Pair Programming and the Level of Knowledge in the Formation of Pairs

Sónia Rolland Sobral(✉)

REMIT, Universidade Portucalense, Porto, Portugal
sonia@upt.pt

Abstract. The pandemic situation COVID-19 requires teachers to rethink some of the activities proposed to students since even though they are physically in the same room at the university, they cannot join and must maintain a social distance. The curricular unit of introduction to programming in the first year of computer courses requires several activities: students only learn by doing. One of the activities that has been used a lot in the academic community is pair programming: a single computer for two students working together. In the case that this article reports the pair works sharing a ZOOM room and alternates the position of driver (writes the code) with navigator (one that comments on the code). This article uses the technique of pair programming for writing top-down, algorithm and traces to verify the behavior of variables along the algorithm. If the pair programming technique has had very interesting results in terms of task speed and code quality far superior to programmers alone, there are some doubts about the problem of the constitution pf the groups. In this article we use two opposite strategies: group students with opposite knowledge or group students with similar knowledge. We found that students who prefer this activity are the worst ones (in both strategies) and the best students when grouped with colleagues with similar knowledge.

Keyword: Pair programming · COVID-19 · CS1 · Work group · Algorithms

1 Pair Work

Pair programming is a strategy that has been widely described: is a practice in which two programmers work collaboratively at one computer, on the same design, algorithm, or code [1–3]. It started being used in industry and became a pedagogical strategy widely used in curricular units as introduction to programming [4–6]. There are reports that reveal that this is an excellent way to improve teaching-learning [7–9] [10, 11]. In this technique there is a computer for a driver (writes code) and a navigator (one that comments on the code written by the colleague) [12]. Reports say that the code is written much faster and more effectively than if both programmers were working alone [13–15].

Using the technique for academic experiences, the way groups are formed can be done in different ways as it can be a very important factor [9]: randomly [16], chosen by students [17], by level of knowledge [7] or by other characteristics [2, 18, 19]. In the case of this article, groups made up of two students were grouped by prior assessment:

Á. Rocha et al. (Eds.): WorldCIST 2021, AISC 1367, pp. 212–221, 2021.
https://doi.org/10.1007/978-3-030-72660-7_21

in a first case with the best students working with the worst ranked, in the second case with groups to be made up of students with similar knowledge.

The experience reported in this article uses the structure of peer programming: physically the students are in the same room, but because they are in a pandemic situation COVID-19, they have to maintain social distance and cannot move elsewhere within classroom. Very similar to the Distributed pair programming form [20] with the two elements at a distance from each other. Each student is on their own computer and they communicate through the ZOOM application [21]: One of the elements is in the driver situation (the one who shares the screen and writes), while the navigator comments on what the colleague writes. The positions alternate during the activity. The activity in question is not a program in a programming language, but a document in a text processing with the top-down, the algorithm and the table with exploratory data.

In this document we detail the experience, list student responses, analyze responses and make a conclusion.

2 The Experience: Two Different Designs

This experiment was carried out as part of an initial programming unit in the first year of two computer courses at the fourth and fifth week of classes, during the pandemic situation of COVID-19.

Classes are divided into two hours of lectures on the ZOOM platform and two blocks of two hours, laboratory. At the beginning of the lectures, an assessment is made in MOODLE lasting 15 min and mostly consisting of true and false questions, multiple choice, or space filling. In the second block of laboratory practical classes it starts with an exercise (sometimes on paper and sometimes in digital form and submitted in MOODLE) with a duration of 40 min, the rest being for the resolution of the same exercise (in a more detailed version) in activity in pairs. Both the MOODLE assessment and the written exercise of the practical class are corrected and correspond to a score from 0 to 20 and are published on the page of the course unit in MOODLE. The average of these assessments (excluding the worst grade of each student from the average) is placed in a list ordered by class and classification. Each student is also assigned a type: A (grades from 16), B (grades [12, 16 [), C ([8, 12 [), D ([4, 12 [) and E (more grades lower than 4). This course tends to have very high dropout rates.

