IRSTI 14.25.07

10.51889/2959-5894.2024.87.3.025

L.Sh. Sabyrkhanova[1*], L.K. Zhaidakbayeva[2], A.B. Seitkhanova[2]

[1]O. Zhanibekov South-Kazakhstan Pedagogical University, Shymkent, Kazakhstan
[2]M. Auezov South Kazakhstan University, Shymkent, Kazakhstan
[*]*e-mail: s.lazzat_777@mail.ru*

USING SCRATCH SOFTWARE TO DEVELOP THE COMPUTATIONAL THINKING OF PRIMARY SCHOOL STUDENTS

Abstract

The article examines the impact of using the Scratch program on the development of computational thinking in younger schoolchildren. The purpose of this study was to teach primary school students the subject of "Digital literacy" tasks "Creating an online game with codes" on the topic "Logical operators" were proposed, and studies on the development of students' computational thinking were developed. Task execution planning provides for the desire to create independently created games, successfully complete the stage of their execution in the Scratch program, and share your tasks in the Scratch program with classmates, introducing them to the network. Some elementary school students got acquainted with the tasks shared by their classmates and showed a high level of computational thinking and motivation to create additional games from them for the online platform. As a result, the influence of students on the skills of computational thinking, communication and independent development is shown. Due to the development of information technology, learning programming at an early age is important because they should not have problems understanding the logic of programming when they reach the age of a bachelor's degree. Using visual, two-dimensional Scratch for this purpose allows children to express their ideas and thoughts interactively. They can create their own projects using blocks with various functions such as motion, sound and animation. This teaching of children to present their ideas and concepts in a certain form shows the impact on the development of their representational abilities, that is, on their computational thinking.

Keywords: Scratch, programming, motivation, computational thinking, coding, project, divergent, convergent, digital content, problem solving.

Л.Ш. Сабырханова[1], Л.К. Жайдакбаева[2], А.Б. Сейтханова [2]
[1]Южно-Казахстанский педагогический университет имени О. Жанибекова, г. Шымкент, Казахстан
[2]Южно-Казахстанский университет имени М.Ауэзова, г.Шымкент, Казахстан

ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ SCRATCH ДЛЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОГО МЫШЛЕНИЯ УЧАЩИХСЯ НАЧАЛЬНОЙ ШКОЛЫ

Аннотация

В статье рассматривается влияние использования программы Scratch на развитие вычислительного мышления младших школьников. Цель данного исследования в обучении учащихся начальных классов предмету "Цифровая грамотность" были предложены задания "Создание онлайн игры с кодами" на тему "Логические операторы", разработаны исследования развития вычислительного мышления учащихся. Планирование выполнения задания предусматривает желание создавать самостоятельно созданные игры, успешно завершая этап их выполнения в программе Scratch, в режиме онлайн делиться своими задачами в программе Scratch с одноклассниками, внедряя их в сеть. Некоторые ученики начальных классов познакомились с заданиями, которыми поделились их одноклассники, и показали высокий уровень развития вычислительного мышления и мотивации к созданию из них дополнительных игр для онлайн-платформы. В результате показано влияние учащихся на навыки вычислительного мышления, общения и самостоятельного развития. В связи с развитием информационных технологий обучение программированию в раннем возрасте важно, потому что они не должны испытывать проблем с пониманием логики программирования, когда они достигают возраста бакалавриата. Использование визуального двумерного Scratch для этой цели позволяет детям выражать свои идеи и мысли в интерактивном режиме. Они могут создавать свои собственные проекты, используя блоки с различными функциями, такими как движение, звук и анимация. Это обучение детей представлению своих идей и концепций в определенной форме, показывает влияние на развитие их репрезентативных способностей, то есть на их вычислительное мышление.

Ключевые слова: Scratch, программирование, мотивация, вычислительное мышление, кодирование, проект, дивергентность, конвергентность, цифровой контент, решение задач.

Л.Ш. Сабырханова[1], Л.Қ. Жайдақбаева[2] , А.Б.Сейтханова [2]
[1]Ө.Жәнібеков атындағы Оңтүстік Қазақстан педагогикалық университеті, Шымкент қ., Қазақстан
[2]М.Әуезов атындағы Оңтүстік Қазақстан университеті, Шымкент қ., Қазақстан

