A Computer Platform to Increase Motivation in Programming Students - PEP

Paula Correia Tavares[1,2], Pedro Rangel Henriques[2] and Elsa Ferreira Gomes [1,3]
[1]*Departamento de Informática, Instituto Superior de Engenharia do Porto, Porto, Portugal*
[2]*Centro Algoritmi & Departamento de Informática, Universidade do Minho, Braga, Portugal*
[3]*INESC TEC, Portugal*

Keywords: Motivation, Program Animation, Automatic Evaluation, Immediate Feedback.

Abstract: Motivate students is one of the biggest challenges that teachers have to face, in general and in particular in *programming courses*. In this article two techniques, aimed at supporting the teaching of programming, are discussed: program animation, and automatic evaluation of programs. Based on the combination of these techniques and their currently available tools, we will describe two possible approaches to increase motivation and improve the success. The conclusions of a first experiment conducted in the classroom will be presented. PEP, a Web-based tool that implements one of the approaches proposed, will be introduced.

1 INTRODUCTION

According to Hundhausen and Douglas (2000) or Proulx (2000) and many other authors, including the various notes included in the Computer Science Curricula of ACM/IEEE Guideline of 2013 (ACM/IEEE, 2013), confirmed by our professional experience teaching courses on *introduction to computer programming*, learn to program is an arduous and complex task that raises many challenges both to teachers and students. The lack of motivation is one of the main reasons for the students' failure in programming courses (Santos and Costa, 2006; Ramos, 2013). There are many reasons for students to fail in learning programming (Proulx, 2000) but the truth is that given the slightest difficulty in understanding the statement, in developing an algorithm or in the use of a programming language, discourage learners and they immediately give up. The project, in which the work here reported is inserted, aims: to understand the actual reasons for the difficulties which arise in the process of teaching/learning computer programming (discussed in (Tavares et al., 2015a)); to study computer-supported approaches to combat this failure (discussed in (Tavares et al., 2015b)); and to suggest ways to combine those approaches to increase the involvement of students in order to overcome such difficulties (Tavares et al., 2016b). In this paper, two techniques designed to support the teaching of programming are presented in particular: an older one, *Program Animation* that aims at taking advantage of our visual acuity and the effect of simulation to help understanding the algorithms and programs; and another, more recent, which focus on the use of systems for the *Automatic Evaluation of Programs* to encourage students to go on working providing them immediate feedback as soon as they finish writing a program. These two topics will be introduced briefly in section 3, aiming to support the combined approaches that are proposed.

As mentioned above, the first objective of this study focus on the difficulty intrinsic to the process of teaching/learning programming and the consequent failure. Thus, the project here discussed is based on deep research study concerned with the failure of learn to program and consequently in developing proposals to increase motivation and self-confidence of the students of introduction to programming courses. Experimental studies at the school level are essential for the development or use of learning aid platforms. These tools will be elements of teaching support to increase students' ability to solve their problem. It is important to help students in the transition from basic knowledge to the comprehension of an algorithmic solution. The goal is to get students to increase their ability to practice programming regularly since the first day, because we believe that in this way their success in

Tavares, P., Henriques, P. and Gomes, E.
A Computer Platform to Increase Motivation in Programming Students - PEP.
DOI: 10.5220/0006287402840291
In *Proceedings of the 9th International Conference on Computer Supported Education (CSEDU 2017) - Volume 1*, pages 284-291
ISBN: 978-989-758-239-4

school will increase. With regard to the resources currently available for animation and automatic evaluation of programs, we propose two alternative approaches, described in section 4 and 5, in order to expedite the process of teaching learning programming. To check these two alternatives, which are complementary, we conducted a classroom test following the first approach, AEv&Anim (summarized in section 4), and developed a tool that materializes the second approach, Anim&AEv (introduced and discussed in section 6). In the next section, we present a short survey about the basic subject underlying all our work: Human Motivation.

2 MOTIVATION

As will be seen below, several theories have been developed to explain the motivation from the beginning of the history of psychology as a science. Because it is a complex phenomenon, the subject has been studied under different prisms (Williams and Williams, 2011; Almeida, 2012). Some of them claim that people are motivated by material rewards, others by increasing their power and prestige in the world, or by an interesting work, enriched environments, recognition, or to be respected as an individual. The fact is that humans in general have very complex needs and desires. Motivation is one of the keys to understand the human behaviour; it acts on the thought, attention, emotion and action of the Human Being, involving desire, effort, dreams and hope (Williams and Williams, 2011; Almeida, 2012).

