Motivating Students to Learn Computer Programming in Higher Education: The SimProgramming Approach

Ricardo Rodrigues Nunes[1](✉), Gonçalo Cruz[2], Daniela Pedrosa[2], Ana Margarida Maia[3], Leonel Morgado[4], Hugo Paredes[5], José Cravino[6], and Paulo Martins[5]

[1] Brazilian Institute of Neuroscience and Neurotechnology (CEPID BRAINN FAPESP) & Faculdade de Ciências Médicas (UNICAMP), Campinas, São Paulo, Brazil
rnunes@unicamp.br
[2] CIDTFF - Universidade de Aveiro & Universidade de Trás-os-Montes e Alto Douro, Aveiro, Vila Real, Portugal
goncaloc@utad.pt, dpedrosa@ua.pt
[3] CIDTFF - Universidade de Aveiro, Aveiro, Portugal
margaridam@utad.pt
[4] Universidade Aberta & INESC TEC, Coimbra, Portugal
leonel.morgado@uab.pt
[5] Universidade de Trás-os-Montes e Alto Douro & INESC TEC, Vila Real, Portugal
{hparedes,pmartins}@utad.pt
[6] Universidade de Trás-os-Montes e Alto Douro & Research Centre Didactics and Technology in the Education of Trainers (CIDTFF), Aveiro, Vila Real, Portugal
jcravino@utad.pt

Abstract. This paper presents an action research study aiming to motivate undergraduate students to develop their computer programming learning skills, particularly within the transition from beginner to proficient level. The SimProgramming motivational approach is presented as a didactic proposal for this context. From the results of this iterative research process, we concluded that SimProgramming is a promising tool for teaching computer programming skills in intermediate classes, with potential to be used and/or applied in other educational contexts.

Keywords: Motivation to learn · SimProgramming · Computer programming · Computer science · Engineering education

1 Introduction

Motivation is a topic of great importance for computer programming teachers. It is quite common to hear them asking questions like "How can I motivate students to learn to code?", "What can I do to get students to actively participate in my programming classes?", or even "What can I do to engage students with educational programming activities?". These and other issues have been debated for many years in higher education, within the context of Engineering [1–3] and Computer Science courses [4–6].

A. Reis et al. (Eds.): TECH-EDU 2020, CCIS 1384, pp. 506–518, 2021.
https://doi.org/10.1007/978-3-030-73988-1_41

In intermediate level courses of two bachelor programmes at the University of the Trás-os-Montes e Alto Douro (UTAD), namely Informatics Engineering (IE) and Information and Communication Technologies (ICT), students typically create small programs by adapting and combining parts of pre-existing code. However, when challenged to develop other programs in advanced programming situations (e.g., when programmers need to write well-structured code leveraging pre-existing structures such as frameworks, libraries and/or APIs, Application Programming Interfaces) students do not seem to find the necessary motivation to learn, hampering their opportunities to realize the long-term benefits of organizing code in a more structured and manageable way. This is often a problem that will leave new graduates underprepared for the labour market, struggling in specific situations where these computer programming skills are fundamental.

Therefore, the main research objective of the study here presented is to tackle this problem, assuming the premise that students are not motivated and do not recognize the importance of better code organization due to their inexperience with team-based approaches to work in long-term software development settings. As an outcome, we present the SimProgramming motivational approach, arguing for the use of community-based learning environments to enable undergraduate students in becoming motivated and benefiting from contact with experienced (professional) programmers in order to succeed. In this approach, personal, behavioral, and environmental factors are dynamically related and strongly influence students' feelings about the needed skills to overcome possible challenges that arise during their learning process [7].

2 Motivation to Learn in Engineering Education

Much research has been conducted in Engineering Education on students' motivations to learn. Some of the most popular theories, adopted by different researchers are: achievement goal [8, 9], interest [10, 11], expectancy-value [12, 13], causal attributions [14, 15], self-efficacy [16, 17], and self-determination [18, 19]. However, it is argued that many of these studies are exploratory and do not clearly define what motivation is, or even that they do not consistently follow any specific theoretical framework [20]. It is essential to point out that the coexistence of these theories with their different aspects demonstrates how complex and multifaceted motivation is. Table 1 presents a summary of the main aspects of some of the motivation to learn theories.

