University of Novi Sad
Technical faculty "Mihajlo Pupin" Zrenjanin

Proceedings of the 12th International Conference on Applied Internet and Information Technologies AIIT 2022

14 October, 2022, Zrenjanin, Serbia

AIIT

International Conference

University of Novi Sad
Technical faculty
"Mihajlo Pupin"
Zrenjanin
Republic of Serbia

XII INTERNATIONAL CONFERENCE ON APPLIED INTERNET AND INFORMATION TECHNOLOGIES

AIIT 2022
PROCEEDINGS

October 14, 2022
Zrenjanin
Serbia

Proceedings publisher and organizer of the conference:
University of Novi Sad, Technical faculty “Mihajlo Pupin”, Zrenjanin, Republic of Serbia

For publisher:
Dragica Radosav, PhD, Full Professor
Dean of Technical faculty “Mihajlo Pupin”,
Zrenjanin, Republic of Serbia

Proceedings editors:
Zeljko Stojanov, PhD
Zoran Kotevski, PhD
Igor Bychkov, PhD

Conference Chairmans:
Zeljko Stojanov, Technical faculty “Mihajlo Pupin”, University of Novi Sad, Zrenjanin, Serbia
Zoran Kotevski, Faculty of Information and Communication Technologies, University “St. Kliment Ohridski”, Bitola, North Macedonia
Igor Bychkov, Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia

Technical preparation of the proceedings:
Zoltan Kazi, PhD; Zeljko Stojanov, PhD

Zoran Kotevski, PhD; Roman Kostromin, PhD

Aleksandra Stojkov, MSc; Milica Mazalica, MSc

Cover design:
Visnja Ognjenovic, PhD

Sinisa Mihajlovic, MSc

e-Proceedings
ISBN 978-86-7672-361-4

CIP - Каталогизација у публикацији
Библиотеке Матице српске, Нови Сад

004(082)(0.034.4)

INTERNATIONAL Conference on Applied Internet and Information Technologies (12 ; 2022 ; Zrenjanin)
Proceedings [Elektronski izvor] / XII International Conference on Applied Internet and Information Technologies AIIT 2022, Zrenjanin, October 14, 2022 ; [organizers] Technical Faculty "Mihajlo Pupin", Zrenjanin. - Zrenjanin : Tehnički fakultet "Mihajlo Pupin", 2022. - 1 elektronski optički disk (CD-ROM) : tekst, ilustr. ; 12 cm

Sistemski zahtevi: Nisu navedeni. - Naslov sa nasl. ekrana. - Bibliografija uz svaki rad.

ISBN 978-86-7672-361-4

а) Информационе технологије - Зборници

COBISS.SR-ID 81418505

Introduction

The objectives of International conference on Applied Internet and Information Technologies are aligned with the recent trends and goals of regional economic development. The focus of the conference is to foster the implementation of Internet and Information Technologies in all areas of human activities. The conference provides a forum for discussion and exchange of experiences between people from universities, research institutions, industry, government, and state agencies. The conference areas and relevant research topics are information systems, software engineering and applications, data science and big data technologies, business intelligence and IT support to decision-making, communications and computer networks, data and system security, distributed systems, Internet of Things and smart systems, embedded systems, computer graphics, IT management, E-commerce, E-Government, E-Education, Internet marketing, and IT practice and experience.

International Conference on Applied Internet and Information Technologies (AIIT 2022) is an annual conference that was held since 2012, based on successful results of the International Conference on Information and Communication Technologies for Small and Medium Enterprises in 2011. In this year the conference was held on October 14 in Zrenjanin, Serbia. The conference was successfully co-organized by 6 institutions from 4 countries - Serbia, North Macedonia, Russia, and Bulgaria. It has been managed in collaboration with 3 co-chairmen from Serbia, North Macedonia, and Russia.

The work during the conference was organized in four sessions: plenary session, online session, oral session, and poster session. In addition, a round table "Technical and soft skills of young IT engineers" with participants from academic organizations and IT industry was successfully organized.

The AIIT 2022 organizing committee would like to thank the authors of the papers for their contributions. All submitted papers were peer-reviewed through the double-blind review process. Each submitted paper was assigned to at least two reviewers from different countries. The organizing committee would like to express special gratitude to the reviewers who greatly contributed to the quality of the papers.