This document reports and compares two different ways of constituting pairs: the first (N = 54) with inverse pairs (A with E, B with D and C together), the second (N = 52) with pairs of the same type. The number of students by type in each of the two experiences is shown by the next table (Table 1).

Each of the pairs was asked that the student with the best rating create a ZOOM room and send an email to their peer. The objective would be for the second to write a top-down, the first the algorithm and again the first to write a table with the execution data provided by the second. The goal is that the driver is the one who writes and the navigator the one who comments on what is being written.

In the case of the first experience, there were five female students, while in the second, only three. The number of students in the Computer Engineering course was 37 and 36, respectively, while the computer students were 17 and 16. In both cases there

Table 1. Number of students by type in each of the two experiences.

Type	Inverse N = 54	Equal N = 52
A	4	4
B	20	19
C	15	20
D	6	7
E	9	2

were 44 students aged 18 to 20 years. In the first experience there were 10 students between 21 and 26 years old, this number being 8 students in the case of activity with pairs of the same type. The features described are shown in the following table (Table 2).

Table 2. Gender, course, and ages of students in each of the activities.

Gender	Inverse N = 54	Equal N = 52
Feminine	5	3
Male	49	49
Course		
Computer Engineering	37	36
Computer Science	17	16
Ages		
[18, 20]	44	44
[21, 26]	10	8

3 Student Responses

The question "What did you think of the Pairs experience?" was answered from 1 (terrible) to 5 (excellent). The average of the two experiences was similar (3.91 and 4.27), however if we analyze the responses by type of student it appears that they are quite different. In the case of the activity with constitution of groups by inverse type, the average rises as the type of student worsens. In the case of activities with the same type of group, all the best students rated the experience as excellent, with the worst evaluations being for students with intermediate knowledge level. The most complete numbers are shown in the following table (Table 3).

To the question "How was the experience of creating the ZOOM room or entering the room created by the colleague?" in the first case the best average is again increasing in relation to the type of student, while in the second case either the best or the worst

Table 3. Averages, answers, question "What did you think of the Pairs experience?" by type.

Type	Inverse	Equal
A	3,50	5,00
B	3,60	4,58
C	3,73	3,65
D	4,33	4,71
E	4,78	4,50
Average	3,91	4,27

students rated the ZOOM room experience as excellent and again it was the students with intermediate knowledge (type C) who evaluated the experience worse (3.75). The numbers by type are shown in the following table (Table 4).

Table 4. Averages, answers, question "How was the experience of creating the ZOOM room or entering the room created by the colleague?" by type.

Type	Inverse	Equal
A	2,50	5,00
B	3,65	4,42
C	4,20	3,75
D	4,50	4,71
E	4,89	5,00
Average	4,02	4,27

Table 5. Question "How was the communication with the colleague in the ZOOM room" by students Type.

Type	Inverse	Equal
A	3,50	4,75
B	4,10	4,21
C	3,93	3,60
D	4,50	4,71
E	4,89	5,00
Average	4,19	4,12

To the question "How was the communication with the colleague in the ZOOM room?" in the first case the best average is again increasing in relation to the type of student, while in the second case the best assessment is of the worst students, followed by the best students (A). The numbers by type of student are shown in the next table (Table 5).

To the question "Do you feel you have more knowledge than your partner?" students classified as type A answered an average of 4.5 in the first experiment and 3 in the second. Interestingly, type E students responded, on average 3.11, to quantify whether the partner has more knowledge than the respondent himself - which is strange since E students have previous evaluation scores below 4 and the partners have grades evaluations from 16 values, as we can see in the table below (Table 6).

Table 6. Question "Do you feel you have more knowledge than your partner?" by students Type.