Much of the research on motivation to learn deals with the importance of the pedagogical context in which the students are inserted, as well as with their involvement and persistence for performing the learning activities [21]. Either oral and written communication, or teamwork skills are considered fundamental to undergraduate students in Computer Science, as future professionals in the field [22]. Thus, many authors propose project-oriented and teamwork activities to motivate and facilitate the development of both students' innovation and complex problem-solving skills that allow them to succeed in the labor market [21].

Concerning the achievement goal theory, students showing a strong commitment with learning present greater autonomy and cognitive strategies [23]. Additionally, they are more likely to seek help from colleagues and teachers when facing learning difficulties, both online and in classroom environments [24]. This is explained either by goal-oriented or by self-efficacy theories on motivation to learn [25].

Table 1. Main aspects of motivation to learn theories.

Theory	Aspects related to the practice of teaching programming
Achievement goal	• Students' academic and social goals influence their motivation; • Students have different goals and consequently different behaviors throughout their studies; • Students concerned with the development of their learning present greater autonomy, elaborate more meaningful strategies, and feel safer in seeking for help from their peers; • Students concerned about their reputation in terms of skills and expertise can also achieve good academic results; • Students enjoy recognition through rewards or feedback; • Tasks need to be challenging, but appropriate to the knowledge levels of students; • Tasks need to be fair, according to the level of the required effort to complete them; • Academic environments that promote more significant interaction among students tend to be better for learning
Interest	• Interest increases the chances of students developing their skills; • Interest can be generated from activities that promote self-study, active learning, collaborative work, and social interaction, for example, through Problem-Based Learning (PBL), serious games, social games, virtual worlds, gamification, etc.; • Constant support and encouragement help to develop students' interest
Expectancy-value	• The value attributed to a task and the expectation of its successful completion can contribute to greater student involvement; • Individual beliefs and perceptions, as well as the socio-cultural environment, influence students' motivation; • The social context, collaboration, and authentic activities closer to the workplace settings promote student motivation
Causal attributions	• Students develop hypotheses (attributions) about the causes that led them to success or failure in carrying out their activities; • Through optimistic attributions, students are confident to succeed in the accomplishment of a task; • Through pessimistic assignments, students have little confidence in completing a task successfully; • Through hostile assignments, students can develop anger and present aggressive behaviors
Self-efficacy	• Students with higher expectations of self-efficacy perform better; • Previous positive experiences, social models, persuasion, and physiological reactions originate and develop students' self-efficacy beliefs; • Reducing stress situations and enhancing positive emotional states help to motivate students; • task structuring and supervision aligned with task complexity enhance students' motivation to learn; • Self-reflection, collaborative work, and activities closer to the workplace environment enhance students' motivation to learn; • Problem-Based Learning (PBL) helps to achieve a more significant task involvement by developing students' positive feelings (e.g., task value recognition, social acceptance)
Self-determination	• Intrinsically motivated students usually take a greater interest in learning; • Extrinsically motivated students generally experience lower academic performance compared to intrinsically motivated students; • Social and environmental settings that meet students' needs in terms of autonomy, competence, and relational skills promote their motivation; • Regulatory processes influence students' motivational behaviors

Self-efficacy, in turn, is an indicator of success in solving mathematical problems [26]. This can be enhanced in Problem-Based Learning (PBL) environments; teachers who use PBL are more likely to assist their students in overcoming difficulties that may negatively impact their self-efficacy beliefs, such as team composition or task difficulty [27]. PBL and its derivatives are reported in the literature as methodological tools that can promote and maintain students' motivational aspects as [28], for example, their situational interest [29].

Pascual [30] describes a PBL approach to increase students' knowledge through a social construction process of learning. His approach aimed to maximize opportunities for knowledge sharing between students and professionals, uniting academia through the creation of communities of practice. These ideas were based on the theory of self-determination. The focus was on increasing students' intrinsic motivation by creating the conditions needed for their social relationships. Various activities and tools were developed during the intervention as, for example, the development of meetings between the communities of students, maintenance engineers and academics, the development of recreational activities inside and outside the university campus, and the development of a web-based decision support system. This author started from the hypothesis that this multimodal approach would enhance active learning and social interactions. As a result of the research carried out, an increase in students' motivation was identified, confirming that communities of practice and social relationships are relevant factors for effective student learning. Autonomy and other psychological needs of self-determination were also related to PBL, which favors social interaction and promoting active learning and self-study in engineering courses [31].