People are driven by very different factors, with varied experiences and respective involvement. The motivation leads to an action directed to a particular goal, being regulated by biological or cognitive, factors of each person. This action is enabled by the needs, emotions, values, goals and personal expectations, constituting a single intentional and multifaceted phenomenon (Ryan and Deci, 2000). According to Susana Ramos and colleague (2011), there are theories of motivation that characterize the individual as unique, but also try to analyse the motivational phenomenon in its origins, evolution and direction. Susana Ramos (2013) says that these theories can be classified in Satisfaction Theories - *Maslow's Hierarchy of Needs, McGregor's Theory X & Y, Herzberg's Two-Factor Theory* - and Theories of Progress - *McClelland's Need Theory and Vroom's expectancy theory*.

Motivate students is one of the biggest challenges that teachers have to face. In programming is particularly difficult. For the teacher play an important role in the learning process that occurs in the classroom, the teacher would have to have control over the external factors that influence the behavior and involvement of students (Callahan, 2010). The level of motivation needed to involve each student in a given task is determined by his expectation for success and the value that the student gives to that particular task. This theory suggests that students can succeed if they dedicate with effort and appreciate the activities in which they enrolled. As Almeida (2012) stated, it's important to understand why students do not have motivation. Many students attribute this problem to the behavior of the teachers and the school in general, with the expectation that they are active elements in their learning. To verify this statement we designed a questionnaire to survey students' actual opinion; as soon as we finish the analysis of the collected answers we will publish the study. On the other hand, the teacher assign the difficulties to the students, with the expectation that they are interested, auto-regulated, with energy to search for knowledge, and responsible for their own motivation. In this way, there is a conflict between students' expectations, and teachers, who expect a general behavior distinct from that, manifested by students (Almeida, 2012). The motivation is not only a unitary phenomenon, which refers to the concept of quantity. More than a lot of motivation, there are variations in levels and motivational guidelines. In this way, it is possible to ask what is the reason that leads to a more or less motivated behavior. To reason about motivational quality it is crucial to consider the attitudes and goals that move people towards an action. A good example is the motivation that compels a student to do his homework. He can do it without any curiosity or interest, simply looking for the approval of the teacher or parent; but, in the other way around, he can be motivated to acquire new knowledge, or face new challenges because he understands that his attitude brings advantage and values; or he can still be motivated because the knowledge acquired will give him a position to attain better grades or a better social life. In this example, the motivation may not vary quantitatively, but its nature (the quality) can be definitively distinct (Almeida, 2012). Distinguish between quantitative and qualitative aspects of motivation enlarges the view on it, as shown in Figure 1.

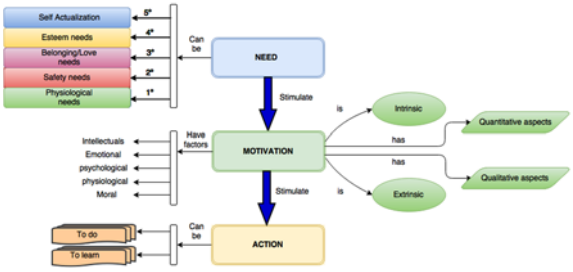

Figure 1: Factors involved in Motivation and its impact.

As can be seen in Figure 1, the motivation is directly related to different factors ranging from, physiological or emotional to intellectual and moral factors.

Motivation is like an impulse, a feeling that moves people to act to obtain their goals. Is what makes the individuals do their best, do what they can to get what they want.

According to the theory of self-determination, the motivation can be intrinsic or extrinsic. The intrinsic motivation does not need any external factor. It derives from the student himself as the dedication, competence, willingness and ability to accomplish a task. Extrinsic motivation is the result of external factors, such as the resources that the student has, the rewards, and the environment where it develops his tasks (Silva et al., 2014). Both work together and the result will set the student's behavior, as shown in the Figure 2.