As conference chairs, we are confident that the AIIT international conference will continue its growth toward the goal of becoming a highly influential conference with great impact on Internet and information technology research and development.

Conference chairs:

Zeljko Stojanov, University of Novi Sad, Technical faculty “Mihajlo Pupin”, Zrenjanin, Serbia

Zoran Kotevski, University “St. Kliment Ohridski”, Faculty of Information and Communication Technologies, Bitola, North Macedonia

Igor Bychkov, Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia

Pair Programming In Primary Education - A Macedonian Case Study

Ilker Ali [1, 2], Blagoj Ristevski [2], and Ilija Jolevski[2]

[1] *Faculty of Informatics, International Vision University – Gostivar, North Macedonia*
[2] *Faculty of Information and Communication Technologies, University "St. Kliment Ohridski" – Bitola, Macedonia*

ilker@vizyon.edu.mk, blagoj.ristevski@uklo.edu.mk, ilija.jolevski@uklo.edu.mk

Abstract
This research examines the use of the pair programming technique in primary school education. This research explains how this technique is usable in primary school subject teaching. Furthermore, it investigates the achievements of pupils with the use of pair programming. A pair programming is one of the agile software development techniques in which two programmers work together on one computer. They call Coder the pilot, while the other they call the co-pilot or the observer. They evaluate each line of code, which the pilot typed. These two developers in our scenario change their roles. When reviewing, the co-pilot also considers the strategy by which the task will proceed. He also thinks of improving the written codes and the problems that may arise during further development. In this way, it avoids loading the pilot with these problems to focus his attention on the technical aspects of completing the current task. The data used in our paper are obtained from the evaluation of the fourth-class student's results.

Keywords:
Pair programming, socialization through programming, learning and education

1. Introduction

Pair programming is made up of extreme programming techniques, but since the early 90's it has been used and described by different experts under different names. In this kind of programming, two software developers work together to develop a program. Just like in the rally, one of the developers sits in the pilot's seat. The developer who writes the codes is called the driver. The second pilot is called an observer (a pilot). The co-pilot's job is to monitor and assist the driver in intervening and finding the right path when the driver has a road problem.

The observer monitors the code writing and suggests strategic approaches. For instance, the direction in which the work goes also considers the actions that need to be taken to prevent future problems. For such a case, we should consider the following example:

Suppose writing of five services. Two of these services take different values from the user and return the answer by searching the database. The other three services receive GUIDs from the user and return records containing these GUIDs, and when the driver writes the first service, he only thinks of writing the service. The goal is to complete the service in the fastest way. Conducting the service in the shortest possible route is not bad and choosing this approach increases efficiency. The observer follows what the driver writes, considers other services, and should guide the driver.

People think linearly when they write code. In other words, when you solve a problem, the mind tries to solve it most shortly. The driver tends to take a shortcut in his first decisions. The observer should intervene if the driver fails. The observer's thinking in this manner allows the driver to be free in his work. With this freedom, the driver focuses more quickly on his work. If we continue with the

above example, we can quickly implement the tactical approach needed to complete the desired service. The point is to remember that the observer is present as a guide and that the driver trusts the observer. There are different variations and types of pair programming:

1.1. Driver and observer variations

Master - Master, can be the first choice for high efficiency and can produce excellent results. The biggest problem in this pairing is that they both are very experienced and will not bother to think of new and creative ideas while solving problems.

Master-Apprentice (student): This is an excellent opportunity for the mentor to prepare and educate the apprentice (Figure 1). In this pair, new and creative ideas can be discovered by the student. In this case, the leading partner transmits existing approaches to the student in response to these new and innovative ideas. This method also allows the qualified partner to challenge existing practices. The bad thing about this technique is that if the apprentice is very passive, they will experience the phenomenon of following the master.

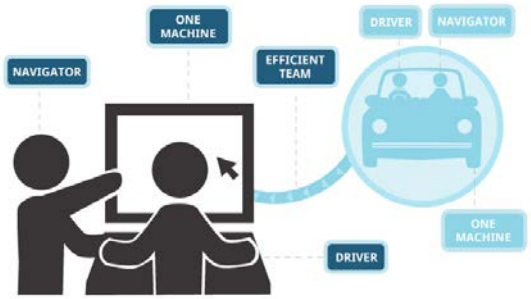

Figure 1: Example pair programming

Apprentice - Apprentice: This pairing can provide good results, but these probably will not be as useful as the previous combinations. Therefore, these are the reasons why this pairing is not recommended.

