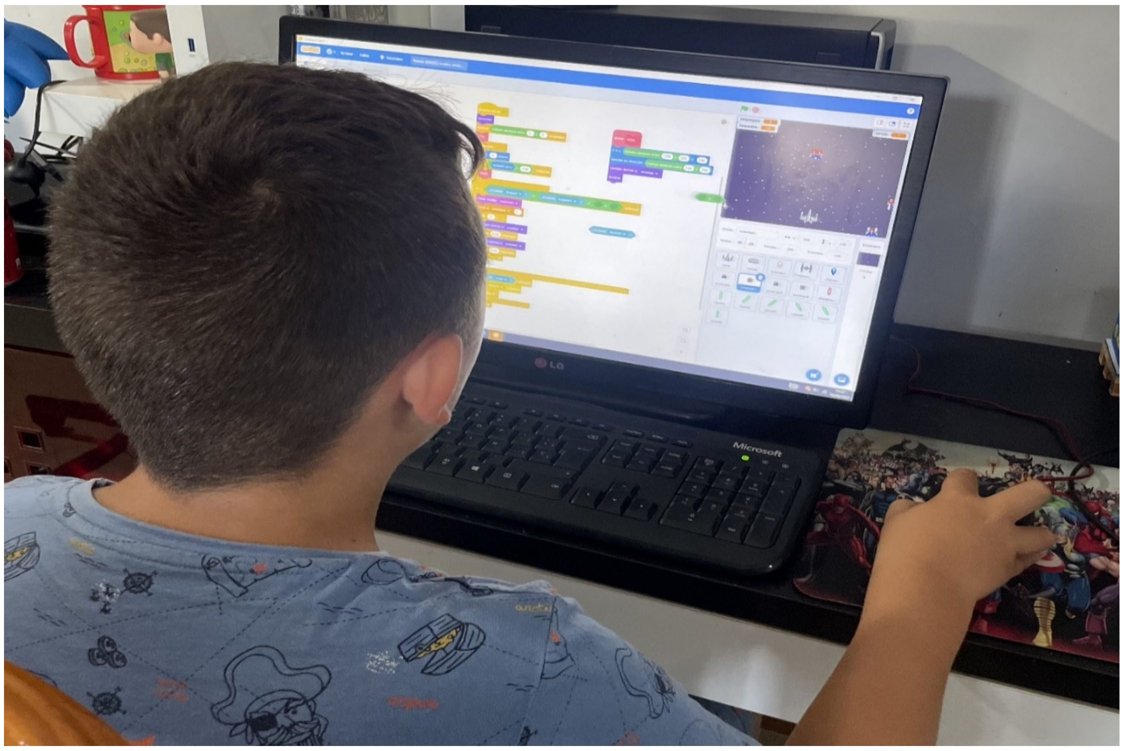


Figure 1. Student under analysis handling the PC smoothly.

The systematic integration of regular practices solidified the application of keyboard shortcuts in the student's everyday workflow.

Overall, the student experienced a brief initial adaptation phase due to their familiarity with tablets. However, this period was overcome within the first 20 min of the session. The rapid adaptation to the use of primary and secondary buttons, opening and closing programs, executing icons, and using keyboard shortcuts indicates that the learning curve was not extensive.

During the introduction to the Scratch phase, various activities were carried out to explore Scratch's capabilities and develop the student's programming skills. In the first activity, Scratch exploration was addressed. This included a general presentation about the Scratch interface, available tools, and basic functions, as well as the creation of simple projects to understand how programming blocks work.

Subsequently, specific exercises called mouse coordination exercises were conducted to improve the student's motor coordination in dragging and dropping blocks.

In the next stage, called first functionalities with blocks, an introduction to the concept of objects and the Scratch drawing palette was given to create custom objects. Additionally, blocks were applied to give basic movements to the created objects.

Next, the creation of basic projects was progressed, involving the gradual development of more complex projects. This phase aimed to apply and consolidate the knowledge acquired during the previous stages. Interactive resources were used, leveraging pre-existing projects in Scratch to allow the student to explore and learn from practical examples. During these practical sessions, immediate feedback and guidance were provided to maximize understanding and retention of concepts.

The student showed enthusiasm with the introduction of objects, represented as images that they could create and modify in Scratch. The idea of being able to assign actions to these objects through blocks was well received. Additionally, they managed to understand the basic Scratch blocks to move objects and create simple scenes. These results indicate a positive response from the student to the practical application of acquired knowledge, suggesting satisfactory progress in their understanding and programming skills with Scratch. Figure 2 shows some of the many simple projects completed by the student.

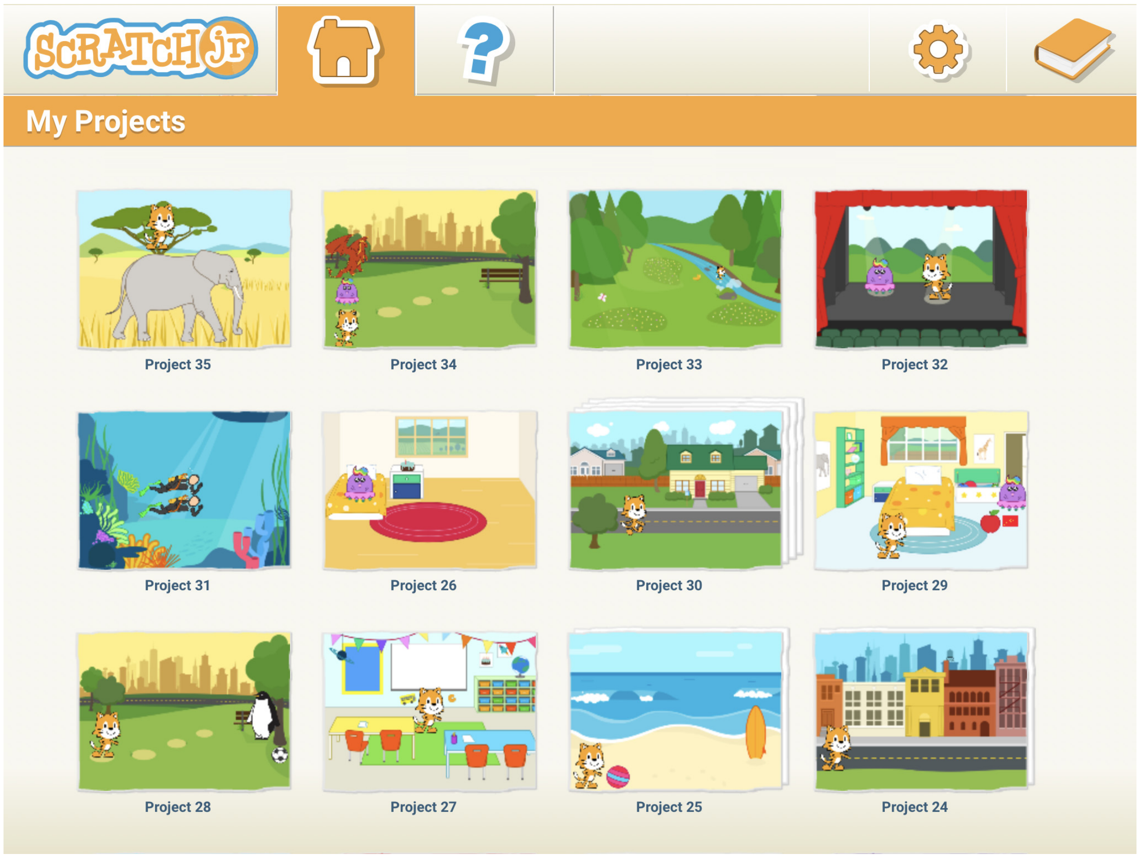


Figure 2. Some simple scenarios and short films created by the student.

