Learning to program in secondary classrooms: Students’ and Teachers’ perceptions of the pair-programming setting.

Arash Issaee
Faculty of Computer Science
University of Vienna
Vienna, Austria
issaeea91@univie.ac.at

Renate Motschnig
Faculty of Computer Science
University of Vienna
Vienna, Austria
renate.motschnig@univie.ac.at

Katrin Göltl
Faculty of Computer Science
University of Vienna
Vienna, Austria
katrin.goeltl@univie.ac.at

***Abstract*— This Research to Practice Full Paper summarizes insights from three cycles of participatory action research on learning to program by developing simple video games in secondary-level classrooms using pair-programming. We found that pair-programming is an effective learning method for programming that additionally helps students acquire various other 21st-century skills in computer science classes at K9 level. As a follow-up to participatory action research, a series of seven interviews with computer science teachers and researchers from North America, Africa, and Europe, who focusg on pair programming, was conducted. The interviews were analyzed using qualitative content analysis by Mayring. Finally, the findings were interpreted and related to collaborative and experiential learning theories. Our mixed-methods study aims at presenting teachers’ attitudes toward the pair-programming setting. Furthermore, it examines if that setting is promotive to problem-solving skills, has an improved learning outcome, fosters a higher motivation, increases satisfaction, social skills, and positive attitudes toward coding.**

Based on the mixed-methods research, practical implications for learning to program in secondary-level classrooms are derived. With this contribution, we want to enlarge teachers’ and researchers’ repertoire to make programming-learning more engaging, more motivative, and more successful. This research should be the foundation for developing a positive image of programming for young learners of any gender and culture.

Keywords—pair-programming; learning to program; problem-solving; mixed methods; action research, qualitative content analysis

I. Introduction

Teamwork is a crucial part of CS learning and benefits students’ learning experience in sharing information and receiving feedback within a social community of peers [1]. Hence, a collaborative environment forms a crucial aspect of designing learning processes. Advantages of a collaborative learning environment are higher achievements, positive relationships among students and healthier psychological adjustment [2].

In pair-programming (PP), a special-case of collaborative working, each student is assigned a specified responsibility, which means that each student engages in programming and, subsequently, checks the coding of their partner. Pair-programming should facilitate the management of sophisticated tasks in programming by two people tackling a specific problem collaboratively and supporting each other with ideas so that they can find a solution with greater ease. Sharing the line of thought with a partner demands specific social skills and active communication. Because learners have to communicate what they know and understand, this improves their learning and indicates what they don't know. By employing this dynamic process, the partners serve as a resource for each other. [3].

When working in pairs, students are more likely to solve problems on their own, without the help of a teacher. Research on collaborative learning indicates positive influences on students of various ages and educational levels. According to Panitz and Panitz [4], there are five fundamental elements involved in cooperative learning: 1. Positive interdependence (meaning that each group member can only succeed when all other members succeed as well), [5] 2. Individual and group accountability, 3. Interpersonal and small group skills, 4. Face-to-face promotive interaction, 5. Group processing. Each group member can be held accountable for contributing to the project. Panitz and Panitz [4] also indicate that collaborative learning (CL) means forming a consensus through collaboration among group members and not to compete and to be better than the other group members. Collaborative learning emphasizes the social nature of learning and the need to teach students how to resolve conflicts, interact appropriately, and actively involve all group members. Since pair-programming is a special-case of a cooperative learning environment, the five elements of cooperative learning mentioned above would equally apply to learning to program in pairs. For instance, pair-programming in academic settings is beneficial to students in increasing students performances and retention in computer science [6] [7]. At the secondary level (K9), students who developed mini-games in pair-programming classes had better outcomes regarding coding- and problem-solving skills than students who solved the same tasks in a solo programming setting. Moreover, most students found that pair-programming accelerated the coding process [8].

In another study of 9- to 13-year-olds, conducted at Kerrisdale Elementary School in Canada, pairs who worked together on one computer, were more successful in solving computer puzzles and more motivated to continue playing than solo players or children playing next to each other on two computers [5].

Werner, Campe, and Denner developed an NSF-supported afterschool program for middle school-aged girls called "Girls Creating Games" to increase their information technology (IT) fluency through game creation [9]. The program was structured around collaborative work and pair programming; 85% of students preferred to work in pairs for the duration of the program. (The project-based program focused on the design and construction of interactive narrative computer games using Macromedia Flash MX – a web-based multimedia software program).