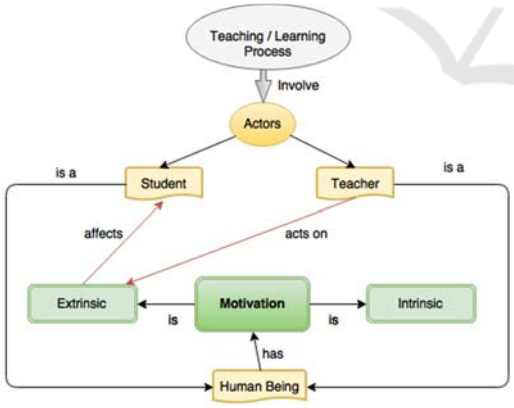

Figure 2: Student/Teacher and Motivation.

According to Deci and Ryan (2000), motivation has different degrees that in a continuum range from demotivation (absence of motivation), to intrinsic going through extrinsic motivation as depicted in Figure 3 (adapted from (Deci and Ryan, 2000)).

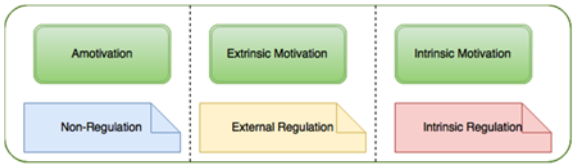

Figure 3: Different degrees of Motivation.

The motivational and emotional factors have a direct relation with the learning process. So, teachers have to understand it to improve their role in the classroom.

3 TECHNIQUES TO SUPPORT THE TEACHING OF PROGRAMMING

Among many techniques that have been studied to help on teaching of programming, two that are particularly interesting will be discussed in this context: *program animation, and automatic evaluation of programs*. For the sake of space, the deep survey of these two areas, in which our work is based, cannot be presented in detail. So, we recommend the reading of (Tavares et al., 2015b).

3.1 Animation

The animation of an algorithm is a type of dynamic visualization of its main abstractions. The importance of this technique lies in the ability to describe in a visible mode the essence or logic of an algorithm (Pereira, 2002). The relevance of algorithms animation in learning is justified because many times the teacher resorts to the use of visual representations to help students understand the essence of algorithms, and the dynamic behavior of programs. The animation can be composed of a set of views more or less interactive. Ari Korhonen (2003) explains how we should apply these techniques in order to help students deal with complex concepts. According to this author, from a pedagogical point of view, it will be more interesting to illustrate the logic and behavior of an algorithm than the implementation details. It must be ensured that this approach causes, at least, one level of progress on learning. This methodology requires an environment that provides feedback about student achievements. This statement partially inspired our proposal to be introduced later. In this context, there are several tools in order to assist students on learning programming, aimed at introducing basic

algorithmic concepts through a familiar and pleasant context. These animation tools were enumerated in a previous paper (Tavares et al., 2015b). The detailed study underlying that survey supported our choice. So, in the proposal below we use the Jeliot tool (Ben-Ari et al., 2011; 2002), an easy to use and effective Java animator.

The animation becomes a facilitator of the learning process since the presentation of abstract concepts is more didactic, improving the quality of the class support materials. It is clear that if the student acquires a good knowledge basis, his performance increases producing better results, and better professional abilities (Santos and Costa, 2006).

3.2 Automatic Evaluation

It is very important to give students the opportunity to practice and solve programming exercises by themselves. However, the maximum effectiveness of this approach requires the teacher's ability to review, mark and grade each solution written by students. Instant feedback is very important for the acquisition of knowledge. Independently of the particular learning strategy, it motivates students. However, in large classes and with few lecture hours, this approach is impractical. Individual feedback may consume too much teacher´s time with risk that students do not benefit from it in due time (Queirós and Leal, 2015).

It is difficult to find a tool that incorporates all the different advantages of the various existing tools. To improve the learning of programming it is important to be able to provide, at least, a combination of a virtual environment and a submission system. First it shall be provided support for solving problems in an individual study, and second, the environment shall support various formats for the submission of questions and evaluation. In other words, the teacher has to provide to students an automatic evaluation system able to return instant feedback.

New tools have emerged to facilitate and enable their use in teaching activities, allowing students to incorporate tests in their work. These tools, that have been surveyed in the above referred paper (Tavares et al., 2015b), increase the level of satisfaction and motivation of students. According to teachers and students, feedback should be provided and detailed as quickly as possible. These tools do not replace the teacher, but provide help and increase the value of time in the classroom. Teachers should be able to select the problems they intend to present to the students according to their level of difficulty (Verdú et al., 2011). Different teachers can adopt different strategies, depending on their specific goals and objectives of the course, especially of their own style and preferences (Joy et al., 2005). So, students must receive feedback at the right time to benefit from it. In our proposal we adopted Mooshak tool (Leal and Silva, 2008), a general purpose evaluation system used in many universities in our country.