Following the Discovery Learning phase of the presented guide, whose main objective was to promote autonomy and the student's ability to discover and understand new programming concepts through active exploration, the activities were designed to encourage creativity, experimentation, and independent problem solving.

In the free exploration projects, open-ended assignments were given, allowing the student to creatively apply the learned programming blocks. The aim was to encourage experimentation and independent problem solving.

Challenges and puzzles constituted another important part, where programming-related challenges and riddles in Scratch were presented. The student was encouraged to solve problems by applying previously learned concepts.

Guided research involved stimulating the student to seek other children's projects, research them, and share opinions online, thus fostering active research and interaction within the online community.

The assessment process was carried out through constructive feedback, including regular evaluation of the projects and challenges completed by the student. Constructive feedback was provided to drive continuous improvement and offer specific guidance on areas of development.

The student responded positively to this section of the guide, creating somewhat more complex projects, and showing enthusiasm and curiosity in continuing to advance and modify the created projects. Additionally, they demonstrated interest in finding and understanding projects already undertaken on the internet, indicating a high level of commitment and motivation to explore and learn autonomously.

In the phase of introducing the student to fundamental concepts of mathematics, including variables, spatial arrangement, and negative numbers, the designed activities aimed to provide a practical and applied understanding of these concepts in the Scratch programming environment. In the first activity on coordinates in a two-dimensional plane, the coordinate system (x, y) and its application in Scratch objects were explained. Practical exercises were conducted involving the positioning of objects on the screen (see Figure 3), providing a tangible understanding of the concept of two-dimensional coordinates. The section on negative numbers addressed the concept of negative numbers in relation to spatial arrangement on coordinate axes and their relevance in programming. Practical exercises were performed to understand how negative numbers influenced the movement of objects. Although the student showed some difficulty in comprehension, they could relate these concepts to the coordinates of objects for their placement on the stage, demonstrating an ability to associate and apply practical knowledge.

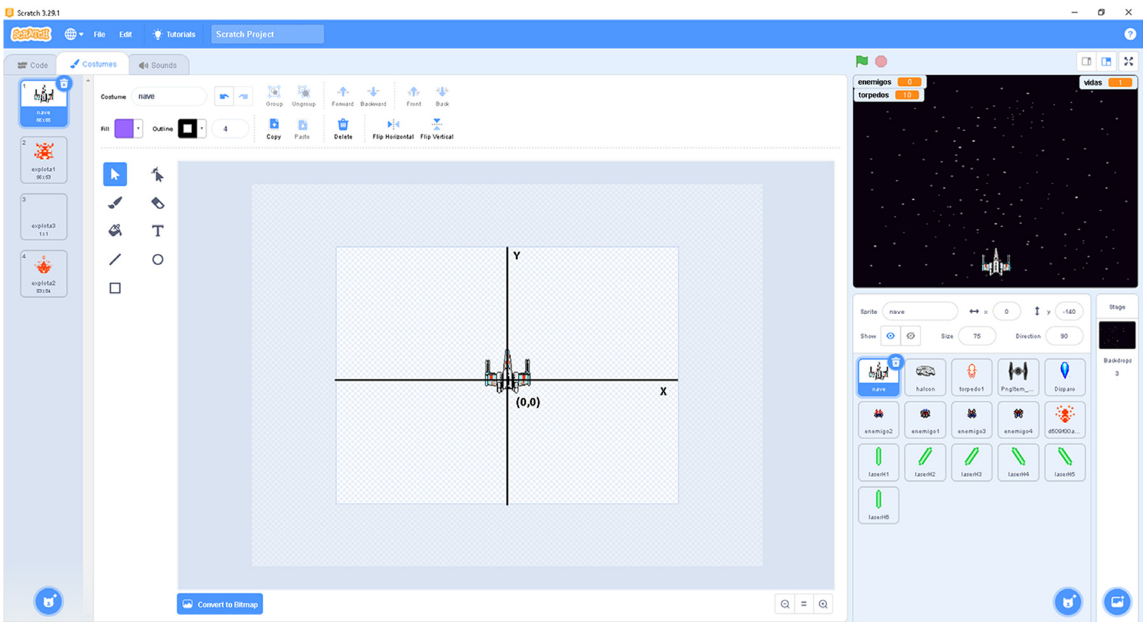


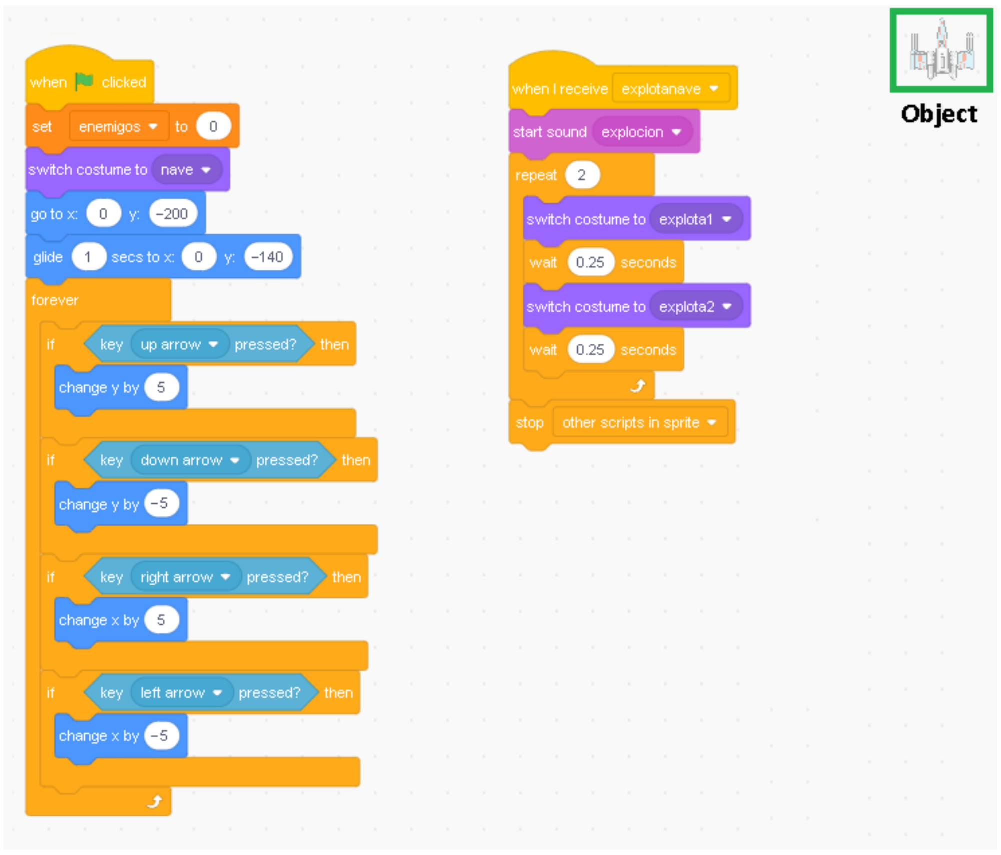
Figure 3. Template with coordinate axes to visually convey the meaning of negative numbers related to the positioning of the object.

In the section dedicated to variables and basic operations, the concept of variables and their role in programming was introduced. Practical examples of how to use and modify variables in Scratch were presented, including basic operations such as addition, subtraction, multiplication, and division. Concepts such as the counter and the reinforcement of the object concept were also addressed. The student assimilated these concepts as a numeric variable whose value is not known in advance, demonstrating understanding of basic operations and the counter concept.