Under adverse conditions, pair programming may also have negative effects. For instance, according to Thomas, Ratcliffe & Robertson, students who report higher levels of self-confidence with programming do not like pair programming as much as other students [10]. In addition, based on our previous study [11], the difficulty level of a task in the pair-programming setting plays an essential role among students at the secondary school level. It may support or hinder improvement in developing a game in pair-programming settings.

In this contribution, we focus on pair-programming for students at the secondary school level (age: 14–15) in computer science classes (CS). In all eight groups, there was approximately the same number of students, the same tasks done, and the same tutorials provided. However, they were using different social settings for programming. In response to the worldwide pandemic in 2020 and 2021, instruction and research proceeded either in virtual or hybrid-learning mode.

To expand the findings from our action research studies, we conducted structured interviews with researchers from three continents to figure out their experiences applying pair programming in their CS classes. This mixed-methods framework allowed us to include the experiences, perspectives, and learning from all stakeholders, especially teachers, students, and researchers, thereby increasing the intersubjectivity and reliability of our research. The findings might interest teachers, instructors/educators, educational researchers, staff responsible for (further) education of computer science teachers, and potentially educational policymakers. We hope to inspire some of these audiences to embark on the challenge of learning to program by using pair-programming settings more effectively.

Ⅱ. Research Approach

Our interest in social influences when solving problems and learning to program has biographical roots. The first author, an advanced Ph.D. student from Western Asia, had held numerous academic courses in which pair-programming proved to be "the right thing to do." His European supervisor comes from the humanistic, person-centered learning tradition in which problem-solving, experiential learning, and a constructive socio-environmental climate [12], [13] are assumed and theorized to improve learning [14], [15]. We included our student assistant, a pre-service teacher, in our research because of her deep interest in the subject matter.

From the previously mentioned aspects, it follows that the theoretical framework for our mixed methods research builds upon theories on collaborative learning [4], [16], [17] and significant, experiential learning [18]. Moreover, students' motivation is reflected in terms of Ryan and Deci's SDT (Self Determination Theory) [19].

Our particular research interest in learning to program in a pair-programming setting can be summarized by the following research questions that guided the study design described in the subsequent parts of this section.

A. Research questions

1. What makes students satisfied, and what makes students frustrated about the pair-programming setting?

2. What kind of influences does pair-programming have on students' learning of programming?

3. Under what conditions does the pair-programming setting lead to specific kinds of improvement (such as better programming results, more satisfaction, higher motivation, better problem-solving) in contrast to the solo programming setting?

4. What kind of influences does pair-programming have on students' programming skills?

B. *Research Design and Instruments*

The mixed-methods research design underlying this paper consists of two parts corresponding to two phases of our research. First, we followed three cycles of participatory action research (PAR). With this, the focus was on students' perceptions of various aspects of pair-programming, complemented by the researcher's (first author's) observations, surveys, and evaluations of learning outcomes in class. Second, to complement the students' and researchers' views with a broader, international perspective, the first author conducted a series of interviews with teachers who practiced and researched pair programming. The following two subsections summarize how we applied PAR and structured our interviews with experts.

C. *Participatory Action Research as Framework*

According to Susman and Evered [20], cited by Baskerville [21], the Action Research method guides researchers in investigating social systems in the form of an inquiry using an interventionalist's viewpoint. "Researchers both observe and participate in the phenomena under study." [21, p.5] The action research, in which practitioners are involved as both subjects and co-researchers, is referred to as Participatory Action Research (PAR) [22] [23].

PAR was applied successfully in educational contexts, where teachers act as reflective practitioners in their courses [22]. They cyclically improve their educational offer, typically by working through five phases [22],[24]: Diagnosing the situation, planning the action, taking it, evaluating any effects, and specifying their learning, before moving into the next cycle.

In the first cycle, we were interested whether pair-programming was applicable in the context of learning to program in secondary level (K9) classrooms. The target audience consisted of 28 students from three CS classes in 2019 and 2020. Our focus was on exploring the setting, the procedure, the advantages, and the limitations of pair-programming computer games among 14-15 years old students in computer science classes. In the first cycle, we conducted surveys in two ways: in a questionnaire form and with a structured focus group.

In the second PAR cycle, we focused on students' learning and motivation. Besides observing students and employing a brief questionnaire, we measured students' learning using the software metric "Lines Of Code" (LOC). In this cycle, we studied 41 students from four different classes during the COVID-19 pandemic in 2020 and 2021.

The third PAR cycle encompassed five CS classes and 55 students at the secondary level. They were instructed and researched, focusing on students' learning, problem-solving, coding, and computational thinking. Students were instructed on how to interact with their peers in a pair-programming setting while collaboratively coding a simple computer game. Amongst others, their accomplishment in terms of correct lines of code was measured.

