Exploring the effects of feedback on the problem-solving process of novice programmers

Hemilis Joyse Barbosa Rocha[1]

[1]Instituto Federal de Alagoas
Campus Viçosa

hemilis.rocha@ifal.edu.br

Abstract. *The research investigates how specific affective states, such as frustration, boredom, and anxiety, influence help-seeking behaviors in problem-solving programming activities. Carried out with 73 beginner programming students divided into two CS1 classes, the study uses an interactive learning environment to collect and analyze data from student interactions. The results reveal that negative affective states are significantly associated with the search for clues that offer ready-made answers to problems. Furthermore, there is a correlation between boredom and anxiety reported by students. It is concluded that negative affective states can motivate students to prefer quick and less challenging solutions, emphasizing the importance of considering the affective dimension in the design of interactive learning environments.*

Resumo. *A pesquisa investiga como estados afetivos específicos, como frustração, tédio e ansiedade, influenciam os comportamentos de busca de ajuda em atividades de programação de resolução de problemas. Realizado com 73 alunos iniciantes em programação divididos em duas turmas CS1, o estudo utiliza um ambiente de aprendizagem interativo para coletar e analisar dados das interações dos alunos. Os resultados revelam que estados afetivos negativos estão significativamente associados à busca por pistas que oferecem respostas prontas para os problemas. Além disso, há uma correlação entre tédio e ansiedade relatados pelos alunos. Conclui-se que estados afetivos negativos podem motivar os alunos a preferir soluções rápidas e menos desafiadoras, enfatizando a importância de considerar a dimensão afetiva no design de ambientes de aprendizagem interativos.*

1. Introduction

In programming learning contexts, where students often face challenges in solving problems, personalized feedback plays a crucial role in improving the learning process [Langer 2011, Raabe et al. 2015, Cavalcanti et al. 2020]. Despite its recognized importance, effective feedback design in problem-solving environments remains a significant challenge in learning environment research[Alves and Jaques 2014]. This challenge requires careful consideration of several parameters described in the literature, adapting to the specific characteristics of the target audience and the domain in question.

At the same time, environments, methodologies, and systems for authoring teaching materials play a crucial role in digital education. The authoring of materials on virtual platforms is not limited to the simple transmission of knowledge but also seeks to engage

DOI: 10.5753/sbie.2024.242737

students and facilitate effective learning. The use of well-structured pedagogical methodologies and interactive systems can significantly improve the quality and effectiveness of these materials. Tools such as virtual learning environments (VLE) and interactive authoring platforms allow educators to create, organize and distribute educational content efficiently, promoting dynamic and personalized learning.

This article presents an exploratory study carried out in a programming course for beginners, investigating the effects of different moments of feedback on problem-solving activities [Caspersen and Bennedsen 2007]. Using an established conceptual framework [Narciss 2008], the study analyzed behavioral and performance data to explore how these elements influence variables such as student effort, persistence, and engagement. A significant finding was the substantial impact of feedback timing on student engagement. Immediate feedback, both reactive and proactive, has been associated with higher levels of engagement, effort, and persistence in solving programming problems. Conversely, situations with no or delayed feedback resulted in greater reliance on external assistance such as peers and online resources. Despite these behavioral variations, no significant differences were observed in students' academic performance between the different feedback moments, suggesting a direct moderate impact on the final learning outcome.

2. Research Context and Conceptual Framework

The first experience in learning programming for many computer students is often frustrating [Sheard et al. 2009]. Some authors, including [Medeiros et al. 2018] point out the volume of works that point to the difficulties of introductory programming as a subject considered difficult to learn and teach.

Some studies on computer education reveal that learning computer programming is a difficult task [Koulouri et al. 2014] [Kunkle and Allen 2016], requiring students to have good problem-solving skills. However, in context, problem-solving is not only creating a solution to a problem [Houghton 2004], but also expressing it in an algorithm using a programming language. This means not only figuring out the answer to a problem but also describing (applying the constructs) step-by-step what a machine would be doing to get the answer. Problem-solving skills are more important than learning various programming languages because technologies for programming computers are constantly changing. [Koulouri et al. 2014] However, learning the problem-solving skills needed for programming has proved to be difficult, mainly because it often requires learning of concepts and abstract thinking, regularly associated with skills to solve mathematical problems.