Visual resources, such as graphics and visual examples, were used to facilitate the understanding of these mathematical concepts. Practical situations in Scratch were created that required the use of variables and basic operations, providing a practical and visual approach to learning.

Although the student experienced some difficulty with the concepts of negative numbers, they managed to assimilate them by relating them to coordinates on the stage. Furthermore, the student demonstrated adequate understanding of variables, basic operations, and other essential mathematical concepts, suggesting positive progress in assimilating these mathematical fundamentals in the context of programming in Scratch.

Once the project of creating the Shmups-style game began, with the aim of gradually applying the concepts learned in the previous sections and introducing new programming logics, in Figure 4, the initial object and its block programming created by the student in a guided manner can be seen. It is important to note that this figure provides information highlighting several important programming concepts found in these blocks. Elements such as positioning, response to events (such as movements triggered by pressing a specific button), assigning values to variables, infinite loops, and "if" conditionals are observed. Finally, function calls are included. The latter was interpreted by the student similarly to the concept of event-action. In this case, when an event occurred, such as the collision of a ship, the "explotanave" or "explode ship" functionality was invoked.



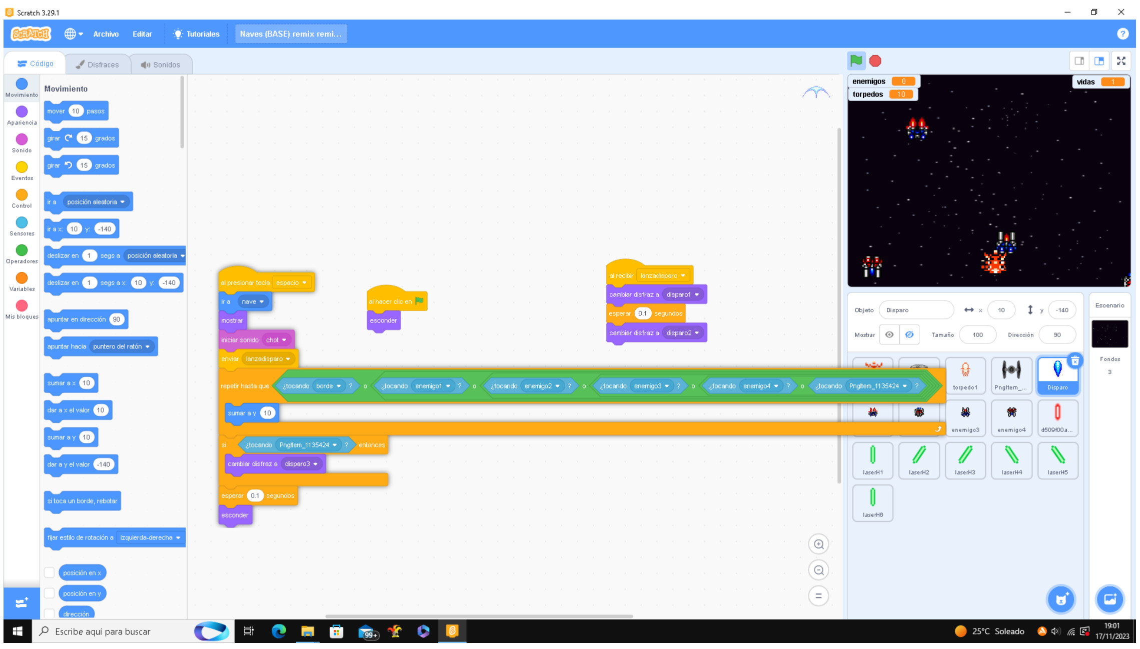
```
when 🏴 clicked
set enemigos ▾ to 0
switch costume to nave ▾
go to x: 0 y: -200
glide 1 secs to x: 0 y: -140
forever
    if key up arrow ▾ pressed? then
        change y by 5
    if key down arrow ▾ pressed? then
        change y by -5
    if key right arrow ▾ pressed? then
        change x by 5
    if key left arrow ▾ pressed? then
        change x by -5
```

```
when I receive explotanave ▾
start sound explocion ▾
repeat 2
    switch costume to explota1 ▾
    wait 0.25 seconds
    switch costume to explota2 ▾
    wait 0.25 seconds
stop other scripts in sprite ▾
```

Figure 4. Code to program the movement of the main spaceship object. Note that left and down movements are performed using negative numbers.

During this stage, individualized guidance was provided to address challenges and facilitate optimal progress in programming learning.

The activities included tutoring sessions, which consisted of scheduling regular meetings with the tutor (parent) to review the student's progress and address questions or difficulties. An open space was created for discussion and clarification of concepts, thus fostering interaction and dialogue. Regarding the adaptation of the approach, a continuous assessment of the student's learning style was carried out to tailor educational strategies as needed. Specific areas requiring more attention or personalized focus were identified and addressed, ensuring an adaptive educational approach. Progress tracking was an essential part, maintaining a detailed record of the student's progress in terms of acquired skills and completed projects. These records were used to adjust teaching strategies and ensure consistent development. During this period, more advanced concepts were introduced, including the combination of programming logic and a more complex utilization of conditionals and loops, as illustrated in Figure 5, where the events for the main ship's shooting object were programmed.



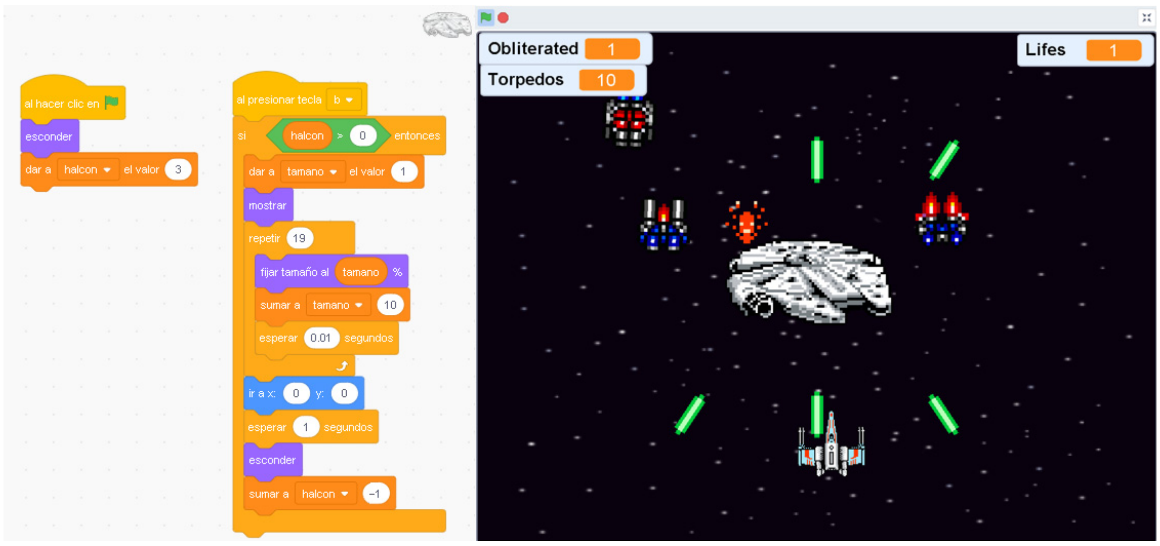
```
al presionar tecla espacio
ir a nave
mostrar
iniciar sonido chot
enviar lanzadisparo
repetir hasta que ¿tocando borde ? o ¿tocando enemigo1 ? o ¿tocando enemigo2 ? o ¿tocando enemigo3 ? o ¿tocando enemigo4 ? o ¿tocando PngItem_1135424 ?
    sumar a y 10
si ¿tocando PngItem_1135424 ? entonces
    cambiar disfraz a disparo3
esperar 0.1 segundos
esconder

al hacer clic en 🏁
esconder

al recibir lanzadisparo
cambiar disfraz a disparo1
esperar 0.1 segundos
cambiar disfraz a disparo2
```

Figure 5. Programming of the main ship's shooting object using a "for" loop and a conditional statement.