D. *Expert Interviews and Qualitative Content Analysis (QCA)*

After spending several years researching pair-programming, we decided to conduct interviews with experienced international researchers who practiced pair-programming in classes at secondary levels of education to supplement our findings.

Participants. The first step was to identify experts who practiced pair-programming with secondary-level students. Therefore, drawing from the literature on pair-programming, we decided to contact authors who had published works on pair-programming focusing on students aged 14-15.

Addressing respondees. The first author emailed an invitation to eleven authors of international topic-related publications, asking them for an online interview. Eight out of eleven authors accepted the invitation. We had to exclude one researcher as he only investigated university-level students. Thus, we considered only seven out of eleven authors in our study. The participants received the interview guide beforehand to allow them to read the questions and, optionally, think about their responses in advance.

Demographics. Out of the seven experts, four were from Austria, two were from South Africa, and one was from the USA. Only those demographic categories relevant to the research questions were in the questionnaire. Keeping the number of demographic categories minimal helped to ensure participants' anonymity.

Research instruments. To keep the interviews focused, two authors of this paper designed an interview guide based on our research questions. The interview guide[1] consists of three main parts reflecting three clusters of aspects concerning the application of pair-programming for learning to program.

Data collection. The first author of this paper conducted the interviews. At first, he asked the interviewees for their consent to publish their statements in an anonymized form. Then he explained

[1] https://drive.google.com/file/d/12ifdSYRaTC36P0G_H4YuWhE-t2FDdtYL/view?usp=sharing

the goals and purpose of the research, asked questions based on the interview guide, and took notes. It is worth mentioning that we conducted all interviews via Zoom[2]. Some interviewees agreed to record the session, which helped to add some details to the notes afterward.

Qualitative content analysis. For data analysis of the interviews, we chose the qualitative content analysis based on Mayring [25]. This method allowed us to use the interview notes as a corpus of the text to be analyzed. The corpus amounted to 2182 words. The unit of analysis was defined to be any meaningful statement. Mixing inductive and deductive category building was consistent with Mayring's approach. [25] This proved supportive since we had already identified some categories when formulating the research questions, yet we also wanted to remain open to developing (sub-)categories based on respondents' answers. The flexibility of Mayring's method also allowed us to use time-saving and effective shortcuts. Additionally, we considered the frequency of statements in each category since it helps to reflect how many teachers had addressed a particular category. All categories were summarized and maintained in an Excel file.

All three authors engaged in suggesting categories. To increase the reliability of the qualitative content analysis, the two main authors of this paper asked a pre-service teacher to participate in category formation and association of statements to categories. Finally, two raters agreed on a category system, the specification of the 27 categories, and typical examples of statements falling into the respective category. Then they classified the individual statements independently and met in a Zoom conference to discuss their categorization and to find agreement on a few statements, they had classified differently. Since one of the anonymous reviewers suggested grouping the experts' statements into larger themes, the first two authors identified five themes that became the new (higher-order) categories and were associated with the previously identified fine-grained categories taking the role of subcategories (see Table II).

Ⅲ. FINDINGS

In the following, we describe the results of PAR and the content analysis in terms of categories, subcategories, and frequency of occurrence that fit into a (sub)category. Doing this shall help the reader to gain insight into the interviewees' experiences. In addition, we interpret the results and reflect on the insights gained from the interviews.

[2] Zoom is a web conferencing platform from Zoom Video Communications, Inc

A. Summary of Findings from Participatory Action Research

Findings from the first cycle, based on the surveys and the results of the focus group, show that students in 5th grade of a computer science class were satisfied and more motivated with the pair-programming setting than with the solo programming one [26]. Developing a mini-game in the pair-programming setting helped students understand the course structure. They found the course more interesting and more satisfying than working alone or starting to develop a more complex game. In addition, students presented collaboratively and were also very motivated by the opportunity of developing the mini-game in pair programming. Both students performing in driver and navigator roles had their duties and responsibilities in pair-programming, so they did not feel bored and did not complain. They even asked us to continue the pair-programming approach in future classes. Students considered the game appropriately challenging. They were satisfied with the mini-games level of difficulty in the pair-programming setting [26].