During this process, at a time, the student will demand some type of assistance, such as feedback. For Cano [Boud and Molloy 2013] feedback is the process by which students obtain information about their work, assessing the similarities and differences between the standards appropriate for any work, and the qualities of the work itself, in order to generate improved work. According to Hattie and Timperley [Hattie and Timperley 2007], on feedback can happen at three levels: task level, process level and self-regulation level.

Shute [Shute 2008] introduces the concept of formative feedback and defines it as information communicated to the student to modify their thinking or behavior in order to improve learning. In the work of Narciss [Narciss 2008] we find the definition of the term tutoring formative feedback (FTI) as feedback strategies that provide feedback components designed to guide the student towards successful task completion. The purpose of this feedback is to help students detect errors, overcome obstacles, and apply more efficient strategies to solve tasks. In addition, the ITF components are presented to the student without immediate knowledge of the correct answer to the task. Narciss's categories largely overlap with the categories used to describe the actions of human tutors when helping students learn to program.

3. Method

The exploratory study was conducted during a summer programming course, involving students from a public technical school in Brazil. It is worth noting that this study was submitted and approved by the Brazilian ethics committee. The course lasted eight weeks and was designed to introduce students to basic programming concepts. Both the course and the study were carried out entirely remotely, using the Discordhttps://discord.com/ environment. Discord was chosen for its real-time communication features, including text, voice, and video chat, as well as allowing the creation of dedicated channels for different course topics and activities. During the course, students participated in synchronous and asynchronous sessions, completed programming exercises, and received real-time feedback from instructors and peers. The platform also facilitated the collection of data on student interaction and their performance in programming activities.

3.1. Participants

We use Google's Form tool to collect registrations and some information about students. Although we counted the enrollment of 34 students, we noticed that 14 students were effective participants in the course and study. Therefore, the sample number for this study is equal to 14. The 14 participants were in the computer technical course and declared themselves to be beginners in programming. There were 4 female and 10 male students. Their mean age was 17 (SD = 0.76). All of them live with their families, with 57% of students not living in the same city as the school and 21% living in rural areas. Although everyone has access to the internet, only 32% have a computer, 14% have a tablet and 84% have a smartphone.

In addition, of the 34 students who applied, 20 were unable to complete the course and study. Of these, 11 students did not participate in at least one stage of the study, experiencing instability with the internet connection and 9 were completely absent. Therefore, to understand the reason for the complete absence of the 9 students, we contacted them and identified the following reasons: 60% claimed problems with the internet, 30% needed to dedicate time to work and 10% did not respond or claimed unavailability. Given this, below we will discuss the data of the 14 effective participants.

3.2. Task and Material

Some authors state that most of the difficulties experienced by beginners in programming are not in understanding the basic concepts of programming logic, but in combining and properly using these concepts in building a program [Lahtinen et al. 2005]. In other words, the difficulty is in "putting the pieces of the program together"[Spohrer and Soloway 1989]. Combining concepts and using them to build a solution is an essential task in solving programming problems. Furthermore, one of the

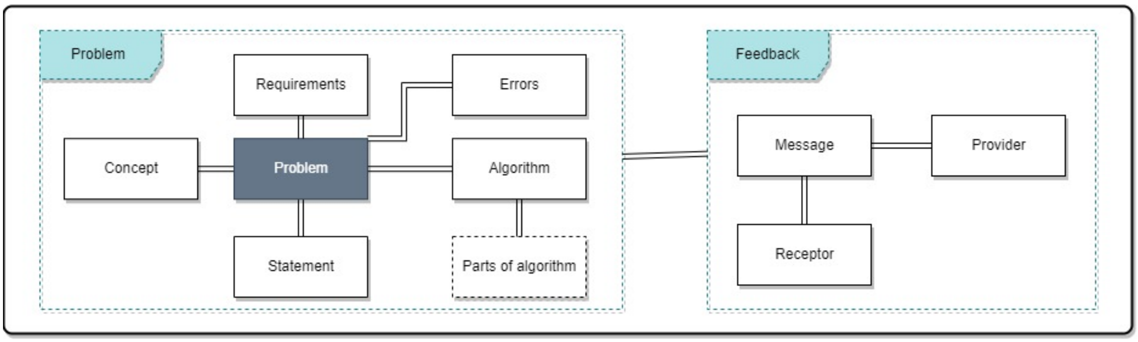

Figure 1. Problem-solving elements

important objectives of learning computer programming is to develop the ability to create a program that solves a specific problem.