The creation of objects such as enemy ships and the user-controlled spaceship was initiated. Other action events were explored, such as shooting upon pressing a button, and additional concepts were developed, including different object collisions and animations. The student gradually assimilated these concepts, easily understanding conditionals and loops. However, they faced difficulties in creatively applying these concepts to address posed problems. Challenges with increasing levels of difficulty were designed to encourage the student's autonomy and enable them to use these concepts more independently. Therefore, despite initial challenges in creatively applying more advanced concepts, the student demonstrated steady progress in assimilating programming logic and its application in creating a Shmups-style game in the Star Wars universe. The strategy of gradual exposure and progressive challenges proved effective in strengthening their understanding and programming skills.

In the next phase of the guide, whose goal was to effectively consolidate and apply advanced programming concepts in more complex projects, with a focus on developing the ShuWars game. The designed activities aimed to foster creativity and the integration of learned elements. During the creation of personal projects, the student was encouraged to explore their creativity; to do this, they were assigned the task of including a new functionality not contemplated in the game. In this case, they were asked to implement an ally to assist in the mission of destroying enemy ships. The specific task involved making an object representing the "Millennium Falcon" gradually appear and launch six laser beams in various directions upon pressing the space key, causing the explosion of enemy ships upon collision.

In Figure 6, the final programming of this new object can be observed, as well as the achieved result.



```
al hacer clic en 🏴
esconder
dar a halcon ▾ el valor 3
```

```
al presionar tecla b ▾
si halcon > 0 entonces
    dar a tamano ▾ el valor 1
    mostrar
    repetir 19
        fijar tamaño al tamano %
        sumar a tamano ▾ 10
        esperar 0.01 segundos
    ir a x: 0 y: 0
    esperar 1 segundos
    esconder
    sumar a halcon ▾ -1
```

Figure 6. Programming a new feature for the video game project.

The student demonstrated a good application of the learned concepts by gradually making the spaceship appear upon pressing the space key and launching laser beams. However, the student faced difficulties in the action of laser collision. In response to this challenge, assistance was provided in the form of hints to guide the student without directly revealing the solution. This strategy was implemented with the aim of encouraging independent problem solving and strengthening their ability to creatively apply the concepts learned. Additionally, the "Presentation and Teaching" activity was included, which incentivized the student to present and explain the projects performed to other children or family members. This activity not only promoted the development of communication skills but also contributed to the consolidation of learning through teaching others. Therefore, this phase provided the student with the opportunity to apply advanced programming concepts in a personal and meaningful project. Although specific challenges were encountered, the provision of hints and the promotion of presentation and teaching to others contributed to their continuous development in the field of programming.

In the evaluation and feedback phase, the main objective was to assess the student's progress and provide constructive feedback for their continuous development in programming. The designed activities aimed to evaluate the implementation of the learned concepts and the ability to apply them independently, and foster self-assessment. Periodic reviews involved regular assessments of the student's work. During these reviews, the implementation of learned concepts was evaluated, providing detailed feedback on positive aspects and areas for improvement. This process allowed continuous assessment of the student's performance. In the individual project activities, specific projects were assigned to assess the student's ability to apply concepts independently. The assessment focused on creativity, problem solving, and technical implementation, providing a more comprehensive measure of the acquired skills. Self-assessment was encouraged as an essential part of the process, where the student reflected on their own progress and set goals for the future. This activity contributed to the development of metacognitive skills, promoting autonomous and conscious learning. As a result, the student was able to create new projects autonomously. These projects included the creation of cartoon-style scenarios or small movies and the proposal of new video games suggested by the student, such as a tag game. In the latter, the student designed a game, this time using a tablet, where one object moved randomly, while another object, controlled by the finger, allowed it to dodge (see Figure 7). This achievement demonstrated the student's ability to propose and successfully execute new games, marking a significant milestone in their programming development.

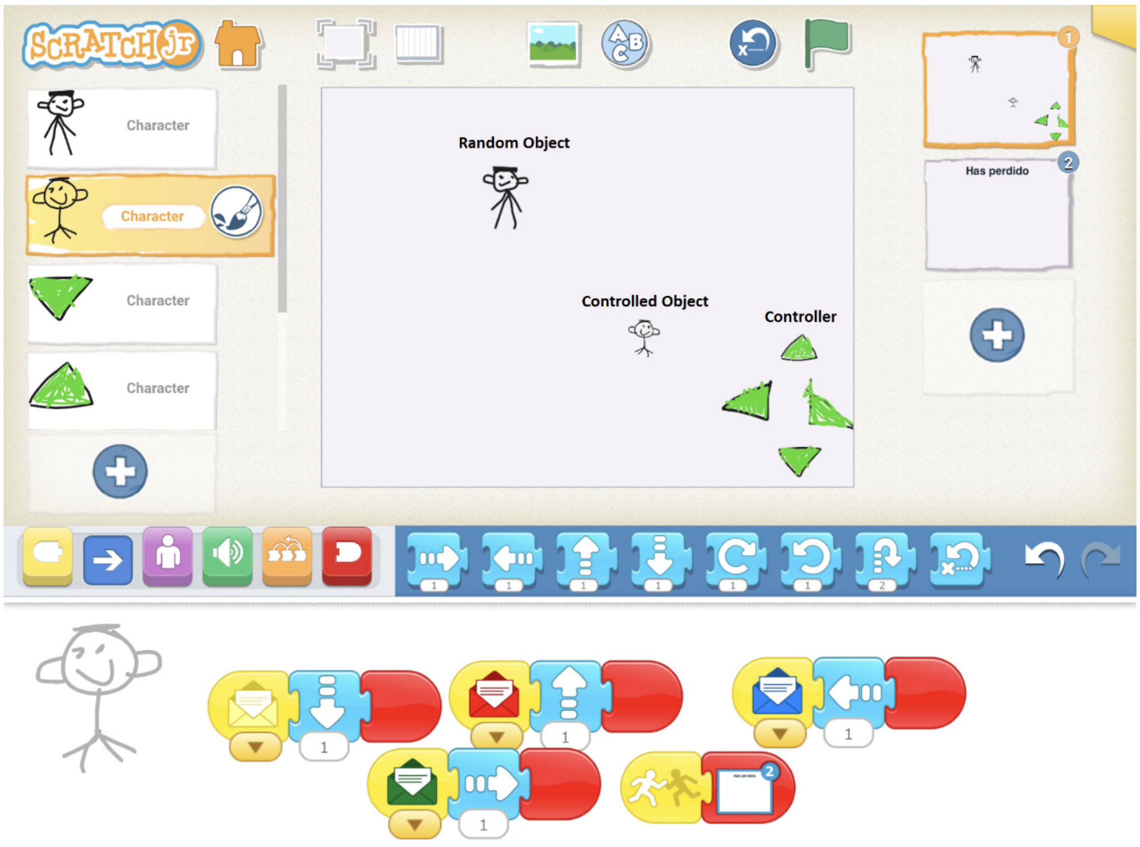


Figure 7. Proposal, design, and execution of an independent program.