БАСТАУЫШ СЫНЫП ОҚУШЫЛАРЫНЫҢ ЕСЕПТЕУ ОЙЛАУЫН ДАМЫТУ ҮШІН SCRATCH БАҒДАРЛАМАЛЫҚ ҚҰРАЛЫН ПАЙДАЛАНУ

Аңдатпа

Мақалада Scratch бағдарламасын қолдану бастауыш сынып оқушыларының есептеу ойлауын дамытуға әсері қарастырылған. Бұл зерттеудің мақсаты бастауыш сынып оқушыларына "Цифрлық сауаттылық" пәнін оқытуда "Логикалық операторлар" тақырыбында "Кодтармен онлайн ойын құру" тапсырмалары ұсынылып, оқушылардың есептеу ойлауын дамытуға зертеу жасалынды. Тапсырманың орындалуын жоспарлануы, Scratch бағдарламасында оларды орындау кезеңін сәтті аяқтай отырып, өз бетінше жасаған ойындарын құруға құлшынысын, онлайн режимде Scratch бағдарламасындағы тапсырмаларын желіге енгізе отырып, сыныптастарымен бөлісуін қарастырады. Кейбір бастауыш сынып оқушылары сыныптастары бөліскен тапсырмалармен танысып, олардан онлайн-платформа үшін қосымша ойындар жасауға есептеу ойлауының дамуы мен мотивацияның жоғарғы деңгейін көрсетті. Нәтижесінде оқушылардың есептеу ойлау дағдыларын, коммуникацияны және тәуелсіз дамуына ықпал етуін көрсетеді. Ақпараттық технологияның дамуына байланысты бағдарламалауды ерте жастан үйрету маңызды,себебі олар бакалавриат жасына жеткенде бағдарламалау логикасын түсінуде қиындықтарға тап болмауы керек. Осы мақсатта визуалды екі өлшемді Scratch-ті пайдалану балаларға өз идеялары мен ойларын интерактивті түрде жеткізуге мүмкіндік береді. Олар қозғалыс, дыбыс және анимация сияқты әртүрлі мүмкіндіктері бар блоктарды пайдаланып өз жобаларын жасай алады. Бұл балаларды өз идеялары мен тұжырымдамаларын белгілі бір формада ұсынуға үйрету, олардың өкілдік қабілеттерін дамытуға, яғни есептеу ойлауына әсерін көрсетеді.

Түйін сөздер: Scratch, бағдарламалау, мотивация, есептеу ойлау, кодтау, жоба, дивергенция, конвергенция, сандық мазмұн, есептерді шешу.

Main provisions

The main points of the article can be summarized as follows:

- The use of Scratch is linked to the development of higher cognitive skills, including analyzing, synthesizing, and evaluating information. Despite the increasing importance of computational thinking in today's world, effective methods for developing these skills among schoolchildren are still not well understood. While using Scratch as a tool for teaching computational thinking offers innovative possibilities, further research is required to identify the best techniques and evaluate their impact on education.
- Representation is highlighted as the ability of children to perceive and interpret the world. Using Scratch allows children to express their ideas interactively through projects using various blocks with functions like motion, sound, and animation. This concrete representation contributes to the development of children's representational abilities.
- This article highlights the significance of using Scratch software to enhance computational thinking in elementary school students. The research and practical examples demonstrate that Scratch not only helps students grasp programming basics but also serves as a powerful tool for fostering computational thinking and sparking their imagination.

Introduction

The definite opposite of computational thinking is creative thinking. According L.I.Shishkina’s analysis of the works of foreign psychologists, it is revealed that creativity is not solely possessed by a limited number of exception geniuses who possess extraordinary talents and effortlessly break free from established norms. Instead, this aptitude is present in almost all individuals to different extents.

The opposite of computational thinking is standard thinking. This is due to the typical (standard, algorithmic) solution of the problems under consideration. It should be noted that this is how a person performs the vast majority of actions and thus solves his problems, and it is also possible to develop skills for solving them and further “automated” actions by standardizing. Computability of thinking is manifested in a non-standard approach to solving typical situations. In particular, such a situation can be an educational task, so the development of computational thinking (as opposed to calculus) in the educational process becomes practically possible.

Mastering standard techniques largely depends on studying "similarity actions" – patterns that the teacher demonstrates and the student repeats. This type of training should be considered ideal because it provides students with the necessary actions and, therefore, competencies.

The problematic situation is that, despite the growing importance of computational thinking in the modern world, the methods of its effective development among schoolchildren remain insufficiently studied. The use of Scratch as a tool for teaching computational thinking can offer innovative approaches, but more research is needed to identify optimal techniques and assess their impact on the educational process.

This article emphasizes the importance of using Scratch software to develop computational thinking in elementary school students. The proposed research and practical examples clearly show that Scratch not only contributes to the assimilation of the basics of programming, but is also a powerful tool for stimulating computational thinking and awakening the imagination of students.

Scratch software demonstrates that it promotes the development of basic skills such as problem thinking, logical reasoning, collaboration, and computational problem-solving by offering students the unique ability to compute interactive projects, games, and stories that allow them to implement their ideas digitally. Students become active participants in the educational process, which indicates their influence on independence and motivation.

Our solving of the problem at the computational level involves deviating from the template. There are these two types of computational thinking:

-divergent – the ability to find multiple solutions to the same problem;

-convergent – the ability to choose the best method of available solutions.

Usually, for developing of students' computational thinking, a certain set of special tasks are offered, the concepts of which differ from the standard ones. The development or selection of such tasks, of course, is performed by the teacher, and in this sense, the development is not systematic and artificial, since it depends on the desire (or unwillingness) of the teacher and the didactic tasks that the teacher sets.