Type	Inverse	Equal
A	4,50	3,00
B	3,30	3,21
C	3,47	2,90
D	2,50	3,57
E	3,11	3,00
Average	3,31	3,12

To the question "Did the Partner help you with the exercises?" in the first case, type D and E students answered 5 (maximum), and type A and average students responded less favorably (3.75). In the second case, it was students of type A and D who evaluated at most, and again those of intermediate knowledge were those who rated the experience worst (3.85). However, both averages were clearly positive (4.57 and 4.44), as shown in the next table (Table 7).

Table 7. Question "Did the Partner help you with the exercises?" by students' type.

Type	Inverse	Equal
A	3,75	5,00
B	4,45	4,74
C	4,53	3,85
D	5,00	5,00
E	5,00	4,50
Average	4,57	4,44

To the question "Did you help your colleague with the exercises?" type A students responded excellent (5) and, interestingly, type E students also responded with excellent (5) in the experience with similar knowledge pairs. Once again, it is Type C students who worst assess the issue, as shown in the next table (Table 8).

Table 8. Question "Did you help your colleague with the exercises?" by student type.

Type	Inverse	Equal
A	5,00	5,00
B	4,55	4,63
C	4,27	3,75
D	4,50	4,57
E	4,11	5,00
Average	4,43	4,33

To the question "Did you feel that the activity was beneficial for your learning?", In the first case, the assessment goes up as the grade of previous assessments goes down. In the case of experience with pairs of similar knowledge, the answers vary between 5 of type D and 3.35 of type C. The numbers are presented below (Table 9).

Table 9. Question "Did you feel that the activity was beneficial for your learning?" by student type.

Type	Inverse	Equal
A	3,3	4,5
B	3,5	4,42
C	3,5	3,35
D	4,8	5
E	4,9	4
Average	3,9	4,08

To the question "Would you like to repeat the experience?" 83% answered yes in both cases, 11.1% answered no in the first case and 9.6% in the second. The remaining answers were in the sense of repeating the experience, but with another partner (5.6 and 7.7%). In the case of A Type students, 25% of students answered yes, but with another partner, while all the others answered yes. All students with the lowest level of knowledge considered the experience to be advantageous and to repeat. As we can see in the next table, the most critical students who apparently did not like the activity were type B (15% and 5%) and C (21% and 20%) students in the case of the first activity and the second (Table 10).

Table 10. Question “Would you like to repeat the experience?” by student type (%).

Type	Inverse			Equal		
	No	Yes	Yes, but	No	Yes	Yes, but
A		0,75	0,25		1	
B	0,15	0,8	0,05	0,05	0,84	0,11
C	0,20	0,80		0,20	0,70	0,10
D		0,83	0,17		1	
E		1			1	
Total	0,11	0,83	0,06	0,10	0,83	0,08

4 Discussion of Results

Both experiences were based on the previous constitution of the working group formed by two students. In the first experiment, the groups had two elements with a previous inverse assessment, while in the second case the previous assessment of the students was similar. The two experiences were developed by a similar number of students (54 and 52 respectively) and with similar characteristics (gender, age, course, and level of knowledge of the students). It was found that when there is inverse knowledge, students with the worst level of knowledge evaluate the experience close to excellent (5) and that this evaluation of students is inversely proportional to their knowledge, with the best students evaluating, on average, the experience with 3.5. In the case of constitution, the similar level, the best students were the ones who enjoyed the experience, as we can see in the following figure (Fig. 1).

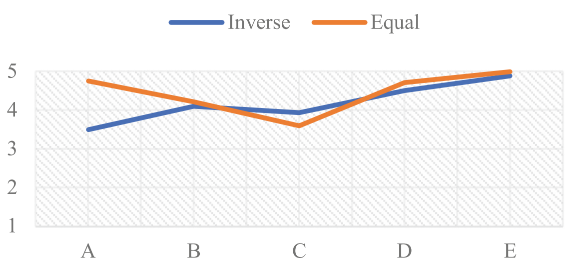

Fig. 1. Evaluation of the two experiences by type of student.