1.2. Remote pair programming

Remote pair programming, or virtual pair programming, means that two software developers are in different locations but they work on the same problem simultaneously. Partners develop code in real-time editors, desktop sharing screens, or remote programming IDE ports. Because they are not in physical contact, they cannot notice their partner's emotions, thus it can cause reasons for problems in sharing information and communication.

1.3. Ping-pong pair programming

Ping-pong pair programming has emerged as a slightly different approach from classic pair programming. This approach, commonly used by extreme programming teams, is implemented as follows in pairs consisting of individuals A and B:

- A writes a test and sees the test he failed.
- B sits in the driver's seat and writes the smallest code snippet to pass the test.
- B writes a new test and sees that the test has not passed.
- A sits in the driver's seat and writes the smallest code snippet to pass the test.
- A writes a new test and sees that the test did not pass.

The process of writing code continues in this way when refactoring is needed, and the driver does the refactoring. The purpose of developing code in this way is to ensure that software developers stay focused.

Pair programming is a continuous code review process.

In pair programming, the aim is to create a flow. A flow is a state of deep thought. Many software developers think that they can catch the flow only when they are alone and silent. For this reason, it is considered that this deep state of thought cannot be captured while communicating with someone. If both friends focus on the same subject, this mental state can be caught.

I can get the same results without the observer.

Creativity is not like mental intelligence. Most of the invention comes from retelling your ideas. Collaboration and communication are important factors for creativity to emerge. Ron Jeffries [1] has said: “Do not be afraid of pair programming: You are not as good as you think, but you are not as bad as you are afraid.”

Everyone uses pair programming. Just what happens under the name ***we are working on a huge mistake*** [2]. Most of the time, a software developer works with another software developer to fix a huge bug. They are then separated because software development is studied as a stand-alone task. Extreme programming has only slightly improved the activity and given it a new name.

The rest of the paper is structured as follows.

2. Related Works

1960s’ Logo programming language was first used as an intellectual thinking educational tool for mathematics [3]. After the Logo programming language, programming to teach thinking skills in primary schools was minimal [4]. In recent years, however, interest in introducing student programs has increased again [5]. This was supported by the availability of visual programming languages such as Scratch [6] [5], Stagecast Creator [7] and Alice [8] [9].

In studies, pair programming is a good technique for students' programming learning and social development [10] [11]. In these studies, it has been shown that pair programming is more advantageous than individual programming in completing the assignment [12]. There are studies conducted to increase students' self-confidence [13]. They stated that students' self-confidence had been increased in pair programming. Many studies have investigated combinations of male and female students and it is shown that pair programming has many benefits.

Increases discipline. Partners are generally in the "do" mode. Partners can take advantage of pair programming if they tend to avoid distraction. For example, if one of the partners is distracted, he will make mistakes that the partner does not usually make; the other will notice. In such situations, it is desirable to pause for a few minutes and then resume work than to work concentrated with many mistakes [14].

Improves code quality. Pair programming uses the experience of two programmers in writing code. The code written in this activity, where different perspectives, problem-solving, and experiences are combined, will therefore be better quality than the code written by one programmer. Conflicts arise in programming because partners do not always think the same. Solving these conflicts ensures the quality of work. The purpose of pair programming is to prevent errors from occurring. The Fig. 2. Diagram below describes the cost difference between the development of a defect and its presence in a production environment [15].