For the framework testing experiment with novice programmers, we consider problem-solving according to two basic components: problem and feedback. In the experiment presented in this article, the problem-solving structure was organized with these two essential components, ensuring a clear focus on both problem definition and the quality of feedback provided.

3.2.1. Problem

Narciss [Narciss 2008] defines four steps for selecting and specifying tutoring informational feedback content. The steps are: (1) selecting and specifying instructional objectives; (2) selecting typical learning tasks and matching them with the necessary learning outcomes; (3) analyzing the requirements for each type of task; and (4) describing typical errors and incorrect steps. Although these steps are identified as prerequisites for selecting and specifying useful feedback information, we consider them more related to the task elaboration. Therefore, they were integrated into the problem development, the first element of the resolution structure specified in Figure 1 and used in this experiment.

According to Figure 1, a problem is structured with four components:

1. Statement: In the process of elaborating the statements, we consider the concepts and desired learning outcomes according to step 2;
2. Requirements: We identified the following items associated with the problem: (a) programming concepts; (b) cognitive operations related to programming concepts and commands implemented in a feedback message, for example, remembering or replacing a command; and (c) cognitive and metacognitive skills of the concepts associated with the problems. This definition aligns with what Narciss specified in step 1;
3. Solution: This is the representation of the solution to the problem in a program using a programming language; and
4. Errors: These are the possible errors made in the problem-solving process to find the solution.

The communication between the components described above happens as follows: A problem has only one statement and vice versa; associated with a problem are several requirements, and a requirement can be associated with more than one problem; a requirement can include several concepts, which in turn can be presented by several programming commands and vice versa; and a set of errors can be associated with a solution that can be made up of several parts.

This framework ensures that each problem is thoroughly developed with clear objectives, necessary requirements, potential solutions, and common errors, providing a comprehensive basis for effective feedback.

3.2.2. Feedback

The second element of the problem-solving framework shown in Figure 1 is feedback. The four components of feedback included in this structure are message, receiver, provider, and timing.

1. Provider: The agent who perceives the need for feedback and decides the timing, content, and form of feedback presentation. This agent can be a human (teacher, tutor, or student) or a software agent, such as an intelligent tutoring system.
2. Receiver: The agent that generates or provokes the need for feedback. Like the provider, the receiver can also be a human (teacher, tutor, or student) or a software agent.

For this study, the provider and receiver elements were instantiated by the human agents, the teacher and the student, respectively. The message and feedback elements were developed based on the framework by Narciss [Narciss 2008].

1. Message: The content of the feedback provided, is tailored to address specific needs and issues identified during the problem-solving process.
2. Timing: The appropriate moments for delivering feedback to maximize its effectiveness and support the learning process.

This comprehensive feedback framework ensures that feedback is timely, relevant, and personalized, enhancing the learning experience and supporting effective problem-solving.

The message constitutes the feedback content that the provider delivers to the receiver [Narciss 2008]. The timing of feedback is crucially linked to the problem-solving process. Feedback can be provided immediately after a need is identified or delayed to a later point. Narciss's framework encompasses these two feedback delivery moments, recognizing their significance in effectively addressing the learners' needs during the problem-solving process.

For the exploratory study, we used the delayed feedback moment and further subdivided the immediate feedback moment into two categories: immediate reactive and immediate proactive. In immediate reactive feedback, the need for feedback is initiated by the receiver and requested from the provider, meaning the provider plays a reactive role. In immediate proactive feedback, while the need for feedback might be generated by the receiver, it is ultimately up to the provider to perceive this need and decide whether or not to deliver feedback. Consequently, we identified three distinct feedback moments for the study: delayed feedback (AO), immediate reactive feedback (IR), and immediate proactive feedback (IP).