The development of information technologies creates favorable and natural conditions for developing of computational thinking. Modern software, which is based on the principles of an object-oriented approach, offers several alternative ways to change the properties of objects on the screen and execute commands. For example, to copy the highlighted part to MS Word, you can use the context menu that appears after right-clicking; you can also left-click on the corresponding icon on the formatting panel or use the keyboard shortcut Ctrl+C or Ctrl+Fn+Ins. The teacher should pay attention to the presence of these features. Then, some students begin to independently look for and apply such "alternative" solutions, in addition to the commands or schemes indicated by the teacher. Thus, divergent thinking develops, the ability to come to non-standard solutions. At the same time, convergence, i.e., finding and choosing the optimal solution, is manifested in the development of “hotkeys” – keyboard shortcuts that allow you to execute commands faster than in the menu or toolbar, as experts in the field of Information Technology do. Note that such development is not limited to a specific discipline and is not limited in time: rather, it occurs throughout the entire process of mastering information technology. Furthermore, through the utilization of the Scratch program, students have the ability to bring their imaginative ideas to life by designing games. As illustrated in Figure 1, an engaging game called Elefun can be presented as an example, specifically tailored for primary school students. This game involves coding and enables the players to control the chosen character's movement both in a straightforward manner идти 10 шагов and by employing code blocks to move right повернуться в направлении 90 or left повернуться в направлении -90. Such activities foster the development of critical thinking skills.

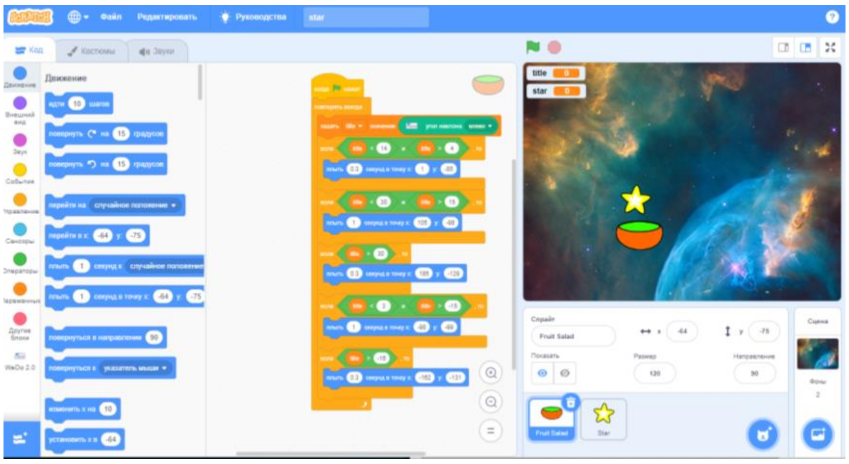
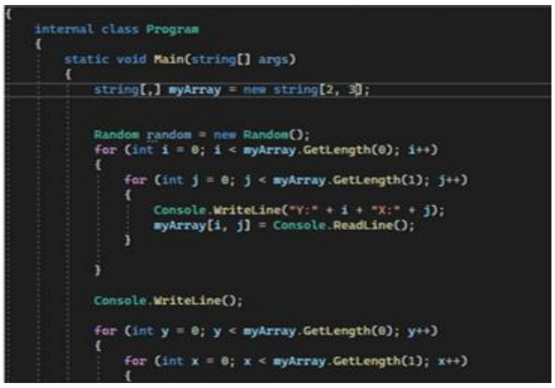


Figure 1. Encoding on Scratch

Coding, as the main trend in the modern world, occupies a leading position in education. It was evoked by the Decree of the Government of the Republic of Kazakhstan dated March 28, 2023 No. 249. In the Concept of Education development of the Republic of Kazakhstan for 2022-2026, free courses will be organized for schoolchildren on the development of digital skills, one of which is coding, including programming, robotics, 3D printing and others. Computer programming is perceived as an important competence for developing skills and computational thinking in addition to logical reasoning. Consequently, its integration at all levels of education, as well as at an early age, is considered valuable, and scientific research is being conducted to study this phenomenon in more detail. In the context of these facts, this study examines the impact of Scratch learning on the development of algorithms and computational thinking skills. It is important to teach students programming at an early age so that they do not have difficulties understanding the logic of programming when they reach the age of an undergraduate. To achieve this goal, Scratch, a visual, two-dimensional programming tool, was developed. For novice programmers, Scratch is the most popular block-based software to facilitate learning programming around the world [1]. In addition, Scratch is widely used in Kazakhstan and was chosen as a block-based coding tool for developing problem-solving abilities, self-efficacy, motivation, and interest. Despite such widespread use, there is not enough scope to understand how information and communication technology (ICT) teachers adopt and use Scratch to teach programming. Today, there are a huge number of types of programming languages. Each of them is used to perform different tasks. One of them is the Scratch programming language. This language is the easiest visual language to learn programming and the easiest way to start programming. Scratch is a program that allows you to create by playing, using blocks of colored code to combine them rather than writing instructions like other programs. Scratch is simple and easy to use, and yet it's a great language for teaching us the basic ideas of other programming languages that we need to use through the game. Scratch makes the process of writing code easy, diverse, and fun. The solution to many modern applied problems related to the need to visualize the results of scientific research presentations is impossible without graphical implementation. When teaching programming is considered, as a rule, students of higher educational institutions come to mind as the target audience. However, programming teaching has been integrated into the curriculum of secondary schools and even primary school and kindergarten classes, along with various applications for obtaining 21st-century qualifications in education [2]. Scratch, a free visual programming tool, was initially created at the Massachusetts Institute of Technology (MIT) in 2003. Through its website, users can share and communicate prepared projects with others [3]. This programming tool supports multiple languages and employs a user-friendly, block-based approach that appeals to individuals unfamiliar with programming. Research has been conducted on Scratch programs to facilitate training in algorithm development and programming. Several studies, including those by Begosso, Silva, Maloney (2013) and colleagues, focused on teaching algorithm development and programming to students aged 8 to 16 using Scratch. Additionally, Zoran and colleagues instructed college students in algorithm development and programming with the Scratch tool. Although Scratch has primarily been utilized for research on its impact on programming learning, there is a limited body of work exploring its potential to enhance students' computational and computational thinking skills, specifically in programming and algorithm development [4]. In recent years, it has been demonstrated that a tool like Scratch has made a great contribution to teaching programming to children (no coding errors or simple detection, no syntax errors, easy debugging, project development using multimedia tools, reduced cognitive load, simple and intuitive interface, etc.) and has become widespread. These programming tools have a structure that makes it easier for beginners to understand and use basic algorithmic forms and also prevents syntax complexity in programming by demonstrating the visual aspects of programming [5, 6]. The learner learns to think, to represent and to solve problems that require combining human cognitive power with computing ability [7, 8, 9].