According to teachers and students, the automatic feedback produced by those systems must be improved; it should be faster and more detailed to enable the improvement of the program quality, getting a nicer solution when the original one is poorly coded or too complex. This feedback is obtained from the execution of a series of tests, showing the results of each run, those who have passed and those who failed. This interactivity with the student tries to engage students in the course and so help improve learning and, consequently, reduce the failure (number of students unsuccessful).

4 AEV&ANIM AND ITS EXPERIMENTAL VALIDATION

In this approach, the student is exposed directly to the resolution with use of automatic evaluation and its feedback; and after that first individual trial, he has to analyze the correct solution using the animation tool. To adopt this approach, the teacher prepares for each subject to teach a number of problems relating to that topic with similar difficulty. For each problem of the set, asks the student to analyze the statement, develop the algorithm and code it, passing to the test it with the AES. After some time, the teacher provides his solution and asks the students to analyze carefully using the animation tool, looking to assimilate the knowledge derived from it.

To validate this, we prepared an experimental setting to get real feedback from its application in classroom. The main objectives for this first experiment were:

- Understand the behavior of students facing a new and different situation;
- Observe if students are involved and motivated;
- Understand the main difficulties faced by first year students, when they are engaged with a programming task: the interpretation of the

statement, the development of algorithms or their coding in a specific language;
- Check the effectiveness of the proposed approach.

This approach assumes that the teacher selects a powerful Animation tool, easy to use, and chooses an AES that is user-friendly and returns a feedback as complete as possible (with a diagnosis for the errors found). It is also desirable that AES comments the code quality. For our experiment, we chose Jeliot and Mooshak, as said above.

In our case, to teach the introductory topic "*sequential numeric processing and conditional and iterative control structures*" we wrote three exercise statements. After deciding the concrete tools to use, the topic of the experimental lesson, and the exercises to solve, it was necessary to write down a careful plan for the lesson, so that all the students enrolled could understand what they are asked to do and how should they proceed—as the experiment conducted in two classes with 25 students is not the focus of this paper we suggest the reading of the paper (Tavares et al., 2016a) for a detailed description and discussion of the results. However we report in the sequel a summary of the main outcomes: it is possible to affirm that the evolution of the behavior of the students during the class of two hours showed that this approach led to a better performance of the students. On one hand, it was noted that the number of students with accepted submissions increased and, on the other hand, that the number of submissions increased and that the number of compilation errors decreased. As students did not gave up soon as the first error appear (they keep searching for a correct solution) we concluded that their motivation has increased; a second effect of the increased effort is the number of base mistakes that have been reduced. Motivation was one of our main concerns. The experiment presented also allowed us to understand the best way to conduct future tests, for example, allowing greater flexibility in the management of time during the lesson. This means that we intend to propose the three exercises at the beginning and allow students to choose the time intervals to use in each of them; In this way, they can explore the animation more deeply if they find it important to sediment knowledge before progressing to a new implementation.

5 ANIM&AEV AND ITS AUTOMATION

While AEv&Anim, described in the previous section, is specially tailored to be applied in the classroom following the traditional teaching method, the next one, that we introduce in this section, is thought to be used in a self-study process, at home. In this approach, first the student is exposed to the analysis of the problem and its resolution, with the support of the Animation; then he goes on to a self-resolution phase using automatic evaluation and its rapid feedback. For each topic to be taught, the teacher will prepare two sets of similar exercises. For the exercises in the first set the teacher discusses the statement, the resolution (outlines an algorithm) and presents the program that solves it so that the student can make his animation and thus analyze / understand the solution.

For the exercises of the second set, after discussing the statement, the teacher asks the students to solve it and test the solution created through an automatic evaluation system. In a third moment, the teacher discusses with the students the feedback received from the evaluator.

Following Anim&AEv, a web-based information system (known as PEP, *Plataforma para o Ensino da Programação*) was developed to support the teacher in laboratory classes and, above all, provide students with the possibility of doing study sessions outside the classroom.