An innovative pedagogical project has been developed in a higher education course taught at the Faculty of Engineering of the University of Porto (FEUP) [32]. In this curricular unit, called Project Management Laboratory (PML), undergraduate and master students developed different learning projects within a simulated business-like environment. During one semester, the students were divided into different teams and created companies to respond to customers' problems. The learning activities were divided into four stages, with two oral and public presentations at crucial moments throughout the project. The project was initiated by the startup stage, where companies were created from kick-off meetings. The second stage was the conception, where weekly project planning, quality control and risk management were carried out, accompanied by written reports. At the end of this stage, intermediate presentations were made in public sessions. The third stage was the software development itself, planning and the development of weekly reports, quality control and risk management. After this stage, final oral presentations were made in public sessions. At the end of the project, during the last stage, self-evaluation and project delivery meetings were made with customers. Students developed knowledge about project management, entrepreneurship, marketing, communication, customer interaction, and teamwork in software projects.

In addition to studies on the motivational impact of PBL on engineering students' learning, research was conducted on the relation between students' motivation with online activities and tools. For example, multimedia resources and discussion forums increased student motivation in blended learning and e-learning environments [33]. Evidence indicates that such approaches increase students' interest and emotional involvement with learning [33]. These results were somehow expected since the main aspects of students' lives, such as leisure, friendships, social interactions, and civic activities, are mediated by these technologies [34]. This was also reflected in studies on the impact of games in students' motivation and learning. An increasing number of research have been conducted on how serious games [35], social games [36], virtual worlds [37], and gamification [38] are related to several personal factors, such as students' motivational, affective, cognitive, and behavioral factors. All these interactive technologies helped develop participatory cultures in which collaboration and networking define what is understood by social and cultural competences [39]. Such technological and social contexts allowed students interaction to search for information, share resources, and develop their curiosity, interest, and involvement with learning [40].

Due to the complexity of factors that influence student motivation and learning, planning, and executing projects in the classroom is a challenge for many teachers. To facilitate these activities, research has been carried out on how models based on the expectancy-value theory can help implement teaching approaches that promote the social aspects of learning, as collaboration and authentic activities closer to the workplace environment [41]. The results of these efforts highlighted the importance of pedagogical practices proposed by teachers, as well as the positive impact that social interactions and the use of technologies have on students' motivation and learning.

3 Research Problem and Question

The main research problem addressed in this paper is how to tackle the difficulty that students have in making the transition from a basic to a proficient computer programming level. Students do not seem to value the benefits that well-structured, organized code can bring in the long run during this transition. It is difficult for them to acknowledge the value of coding while thinking about its architectural features, mainly because they can make their programs work anyway, without this extra effort.

In this context, knowing that motivation has been identified as an important factor for students' learning success, we assume that the evidence generated by research on this topic can help to tackle the identified research problem. Thus, the research question addressed in our study was: How to bring the knowledge about learning motivation into the context of higher education intermediate computer programming classes in order to support students' transition from basic to proficient level?

4 Research Method

An action-research study was carried out within the curricular unit of "Programming Methodologies III" (PMIII), over three iterations/academic years, at the UTAD. The curricular unit has a duration of 4 months and is a mandatory subject of the second year of the undergraduate programmes in IE and ICT, with the main goal of introducing software architecture concepts to support the development of students' code organization skills.

An approach was designed and developed from a set of different learning problems related to the topics covered in the course. In Table 2, some instances of the problems are presented. For them, students, in groups, were asked to propose a solution. The activities were developed using PBL. Each group was assigned with a specific problem involving an architectural pattern related to MVC (Model-View-Controller) and pre-existing structures such as frameworks, libraries, or Application Programming Interface (API). At the end of each edition, the students' groups have to develop a written document explaining the approaches used to develop their code with the application of an architectural pattern with the pre-existing structure related to the assigned problem.

Throughout the research project, quantitative and qualitative data were collected from submitted files and logs in the information systems adopted in the curricular unit (e.g., Wiki PBWorks and Moodle), online questionnaires, semi-structured interviews, audiovisual recordings of some face-to-face activities and direct observation.

In the first iteration [42], students were challenged to solve the problem assigned with theoretical and practical components by researching technical-scientific literature, and by engaging with more experienced programmers in different social networks and/or online communities of practice. The goal was to motivate them to develop their computer programming skills by meeting and asking for help from other professionals, members

Table 2. Some of the specific problems assigned to student teams.