In the second cycle, technical outcomes of students' programming were examined by using the Lines of Code (LOC) metric. In this paradigm, students were asked to answer several C# programming tasks in both pair and solo programming settings. Overall, by counting the number of students' correct answers in each setting, we could evaluate students' learning to program. Findings from this cycle indicate that the pair-programming setting (CS1and CS2) was superior to solo-programming (CS3 and CS4) regarding students' learning of programming and problem-solving. CS1 included nine students. Thus, we could make three pairs plus a group of three students as an exception. In this way, we collected 16 answers from students: ((3 pair * 4 questions) + (1 group * 4 questions)). CS2 included twelve students (six pairs). Thus, we collected 24 answers from this class (6 pairs * 4 questions). After counting the number of the correct answers (LOC), 19 correct answers were identified. To sum up, 35 correct answers out of 40 answers have been collected from pair-programming classes. Regarding solo-programming settings (CS3 and CS4), 45 correct answers out of 72 answers have been collected from the two solo-programming classes. In addition, in the solo-programming classes, a lack of collaboration and problem-solving was observed [8].

Finally, in the third cycle, we focused on students' problem-solving skills besides measuring their coding performance. To find out if pair-programming leads to an improvement in problem-solving skills among students, a survey was conducted at the end of the programming task. We asked students whether they could resolve at least one upcoming issue with their peers in the pair-programming setting or not. Based on the students' responses, this setting leads to improved problem-solving skills for all participants in the third cycle, as all of them responded to this question affirmatively. Besides that, teachers and authors observed that students searched for a solution when they got stuck with their assignments. Some googled, whereas others consulted their peers or other classmates. Furthermore, students avoided asking their teacher or the researcher too many questions regarding their programming issues.

B. *Findings from the Analysis of Expert Interviews with Teachers and Researchers*

Based on the teachers' responses and our four research questions, we describe the results of the content analysis regarding categories, subcategories, representative statements, and the number of statement occurrences to facilitate the reader's insight into the teachers' findings and experiences.

Demographic data. We asked the respondents three questions to gather basic information about their experiences. The first was: "How long have you been using the pair-programming approach in secondary schools?" Three teachers have been using this approach for more than four years, one teacher began using this approach two years ago, and the remaining three teachers have been working with the pair-programming setting for three to four years.

The second question was related to the tools used in schools. Although, Unity™ and Delphi are displayed as the most used software in Fig. 2, it's worth mentioning that the teachers who mentioned Delphi used it several years ago, while those who mentioned Unity™ used this software in the last few years. Finally, each teacher used more than one tool during the pair-programming lessons.

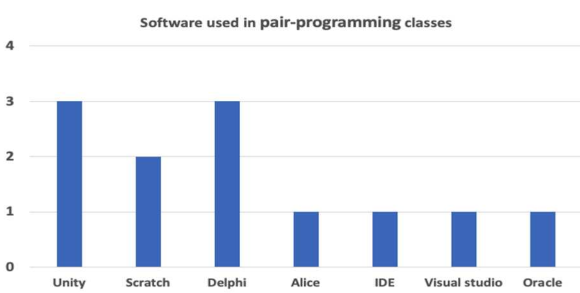

Fig. 1. Software used in classes.

Regarding the programming language used, 4 teachers used C#, and 3 used Java. C++, C, Pythons, PHP. Pascal was used by one teacher, respectively.

Qualitative content analysis. Table. II displays the (sub-)categories, the frequency of occurrence ('#'), and characteristic sample statements.

TABLE Ⅱ. Categories and subcategories with sample statements

Category/ - Subcategory	**Sample statement**	#
Social, emotional, and motivational aspects of PP		**39**
- Increased motivation	Students were much more motivated when doing PP	7
- Higher satisfaction	Being able to accomplish more in PP raises the students' satisfaction. Sharing their success increases their satisfaction	7
- Collaboration and teamwork	When one student gets stuck, the other could assist him/her. Students preferred to work together instead of on their own	7
- Increased engagement	They were enthusiastic and engaged in working together. [the engagement] is 100% better than individual work because of the interaction.	6
- Differences between PP and SP related to social aspects	It [PP] gives students a better picture of programming and encourages problem-solving together in a collaborative setting. In PP classes in the end there was no isolation and everyone felt comfortable asking other people for help.	6
- Motivation is based on factors related to the team'	If they didn't like their partner, they didn't enjoy it When they should collaborate with others, their motivation changed.	3
- Social Aspects	Students get to know each other better, which causes them to find friends.	3
Conditions and features impacting the outcome of PP		**34**
- PP-instructions	Describing the roles of driver and navigator; Explaining the rules in PP	9
- Introducing social aspects and rules for collaborative working	Explaining how to communicate and how to listen; Having students work together over MS Teams.	6
- Pairing is done by the students	They could choose their partner. It was done voluntarily.	5
- Pairing is done by the teacher	I [the teacher] decided based on the student's suggestions. I [the teacher] chose the partners myself so that some students wouldn't feel bad if no one picked them	5
- Context is dependent on the complexity and length of activities	I use PP in projects for sure. When students have to accomplish longer projects, that require working together creatively, I use PP.	3
- Context is dependent on age- and knowledge	With older groups; In the beginning, it is important to use SP.	3