3.3. Procedure

The exploratory study was conducted within a summer programming course. The course lasted 15 days, totaling a workload of 60 hours. For the course, we developed 30 programming problems, each with four resolution activities, amounting to a total of 120 problem-solving activities. Additionally, for each activity, we predefined nine types of feedback, producing a total of 1080 feedback instances. The experiment was structured into three stages: planning, execution, and assessment.

1. Planning Stage: In the planning stage, we meticulously designed the programming problems and the associated feedback types. The objective was to cover a broad spectrum of problem-solving scenarios and feedback responses to gather comprehensive data on the students' interactions and learning outcomes. This stage also involved setting up the necessary technical infrastructure, including the learning management system and feedback delivery mechanisms, to ensure a seamless execution phase.
2. Execution Stage: During the execution stage, the course was conducted over 15 days. Students engaged in problem-solving activities and received feedback according to the predefined types. The feedback was categorized into three distinct moments: delayed feedback (AO), immediate reactive feedback (IR), and immediate proactive feedback (IP). Immediate reactive feedback was provided when the need was initiated by the receiver, while immediate proactive feedback was at the discretion of the provider, based on their perception of the student's needs. This dual approach allowed us to observe the impact of different feedback timings on the students' learning processes.
3. Assessment Stage: In the analysis stage, we meticulously evaluated the collected data to assess the effectiveness of the feedback types and timings. This involved analyzing the student's performance in the problem-solving activities, their interactions with the feedback, and any changes in their affective states. The goal was to identify patterns and correlations that could inform future instructional designs and feedback strategies in programming education.

Overall, the study aimed to explore the nuanced effects of different feedback moments and types on students' learning experiences in a programming context.

3.3.1. Planning

In the planning stage of the study, we developed the programming problems, cataloged the types of feedback, configured the problem-solving marathons, and formulated the research questions. The problem-solving marathons were the sessions in which the problems were presented to the students for resolution. Each marathon consisted of five problems, with different feedback moments associated with each one. We used Google Forms to set up each marathon, creating a form for each and recording the statements of the five problems. Each form included spaces for students to submit their algorithmic solutions. All students received the same problem sets in each marathon to ensure consistency in the study.

The study featured four distinct marathons:(i)Marathon Without Feedback (MNF): Students solved problems without receiving any feedback; (ii)Marathon with Delayed Feedback (MFA): Feedback was provided after a delay, allowing students time to work through the problems independently before receiving guidance; (iii)Marathon with Immediate Proactive Feedback (MFIP): Feedback was proactively provided by the instructor based on the instructor's perception of the student's needs, regardless of whether the students requested ii; (iv)Marathon with Immediate Reactive Feedback (MFIR): Feedback was provided immediately upon the student's request, responding reactively to their needs.

By structuring the marathons in this way, we aimed to explore the impact of different feedback timings and types on the students' problem-solving processes and learning outcomes.

3.3.2. Execution

In the experiment execution stage, students participated in the marathons in the following order: 1st MNF, 2nd MFIR, 3rd MFA and 4th MFIP. Each marathon was carried out in three stages: introduction, problem solving and evaluation.

MNF: In the first marathon, students solved the problems without receiving or requesting feedback from the teacher during the solving process. This marathon was held in a synchronous meeting on the Discord platform, lasting 3 hours. In the introduction (20 minutes), students received detailed instructions about the marathon and a link to access the problems and evaluation form on Google Forms. Then, they began solving the problems (2 hours and 20 minutes) and, at the end, they answered the marathon evaluation form (20 minutes).

MFIR: In the second marathon, the moment of immediate reactive feedback (FIR) was applied. For this, five groups of two or three students were created, with each group attended at different times, totaling 3 hours each. On Google Meet, each student was treated individually and could request feedback at any time via chat or video. In each 3-hour session, the first 20 minutes were dedicated to instructions about the marathon and resolving questions, along with the link to the problems and the evaluation form on Google Forms. Then, problem solving took place (2 hours and 20 minutes) and, at the end, they responded to the evaluation form (20 minutes). As students were attended at 5 different times, this marathon lasted a total of 15 hours, spread over 3 days.