Problems
Write a detailed document that explains how to apply the MVC (Model-View-Controller) architectural standard to application development with libOpenMetaverse. This document should complement this explanation with concrete examples of the various forms of application that design
Write a detailed document that explains how to apply the MVP (Model-View-Presenter) architectural standard to the development of applications in the Windows Phone Application Platform, with the XNA framework. This document should complement this explanation with concrete examples of the various ways they conceive for applying the standard
Write a detailed document that explains how to apply the MVVM (Model-View-ViewModel) architectural pattern to the development of applications with Windows Forms. This document should complement this explanation with concrete examples of the various ways they conceive for applying the architectural pattern MVVM

of these online communities. The results of this first iteration showed that most groups were unable to solve their assignments successfully. Of the twenty groups (a total of 62 students) that were initially enrolled in the project, nineteen groups (a total of 59 students) performed some tasks. Seven groups performed the activities during all phases of the project, with four of them reaching an acceptable quality level regarding the learning outcomes achieved at the end. The overall quality of the reports from these groups that completed the project showed that the students acquired significant aspects on the themes of the proposed assignments and were able to provide useful examples of the approaches they developed; however, this effort was not reflected in the final grade of the curricular unit, which were much more dependent on the two written tests that students had to perform than the project itself- the latter only counted 20% for the final grade.

The second iteration occurred in the following year [43]. The activities were implemented with new teaching and learning strategies: they were more structured throughout the project (with weekly tasks and strict deadlines), and supervised by two tutors to provide support to the students through follow-up and feedback, which also facilitate three group dynamics conducted within the classroom. Compared to the previous cycle, it was found that more groups participated actively throughout the project. More specifically, nine of the twenty-one groups that started the project continually developed their activities and received feedback in the online communities of practice until the end. From the results of this second iteration, it was possible to identify and offer possible solutions to the students' main problems concerning their learning process, namely the lack of time to work, the lack of feedback on the development of the project, and the low motivation to complete the assigned activities. Based on these results, a motivational approach called SimProgramming was developed, which simulates a business-like environment and was then implemented in the third and last cycle of this research.

5 The SimProgramming Motivational Approach

Some significant changes were implemented in the third research iteration, reported already somewhere [44, 45]. One of them was creating a community of practice with students, alumni, and invited external programmers, instead of asking students to participate in the external online communities, and thus promoting a more significant interaction and strengthening the relationships among the different project participants. Better project management practices were also adopted to identify potential problems faced by students at an earlier stage, allowing the provision of supportive guidance by the teaching team according to their needs and, consequently, helping them to obtain better results. Therefore, it was also proposed a reformulation of the interaction and assessment strategies previously adopted, and a business-like environment was designed, and a popular project management method used in the workplace, known as Scrum, was implemented [46]. Other aspects, like continuous feedback, self-evaluation and hetero-evaluation strategies were also implemented. Below, the SimProgramming motivation approach is described according to its main conceptual foundations and application phases.

5.1 Conceptual Foundations

The SimProgramming approach is based on four conceptual foundations: (1) a business-simulated environment for learning, (2) self-regulated learning; (3) Co-regulated learning; and (4) formative assessment.

A Business-Simulated Environment for Learning (Situated learning). In this environment, each participant plays a role within a business roleplay context. The teacher plays the role of CEO - Chief Executive Officer (or General Manager), globally responsible for evaluating the progress of the projects, clarifying the specific doubts that students have and guiding them from the analyses made on the presentations in the face-to-face meetings and on the reports submitted. Tutors or Teaching Assistants take the role of project managers, responsible for closer monitoring, regular feedback, and mentoring. In all teams, one student plays the role of team coordinator, communicating with management (general manager and project managers). Among his duties are ensuring integration of other students/participants in the team, making sure that they maintain an overall view of the project and its status, as well as reporting issues, difficulties and doubts experienced by the teams to the managers (i.e., acting as a link). Team coordinators, like the project managers, are motivating agents for students. The proposal of simulating a business environment is based on the available literature about the influence that the pedagogical context has on the students' motivation.