In the phase of demonstration and teaching to others, the main objective was to reinforce the student's learning by teaching what they had learned to others, thus consolidating their programming knowledge. Specific activities were carried out to encourage the creation of educational videos and the organization of teaching sessions.

During the educational video creation activity, the student was encouraged to create explanatory videos where they shared programming concepts with a wider audience. This task aimed not only at developing communication skills, but also the ability to synthesize information in a clear and understandable manner. The creation of these videos allowed the student to express their programming knowledge in an accessible way for others. The student conducted some sessions where they had the opportunity to teach other children, specifically two elementary school students aged 8 and 10 (see Figure 8).

Although due to the student's young age and limited experience, and although the student was not a formal educator, the student managed to convey the basic use of the Scratch program and some fundamental programming concepts learned in previous stages. This experience aimed not only to reinforce the understanding of the concepts by explaining them to the student's peers, but also provided the student with the opportunity to gain confidence in the student's abilities. Thus, this phase allowed the student to apply and consolidate the knowledge acquired by teaching others. In the final phase, the main objective was to encourage the documentation of the learning process and the creation of a digital portfolio. Specific activities were implemented to incentivize continuous record-keeping and the creation of a portfolio highlighting the student's projects, skills, and progress.

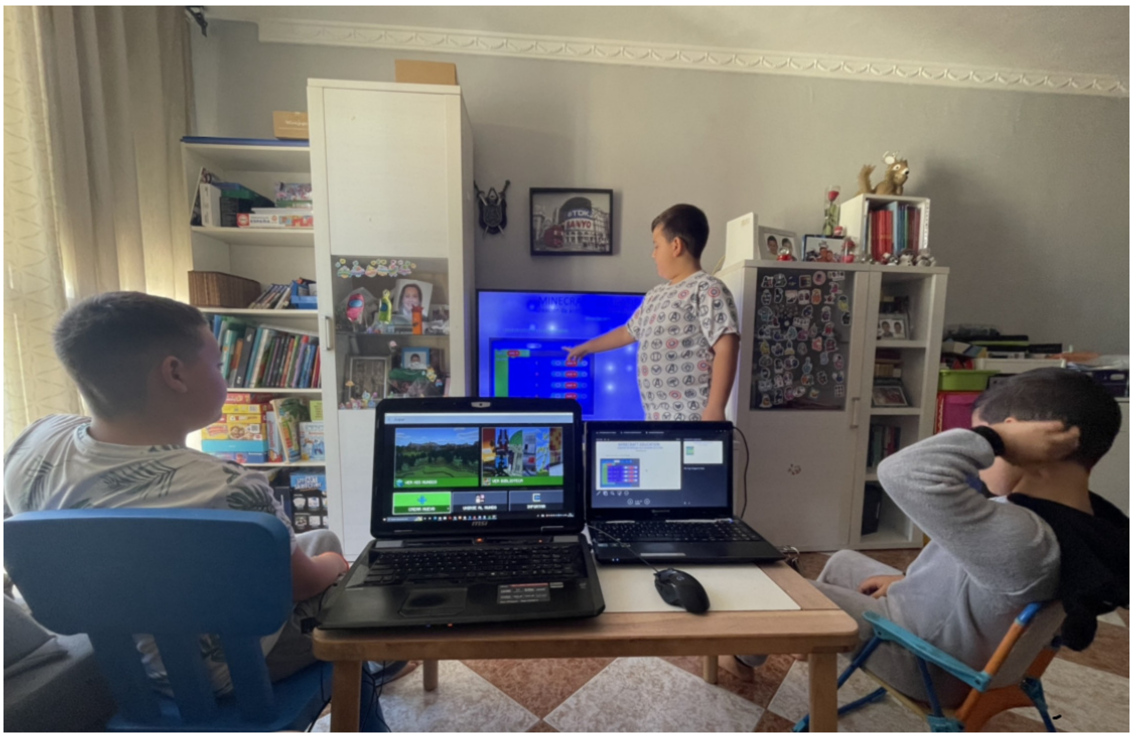


Figure 8. The student's transmission of what they have learned to other classmates and friends.

Regarding continuous record keeping, the student was encouraged to maintain a constant record of the activities, challenges overcome, and achievements. The use of learning journals or blogs was suggested as tools to document reflections and learnings throughout the programming journey. This practice not only contributed to the consolidation of learning but also allowed the student to continuously reflect on their progress. Additionally, the student was guided in creating a digital portfolio that showcased the developed projects and skills. It is important to note that the creation of an anonymous portfolio was emphasized since the student was underage, aiming to showcase the knowledge and projects developed in a closed environment without exposing the student publicly. This approach ensured the student's safety and privacy while providing a controlled platform to share achievements. The student's response was enthusiastic, showing excitement to explain what they had learned and present the projects worked on. The creation of this anonymous portfolio not only served as a valuable resource to highlight skills and projects, but also provided the student with a platform to express their passion and enthusiasm for programming.

In summary, the documentation and creation of a portfolio phase closed the learning cycle, providing the student with the necessary tools to continue developing and sharing achievements safely and in a controlled manner. This approach could contribute to the understanding of programming and cultivate the ability to continuously reflect on one's own learning.

7.1. Performance Results

Following the questionnaire shown in the section "Performance Evaluation Questionnaire" in the Supporting Information, the student's performance was assessed for the following items:

I. Mouse and Keyboard Handling;
II. Operating System Navigation (Windows);
III. Keyboard Shortcuts;
IV. Scratch Exploration and Project Creation;
V. Coordination and Initial Block Functionalities;

VI. Creation of Basic Projects and Gradual Development;
VII. Application of Essential Mathematical Concepts;
VIII. Participation in Tutoring Sessions and Adaptation of Approach;
IX. Creativity in Personal Projects and Teaching Skills;
X. Self-assessment and Documentation in the Portfolio.

Each item was evaluated by the educator with two quantitative questions, each rated on a scale from 1 to 5, where the meaning of each score is specified in the "Performance Evaluation Questionnaire" section of the Supporting Information. Therefore, the average of the scores obtained for each item was taken for each category, and the result can be seen in Figure 9.

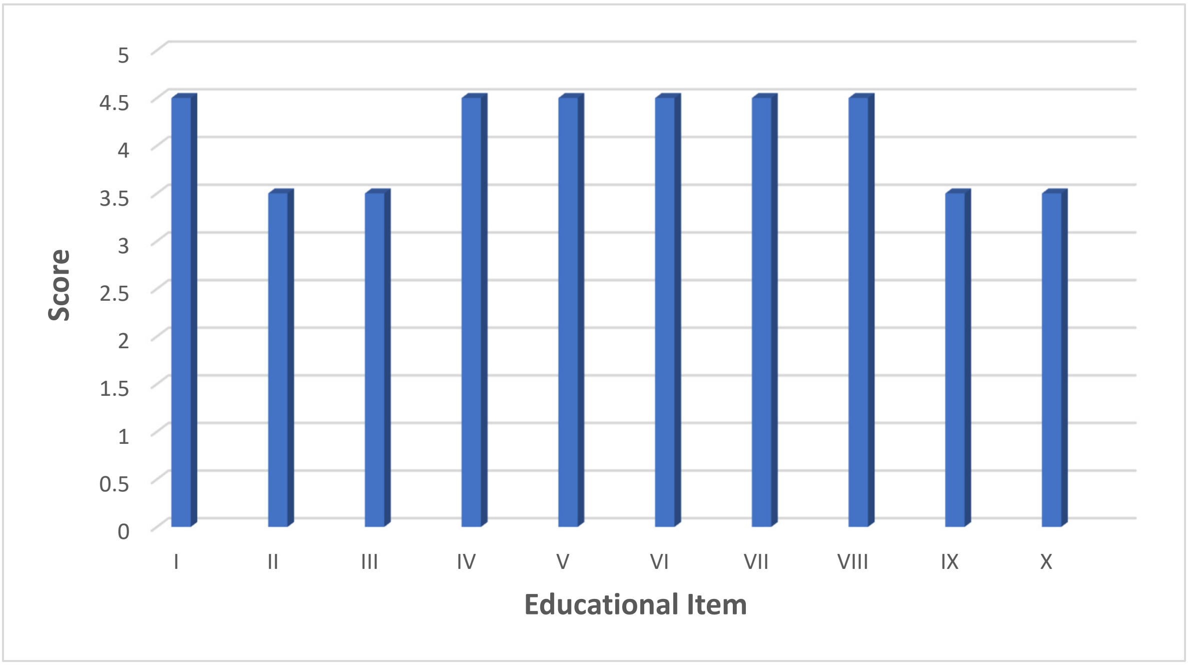


Figure 9. Results of performance in programming learning following the guide.