MFA: In the third marathon, we studied the application of delayed feedback (FA). Students solved all problems without access to feedback during the process. However, at the end and after submitting the solutions, they received feedback on each problem. This marathon was held in a synchronous meeting on Discord, lasting 3 hours. In the introduction (20 minutes), students received detailed instructions about the marathon, mainly about when feedback would be provided, and a link to the problems and the evaluation form in Google Forms. Then, they began solving the problems (2 hours and 20 minutes) and, at the end, they answered the evaluation form (20 minutes).

MFIP: In the last marathon, we implemented proactive immediate feedback (FIP). The students were divided into seven pairs, with each pair attending at different times. In individual meetings on Google Meet, each student shared their screen, allowing the teacher to follow the entire resolution process and proactively provide feedback when necessary. Each 3-hour session began with 20 minutes of guidance on the marathon, resolving doubts and providing links to the problems and the evaluation form on Google Forms. Then, they began solving the problems (2 hours and 20 minutes) and, in the end, they answered the evaluation form (20 minutes). As students were attended at 7 different times, this marathon lasted a total of 21 hours, spread over 4 days.

3.3.3. Assessment

Table 1. Means and Standard Deviations of Measurements Moments of Feedback and DFA

	n	Achievement		Effort		Persistence	
		Mean	SD	Mean	SD	Mean	SD
No Feedback (NF)	14	2.70	0.00	43.77	0.00	17.12	0.00
DFA Low (NF)	2	3.00	0.00	50.53	6.97	16.77	2.55
DFA Medium (NF)	3	2.67	0.47	41.65	13.79	17.55	3.44
DFA High (NF)	9	2.44	2.44	39.12	19.17	17.05	3.89
Delayed Feedback (AO)	14	2.99	0.00	44.72	0.00	15.37	0.00
DFA Low (AO)	2	3.50	0.50	40.70	16.81	14.63	4.20
DFA Medium (AO)	3	2.67	1.23	46.09	15.70	15.23	4.39
DFA High (AO)	10	2.80	1.25	47.36	25.96	16.26	4.54
Immediate Reactive Feedback (IR)	14	4.42	0.00	83.28	0.00	12.49	0.00
DFA Low (IR)	8	4.25	0.66	80.16	9.46	12.73	3.80
DFA Medium (IR)	4	4.50	0.87	82.72	10.11	11.83	4.09
DFA High (IR)	2	4.50	0.50	86.95	9.56	12.90	4.20
Immediate Proactive Feedback (IP)	14	4.66	0.00	89.56	0.00	11.10	0.00
DFA Low (IP)	7	4.57	0.49	90.62	6.30	10.80	3.92
DFA Medium (IP)	3	4.67	0.47	87.83	0.88	11.09	4.23
DFA High (IP)	4	4.75	0.43	90.22	8.15	11.40	3.83

In a 4x4 between-subjects crossover study design, feedback conditions and demand for assistance (DFA) were considered independent variables. Although each marathon had a different time for providing feedback, students were free to seek other types of assistance, such as consulting with peers and searching for information on the Internet. Given this distribution, students were divided into three DFA groups: low DFA (N=12), medium DFA (N=44), and high DFA (N=60). There was no significant difference between the groups' scores.

For each marathon, students were asked to rate three items related to Demand for Assistance (DFA) capabilities during the problem-solving process. These items were: "I wish I had received more help when solving problems" (strongly agree – strongly disagree), "I have sought help from colleagues when solving one or more problems" (very often – rarely), and "I have sought help on the Internet while solving one or more problems" (very often – rarely). Students responded to each of these items on a response scale that ranged from 1 (representing the negative pole) to 5 (representing the positive pole). For this study, scores for these DFA items had a Cronbach's Alpha of 0.82. Thus, to calculate each person's score, the minimum DFA score was 12 and the maximum was

60. Analysis of the empirical distribution of these DFA scores revealed that, on average, 3 students rated the three DFA items with a center value of 44. Given this distribution, students were divided into three DFA groups: low DFA (N=12), medium DFA (N=44), and high DFA (N=60). There was no significant difference between the groups' scores.

3.4. Measures

In each marathon, students had the autonomy to decide how much time they dedicated to solving each problem. Therefore, the time spent on solving a particular problem can reflect the level of effort students were willing to invest in developing a solution [Narciss 2004]. This total problem-solving time was chosen as an indicator of *effort* and was calculated for each student by summing the minutes spent working on each problem. Additionally, students had the flexibility to choose whether to attempt and complete each of the problems provided in every marathon. The total number of problems a student engaged with was thus considered a measure of their *engagement*. We assessed the percentage of problems completed as a measure of *persistence*.