Self-regulated Learning. SimProgramming promotes active learning, developing students' self-regulation, by enabling a study routine through weekly task deliveries, keeping students focused on course content. Weekly reports allow students to reflect on their work, planning what to do the following week, and identifying negative factors that prevent them from achieving individual and team objectives. This conceptual foundation also supports developing students' skills to avoid procrastination and improve time management. Thus, they are encouraged to adopt a study routine by promoting a motivational context for learning. In this way, students can gradually develop their accountability, doing their activities regularly and not delaying them to the last moment.

Co-regulated Learning. Students with little experience have only general ideas about the disciplinary field. When they acquire new responsibilities, performing more complex tasks, they gradually develop their identities against other participants in communities of practice. As they interact with others, they increase the opportunities to learn through their contributions and feedback. Thus, the teaching team must provide support to develop social and group interactions, guiding students' participation and gradual involvement in such communities. This development of co-regulation includes suggesting specific strategies for interaction. Such actions support the development of students' sense of being, informal interactions, and debating. This can also be promoted and monitored through social media groups created specifically for the curricular unit. Such an internal community can be maintained over the years, becoming a source of interaction between new students, former students still studying (e.g., MSc or PhD students), and graduates already in the labor market. Some of these alumni can play a role of business consultants, providing support and advice to the new teams.

Formative Assessment. This process includes motivational mentoring and constant feedback on the project development status, for example, on whether the work is progressing or deviating from initial expectations and goals. For this, weekly meetings are held based on Scrum - an agile method for planning and managing software development projects at the workplace. This support is provided face-to-face through weekly meetings that are scheduled with the team coordinators, in which three specific topics are addressed: 1) "What did the team do during the past week?"; 2) "What will be done during the current week?" and; 3) "What is preventing the conclusion of the activities? When specific problems (technical, personal, or others) are detected, meetings are scheduled with the teams that presented such difficulties, case by case. In addition to these meetings, all students need to answer the same three specific topics in an online form and submit it on a weekly basis. In addition to these forms, all students also need to submit reports on their interactions in the communities of practice on two specific occasions of the project (at the end of phases 1 and 2 - please see below in '5.2 Application stages'). These reports serve as formative assessment tools. At the end of the project, students make their self- and hetero-evaluation of their teammates. All these reports are excellent opportunities for students to reflect on their learning activities. There are still three presentations at the end of each phase of the project made by students. These presentations are evaluated by the teaching team and serve as a formative resource for guiding the students projects' progress. It is important to note that each meeting with the students is an opportunity to talk informally about the project and guide them on it.

5.2 Application Stages

The SimProgramming approach develops over 4 main stages, based on the conceptual foundations presented above. These phases represent the result of the reflections made by researchers on the entire action-research process. Table 3 presents a summary of the activities carried out during the four stages.

Table 3. Summary of SimProgramming stages and activities.

Stages	Activities
Stage 1 (design)	Organization of the teams; literature review; interaction in the communities of practice; initial presentation; weekly report
Stage 2 (development)	Interaction in the communities of practice; intermediate presentation; weekly report; report of interactions in the communities
Stage 3 (refinement)	Final presentation; final report
Stage 4 (closure)	Final report improved; self-evaluation and hetero-evaluation

Throughout all the project stages, weekly meetings are scheduled between project managers (tutors) and team coordinators, providing motivational support and clarification on more technical questions. When internal problems in any of the student's groups are identified, project managers can also schedule meetings with these teams or with any specific student to make more targeted interventions. Students also submit reports at the end of each stage, lasting one week, detailing what they have done, what they will do, and what eventually prevents them from performing the planned activities. On two occasions (end of Stage 1 and Stage 2), students still submit reports about their interactions in the communities of practice.

6 Results and Conclusion

According to the third cycle of this action-research study, in which SimProgramming was implemented, data was collected from a variety of sources, namely the students' forms filled in and submitted online, the notes from the direct observations of researchers, the audiovisual recordings of face-to-face meetings, the records of messages exchanged between the different actors, as well as the logs of the interactions in the information systems adopted for the curricular unit. During the application of the approach, it was possible to gather evidence on its adequacy to the proposed goals, and it was concluded that the SimProgramming proved to be an important motivational tool for students to develop their computer programming skills. In this cycle, more students participated actively in the project, and the results were reflected in the final grades of the curricular unit. Of the fifteen teams, eleven maintained a constant quality performance on all activities throughout the project. Of the four remaining teams, two worked regularly, but the quality of their reports was below the expected level, and the other two groups did not even start the project. In a total of 97 students, 66 completed the tasks, and 59 performed well in the learning activities. Table 4 summarizes the work developed by each team.