7.2. Results on Student Perception

To assess the student's perception, a semi-structured interview was conducted with the questions outlined in the "Perception Evaluation" section in the Supporting Information.

The codification of the transcript of this interview was:

(1) ***Love for Learning:***

Quotes: "I love learning new things in Scratch because each time I discover how to do more things".

Theme: The student expresses a genuine love for learning and highlights the excitement of discovering new possibilities in Scratch.

(2) ***Ease with Mouse and Keyboard:***

Quotes: "I have no problems using the mouse and keyboard; it was weird at first, but not anymore".

Theme: The student indicates comfort and ease with using the mouse and keyboard, overcoming initial challenges.

(3) ***Fun in Creativity:***

Quotes: "It's fun to create new characters and make them do things like dance or jump".

Theme: The student finds joy in the creative aspect of programming, particularly in creating characters and making them perform actions.

(4) Sense of Achievement:

Quotes: "When I finish a game or movie, I feel really good, and I want to show it to everyone".

Theme: Completion of projects brings a sense of accomplishment, and the student desires to showcase their work to others.

(5) Sharing and Teaching:

Quotes: "I want to showcase my projects and teach how I made them".

Theme: The student expresses a desire to share projects and teach others, indicating a sense of pride and willingness to contribute knowledge.

(6) Collaborative Problem-Solving:

Quotes: "Sometimes, I don't know how to do something, but my dad helps me, and we figure it out".

Theme: Collaboration with the parent (dad) in problem-solving, highlighting a supportive learning environment.

(7) Independent Exploration:

Quotes: "I like figuring out how to do other things on my own".

Theme: The student enjoys independent exploration, demonstrating a self-directed learning approach.

(8) Positive Learning Outcome:

Quotes: "I have learned a lot, and I like it".

Theme: The student acknowledges significant learning outcomes and expresses overall satisfaction with the learning process.

(9) Excitement for Future Learning:

Quotes: "I want to keep making video games and movies".

Theme: The student expresses enthusiasm for continuing the learning journey and creating more projects in the future.

(10) Positive Perspective on Mistakes:

Quotes: "Making mistakes helps you learn".

Theme: The student sees mistakes as a positive aspect of the learning process, emphasizing the role of errors in the learning experience.

8. Discussion

The implementation of the educational guide for the introduction to programming through Scratch has revealed several notable aspects in the learning process of the elementary school student. Throughout the different phases of the educational methodology, positive results and challenges were observed that are worth discussing to better understand the effectiveness of the proposed approach.

The combination of various educational methodologies, such as Discovery Learning, Hands-On Learning, Collaborative Learning, and Personalized Methodology, has proven to be effective in the context of teaching programming. The comprehensive approach not only focused on the acquisition of technical skills but also on the development of cognitive skills, creativity, and autonomy. The application of these methodologies synergistically addressed different learning styles and fostered a deep understanding of programming concepts.

The student experienced a brief initial adaptation phase due to familiarity with tablets compared to desktop PCs. However, the guide successfully overcame this barrier in a short period, highlighting the importance of practical and playful activities to facilitate the transition. The introduction of interactive games and creative projects significantly contributed to the student's rapid adaptation to the programming environment.

The teaching strategy, based on a gradual approach and progressive challenges, allowed the student to gradually assimilate more advanced programming concepts. The ShuWars game creation phase was particularly significant as it provided an opportunity to practically apply the acquired knowledge in meaningful projects. Although facing creative challenges, the strategy of providing hints instead of direct solutions proved effective in fostering independent problem solving.

The implementation of periodic evaluations, both quantitative and qualitative, provided a detailed understanding of the student's progress. Constructive feedback and continuous adaptation of the educational approach in tutoring sessions contributed to maximizing learning and addressing specific areas that required attention. Additionally, self-assessment and documentation in the portfolio not only served as tracking tools, but also encouraged metacognitive reflection and learning consolidation.

The inclusion of activities that encouraged the student's creativity, such as adding new features to the game and teaching others, proved to be crucial. These activities not only allowed the student to apply skills creatively, but also contributed to the development of communication and teaching skills.

Performance results show a consistent improvement in each evaluated item, indicating a solid understanding and application of programming concepts. The student's perception, expressed through the interview, reveals a continuous enthusiasm for learning and a growing confidence in programming skills.

9. Conclusions

The conclusions drawn from the implementation of the guide designed for the introduction to programming through Scratch reflect positive responses to the initial research questions, providing a comprehensive view of the effectiveness of the adopted educational approach.

The guide proved highly effective in facilitating elementary school students in acquiring skills for desktop PC use. Through the interactivity and playful nature of Scratch, students quickly overcame the initial adaptation phase, highlighting the importance of practical activities and interactive games in familiarizing them with desktop technology. The positive assessment in mouse and keyboard handling, as well as operating system navigation, supports the guide's ability to facilitate this process.

The guide not only facilitated, but also empowered, the acquisition of basic mathematics knowledge for programming learning. The gradual application of essential mathematical concepts, contextually integrated into Scratch project creation, resulted in positive performance in applying these concepts. The strategy of learning by doing, combined with creative activities, helped the student to relate mathematics to practical situations and problem solving, establishing a strong connection between both disciplines.

The guide not only successfully introduced elementary school students to the world of computer programming but also achieved good performance in learning fundamental concepts. The combination of educational methodologies, such as Discovery Learning and the creation of gradual projects, allowed a gradual assimilation of more advanced concepts. Positive evaluations in areas such as exploring Scratch, project creation, and applying essential concepts indicate a solid understanding and application of programming.

The student's perception, revealed through interviews, reflected not only a high level of motivation and satisfaction, but also a positive connection with the learning process. Willingness to face challenges, joy in discovering new capabilities, and the desire to share projects indicate a significant commitment to programming. Furthermore, self-assessment and documentation in the portfolio provided valuable tools for the student to reflect on their progress, thus positively consolidating the learning experience.

In summary, the designed guide has proven to be an effective educational tool to facilitate the access of elementary school students to desktop PC use, acquire basic mathematics knowledge, successfully enter the field of computer programming, and ultimately experience a positive and satisfying educational journey. The combination of pedagogical approaches, individualized adaptation, and continuous documentation has contributed to the success of the implementation, supporting the relevance and effectiveness of the proposed comprehensive educational approach.