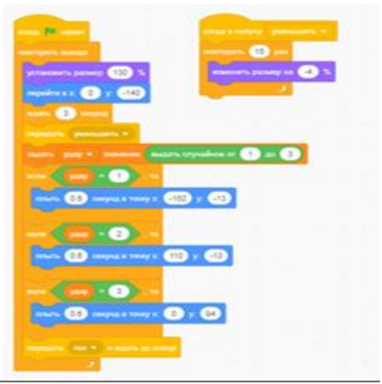
Research methodology

Coding is a whole series of commands written to provide communication between electronic and mechanical devices, computers, phones, tablets, and other devices, as well as people, and, as a result, to perform tasks step by step [10]. In other words, coding is a collection of programs written using programming languages, algorithmic methods, codes, and code blocks to solve one or more problems between the user and the computer, in which the coding is primarily based on one or more problems. To solve this problem, if there is one, the computer should provide the user with feedback. Communication is done by using programs written using encoding to get this response. The step-by-step program is solved using algorithmic methods for solving the problem. Then coding is performed using any programming language, and the solution to the problem is found. As shown in Figure 2, coding tools are divided into text and block tools.



```
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[,] myArray = new string[2, 3];


            Random random = new Random();
            for (int i = 0; i < myArray.GetLength(0); i++)
            {
                for (int j = 0; j < myArray.GetLength(1); j++)
                {
                    Console.WriteLine("Y:" + i + "X:" + j);
                    myArray[i, j] = Console.ReadLine();
                }

            }

            Console.WriteLine();

            for (int y = 0; y < myArray.GetLength(0); y++)
            {
                for (int x = 0; x < myArray.GetLength(1); x++)
                {
```

Encoding of text



Coding using bloks

Figure 2. Encoding Types

People who write a program in text coding write the program in text form according to the language they use. On the other hand, in block coding, there are blocks corresponding to each code. These blocks are continuously queued using drop logic, creating a program.

Incorrect use of programming tools in teaching students programming leads to the fact that students are biased towards programming. To avoid this bias, you need to choose a programming tool that is suitable for all age levels. To make learning programming easier for students, you should use simple programming tools. Programming is divided into text-based and block-based programming. Since learning a program is compared to learning a new language, students may be suitable first to him with prejudice. To do this, foremost, it is preferable to use visually weighted programming tools when coding. Block programming tools make learning enjoyable and also increase student motivation. Abstract concepts are visualized using block tools, which makes programming easier for anyone new to coding. It's easier to use blocks than to write code, especially at the beginner level. Block tools allow students to visualize and understand some situations and problems that they cannot implement. This makes writing code more fun and easier to understand. Among the main block-coding tools is Scratch. The most important features of this tool are that it is free and has visual effects that will make it easier for people at all levels to learn coding. On Scratch, when programming, each child gets to show the abilities of their creativity. Because of the environment, Scratch programming can make different cartoons, games, animated postcards, and presentations. Also, by inserting various photos into their graphic characters and adding sound, they can create interesting fictional fairy tales. These activities are carried out by students through the capabilities of a multimedia program. In the Scratch programming environment, every child can show their creativity. Because in the Scratch programming environment, you can prepare various cartoons, games, animated postcards, and presentations. He can also create interesting stories and fairy tales with the help of sound, create a variety of drawings, and perform graphic processing on his characters. All this is done by students using the multimedia capabilities of this program. It promotes computational thinking in the programming environment by setting your characters in motion, drawing, and working with different sounds.

This research endeavors to enhance pedagogical approaches in alignment with the requisites of Educational Research for the progression and ongoing enhancement of educational methodologies.

The principal objectives of this inquiry encompass:

-Furnishing students with a common medium for communication: object-oriented programming.

-Ensuring the operational viability and evolution of endeavors connected to programming originating from a rudimentary level.

-Fostering computational ideation through the resolution of challenges utilizing the Scratch programming environment.

This research was conducted involving two distinct cohorts of primary school students at the 'Bakdaulet' institution located in the city of Shymkent. The study included a combined group of 67 students, consisting of 31 women and 36 men. The average age of participating students was 9 years. In the present action plan, denoted as Figure 3 a tripartite approach was employed. The initial stage, designated as the first phase, entailed a comprehensive assessment of various methodologies relevant to the preliminary utilization of Scratch. Subsequently, the second phase was dedicated to the development of miniature gaming applications and fostering educational engagement within compact learning cohorts. The third and final phase was dedicated to disseminating the accomplished work to a wider online audience of users.