It is important to emphasize that the SimProgramming approach implied significant changes to the teaching practice, especially in terms of the relationship between teaching staff and students, both inside and outside the classroom. Maintaining a good relationship between all the actors guaranteed a safe and supportive environment that was of the utmost importance for the development of students' learning and academic success.

Table 4. Summary of students' teams in SimProgramming approach.

Team	Assignment development	
	No. members	Summary
T1	6	Regular individual and team task assignment completion. They created a Facebook group for internal interaction
T2	7	Regular individual assignment completion (except the last two weeks). Team tasks performed regularly
T3	7	Irregular individual assignment completion. The team delivered just two reports. One student quit the project and two others got negative grades. Tutor interventions were necessary via face-to-face meetings for feedback
T4	7	Irregular individual assignment completion. They carried out all team tasks but didn't perform the first presentation. Three students quit the project. Tutor interventions were necessary via face-to-face meetings for feedback
T5	7	Initially they weren't achieving the goals, but after replacing the coordinator (the original one quit the project), the team showed better results. Tutor interventions were necessary via face-to-face meetings for feedback
T6	4	Regular individual and team task assignment completion. Tutor interventions were necessary via face-to-face meetings for feedback
T7	7	The team quit the project. The students didn't respond to invitations for meetings with the tutors
T8	7	Regular individual and team tasks assignment completion
T9	6	Regular delivery (with some delays) of individual and team task assignments. One student quit the project. The coordinator was very committed and impactful for the success of the team. They created a Facebook group for internal interaction
T10	6	Regular individual and team tasks assignment completion. Two students quit the project
T11	6	The team quit the project. The students didn't answer invitations for meetings with tutors
T12	7	The team didn't work, and five students quit. Two students met with tutors and were instructed to perform compensatory activities. The other students didn't respond to tutors' invitations
T13	7	Regular for most assignments: individual (with the exception of one student) and team tasks
T14	8	Three students were regularly delivering individual and team task assignments. Tutor interventions were necessary via face-to-face meetings for motivation and feedback. Five students quit the project
T15	5	Regular individual and team tasks assignment completion. One student quit the project

Throughout the project, a relationship of trust between the teaching team and the students was built.

Thus, as the main contribution of this research, the SimProgramming approach is presented as a tool to support learning motivation in computer programming higher education courses, and specifically within the transition from basic to proficient level of programming skills. Another contribution is the whole research-action process developed in the context of this study, which educators can use as a guiding plan and method for the development of innovative pedagogical approaches in the computer education field.

Acknowledgments. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project UIDB/50014/2020.