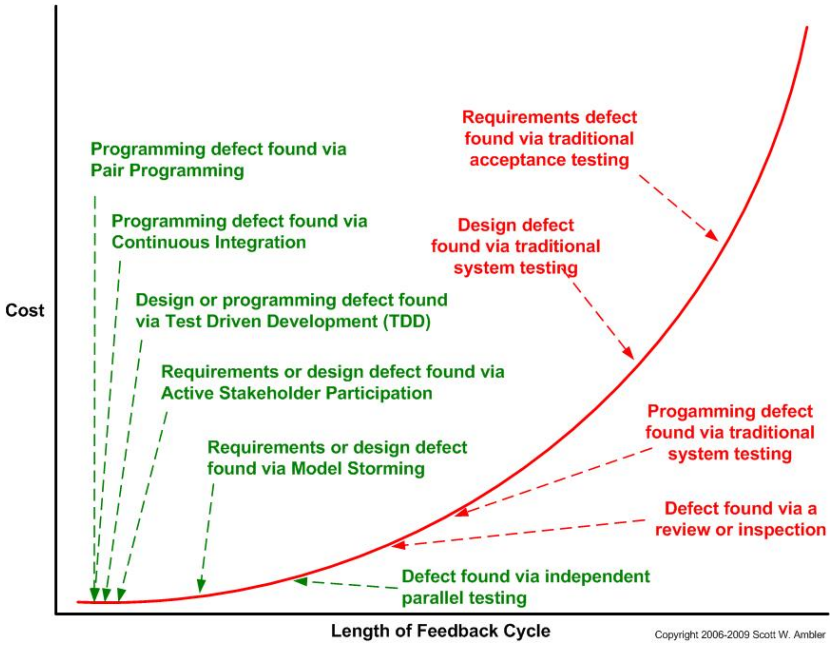

Figure 2: Length of the feedback loop [15]

According to the diagram, when an error is detected while performing pair programming, the replacement cost is very low. When this error is found during the continuous integration phase, the cost increases slightly. When a design error or programming error is found when performing test development, the cost increases again slightly. If there is a problem in identifying needs, if the project stakeholders actively participate and check the product, the error may still be found; of course, development and testing are carried out, and thus the price increases slightly.

Suppose the error is found in the production environment company, so you pay the highest price at this stage to correct the error. While the system has been fully tested, the cost of the change needed to correct the error is greater. It will also damage the trust of the end-user.

The programming process is easier. Partners must be in persistent communication and conversation. There is one point to which we need to draw attention. If the driver is in good condition and doing well in coding, if he does not make a mistake or has no better idea, the observer should not interrupt. The driver and the observer need to know how to communicate.

Increases motivation. Using digital tools has a positive impact on the teaching process and increases students' knowledge and motivation [16]. Well done pair programming is more fun than just writing code. Bad pair programming can be annoying.

Common ownership of codes. When everyone is working together on a project in pair programming partners they often change, and then that person knows the whole code. It is a great way to spread the word.

The feeling of being a team. In the Calamus program, independent developers know each other better. Pair teams move faster to become a team.

Less division. When people see two people working together, they are less willing to ask questions or share their work.

3. Research Methods

This study examined the advantages and disadvantages of individual and pair programming. We analyzed students' results after pair programming.

For this research, we use the following research methods:

- Independent programming and pair programming for students aged 11 years

- Scratch program
- A comparison of the results made with pair programming and self-programming.

1.4. Course flow

Daily preparation:	
Subject:	Informatics
Department	5th dept.
Teaching topic	Video games
Teaching unit	Video games
Type of class	New teaching content
Objectives	Introduction to programming through games. Introduction to the basic features of games. Understanding the concepts of background, characters, and movements in games Expected results: Students to get acquainted with the Scratch program Basic modules of the program Save and open the file To be able to use blocks To be able to put a background to the game
Teaching forms	Frontal, individual In pair programming: frontal and work in pairs
Teaching resources	Computer, projector
Course flow	
Introductory part	The teacher asks the students: What games do they play, what are computer games, how are these games made, do you want to make games? ... By asking the questions, the teacher determines the students' prior knowledge. The teacher explains what video games are. What were the first computer games? How were the games developed?
Main part	
Teacher activities:	• The teacher uses the frontal method and the projector as teaching helpers to demonstrate the program's work. The teacher explains the Scratch program through a presentation. • Opening the program and select the language of use • Getting to know the character, • With a mouse clicking to the first character that appears automatically and moving to the desired position. • Modules in the program (code, costumes, sounds, game screen, game background) • The teacher explains how to add or change the desktop background. • The task is set and at the same time the teacher shows how to work with the program Scratch, to set a background on the screen, "No movement, no game" using blocks to add character movement • With the help of the blocks, commands are given to the character: -