10. Limitations

This study, while providing insights into the learning process of computer programming logic in an 8-year-old elementary school student through the Scratch program, has several limitations. The focus on a single participant, due to logistical and resource constraints, may limit the generalizability of the findings to a more diverse population of 8-year-olds. The absence of gender diversity with only a male participant restricts the understanding of potential variations influenced by gender. The study's homogeneous educational setting, conducted in a home environment with a parent as the educator, may not fully capture the impact of different learning environments.

11. Future Research Directions

While this study provides valuable insights into the effectiveness of the programming introduction guide through Scratch for a single student, there are several avenues for future research that could enhance the comprehensiveness and applicability of the findings:

Scaling to a Larger Cohort: Conducting a similar study with a larger and more diverse group of students would allow for a broader understanding of the guide's impact across different demographics, educational backgrounds, and learning styles.

Comparative Studies: To assess the relative effectiveness of the Scratch-based approach, future research could involve comparative studies with other programming education methods or tools, providing a more nuanced understanding of the guide's strengths and weaknesses.

Longitudinal Analysis: Tracking the progress of students over an extended period could offer insights into the long-term retention of programming skills and the sustained impact of the guide on students' academic and cognitive development.

Parental Involvement Studies: Exploring the role of parental involvement in the learning process, especially in a home-based setting, could provide valuable insights into the dynamics of parent–student interactions and their influence on programming skill acquisition.

Adaptation for Different Ages: Investigating the guide's adaptability for students of different age groups could contribute to tailoring educational strategies to specific developmental stages, ensuring the guide's relevance across various educational levels.

Incorporating Peer Learning: Assessing the impact of collaborative learning environments, where students work together on programming projects, could reveal the potential benefits of peer interactions in enhancing the learning experience.

Exploration of Learning Platforms: As technology evolves, exploring the guide's implementation on different learning platforms or environments, such as virtual classrooms or blended learning models, could provide insights into its versatility and adaptability.

Inclusive Education Considerations: Investigating how the guide accommodates students with diverse learning needs and styles would contribute to creating more inclusive programming education resources.

Integration with Formal Curricula: Studying the integration of the guide into formal educational curricula could provide insights into its alignment with academic standards and its potential as a supplementary resource for schools.

Continuous Guide Enhancement: Considering the rapid evolution of technology and educational methodologies, continuous research into updating and enhancing the guide to reflect current best practices and technological advancements is crucial.

Exploring these future research directions would contribute to a more comprehensive understanding of the guide's effectiveness and its potential to shape the landscape of programming education for elementary school students.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/digital4010002/s1.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: I want to express my heartfelt gratitude to my son for his determined and joyful participation in this study. His dedication and enthusiasm in delving into the world of programming have filled my heart with pride and joy. I am incredibly thankful for his infectious laughter, his moments of discovery, and for being the inspiration behind this project. This gratitude is not only directed at you, my dear son, but also to the wonderful person you are becoming. Your dedication and love for learning are a constant reminder of how incredible you are to this father who loves and admires you.

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

References

1. Alam, A. Educational Robotics and Computer Programming in Early Childhood Education: A Conceptual Framework for Assessing Elementary School Students' Computational Thinking for Designing Powerful Educational Scenarios. In Proceedings of the 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), Villupuram, India, 25–26 March 2022; pp. 1–7.
2. McClure, J.; Pilgrim, J. Implementing a 1:1 technology program in a rural, public school: A study of perceptions of technology integration. *J. Res. Technol. Educ.* **2022**, *54*, 302–316. [CrossRef]
3. Kazimoglu, C. Enhancing Confidence in Using Computational Thinking Skills via Playing a Serious Game: A Case Study to Increase Motivation in Learning Computer Programming. *IEEE Access* **2020**, *8*, 221831–221851. [CrossRef]
4. Harris, T.O. High School Students' Motivation for Engagement and Academic Success: A Case Study. Ph.D. Thesis, Grand Canyon University, Phoenix, AZ, USA, 2020.
5. Shavelson, R.J.; Towne, L. (Eds.) *Scientific Research in Education*; National Academies Press: Washington, DC, USA, 2002; ISBN 0-3090-8291-9.
6. Hecht, C.A.; Murphy, M.C.; Dweck, C.S.; Bryan, C.J.; Trzesniewski, K.H.; Medrano, F.N.; Giani, M.; Mhatre, P.; Yeager, D.S. Shifting the mindset culture to address global educational disparities. *npj Sci. Learn.* **2023**, *8*, 29. [CrossRef] [PubMed]
7. Ross, D.E.; Smalls, Y. *Classroom-Based Interventions to Reduce Academic Disparities between Low-Income and High-Income Students*; Wallace, B.C., Ed.; Springer Publishing Company: New York, NY, USA, 2008; pp. 461–2007. ISBN 978-0-8261-0313-0.
8. Jamil, M.G.; Isiaq, S.O. Teaching technology with technology: Approaches to bridging learning and teaching gaps in simulation-based programming education. *Int. J. Educ. Technol. High. Educ.* **2019**, *16*, 25. [CrossRef]
9. Perera, R. *Reforming School Discipline: What Works to Reduce Racial Inequalities*; Brookings Institution: Washington, DC, USA, 2022.
10. Cohn, T.A. Estimating contaminant loads in rivers: An application of adjusted maximum likelihood to type 1 censored data. *Water Resour. Res.* **2005**, *41*. [CrossRef]
11. Bresnihan, N.; Bray, A.; Fisher, L.; Strong, G.; Millwood, R.; Tangney, B. Parental Involvement in Computer Science Education and Computing Attitudes and Behaviours in the Home: Model and Scale Development. *ACM Trans. Comput. Educ.* **2021**, *21*, 1–24. [CrossRef]
12. Families First—Our Program. 2023. Available online: https://www.families-first.org/wp-content/uploads/2023/12/2023-Annual-Report.pdf (accessed on 30 June 2023).
13. McEvoy, O. Households by Number of Children, as a Percentage of Households with Children in the European Union in 2021. 2023. Available online: https://www.statista.com/statistics/933981/households-by-number-of-children-europe/#:~:text=In2021,49.4percentof,havingthreeormorechildren (accessed on 17 November 2023).
14. Bella, N. VII. Impact of Demographic Trends on the Achievement of the Millennium Development Goal of Universal Primary Education. 2002. Available online: https://api.semanticscholar.org/CorpusID:156608676 (accessed on 28 November 2023).
15. Kotschy, R.; Suarez Urtaza, P.; Sunde, U. The demographic dividend is more than an education dividend. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 25982–25984. [CrossRef]
16. Scratch: Programming Learning without Deep Coding Knowledge. 2023. Available online: https://scratch.mit.edu/ (accessed on 13 December 2023).