- Student's decision	Some students decide to do it [programming] by themselves and some do it in PP. We allow them to choose.	2
- Practical reasons	I chose to do PP because of technical reasons, as there were fewer computers than students.	1
Outcome of PP		**33**
- Problem-solving skills acquired	They could solve problems with their friends. In PP the problem-solving skills increased.	6
- Observation of improved learning	It is difficult to measure but they have improved in learning. We observed that kids talked to other pairs about their problems.	6
- Reasons for improved coding-related skills in the PP setting	It is easier to work in pairs, and students feel more comfortable asking their peers. when they have a problem.	6
- Differences between PP and SP related to the programming output	PP is fostering a more structured programming approach. With PP you work faster and go through the program more often.	5
- Learning of skills related to team-/pair-work or other social factors	They learn from each other and discuss together. Teamworking skills can only be developed in the setting of teamwork so that aspect […] is definitely higher in PP.	4
- Learning improvement	Students understand the subject better. Students can learn better together.	4
- Evidence for learning improvement	They improved, as I measured their learning.	2
Experience with PP		**17**
- Always PP	I will never use solo programming in the future. I use it in all learning situations.	9
- Reasons for decrease in satisfaction	Sometimes their satisfaction drops because they could not reach the outcome by themselves.	2
- Other benefits of PP	PP may support students to become more creative. PP encourages girls too.	6
Strategies for assessing coding skill		**11**
- Assessment based on oral presentation/explanation/colloquium	After students finished their tasks, I measured their learning based on a presentation including a colloquium. Students presented the project.	5
- Assessment based on exams/tests'	I made a pre-and post-test using simple Sudoku.	2
- Other ways of assessment	Checking the completion of all steps in a project; Using a checklist to measure students' engagement.	4
		134

Social, emotional, and motivational aspects of pair-programming (39 mentions). This category subsumes any statements addressing 'Increased motivation' (7), 'Higher satisfaction' (7), 'Collaboration and teamwork' (7), 'Increased engagement'(6), 'Differences between PP and SP related to social aspects'(6), 'Motivation based on factors related to the team' (3), and 'Social Aspects' (3). With 39 statements, this category was the most frequently addressed one. This result corroborates Panitz and Panitz's principles of cooperative learning, which is exemplified by: "When one student gets stuck, the other could assist him/her." The result confirms the motivational forces of "relatedness" as an important dimension of Ryan and Deci's Self-determination Theory (Ryan and Deci 2000). For example, a teacher noted: "Students were much more motivated when doing PP." Moreover, teachers perceived that "Students preferred to work together instead of on their own."

Conditions and features impacting the outcome of PP (34). This category subsumes all statements describing any conditions, arrangements, regulations, and practices followed in the PP classes. It consists of the subcategories: 'PP-instructions' (9), 'Introducing social aspects and rules for collaborate working' (6), 'Pairing done by the students' (5), 'Pairing done by the teacher (5)', 'Context is dependent on the complexity and length of activities' (3), 'Context is dependent on age- and knowledge' (3), 'Student's decision' (2), and Practical reasons (1). With 34 statements, this category was the second most frequently addressed one. As the names of the subcategories suggest, this practically oriented category throws light on how the PP setting can be prepared and implemented and which practices should be chosen under given circumstances. Sample statements deal with the description of the roles of driver and navigator, the strategy to form student pairs, and the consideration of the complexity of the task to be solved when deciding between PP and SP to optimize the learning experience and outcome. We found broad concordance of the international teachers' insights and practices with our experience in the three cycles of PAR.