	Blocks to move: go 10 steps, the number 10 is manually increased or decreased. - Blocks for appearance: Speak "Hello!" 2 seconds, figure 2 seconds increases ... - after each change teacher tests by clicking the green flag so students can notice the changes. • The teacher explained how to add another character • In the second part of the class, the students start working according to the instructions they received from the teacher. • When they can't solve the problem, they ask for help from the teacher. • Finally, the teacher explains how to save the file • At the end of the lesson, the teacher distributes a questionnaire to get feedback on the students' work.	
Student activities:	Using Pair programming Two students work on each computer. One of the students works with the keyboard and mouse, while the other observes and corrects, reminds, or gives ideas for further work.	Without the use of pair programming Students work independently; each student works on one computer.
Final part	Students show their work. The teacher values and motivates the students' work. The students fill in the following survey. Survey Did you like the lesson: Yes No During the work which part you needed help from the teacher: • Open the Scratch program • Set the wallpaper on the desktop • Merging blocks • Character movement • Character talk • Saving the game	

To examine the benefits of pair programming, 5th graders were divided into two groups at different times. The first group of 50 students worked independently while the second group of 50 students was divided into 2 or 5 pairs. Both groups are given the same tasks and explanations.

The introduction and the beginning of the main part of the lesson are held in the same way in both cases. In self-programming, the student tries to do the task given by the teacher. Starting the program and selecting the language was not a problem for students. After starting the program students had to upload an image to the background of the game.

- Set a background image

Independent programming	**Pair programming**
Almost half of the students could not remember how to set the background image in the game. To proceed to the next step, they waited for a reminder from the teacher. The help from the teacher took place by intervening next to the computer itself.	With this way of working, the students who worked in pairs completed this task.

- Using blocks of pairs to move the character, changing direction at any angle, and the first tests to run the game

Independent programming	**Pair programming**

Many of the students entered the movement blocks but forgot to enter the start block. The teacher had to intervene in most of the computers to start the game. They successfully merged blocks but had difficulty learning how to delete blocks entered incorrectly.	Students added blocks; few students needed to be reminded of the starting block. They have successfully joined the blocks. With a little help, they realized that changing the number of blocks increased or decreased the movement by changing the sign of the number changes and the direction of movement.

- Using blocks to make characters speak

Independent programming	**Pair programming**
Few students began to try these blocks, but they couldn't make real communication due to a lack of time (Fig. 3).	Some of the students needed a little help from the teacher. After completing this task, the students had more time and were interested in entering another character to establish communication between characters. They began to add additional characters and extra seconds for communication between characters to be real with the explanation. They were interested in increasing and decreasing the size of the character, and they successfully managed that (Fig. 4).

- Results of the students' work

Independent programming	**Pair programming**
	
Figure **3:** Self-programming example	Figure 4: Pair programming example

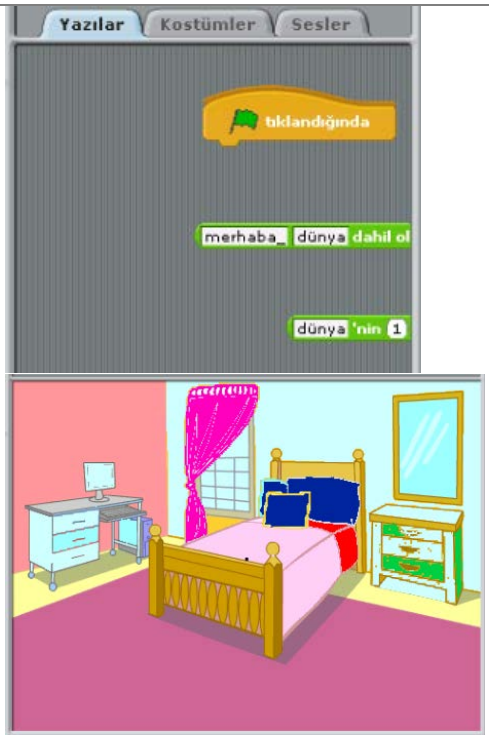

Figure **5:** Self-programming example

Figure 6: Pair programming example

Figure 3 shows that the student cannot remember how to enter the conversation blocks with the character.

Figure 7: Independent programming.

While in Figure 4, they use the pair programming technique, two students are in front of the computer, and it is obvious that they managed to insert a picture in the background of the scene and inserted another character, speech blocks, and time was required for the speech synchronization. So, one of the students remembers how to do it. The goal here is achieved, and both students learn how to solve this problem.

In the first case, the student waits for help from the teacher. Furthermore, if the teacher doesn't take or they are not successful in managing to take his time to help the students, the students will not be able to solve the problems independently. So the students will not be able to learn the material completely.