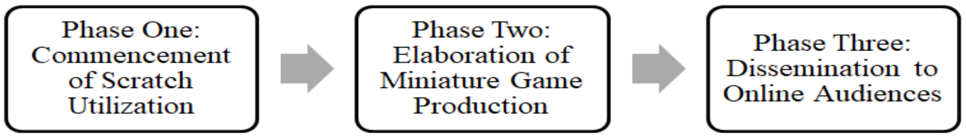

Phase One: Commencement of Scratch Utilization
Phase Two: Elaboration of Miniature Game Production
Phase Three: Dissemination to Online Audiences

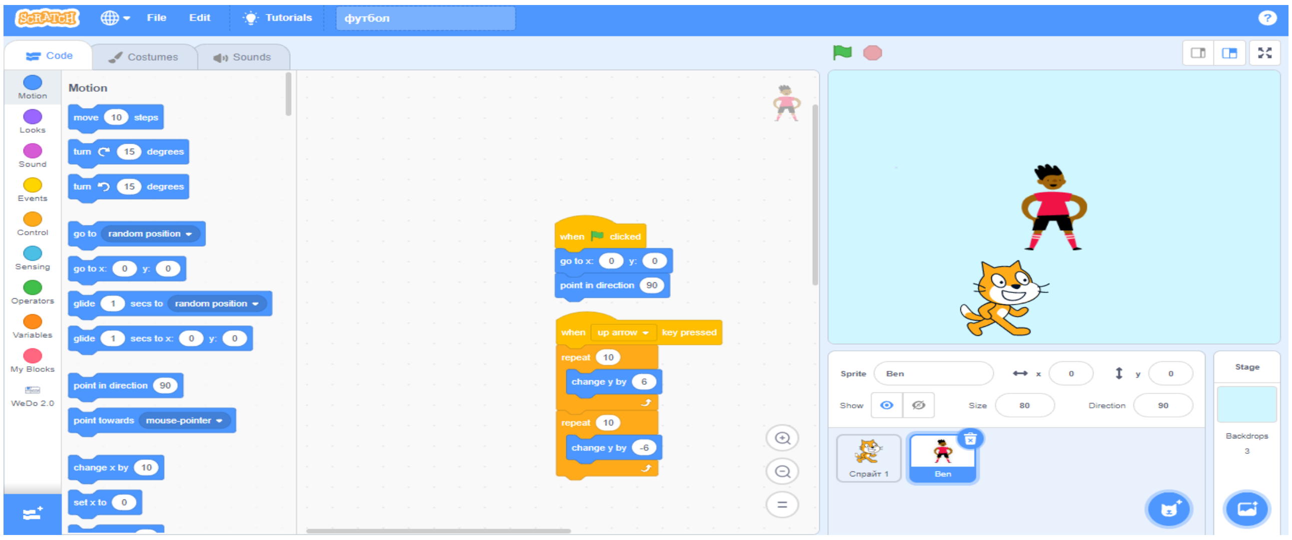
Figure 3. Phases of the plan

In the initial phase of the study, a comprehensive assessment of various methodologies for implementing Scratch software within the classroom setting was conducted. Subsequently, a curriculum comprising six introductory lessons, as detailed in Table 1 was devised.

Table 1. Activities plan

Lesson	*Activities*
1,2	*Introduction to Scratch*
3,4	*Activities 1,2,3,4,5,6*
5,6	*Explore Evaluate*

Each lesson was designed to encompass hands-on practical exercises, as illustrated in Figure 4 intended for individual students to actively engage in during classroom sessions.



```
when [flag] clicked
go to x: 0 y: 0
point in direction 90

when up arrow key pressed
repeat 10
    change y by 6
repeat 10
    change y by -6
```

Figure 4. Phases of the plan

In the initial phase, denoted as the adaptation to Scratch, each student successfully completed the prescribed activities. It was noteworthy that the students exhibited a high level of motivation within the classroom setting during this phase. Some students demonstrated swift task completion, affording them additional time to generate supplementary examples and expand their computational prowess. Notably, all students exhibited a strong affinity for learning with Scratch and expressed enthusiasm for manipulating the 'sprites' within the Scratch environment. Moreover, students derived substantial educational benefit from peer interactions within the class, gaining valuable insights from their classmates. This phase also revealed that students recognized the potential of seeking assistance from their peers rather than solely relying on the instructor, as peers of similar age often offered unique and computational problem-solving approaches and novel narrative concepts that surpassed those of the teachers. In the subsequent phase, encompassing six lessons documented in Table 2, the focus was directed toward the creation of a miniature game. In classroom 1, a template provided in 'Annex I' served as a reference point, prompting students to conceive and execute a miniature game grounded in their imaginative capacities.

Table 2. Activities plan

Lesson	*Activities*
1,2	*Mini game 1*
3,4	*Mini game 2*
5,6	*Explore Evaluate*

In this study, all participating students successfully generated their own mini-games, with some of them demonstrating a propensity to create multiple mini-games. Notably, the students exhibited considerable enthusiasm throughout the game creation process, fueled by the prospect of their games being actively played. Upon completion of this phase, they eagerly engaged in showcasing their independently crafted games to their peers. The third phase of the study encompassed four instructional sessions, as delineated in Table 3.