17. Maloney, J.; Resnick, M.; Rusk, N.; Silverman, B.; Eastmond, E. The Scratch Programming Language and Environment. *ACM Trans. Comput. Educ.* **2010**, *10*, 16. [CrossRef]
18. Kalelioglu, F.; Gulbahar, Y. The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Inform. Educ.* **2014**, *13*, 33–50. [CrossRef]
19. Oliemat, E.; Ihmeideh, F.; Alkhawaldeh, M. The use of touch-screen tablets in early childhood: Children's knowledge, skills, and attitudes towards tablet technology. *Child. Youth Serv. Rev.* **2018**, *88*, 591–597. [CrossRef]
20. Wheatley, K. Increasing Computer Use in Early Childhood Teacher Education: The Case of a "Computer Muddler. *Contemp. Issues Technol. Teach. Educ.* **2002**, *2*, 509–530.
21. Mumtaz, S. Children's enjoyment and perception of computer use in the home and the school. *Comput. Educ.* **2001**, *36*, 347–362. [CrossRef]
22. Bröhl, C.; Rasche, P.; Jablonski, J.; Theis, S.; Wille, M.; Mertens, A. *Desktop PC, Tablet PC, or Smartphone? An Analysis of Use Preferences in Daily Activities for Different Technology Generations of a Worldwide Sample BT—Human Aspects of IT for the Aged Population. Acceptance, Communication and Participation*; Zhou, J., Salvendy, G., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–20.
23. Pruet, P.; Ang, C.S.; Farzin, D. Understanding tablet computer usage among primary school students in underdeveloped areas: Students' technology experience, learning styles and attitudes. *Comput. Human Behav.* **2016**, *55*, 1131–1144. [CrossRef]
24. Power, R. *Technology and the Curriculum: Summer 2018*; University of Ontario Institute of Technology: Oshawa, ON, Canada, 2019.
25. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books, Inc.: New York, NY, USA, 1980; ISBN 0465046274.
26. Boyle, M. The history of Mr. Papert. *J. ISTE Spec. Interest Group Logo-Using Educ.* **1999**, *17*, 8–12.
27. Kalelioğlu, F. A new way of teaching programming skills to K-12 students: Code.org. *Comput. Human Behav.* **2015**, *52*, 200–210. [CrossRef]
28. Giannakoulas, A.; Xinogalos, S. A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Educ. Inf. Technol.* **2018**, *23*, 2029–2052. [CrossRef]
29. Abesadze, S.; Nozadze, D. Make 21st Century Education: The Importance of Teaching Programming in Schools. *Int. J. Learn. Teach.* **2020**, 158–163. [CrossRef]
30. Kaplancali, U.T.; Demirkol, Z. Teaching Coding to Children: A Methodology for Kids 5+. *Int. J. Elem. Educ.* **2017**, *6*, 32–37. [CrossRef]
31. Unahalekhala, A. Young Children's ScratchJr Coding Projects: ASSESSMENT and Support. 2023. Available online: https://www.raspberrypi.org/blog/childrens-scratchjr-projects-assessment-support/ (accessed on 18 August 2023).
32. Talaee, E.; Noroozi, O. Re-Conceptualization of "Digital Divide" among Primary School Children in an Era of Saturated Access to Technology. *Int. Electron. J. Elem. Educ.* **2019**, *12*, 27–35. [CrossRef]
33. Wei, X.; Lin, L.; Meng, N.; Tan, W.; Kong, S.-C. Kinshuk The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Comput. Educ.* **2021**, *160*, 104023. [CrossRef]
34. Stigberg, H.; Stigberg, S. Teaching programming and mathematics in practice: A case study from a Swedish primary school. *Policy Futures Educ.* **2019**, *18*, 483–496. [CrossRef]
35. Lin, P.-H.; Chen, S.-Y. Design and Evaluation of a Deep Learning Recommendation Based Augmented Reality System for Teaching Programming and Computational Thinking. *IEEE Access* **2020**, *8*, 45689–45699. [CrossRef]
36. Ray, B.D. *Homeschooling in the United States: Growth with Diversity and More Empirical Evidence*; Oxford University Press: Oxford, UK, 2022. [CrossRef]
37. Williams, C.; Alafghani, E.; Daley, A.; Gregory, K.; Rydzewski, M. Teaching Programming Concepts to Elementary Students. In Proceedings of the 2015 IEEE Frontiers in Education Conference (FIE), Washington, DC, USA, 21–24 October 2015; pp. 1–9.
38. Bruner, J.S. The act of discovery. *Harv. Educ. Rev.* **1961**, *31*, 21–32.
39. Kolb, D. *Experiential Learning: Experience as the Source of Learning and Development*; Prentice-Hall: Hoboken, NJ, USA, 1984; Volume 1, ISBN 0132952610.
40. Dillenbourg, P.; Baker, M.; Blaye, A.; O'Malley, C. *The Evolution of Research on Collaborative Learning*; Elsevier: Amsterdam, The Netherlands, 1996.
41. Hattie, J.; Timperley, H. The Power of Feedback. *Rev. Educ. Res.* **2007**, *77*, 81–112. [CrossRef]
42. Thomas, J. A Review of Research on Project-Based Learning. 2000. Available online: http://www.bobpearlman.org/BestPractices/PBL_Research.pdf (accessed on 1 January 2000).
43. Flavell, J.H. Metacognition and Cognitive Monitoring: A New Area of Cognitive-Developmental Inquiry. *Am. Psychol.* **1979**, *34*, 906–911. [CrossRef]
44. Bonwell, C.C.; Eison, J.A. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports*; George Washington University: Washington, DC, USA, 1991.
45. Scratch Starter Projects. 2023. Available online: https://scratch.mit.edu/starter-projects (accessed on 13 December 2023).
46. Maxwell, J.A. *Qualitative Research Design: An Interactive Approach*; Sage Publications: Washington, DC, USA, 2013.
47. Kaplan, B.; Duchon, D. Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study. *MIS Q.* **1988**, *12*, 571–586. [CrossRef]

48. Huyler, D.; McGill, C. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, by John Creswell and J. David Creswell. Thousand Oaks, CA: Sage Publication, Inc. 275 pages, $67.00 (Paperback). *New Horizons Adult Educ. Hum. Resour. Dev.* **2019**, *31*, 75–77. [CrossRef]
49. Hollweck, T. *Case Study Research Design and Methods*, 5th ed.; Sage Publication, Inc.: Thousand Oaks, CA, USA, 2014; 282p.
50. Hattie, J.; Yates, G.C.R. *Visible Learning and the Science of How We Learn*, 1st ed.; Routledge: Oxfordshire, UK, 2013.
51. Clancy, M.; Linn, M. Case studies in the classroom. *ACM SIGCSE Bull.* **1992**, *24*, 220–224. [CrossRef]
52. Plavnick, J.; Ferreri, S. Single-Case Experimental Designs in Educational Research: A Methodology for Causal Analyses in Teaching and Learning. *Educ. Psychol. Rev.* **2013**, *25*, 549–569. [CrossRef]
53. Patton, M.Q. *Qualitative Research and Evaluation Methods*; Sage Publications: Thousand Oaks, CA, USA, 2002.
54. Campbell, D.T.; Stanley, J.C. *Experimental and Quasi-Experimental Designs for Research*; Rand McNally & Company: Chicago, IL, USA, 1963.
55. Creswell, J.W.; Plano Clark, V.L. *Designing and Conducting Mixed Methods Research*; SAGE Publications: Thousand Oaks, CA, USA, 2011; ISBN 9781412975179.
56. Fuchs, L.S.; Fuchs, D.; Hamlett, C.L.; Walz, L.; Germann, G. Formative Evaluation of Academic Progress: How Much Growth Can We Expect? *School Psych. Rev.* **1993**, *22*, 27–48. [CrossRef]
57. Bell, S. Project-Based Learning for the 21st Century: Skills for the Future. *Clear. House A J. Educ. Strateg. Issues Ideas* **2010**, *83*, 39–43. [CrossRef]
58. Black, P.; Wiliam, D. Developing the theory of formative assessment. *Educ. Assess. Eval. Account.* **2009**, *21*, 5–31. [CrossRef]
59. Wolcott, H.F. *Ethnography: A Way of Seeing*; Rowman & Littlefield: Lanham, MA, USA, 2008.
60. Miles, M.B.; Huberman, A.M.; Saldaña, J. *Qualitative Data Analysis: A Methods Sourcebook*; Sage Publications: Thousand Oaks, CA, USA, 2014.
61. Kemmis, S.; McTaggart, R. *The Action Research Planner*; Deakin University: Geelong, Australia, 1988.