4. Discussion

This research observes another useful side of pair programming in education: the teacher intervenes less when solving students' problems individually. The first technique shows that assistance among the members or/of each student on an individual basis takes a lot of time because the teacher must first see where the mistake is made and then decides to explain how the problems might be solved. By doing this, they will have more time to explain to the whole classroom.

In both techniques of work, the students are interested in learning how to start the game. They are active in the classroom. They ask for help when they can't solve the problem. Firstly, they turn to the teacher. If the teacher is busy and somehow tries to get help from their friends. When the problem is solved, or something new is found, they want to show their friends a different background or character.

During pair programming in terms of self-programming, we find that the teaching process progresses faster. Creativity is greater; fewer pupils often turn to the teacher for help because one of the partners, in most cases, remembers the solution. In the beginning, some partners have misunderstood as "who would be the driver and who the observer", but this problem is unraveling with a mutual conversation or a conversation with the teacher.

Objective remarks

At the end of the class, the students filled out a questionnaire about the class. To the first question, "Did you like the class?" "All students answered YES.

The second question is essential for obtaining an objective answer to the advantages or disadvantages of pair programming over self-programming. For this reason, we ask the students, "When you asked a teacher for help?". The answers to this question are given below in table 1. The first column contains the task chapters given. The number of students asking for help regarding the given task is given in independent programming in the second column. The number of students asking for help regarding the given task is given during pair programming in the third column.

Table 1:

Statistical data from the survey

	Independent programming	Pair programming
Open the Scratch program	4	0
Setting the background	19	5
Merging blocks	15	4
Character movement	26	9
Character talk	38	18
Saving the game	32	16

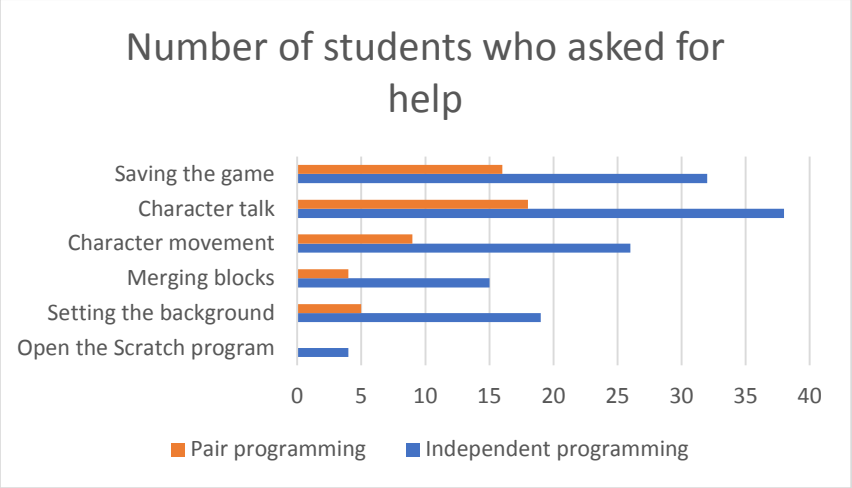

Figure 8: The number of requests for help

Another significant finding in our study is that pair programming is a useful approach to fostering computational and logical thinking for fifth graders. This result of our research matched the results in [17]. However, many factors affect the effectiveness of pair programming. According to [17], during programming, students are exposed to computational thinking. Computational thinking involves problem-solving, designing systems, and understanding human behavior, relying on fundamental concepts of computer science [17]. Many researchers believe that computational thinking is an essential skill for almost everyone in the digital age, not just computer scientists. More importantly, computer thinking is consistent with many aspects of 21st-century competencies, such as creativity, critical thinking, and problem-solving [18]. Thus, it is not surprising that many educators argue that programming provides an important context and opportunities for the development of computational thinking for K-12 students [6] [19] [5] [20]. According to McDowell (2006) [11] a study of students, with the programming language logo and the use of pair programming, found significantly better results obtained than those who work alone.

Another finding of our study is that using the pair programming technique, many of the problems are solved by the students themselves through mutual dialogue without waiting for help from the teacher. We detect that this gives the teacher more time to explain other things about the subject. Also is discovered that working together or solving problems allows students to get socialized faster and more. They teach that helping is positive behavior and leads to better success.