Likewise, peer-to-peer communication is much better for students with lower knowledge when they belong to a group with a better student, but it is also better for some with better knowledge who pair with students with similar knowledge and the worst students. Grouped with another of the worst students. As in the question analyzed in the previous figure, intermediate knowledge students who worst classify the communication experience, as can be seen in the following figure (Fig. 2).

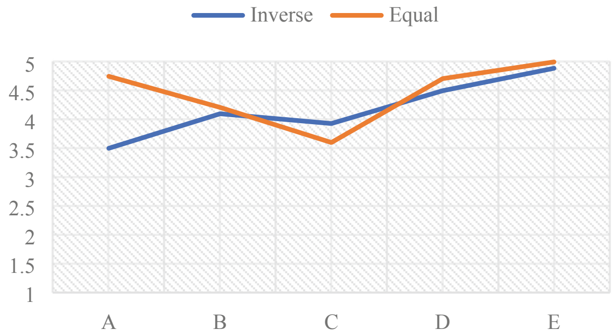

Fig. 2. Communication by students type and experience.

When students are asked how they evaluate the help they gave to their partner and if they feel they were helped by a colleague: the worst students and the best students in the second situation say that they helped a lot and were helped a lot, just like the worst students in the first experience. As we can see in the following figure (left Inverse, right Equal knowledge), once again it is the intermediate students who worst rate the item help and be helped (Fig. 3).

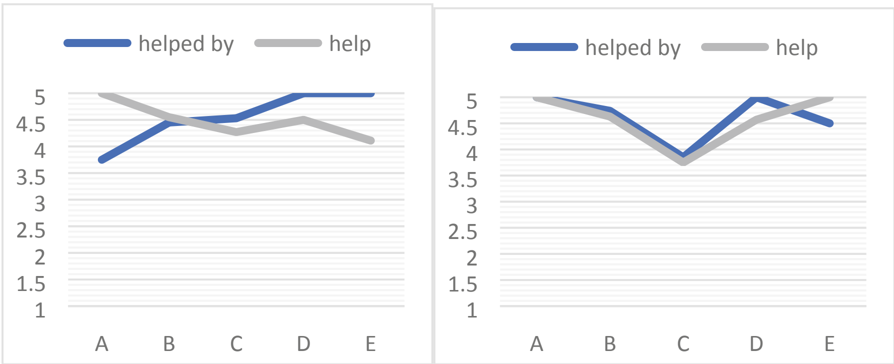

Fig. 3. Help and Help by partner (left Inverse, right Equal knowledge).

It is concluded that the worst students are also the ones that better classify the interest of the activity, either in a group situation or in the other. The same is true for better students who work with better students. And the same conclusion is visible for the students' interest to repeat experience with partners again: the worst are unanimous (100%) in wanting to do the activity again, while the best students (and in large part) want to do the activity with someone colleague of the same level of knowledge. Once again, it is the intermediate knowledge students who worst evaluate the experiences, with 20% not enjoying the experience.

5 Conclusions

The pandemic situation COVID-19 requires that several pedagogical strategies be rethought. If a large part of the time, students are physically apart from each other, even when they are in the same classroom, they cannot be physically close groups because there is a need to maintain physical distance. Pair work seems to be an excellent strategy for introductory programming classes, whether in the phase of using programming languages or in the phase of algorithm design and use of other tools. In this experience, the pairs used the ZOOM to communicate even if they were in the same room with the mandatory social distance.

There have been many experiences related to pair programming and it has been widely described. There is an attempt to make pairs compatible in some situations, in others the groups are randomly constituted. Many of the experiences use characteristics of the students, others their previous evaluation.