References

1. Ibrahim, A., Aulls, M.W., Shore, B.M.: Teachers' roles, students' personalities, inquiry learning outcomes, and practices of science and engineering: the development and validation of the McGill attainment value for Inquiry engagement survey in STEM disciplines. Int. J. Sci. Math. Educ. **15**(7), 1195–1215 (2017)
2. Lucke, T., Dunn, P.K., Christie, M.: Activating learning in engineering education using ICT and the concept of 'Flipping the classroom.' Eur. J. Eng. Educ. **42**(1), 45–57 (2017)
3. López-Fernández, D., Ezquerro, J.M., Rodríguez, J., Porter, J., Lapuerta, V.: Motivational impact of active learning methods in aerospace engineering students. Acta Astronaut. **165**, 344–354 (2019)
4. Dicheva, D., Dichev, C., Agre, G., Angelova, G.: Gamification in education: a systematic mapping study. J. Educ. Technol. Soc. **18**(3), 75–88 (2015)
5. Hao, Q., Barnes, B., Wright, E., Branch, R.M.: The influence of achievement goals on online help seeking of computer science students. Br. J. Edu. Technol. **48**(6), 1273–1283 (2017)
6. Gorson, J., O'Rourke, E.: How do students talk about intelligence? An investigation of motivation, self-efficacy, and mindsets in computer science. In: Proceedings of the 2019 ACM Conference on International Computing Education Research, pp. 21–29. Association for Computing Machinery, New York (2019)
7. Slavich, G.M., Zimbardo, P.G.: Transformational teaching: theoretical underpinnings, basic principles, and core methods. Educ. Psychol. Rev. **24**(4), 569–608 (2012)
8. Hakulinen, L., Auvinen, T.: The effect of gamification on students with different achievement goal orientations. In: The Proceedings of the 2014 International Conference on Teaching and Learning in Computing and Engineering (LaTiCE), pp. 9–16. IEEE, Kuching (2014)
9. Banu, E.A., Hunsu, N.: Exploring the role of students' achievement goals and learning approaches in academic performance. In: the Proceedings of the 126th American Society for Engineering Education Annual Conference & Exposition, pp. 1–7. ASEE, Florida (2019)
10. Clapper, T.C.: Situational interest and instructional design: a guide for simulation facilitators. Simul. Gaming **45**(2), 167–182 (2014)
11. Rowe, J.P., Shores, L.R., Mott, B.W., Lester, J.C.: Integrating learning and engagement in narrative-centered learning environments. In: the Proceedings of the 10th International Conference on Intelligent Tutoring Systems - Volume Part II, pp. 166–177. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13437-1_17
12. Chen, J.L.: The effects of education compatibility and technological expectancy on e-learning acceptance. Comput. Educ. **57**(2), 1501–1511 (2011)
13. Wu, F., Fan, W., Arbona, C., de la Rosa-Pohl, D.: Self-efficacy and subjective task values in relation to choice, effort, persistence, and continuation in engineering: an Expectancy-value theory perspective. Eur. J. Eng. Educ. **45**(1), 151–163 (2020)
14. Vivian, R., Falkner, K., Falkner, N.: Computer science students' causal attributions for successful and unsuccessful outcomes in programming assignments. In: the Proceedings of the 13th Koli Calling International Conference on Computing Education Research, pp. 125–134. Association for Computing Machinery, New York (2013)
15. Duron, T., Gallon, L., Aniorte, P.: Modelling learner's perseverance in education software. In: The Proceedings of the 35th Information Systems Education Conference (ISECON 2019), Galveston, Texas, United States, pp. 34–38 (2019)
16. Stolk, J., Harari, J.: Student motivations as predictors of high-level cognitions in project-based classrooms. Act. Learn. High. Educ. **15**(3), 231–247 (2014)
17. Starkey, E.M., Hunter, S.T., Miller, S.R.: Are creativity and self-efficacy at odds? An exploration in variations of product dissection in engineering education. J. Mech. Des. **141**(1), 012001 1–012001 11 (2019)