5. Conclusion and Future Works

In this paper, we explained the advantages of using pair programming, emphasizing the use of this technique as a teaching technique. According to the research done, we conclude that pair programming in teaching gives huge benefits. In our research, we find that the teaching units are successfully realized. We obtained that the students' creativity and motivation are increased, and pair programming technique develops socialization among the pupils. One of the most significant benefits of pair programming we detect is working together with the partner, merging ideas, group work, and strengthening friendships.

The results obtained in this study are consistent with the previous research. Pair programming has advantages over individual programming. Problem-solving and finding faults take less time for pupils. Similar results have been obtained in other studies [21].

The results of this research with students from 10 years of age suggest that the issue of pair programming in education should be examined with younger and older students. This type of research should be done with primary, secondary, and university students. Research on the socialization effect

of pair programming on students, the impact of helping each other, and the importance of friendship should be conducted. We used the Scratch programming language in this research. However, how other programming languages affect students should also be investigated. After Scratch block programming languages, the contribution of pair programming in writing programs with codes should be investigated. To better understand the contribution of pair programming in education, this technique must be tested with a larger amount of data. Students' achievements at different ages should be examined. Teachers' opinions about pair programming should also be analyzed.

References

[1] R. Jeffries, "ronjeffries.com," 2018. [Online]. Available: https://ronjeffries.com/articles/018-01ff/xp-revisited-1/.

[2] C. Yilmaz, "Pair Programming Üzerine Eleştiriler," 18 03 2016. [Online]. Available: http://www.yilmazcihan.com/pair-programming-uzerine-elestiriler/. [Accessed 2020].

[3] L. B. Feurzeig W., "Programming-languages as a conceptual framework for teaching mathematics," *Interactive Learning Environments,* p. 19, 2011.

[4] S. Grover and R. Pea, "Computational Thinking in K–12 A Review of the State of the Field," *Educational Researcher,* 2013.

[5] Y. Kafai and Q. Burke, "Computer Programming Goes Back to School," *Phi Delta Kappan,* 2013.

[6] B. Karen and R. Mitchel, "New frameworks for studying and assessing the development of computational thinking," *AERA,* 2012.

[7] J. Denner, L. Warner and E. Ortiz, "). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?," *Computers & Education,* 2012.

[8] M. J. Conway, Alice: Easy-to-Learn 3D Scripting for Novices, University of Virginia, 1998.

[9] K. Caitlin and P. Randy, "Using storytelling to motivate programming," *ACM,* 2007.

[10] T. DeClue, "Pair programming and pair trading: effects on learning and motivation in a CS2 course," *Journal of Computing Sciences in Colleges 18,* pp. 49-56, 2003.

[11] C. McDowell, B. Hanks and L. Werner, "Experimenting with Pair Programming in the Classroom," in In Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education.

[12] L. Williams and R. Upchurch, "In support of student pair-programming.," *ACM SIGCSE Bulletin. 33,* pp. 327-331, 2001.

[13] H. ÇAL and G. CAN, "The Influence of Pair Programming on Secondary School Students' Confidence and Achievement in Computer Programming," *Trakya Eğitim Dergisi,* pp. 243-259, 2020.

[14] A. Cockburn and L. Williams, "The Costs and Benefits of Pair Programming," 2009.

[15] S. W. Ambler, "Agile Modeling," 2009. [Online]. Available: http://www.agilemodeling.com/essays/costOfChange.htm.

[16] A. Selimi, M. Saracevic and A. Useini, "Impact of Using Digital Tools in High School Mathematics: A Case Study in North Macedonia," *Universal Journal of Educational Research,* pp. 8(8), 3615 - 3624, 2020.

[J. M. Wing, "Computational Thinking," *COMMUNICATIONS OF THE ACM,* 2006.

17]

[. M. e. a. Binkley, "Defining Twenty-First Century Skills. In: Griffin P., McGaw B., Care E. 18] (eds) Assessment and Teaching of 21st Century Skills," *Springer, Dordrecht,* 2012.

[Y. L and J. Hwee, "Review on teaching and learning of computational thinking through 19] programming: What is next for K-12?," *Computers in Human Behavior,* 2014.

[Resnick, "Scratch: Programming for Everyone," *Association for Computing Machinery,* 2009. 20]

[B. Grant, E. Marlin and W. Tim , "The Effects of Pair-Programming on Individual 21] Programming Skill," in *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science*, Portland, 2008.