In the experiences described in this article, the classifications of the two weekly activities in the course were used: in the first case the students were grouped by inverse knowledge, and in the second case by similar knowledge. The most visible conclusion of this article is that students with the worst knowledge enjoy working in groups with students who have more knowledge than they do; and that both (students with more knowledge and students with less knowledge) appreciate pair work in groups made up of pairs at the same level as theirs. Intermediate level students are those who enjoy the experience less, and in both cases studied, their group was formed with another intermediate level colleague. In a next experiment we will try to divide students into four or six levels (or an even number of levels) so that intermediate students do not always stay with intermediate students.

The work strategy (or programming) by pairs seems to us to be an excellent alternative to activity in units of introduction to programming, and it tends to appear that the ideal is to form groups of two students with a similar level of knowledge.

References

1. Nagappan, N., Williams, L., Ferzli, M., Miller, C., Balik, S.: Improving the CS1 experience with pair programming. ACM SIGCSE Bull. **35**(1), 359–362 (2003)
2. Berenson, S., Slaten, K., Williams, L., Ho, C.-W.: Voices of women in a software engineering course: reflections on collaboration. J. Educ. Resour. Comput. **4**(1), 3-es (2004)
3. Choi, K., Deek, F., Im, I.: Exploring the underlying aspects of pair programming: the impact of personality. Inf. Softw. Technol. **50**(11), 1114–1126 (2008)
4. Sobral, S.R.: Is pair programing in higher education a good strategy? Int. J. Inf. Educ. Technol. **10**(12), 911–916 (2020)
5. Muller, M., Tichy, W.: Case study: extreme programming in a university environment. In: International Conference on Software Engineering (2001)
6. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston (1999)
7. Katira, N., Williams, L., Wiebe, E., Balik, S., Gehringer, E.: On understanding compatibility of student pair programmers. In: SIGCSE (2004)
8. Muller, M.: Two controlled experiments concerning the comparison of pair programming to peer review. J. Syst. Softw. **78**(2), 166–179 (2005)

9. Bevan, J., Werner, L., McDowell, C.: Guidelines for the use of pair programming in a freshman programming class. In: Software Engineering Education Conference (2002)
10. Williams, L., McDowell, C., Nagappan, N., Fernald, J., Werner, L.: Building pair programming knowledge through a family of experiments. In: International Symposium on Empirical Software Engineering (2003)
11. Nosek, J.: The case for collaborative programming. Commun. ACM **41**(3), 105–108 (1998)
12. Nawrocki, J., Wojciechowski, A.: Experimental evaluation of pair programming. In: European Software Control and Metrics (2001)
13. Williams, L.: Integrating pair programming into a software development process. In: Conference on Software Engineering Education and Training. In Search of a Software Engineering Profession, Charlotte (2001)
14. Hulkko, H., Abrahamsson, P.: A multiple case study on the impact of pair programming on product quality. In: International Conference on Software Engineering (2005)
15. Williams, L., Kessler, R., Cunningham, W., Jeffries, R.: Strengthening the case for pair programming. IEEE Softw. **17**, 19–25 (2000)
16. Carver, J., Henderson, L., He, L., Hodges, J., Reese, D.: Increased retention of early computer science and software engineering students using pair programming. In: Software Engineering Education Conference (2007)
17. Thomas, L., Ratcliffe, M., Robertson, A.: Code warriors and code-a-phobes: a study in attitude and pair programming. In: SIGCSE (2003)
18. Williams, L., Layman, L., Osborne, J., Katira, N.: Examining the compatibility of student pair programmers. In: AGILE Conference (2006)
19. Werner, L., McDowell, C., Hanks, B.: Pair-programming helps female computer science students. ACM J. Educ. Resour. Comput. **11**(4), 4 (2004)
20. Baheti, P., Gehringer, E., Stotts, D.: Exploring the efficacy of distributed pair programming. Lecture Notes in Computer Science, pp. 208–220 (2002)
21. Zoom Video Communications, Inc., "ZOOM" (2020). https://zoom.us/