18. Hartmann, A., Gommer, L.: To play or not to play: on the motivational effects of games in engineering education. Eur. J. Eng. Educ. 1–25 (2019).
19. Zhao, L., Lu, Y., Wang, B., Huang, W.: What makes them happy and curious online? An empirical study on high school students' internet use from a self-determination theory perspective. Comput. Educ. **56**(2), 346–356 (2011)
20. Brown, P.R., McCord, R.E., Matusovich, H.M., Kajfez, R.L.: The use of motivation theory in engineering education research: a systematic review of literature. Eur. J. Eng. Educ. **40**(2), 186–205 (2015)
21. Johri, A., Olds, B.M.: Situated engineering learning: bridging engineering education research and the learning sciences. J. Eng. Educ. **100**(1), 151–185 (2011)
22. Adams, R., et al.: Multiple perspectives on engaging future engineers. J. Eng. Educ. **100**(1), 48–88 (2011)
23. Chen, C.H., Wu, I.C.: The interplay between cognitive and motivational variables in a supportive online learning system for secondary physical education. Comput. Educ. **58**(1), 542–550 (2012)
24. Vogt, C.M.: Faculty as a critical juncture in student retention and performance in engineering programs. J. Eng. Educ. **97**(1), 27–36 (2008)
25. Hutchison, M.A., Follman, D.K., Sumpter, M., Bodner, G.M.: Factors influencing the self-efficacy beliefs of first-year engineering students. J. Eng. Educ. **95**(1), 39–47 (2006)
26. Pajares, F., Miller, M.D.: Role of self-efficacy and self-concept beliefs in mathematical problem solving: a path analysis. J. Educ. Psychol. **86**(2), 193–203 (1994)
27. Schaffer, S.P., Chen, X., Zhu, X., Oakes, W.C.: Self-efficacy for cross-disciplinary learning in project-based teams. J. Eng. Educ. **101**(1), 82–94 (2012)
28. Bédard, D., Lison, C., Dalle, D., Côté, D., Boutin, N.: Problem-based and project-based learning in engineering and medicine: determinants of students' engagement and persistance. Interdisc. J. Probl. Learn. **6**(2), 7–30 (2012)
29. Rotgans, J.I., Schmidt, H.G.: Problem-based learning and student motivation: the role of interest in learning and achievement. In: One-Day, One-Problem, pp. 85–101. Springer, Singapore (2012). https://doi.org/10.1007/978-981-4021-75-3_5
30. Pascual, R.: Enhancing project-oriented learning by joining communities of practice and opening spaces for relatedness. Eur. J. Eng. Educ. **35**(1), 3–16 (2010)
31. Stolk, J., Martello, R., Somerville, M., Geddes, J.: Engineering students' definitions of and responses to self-directed learning. Int. J. Eng. Educ. **26**(4), 900–913 (2010)
32. Laboratório de Gestão de Projetos Homepage (Challenge). https://lgp.fe.up.pt. Accessed 20 Nov 2017
33. Sun, J.C.-Y., Rueda, R.: Situational interest, computer self-efficacy and self-regulation: their impact on student engagement in distance education. Br. J. Edu. Technol. **43**(2), 191–204 (2012)
34. Palfrey, J., Gasser, U.: Born Digital: Understanding the First Generation of Digital Natives. Basic Books, New York (2008)
35. Fonseca, B., et al.: PLAYER - a European project and a game to foster entrepreneurship education for young people. J. Univ. Comput. Sci. (J.UCS) **18**(1), 86–105 (2012)
36. Fonseca, B., et al.: BIZZY – a social game for entrepreneurship education learning and collaboration technologies. In: The Proceedings of the International Conference on Learning and Collaboration Technologies, pp. 33–41. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07485-6_4
37. Morgado, L., Varajão, J., Coelho, D., Rodrigues, C., Sancin, C., Castello, V.: The attributes and advantages of virtual worlds for real world training. J. Virtual Worlds Educ. **1**(1), 15–35 (2010)

38. Dicheva, D., Irwin, K., Dichev, C., Talasila, S.: A course gamification platform supporting student motivation and engagement. In: The Proceedings of the International Conference on Web and Open Access to Learning (ICWOAL), Dubai, United Arab Emirates, pp. 1–4. IEEE (2014)
39. Jenkins, H., Clinton, K., Purushotma, R., Robison, A.J., Weigel, M.: Confronting the challenges of participatory culture: media education for the 21st century. In: John, D., Catherine T.M. (eds.) Foundation Reports on Digital Media and Learning. The MIT Press, Cambridge (2006)
40. Arnone, M.P., Small, R.V., Chauncey, S.A., McKenna, H.P.: Curiosity, interest and engagement in technology-pervasive learning environments: a new research agenda. Educ. Technol. Res. Dev. **59**(2), 181–198 (2011)
41. Vekiri, I.: Information science instruction and changes in girls' and boy's expectancy and value beliefs: in search of gender-equitable pedagogical practices. Comput. Educ. **64**, 104–115 (2013)
42. Morgado, L., et al.: Social networks, microblogging, virtual worlds, and Web 2.0 in the teaching of programming techniques for software engineering: a trial combining collaboration and social interaction beyond college. In: The Proceedings of the Global Engineering Education Conference (EDUCON), pp. 1–7. IEEE, Marrakech (2012)
43. Nunes, R.R., et al.: Enhancing students' motivation to learn software engineering programming techniques: a collaborative and social interaction approach. In: the Proceedings of the International Conference on Universal Access in Human-Computer Interaction, pp. 189–201. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20684-4_19
44. Nunes, R.R., et al.: SimProgramming: uma abordagem motivacional para a aprendizagem de alunos intermediários de programação. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação, pp. 1099–1110. Sociedade Brasileira de Computação - SBC, Recife, Pernambuco (2017)
45. Pedrosa, D., Cravino, J., Morgado, L., Barreira, C., Nunes, R.R., Martins, P., Paredes, H.: Simprogramming: the development of an integrated teaching approach for computer programming in higher education. In: The Proceedings of the 10th International Technology, Education and Development Conference, pp. 7162–7172. IATED Academy, Valencia (2016)
46. Schwaber, K.: Agile Project Management with Scrum. Microsoft press, United States of America (2004)