4. Results and discussions

Feedback serves a crucial role in education, as it must deliver valuable information in a manner that students can effectively utilize to bridge the gap between their current and desired states of learning.

In this section, we present the analysis of data derived from the exploratory study. We examined the impact of each feedback moment on demand for assistance (DFA) in relation to effort, persistence, and engagement using a 4 (feedback moment) x 4 (DFA) multivariate analysis of variance (MANOVA). Effect sizes were found to be moderate to strong (0.65, 0.17, 0.52, 0.34), indicating significant impacts. Significant effects were further explored using univariate analyses of variance (ANOVA) with post-hoc pairwise comparisons to identify specific differences between marathons. In these analyses, we conducted pairwise comparisons for each of the 12 DFA conditions.

Table 1 presents descriptive statistics for all measures across feedback moments. Our analysis revealed significant correlations between (a) problem engagement, effort, and persistence, and (b) feedback timing and demand for assistance. These findings underscore the nuanced relationship between feedback provision timing and students' engagement levels in problem-solving activities.

Multivariate effects of feedback timing and demand for assistance (DFA)

Using MANOVA, we found a significant overarching effect of DFA, Wilks' lambda = 0.92, F(10, 65) = 1.62, p = 0.03, eta^2 = 0.65. This result indicates that there are differences between DFA groups at different feedback provision timings. Next, we discuss ANOVAs to explore the effects of feedback moments.

Univariate effects of feedback timing

Univariate tests provided significant feedback moment effects for persistence, $F(3, 65) = 5.43, p = 0.01, \eta^2 = 0.17$, and effort, $F(2, 65) = 4.62, p = 0.02, \eta^2 = 0.17$.

Univariate follow-up tests revealed significant differences between marathons and feedback timing for both persistence and effort. Specifically, we observed that during delayed or no feedback moments, students sought more assistance from colleagues and the

Internet. They also expressed a greater desire for additional assistance. Conversely, during the two immediate feedback moments, students showed higher levels of engagement, effort, and persistence.

Table 1 provides a detailed view of the means and standard deviations of feedback moment and Demand for Assistance (DFA) measurements in different experimental conditions. When analyzing the data, it was observed that the different moments of feedback had varying impacts on the students' performance in terms of achievement, effort, and persistence during the problem-solving marathons.

Initially, in the groups that did not receive feedback or received late feedback (NF and AO), students showed moderate performance, with comparable levels of effort and persistence. These results suggest that the absence or delay in feedback can limit students' engagement and motivation in completing the proposed problems. On the other hand, groups that received immediate feedback, both reactive (IR) and proactive (IP), demonstrated significant improvements in all measures evaluated. Students who participated in these conditions showed the highest levels of achievement, effort, and persistence. Specifically, those who received immediate proactive feedback (IP) achieved the highest scores, indicating that direct and timely intervention from instructors or intelligent tutoring systems were highly beneficial to learning.

These results highlight the critical importance of timing and quality of feedback in promoting a more effective and engaging educational environment. Providing immediate feedback not only made it easier to correct errors and improve technical performance but also encouraged greater persistence in solving problems. Therefore, educational strategies that prioritize the rapid and relevant delivery of feedback can play a crucial role in closing learning gaps and increasing the effectiveness of teaching and learning processes.

Given these results, there are several interesting directions that can be explored based on the findings of this study on feedback in programming learning environments:

1. Exploration of Other Educational Contexts: Investigate how different educational contexts, such as different education levels or different programming disciplines, respond to the implementation of real-time feedback. This could help generalize the results found in this study beyond the specific environment studied.
2. Impact of Feedback on Motivation and Self-Directed Learning: Explore how the type and timing of feedback influences not only academic performance, but also students' motivation and their ability to learn in a self-directed way. This could include investigating how feedback affects students' self-efficacy and their willingness to solve problems independently.
3. Development of Intelligent Feedback Systems: Invest in research into the development of intelligent feedback systems that can dynamically adapt to individual student needs. This includes exploring artificial intelligence and machine learning techniques to personalize feedback based on students' past performance, learning style, and areas of difficulty.
4. Long-Term Assessment of the Effects of Feedback: Conduct longitudinal studies to assess the effects of feedback over time, not only in terms of immediate performance, but also in terms of retention of knowledge and skills. This could reveal insights into the sustainability and durability of the benefits of feedback in educational settings.