6 PEP SYSTEM

PEP platform will allow: (i) the teacher to carry and maintain the exercises (organized by topics and difficulties) to be used in each session, as well as to plan the sessions; (ii) the student runs one or more sessions to practice a particular theme, animating the exercises and then solving them and testing them with immediate feedback. PEP will also allow the teacher to receive back information about how each student's work session was performed (date and time, sequence of solved exercises, time spent, etc.). As can be seen in Figure 4, the system will consist of two main components: the Back-office (BO) and the Front-office (FO).

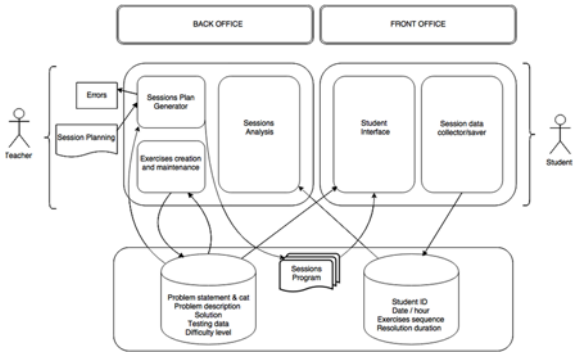

Figure 4: PEP System Architecture.

The first one supports all database management tasks with the questions that will later be used by the second module to construct the sessions that will be presented to the students. In the BO only the teachers will have access and it is from there that they can manage exercises (create, edit and delete), plan sessions, as well as analyze the sessions submitted by the students. In addition BO has two more essential components: the compiler that reads the formal specification of each session (written in a specific domain language, DSL, which we have created specifically for this purpose) and generates the necessary code for the FO to mount the sessions; and the analysis module that recover from the database the information related to each session of each student and presents it to the teacher so that it can follow the learning process.

The Front-Office is intended for students, where the sessions will be listed as well as saved the information about each session, i.e. the identification of the student who performs the session, the date, the sequence of exercises, as well as the duration. All this information will be stored in the database for later possible analysis.

In Figure 5 there is a diagrammatic representation of the interaction flow possible in the interface presented to the student.

A simple access control mechanism is implemented through the registration in the platform. After the choice of the session, the user will be confronted with two new options: part 1 that uses animation techniques; and part 2 supported by an automatic evaluation system to test students solutions (recommended only after completing part1).

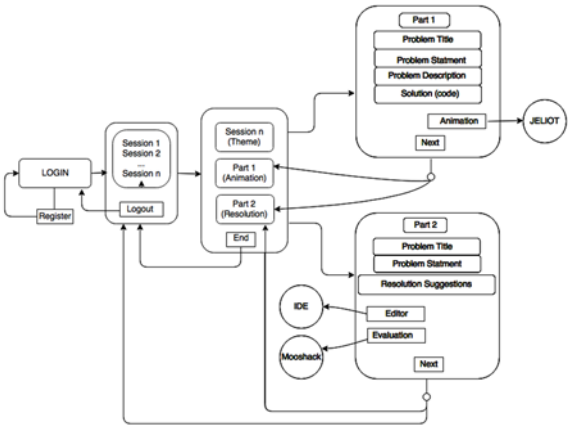

Figure 5: Interaction Flow Diagram.

If student choose Part 1 (Animation), he has access to a description of the problem as well as its solution (Java code). He shall use Jeliot tool to animate the resolution of the problem to a deep understanding of the solution provided. In part 2 (Evaluation), only a description of a problem to solve is presented to the student who is then asked to solve it. Here the Mooshak Automatic Assessment tool is used so that students can verify whether their resolution is correct or not.

PEP Front-Office is illustrated by the screenshots shown in Figures 6 through 9.

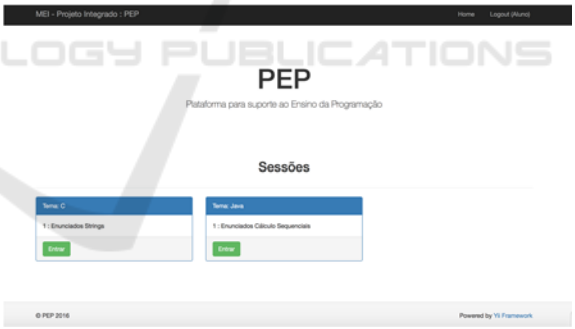

Figure 6: Listing course Sessions.