Outcome of PP (33). This category consisting of 33 statements, is considered the third most frequently addressed. It collects the mentions of any gain the interviewees attributed to the PP-setting. The various outcomes were associated with the following subcategories; 'Problem-solving skills acquired' (6), 'Observation of improved learning' (6), 'Reasons for improved coding-related skills in the PP setting' (6), 'Differences between PP and SP related to the programming output' (5), 'Learning of skills related to team-/pair-work or other social factors' (4), 'Learning improvement' (4), and 'Evidence for learning improvement' (2). Characteristic statements were: "They [students] learn from each other and discuss together." "They could solve problems with their friends." "Teamworking skills can only be developed in the setting of teamwork." The category 'outcome of PP' very directly matches the principles of experiential learning [18]. For example, by the experience of solving real problems, students acquire problem-solving skills. Moreover, the observation that students learn from each other and solve problems together corroborates collaborative learning. In particular, Panitz and Panitz's elements of 'Interpersonal and small group skills, and 'Face-to-face promotive interaction' are mirrored by the interviewees' statements.

Experience with PP (17). This category catches all mentions of any experience with PP that were not covered by any of the previous categories. The strongest subcategory 'Always PP' (9) includes 17 statements, such as "I use it in all learning situations.". Other subcategories are 'Reasons for decrease in satisfaction' (2) with a thoughtful sample statement: "Sometimes their [the students'] satisfaction drops because they could not reach the outcome by themselves." On the positive side, 6 statements were associated with the subcategory 'Other benefits of PP'. As an example, consider: "PP encourages girls, too."

Strategies for assessing coding skills (11). The last category includes statements regarding means to evaluate how well students could produce or understand code. The respective strategies are differentiated by the subcategories 'Assessment based on oral presentation/explanation/colloquium' (5), 'Assessment based on exams/tests' (2), and 'Other ways of assessment' (4). A sample statement of the largest subcategory – oral assessment – is: "Students presented the project." This example also shows the connection to experiential learning, assuming that students would learn/gain more from presenting their achievements and shortcomings than from taking an exam requiring them to recollect what they had learned.

Ⅳ. DISCUSSION

A. Limitation

There are several limitations to our study that emerged during the interviews. Firstly, the sample of experts to be interviewed was relatively small. Moreover, it consisted of seven university professors and instructors identified by the authors of this paper based on our literature review on pair programming. This implies a positive attitude towards pair-programming among the interviewees and could therefore be seen as reflecting an a priori bias toward emphasizing the benefits of pair-programming. Further interviews with teachers should be conducted. Regarding our Participatory Action Research, we see a limitation in the fact that all research had been conducted in few secondary schools in the city of Vienna/Austria. The fact that all schools were situated in a large city and the teachers originated from Western European culture might have also influenced the results. More research across cultures would be needed for generalizing the results.

B. Contribution

Despite the limitations mentioned above, the three cycles of Participatory Action Research and the findings from the interviews provide evidence-based responses to our four research questions.

RQ1) To address the first research question: "What makes students satisfied and what makes students frustrated about the pair-programming setting?" we first refer to the interview question in which we asked teachers about their observations of student satisfaction. Almost all the interviewees believed that the pair-programming approach increased students' satisfaction. One teacher said: "I observed that they were sharing their success in pair-programming, thus, it increased their satisfaction." Another interviewee stated: "In pair-programming, satisfaction increases when students accomplish more," and another one said: "Pair-programming increased students' satisfaction as well as their motivation." However, there were also some statements expressing that pair-programming could lead to frustration among students, which is exemplified by: "When not having it all accomplished by oneself, it decreases their satisfaction," or "Sometimes their satisfaction drops because they could not reach the outcome by themselves." With the vast majority of the 39 statements falling into the category 'Social, emotional, and motivational aspects of pair programming' being affirmative of pair work, this result confirms the importance of SDT's notion of relatedness as a source of motivation [19], Yet, if a relationship is perceived as an obstacle to autonomy, self-determination appears to be preferred.

Referring to the first cycle of PAR, [27] results from students' questionnaires and a focus group discussion showed a significant improvement in students' satisfaction during the pair-programming activity compared to individual work. All students agreed with their classmate who said: "We like pair-programming because we could work faster." Hearing this student's opinion can be seen as a confirmation of the observation that pair-programming positively influences speed when developing the game. One student described the mini-game with the adjective "cool", which was very appealing. Last but not least, according to the survey among students, the pair programming approach could lead to frustration if the tasks are too simple. If so, less engagement was observed [17]. These results from the first cycle of PAR are absolutely in tune with the findings of the expert interviews described above.