5. Multimodal Feedback Integration: Investigate how combining different feedback modalities (e.g., textual, visual, auditory) can improve the effectiveness of feedback and meet the varied needs of learners. This could include experiments that compared the impact of different types of multimodal feedback on various learning metrics.

These research directions have the potential to deepen our understanding of the role of feedback in promoting more effective and adaptive learning environments, contributing to the continued development of evidence-based educational practices.

5. Conclusion

In this paper, we focused on the issue of giving feedback to students who are learning programming, in an introductory course for novice programmers. Thus, we presented an initial exploratory study to investigate which and how different feedback elements affect students in programming problem-solving, having considered a solid and well-known conceptual framework as theoretical support. In this sense, we used data collected from a small group of students who went through a course where four different ways of giving feedback were explored. Hence, we found relevant relationships between studied variables, involving behavioral data and performance measures.

The study revealed that the moment in which feedback is provided significantly influences student engagement. Immediate feedback, both reactive and proactive, has been associated with higher levels of engagement, effort, and persistence in solving programming problems. In contrast, situations without feedback or with delayed feedback resulted in greater demand for external assistance, such as colleagues or the internet. No significant differences were found in academic performance between the different feedback moments, suggesting a direct moderate impact on final performance. It is recommended for future research to explore other educational contexts, the impact of feedback on motivation and develop intelligent feedback systems.

References

Alves, F. P. and Jaques, P. (2014). Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 3, page 51.

Boud, D. and Molloy, E., editors (2013). *Feedback in higher and professional education: understanding it and doing it well*. Routledge.

Caspersen, M. E. and Bennedsen, J. (2007). Instructional design of a programming course: a learning theoretic approach. In *Proceedings of the third international workshop on Computing education research*, pages 111–122.

Cavalcanti, A. P., de Mello, R. F. L., de Miranda, P. B. C., and de Freitas, F. L. G. (2020). Análise automática de feedback em ambientes de aprendizagem online. In *Anais do XXXI Simpósio Brasileiro de Informática na Educaçao*, pages 892–901. SBC.

Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1):81–112.

Houghton, W. (2004). *Learning and Teaching Theory for Engineering Academics*. Engineering Subject Centre.

Koulouri, T., Lauria, S., and Macredie, R. D. (2014). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Trans. Comput. Educ.*, 14(4).

Kunkle, W. M. and Allen, R. B. (2016). The impact of different teaching approaches and languages on student learning of introductory programming concepts. *ACM Trans. Comput. Educ.*, 16(1).

Lahtinen, E., Ala-Mutka, K., and Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *Acm sigcse bulletin*, 37(3):14–18.

Langer, P. (2011). The use of feedback in education: a complex instructional strategy. *Psychological Reports*, 109(3):775–7848.

Medeiros, R. P., Ramalho, G. L., and Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2):77–90.

Narciss, S. (2004). The impact of informative tutoring feedback and self-efficacy on motivation and achievement in concept learning. *Experimental Psychology*, 51(3):214–228.

Narciss, S. (2008). Feedback strategies for interactive learning tasks. In Spector, J. M., Merrill, M. D., Van Merrienboer, J. J. G., and Driscoll, M. P., editors, *Handbook of research on educational communications and technology*, pages 125–143. Erlbaum, 3rd edition.

Raabe, A., de Jesus, E. A., Hodecker, A., and Pelz, F. (2015). Avaliaçao do feedback gerado por um corretor automático de algoritmos. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 358.

Sheard, J., Simon, S., Hamilton, M., and Lönnberg, J. (2009). Analysis of research into the teaching and learning of programming. In *Proceedings of the fifth international workshop on Computing education research workshop*, pages 93–104.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

Spohrer, J. C. and Soloway, E. (1989). Simulating student programmers. In *IJCAI*, volume 89, pages 543–549. Citeseer.