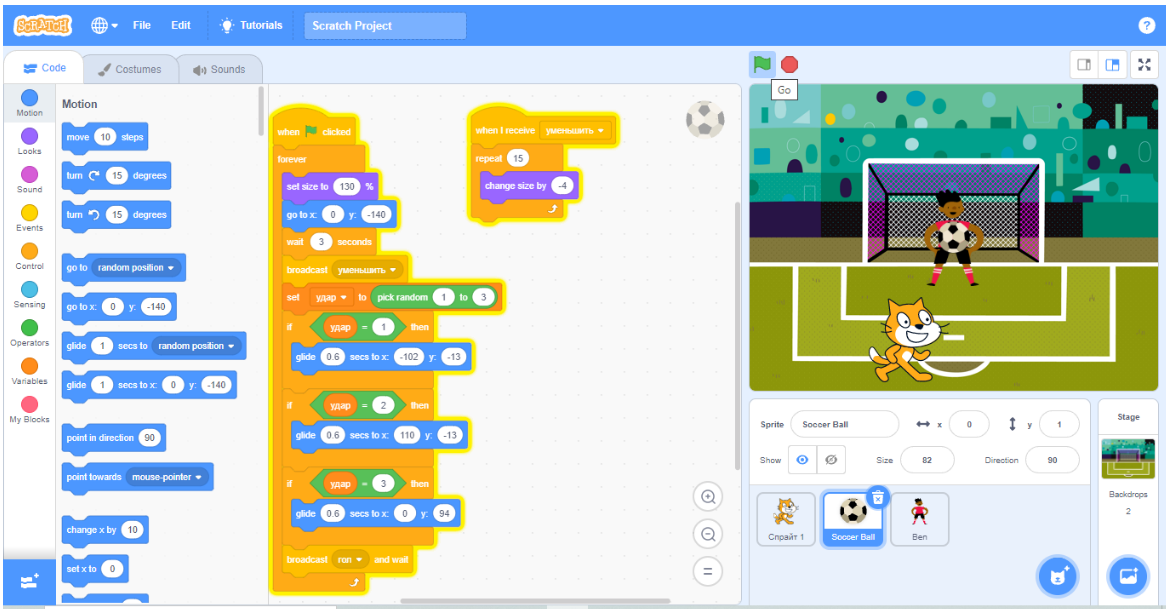
Table 3. Activities plan

Lesson	*Activities*
1,2	*Create an account (scratch.mit.edu)* *Share their projects*
3,4	*Create a project on-line* *Explore the site on-line Evaluate*

During this phase, students were tasked with the responsibility of disseminating their work over the Internet. To this end, each student visits the Scratch website (scratch.mit.edu), where they create their own accounts and start sharing their mini-games with other online peers. Interestingly, some students have demonstrated a high level of motivation on the website, not only playing games created by their peers, but also enjoying them, as well as continuing to create additional games for this online platform. In light of the aforementioned observations, it becomes evident that the Scratch website serves as a pivotal tool for these students. It functions as a conducive space where they can construct interactive activities, distribute their creations to a wider audience, and actively engage in a learning process that commences from the ground up.

Results of the study

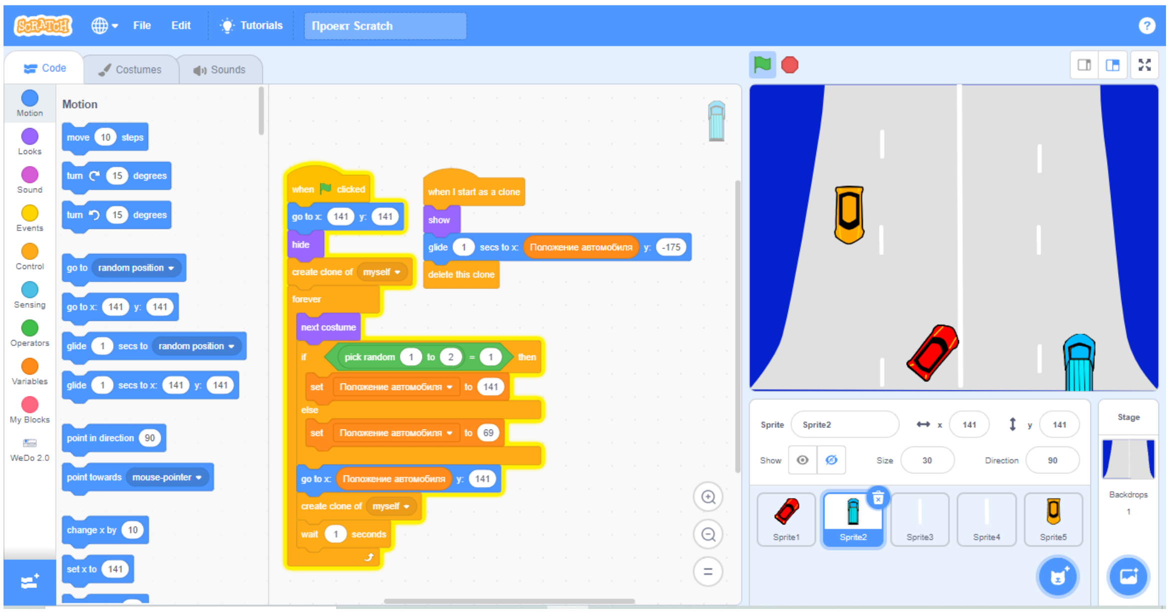
This report presents significant findings derived from the research investigation. During the initial phase (referenced to as Figure 5), a subset of students exhibited their computational abilities.



```
when [flag] clicked
forever
    set size to (130) %
    go to x: (0) y: (-140)
    wait (3) seconds
    broadcast (уменьшить)
    set [удар] to (pick random (1) to (3))
    if <(удар) = (1)> then
        glide (0.6) secs to x: (-102) y: (-13)
    if <(удар) = (2)> then
        glide (0.6) secs to x: (110) y: (-13)
    if <(удар) = (3)> then
        glide (0.6) secs to x: (0) y: (94)
    broadcast (гол) and wait

when I receive [уменьшить]
repeat (15)
    change size by (-4)
```