RQ2) Regarding the second research question: “What kind of influences does pair-programming have on students’ learning of programming?” we refer to the category ‘Outcome of PP’ holding 30 statements. One instructor said: ”Students can learn better together.“ Another one said: ”It [pair-programming] may support students to be more creative.“

Additionally, referring to the second cycle of PAR [8], pair-programming helped students understand programming more thoroughly as they consulted each other and engaged in a collaborative setting. Students felt more comfortable asking their peers than a lecturer or teacher, pointing to promotive interaction and teamwork skills as targets of collaborative learning. In a coding exercise, 35 answers out of 40 answers were correct in pair-programming classes, which amounts to 87.5% of correct answers. In contrast, only 45 correct answers out of 72 answers (amounting to 62.5%) were collected from solo-programming classes. This showed that students’ learning improved by pair-programming.

RQ3) In the third research question we asked: “Under what conditions does the pair-programming setting lead to specific kinds of improvement such as better programming results, more satisfaction, more motivation, and better problem solving, compared to the solo-programming setting?” Referring to the category "Conditions and characteristics that affect the outcome of PP", it is vital that students understand the concept of pair-programming and take on their roles as driver and navigator. Therefore, a complete description of the roles in the PP setting is required. In other words, a teacher must be well prepared in advance when employing PP in class. In the first PAR cycle, we observed that the difficulty level of a pair-programming task affects students' learning, motivation, and satisfaction. In this context, students with too simple tasks and projects did not tend to collaborate with their peers because they preferred to solve them alone. This corroborates with STD's motivation coming from autonomy and the need to increase one's competence. Interestingly, with simple tasks, the need for autonomy seems to dominate the need for relatedness.

RQ4) In the fourth research question we asked: “What kind of influences does pair-programming have on students’ programming skills?” We found ample evidence that students learned programming in the pair-programming approach better. This statement is based on two facts:

- the students’ programming outcome was measured by the first author of this paper in the second [8] and third cycles of PAR using the LOC metric.
- Considering the category ‘*Strategies of assessing coding skills’*, interviewees measured students’ programming skills using various assessment strategies such as exams, pre/post-tests, or a presentation with the colloquium. One teacher said: “They improved,” as he measured their learning. Another said, “Students understand the subject better.” None of the interviewed teachers used the Lines of Code measure, still, making it a unique contribution of the authors’ approach toward measuring the outcome.

In a nutshell, the mixed methods provide ample evidence that – given the preconditions identified in this paper are met - pair-programming is an effective learning method in computer science and a valuable target for further research.

V. Conclusions and Further work

This paper investigated the pair-programming approach targeting students at the secondary school level. A mixed research approach was used to find responses to the research questions. The research goal of this paper was to find out if the pair-programming setting influences various factors such as motivation, satisfaction, learning to program, and problem-solving. This was examined through three cycles of Participatory Action Research that were reviewed in this paper and complemented by subsequent expert interviews with international teachers who practiced pair-programming with their students. The expert interviews were analyzed by qualitative content analysis. In a nutshell, the two research approaches brought concordant as well as complementary results: We found that secondary school students (K9) achieved better programming results, better codes, and higher motivation in pair-programming than in traditional solo-programming classes. Furthermore, our data gathered in the third PAR cycle showed that the pair-programming setting had not only a positive impact on students' learning to program but also on their satisfaction, motivation as well as problem-solving. We conclude that learning to program via pair-programming constitutes a path to promoting students’ 21’s century skills. If instructed properly, it may help to increase the number of students for whom learning to program is connoted with a positive social and learning experience.

Acknowledgment

Thanks to the professors and teachers for their contribution to the interviews and their support. Furthermore, the authors thank the University of Vienna for supporting the research and its presentation at the Frontiers in Education Conference 2022 (FIE).

REFERENCES

[1] P. Sancho-Thomas, R.Fuentes-Fernández, & B. Fernández-Manjón, Learning teamwork skills in university programming courses. Computers & Education, 53(2), 517- 531, 2009.

[2] D. W. Johnson, Cooperative Learning: Increasing College Faculty Instructional Productivity. ASHE-ERIC Higher Education Report No. 4, 1991. ASHE-ERIC Higher Education Reports, George Washington University, One Dupont Circle, Suite 630, Washington, DC 20036-1183, 1991.

[3] K. A. YERION, & J. A. RINEHART, Guidelines for collaborative learning in computer science. ACM SIGCSE bulletin. 27(4):29-34, December 1995.

[4] T. Panitz, & P.Panitz, Encouraging the use of collaborative learning in higher education. University teaching: international perspectives, 161-201, 1998.

[5] K. Inkpen, K. S. Booth, M. Klawe & R. Upitis, Playing together beats playing apart, especially for girls. In The First International Conference on Computer Support for Collaborative Learning (pp. 177–181), 1995.

[6] K. Z. Watkins, & M. J. Watkins, Towards minimizing pair incompatibilities to help retain under-represented groups in beginning programming courses using pair programming. Journal of Computing Sciences in Colleges, 25(2), 221-227, 2009.

[7] C. McDowell, L.Werner, H. Bullock, & J. Fernald, Pair-programming improves student retention, confidence, and program quality. Communications of the ACM, 49(8), 90–95, 2006.

[8] A. Issaee, R. Motschnig and O. Comber, "Pair- versus solo-programming of mini-games as a setting for learning to program: An Action Research approach," in *2021 IEEE Frontiers in Education Conference (FIE)*, Lincoln, NE, USA, 2021 pp. 1-9. doi: 10.1109/FIE49875.2021.9637178

[9] L.L. Werner, S. Campe, and J. Denner. Middle school girls + games programming = information technology fluency. Proceedings of the 6th conference on Information technology education (Newark, NJ, USA, 2005), 301-305, 2005.

[10] L. Thomas, M.Ratcliffe, & A. Robertson, Code warriors and code-a-phobes: A study in attitude and pair programming. SIGCSE. Reno, NV, 2003.

[11] A. Issaee, “Under what conditions does the pair-programming setting lead to kind of improvement in developing a game at secondary school?.” In E. Langran (Ed.), Proceedings of SITE Interactive Conference 2020 (pp. 486-498). Association for the Advancement of Computing in Education (AACE)., 2020.

[12] C. R. Rogers, "The interpersonal relationship: The core of guidance." Harvard educational review, 1962.

[13] R. Motschnig-Pitrik, and G.T. Barrett-Lennard, Co-actualization: A new construct for understanding well-functioning relationships, Journal of Humanistic Psychology, 50(3): 374-98. New York, USA: SAGE, 2010.

[14] C. R. Rogers, A Way of Being. Boston: Houghton Mifflin Co, 1980

[15] C. R. Rogers, Freedom to Learn for the 80’s. Columbus, OH: Merrill Publishing, 1983.

[16] B. Barron, Achieving coordination in collaborative problem-solving groups. Journal of the Learning Sciences, 9(4), 403–436, 2000.

[17] C. K. K. Chan, Collaborative knowledge building: Towards a knowledge creation perspective. In C.E. Hmelo-Silver, C. A. Chinn, C. K. K, 2013.

[18] R. Motschnig and J., H., Cornelius-White “Experiential/Significant Learning (C. Rogers),” in Encyclopedia of the Sciences of Learning. Seel, M. Norbert, Eds. (4300 p.; 100 illus; in 7 volumes). Print (Book): ISBN 978-1-4419-1427-9, eReference: ISBN 978-1-4419-1428-6, 2012.

[19] R. M. Ryan, & E. L, Deci, Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, *55*(1), 68, 2000.

[20] G. I. Susman and R. D. Evered, “An Assessment of the Scientific Merits of Action Research,” Administrative Science Quarterly, vol. 23, no. 4, p. 582, 1978.

[21] R. L. Baskerville, “Investigating Information Systems with Action Research,” in Communications of the Association for Information Systems, vol. 2, available online: http://cais.isworld.org/articles/2-19/, 1999.

[22] S. Ottosson, “Participation action research - A key to improved knowledge of management,” Technovation, vol. 23, pp. 87-94, 2003.

[23] R. Motschnig-Pitrik. Participatory Action Research on a Blended Learning Course on Project Management Soft Skills. Proceedings of 36th Frontiers in Education Conference, (FIE) San Diego, California, IEEE Press. 2006.

[24] R. Motschnig, D. Pfeiffer, A. Gawin, P. Gawin, M. Steiner, and L. Streli, “Enhancing Stanford Design Thinking for Kids with Digital Technologies A Participatory Action Research Approach to Challenge-Based Learning,” 2018 IEEE Frontiers in Education Conference (FIE), 2018.

[25] P. Mayring, “Qualitative content analysis: theoretical foundation, basic procedures and software solution,” Social Science Open Access Repository (SSOAR), 2014.

[26] A. Issaee, “Under what conditions does the pair-programming setting lead to kind of improvement in developing a game at secondary school?.” In E. Langran (Ed.), Proceedings of SITE Interactive Conference 2020 (pp. 486-498). Association for the Advancement of Computing in Education (AACE)., 2020.

[27] A. Issaee, R. Motschnig, O.Comber, "LEARNING TO CODE IN CLASS BY PAIR-PROGRAMMING OF GAMES.", DisCo: E-learning: Unlocking the Gate to Education around the Globe - 14th conference reader, Prague, 2019.