Figure 5. Some results of the 1st phase

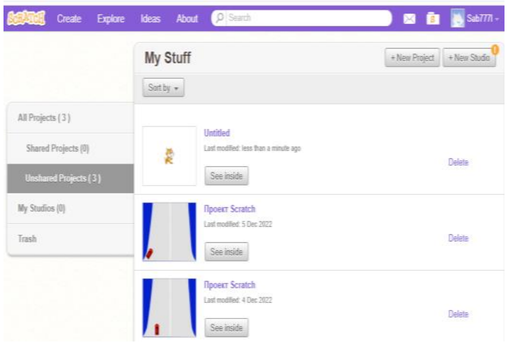
In the subsequent phase (referenced to as Figure 6), there was observable development in the manifestation of this creativity.



```
when [flag] clicked
go to x: 141 y: 141
hide
create clone of myself
forever
    next costume
    if pick random 1 to 2 = 1 then
        set Положение автомобиля to 141
    else
        set Положение автомобиля to 69
    go to x: Положение автомобиля y: 141
    create clone of myself
    wait 1 seconds

when I start as a clone
show
glide 1 secs to x: Положение автомобиля y: -175
delete this clone
```

Figure 6. Some results of the 2nd phase

In the third phase (indicated as Figure 7), the activities created were made available online, and it was observed that these activities were highly engaging and enjoyable to all participating students. During this phase, some students faced challenges in completing their assigned tasks, requiring additional time to conclude the activities.



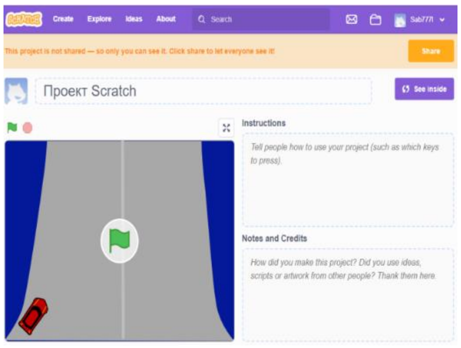


Figure 7. Some results of the 3rd phase

In general, the use of Scratch software in primary school is an effective tool for the development of computational thinking in students. It not only teaches programming but also promotes the development of computational skills, research thinking, communication, and independence. By supporting students in their research and computational efforts, we help them become strong and independent thinkers, ready for future challenges and achievements.

Discussion

As a result, this article emphasizes the importance of using Scratch software for the development of computational thinking in primary school students. The presented research and practical examples clearly demonstrate that Scratch not only contributes to mastering the basics of programming, but is also a powerful tool for stimulating computational thinking and awakening the imagination of students. Scratch software provides a unique opportunity to computational interactive projects, games, and stories, which allows students to implement their ideas digitally. It promotes the development of key skills such as problem thinking, logical reasoning, collaboration, and computational problem-solving. Students become active participants in the learning process, which contributes to their independence and motivation. We tested it in Scratch to see the impact of developing computational thinking. The idea of the approbation is to show the positive impact of completing tasks in scratch on the development of students' computational thinking. Diagnostic work was carried out, the purpose of which was to determine the initial level of development of students' computational thinking. A correctly completed task was rated at 10 points. The results of the phases of the task carried out on the topic of "Logical operators" are shown in Table 4. The results of the diagnostic work are shown in Figure 8.

Table 4. The results of the phases of the task carried out on the topic of "Logical operators"

Creating an online game with codes	*Phase one*	*Phase two*	*Phase three*
Task	*8,5*	*9,5*	*9*
Independent work	*8*	*10*	*9*

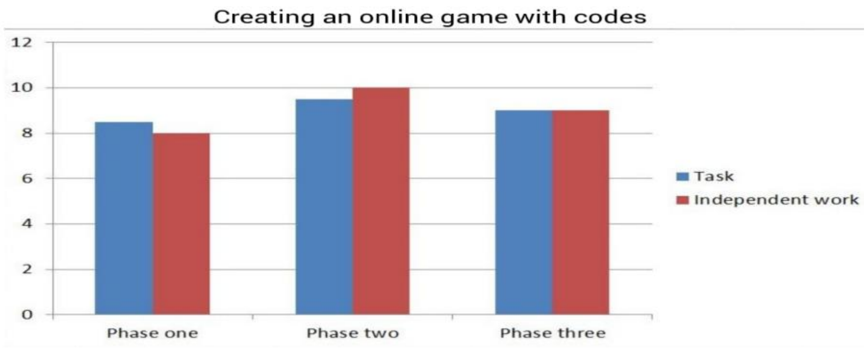


Figure 8. Histogram based on the results of diagnostic work

Based on these results, an analysis of the tasks was carried out and the following conclusions were obtained. The tasks of ordering actions in the algorithm, executing the branching algorithm, and uploading to the online platform have been successfully completed. In addition, the use of Scratch allows you to differentiate training, adapt tasks to the level of each student, and ensure the individual development of their computational potential. This helps to create an inclusive educational environment where every student has the opportunity to reveal their abilities and achieve success. In light of the rapid development of information technology and digital literacy, the use of Scratch software is becoming increasingly relevant for primary schools. By giving students the opportunity to be computational and proactive, we prepare them for the challenges of the modern world, where the skills of analysis, problem-solving, and innovation are in demand.

Conclusion

This research demonstrates the feasibility of incorporating Scratch into the information and communication technology (ICT) curriculum through a structured, progressive pedagogical approach. In conclusion, the use of Scratch software to develop the computational thinking of primary school students proves to be a highly effective and beneficial approach. Throughout this term, we have explored the various aspects of this innovative educational tool and its impact on young minds.

Firstly, Scratch software fosters creativity by enabling students to design, create, and share their interactive stories, games, and animations. This process empowers children to think outside the box, encouraging them to explore their imaginations and express their unique ideas in a digital format. By engaging with Scratch, students develop problem-solving skills as they encounter challenges and find innovative solutions to bring their visions to life. Moreover, the platform's user-friendly interface makes it accessible to students of all skill levels, ensuring that even those without prior coding experience can participate and thrive. This inclusivity fosters a sense of confidence and achievement among students as they witness their projects evolve from simple concepts to fully functional creations. As they gain familiarity with coding concepts and logical thinking, children are better equipped to tackle complex challenges in other subjects and future endeavors.

In addition, the collaborative nature of Scratch promotes teamwork and communication. Students can work together, exchange personal opinions and learn from each other. This cooperative learning environment not only enhances computational thinking but also instills essential social skills, preparing them for success in a rapidly evolving world where teamwork and adaptability are highly valued. In addition to the cognitive benefits, integrating Scratch into the primary school curriculum can significantly enhance engagement and enthusiasm for learning. The interactive and enjoyable nature of coding motivates students to actively participate in their education, fostering a positive attitude towards learning and boosting overall academic performance.

In this regard, it is important for both teachers and parents to understand the importance of developing computer thinking from an early age. By using Scratch software with elementary school students and developing computational thinking, we can create a solid foundation for future generations to succeed in various fields. The skills developed in this process go beyond just mastering software; they from a solid foundation for critical thinking, problem solving, and innovation by preparing students for Lifelong Learning and active contribution to society. In conclusion, using Scratch software to develop the computational thinking of primary school students is a valuable and forward-thinking educational approach. By embracing technology as a tool for creativity, we can empower young minds to become the innovators and problem solvers of tomorrow, shaping a brighter and more promising future for all.

References

[1] Zhang L. & Nouri J. (2019). A systematic review of learning computational thinking through Scratch in K-9 // Computers & Education. Vol. 141, pp. 1-25. URL: https://doi.org/10.1016/j.compedu.2019.103607

[2] Fessakis G., Gouli E., Mavroudi E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study // Computers & Education. Vol. 63, pp. 87–97. URL: https://doi.org/10.1016/j.compedu.2012.11.016

[3] Resnick M., Maloney J., Hernandez A., Rusk N., Eastmond E., Brennan K., Kafai Y. (2009). Scratch: Programing for all // Communications of the ACM. Vol. 52(11), pp. 60-67. URL: https://doi.org/10.1145/1592761.1592779

[4] Begosso L., Silva P. (2013). Teaching computer programming: A practical review // Paper presented at IEEE Frontiers in Education Conference (FIE), Oklahoma City, USA. pp. 508-510. URL: https://doi.org/10.1109/FIE.2013.6684875

[5] Aytekin A., Çakır F. S., Yücel Y. B., Kulaözü İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler [Coding science that gives direction to the future and some methods that can be used in coding learning]. Avrasya sosyal ve ekonomi araştırmaları dergisi. №5(5), 24-41. [in Turkish] URL: https://dergipark.org.tr/en/download/article-file/591508

[6] Kyriazos T. A., Stalikas A. (2018). Applied Psychometrics: The Steps of Scale Development and Standardization Process // Psychology. Vol. 9, pp. 2531-2560. URL: https://doi.org/10.4236/psych.2018.911145

[7] Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. Educational Psychologist. Vol. 50(4), pp. 313-334. URL: https://doi.org/10.1080/00461520.2015.1124 022

[8] Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12. Computers in Human Behavior. Vol. 41, pp. 51-61. URL: https://doi. org/10.1016/j.chb.2014.09.012

[9] Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. arXiv Preprint arXiv:1801.09258, en M. S. Khine (Ed.), Computational Thinking in STEM: Foundations and Research Highlights. Springer, Cham. URL: https://doi.org/10.1007/978-3 319-93566-94

[10] Sterling L. (2015). An education for the 21st century means teaching coding in schools. Retrieved from the conversation. [Electron resource]. URL:https://theconversation.com/an-education-for-the-21st-century-means-teaching-coding-in-schools-42046 (date of access 21.01.2023).