Figure 7: Access to Parts 1 and 2 of a Session.

Figure 8: Part1-navigation buttons to invoke the Animator.

Figure 9: Part1- Problem statement.

PEP is an easy to use tool, accessible as a Web application that can be used to increment motivation for self-study activities -- we strongly believe this will improve the overall learning process.

7 CONCLUSIONS

This paper discusses the need to find ways to improve the teaching / learning process in courses of introduction to programming. We studied and discussed methods to act as external factors in the extrinsic motivation and self-confidence of students who easily disinterest when faced with difficulties. In this sense, a combination of two existing tools has been proposed in order to get more a effective method: the animation of programs, to help understand how programs really behave to solve the problems; and the automatic evaluation of programs, to provide students with immediate feedback on the solutions they develop. Two approaches (AEv&Anim and Anim&AEv) were then presented that combine in opposite orders the two techniques mentioned. The first of these has already been the subject of an experimental analysis in the classroom, as described; the second led to the implementation of the Platform for Teaching Programming (PEP), which may be used to help the teacher in class or the student at home. As future work we plan to define new classroom experiments and also to study how PEP can be improved with gamification artefacts to increase students motivation.

ACKNOWLEDGEMENTS

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

REFERENCES

ACM/IEEE, 2013. Computer Science Curricula 2013 -- Curriculum Guidelines for Undergraduate Degree *Programs in Computer Science*, Final Report.

Almeida, D., 2012. A motivação do Aluno no Ensino Superior: um estudo exploratório. *Dissertação apresentada ao programa de Mestrado em Educação da Universidade Estadual de Londrina.*

Ben-Ari, M., Myller, N., Sutinen, E., Tarhio, J., 2002. Perspectives on Program Animation with Jeliot , *S. Diehl (Ed.): Software Visualization, LNCS 2269, pp. 31–45*, Springer-Verlag Berlin Heidelberg 2002.

Ben-Ari, M., Bednarik, R., Ben-Bassat, R., Ebel, G., Moreno, A., Myller, N., Sutinen, E., 2011. A decade of research and development on program animation: The Jeliot experience, *Journal of Visual Languages & Computing, Volume 22, Issue 5, October 2011, pp. 375-384*, http://dx.doi.org/10.1016/j.jvlc.2011.04.004.

Callahan, M., 2010. How Do I Motivate My Students? Teaching Resources, Texas Tech University.

Compos, E., Ramos, S., 2011. Teorias da Motivação. Dissertação do 2º Ciclo em Gestão de Recursos Humanos e Comportamento Organizacional (ISMT), pp. 125-142.

Hundhausen, C., Douglas, S., 2000. Using Visualizations to Learn Algorithms: Should Students Construct Their Own, or View an Expert's? *Proceedings 2000 IEEE International Symposium on Visual Languages IEEE Computer Society Press, Los Alamitos.*

Joy, M., Griffiths, N., Boyatt, R., 2005. The BOSS Online Submission and Assessment System. *Journal on Educational Resources in Computing, Volume 5 Issue 3,* September 2005.

Korhonen, A., 2003. Visual Algorithm Simulation. Dissertation for the degree of Doctor of Science in Technology. *At Helsinki University of Technology* (Espoo, Finland), November 2003.

Leal, J., Silva, F., 2008. Using Mooshak as a Competitive Learning Tool. *ACM-ICPC Competitive Learning Institute Symposium (CLIS 2008),* April 2008.

Moore, R., Lopes, J., 1999. Paper templates. In *TEMPLATE'06, 1st International Conference on Template Production*. SCITEPRESS.

Pereira, M., 2002. Systematization of Programs Animation, Sistematização da Animação de Programas. Dissertação submetida à Universidade do Minho para obtenção do grau de doutor em Informática, ramo Tecnologia da Programação, December 2002.

Proulx, V., 2000. Programming patterns and design patterns in the introductory computer science course. *Proceedings of the thirty-first SIGCSE technical symposium on Computer science education,* pp.80-84. New York.

Queirós, R., Leal, J., 2015. Ensemble: An Innovative Approach to Practice Computer Programming. In R. Queirós (Ed.), *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming* (pp. 173-201). Hershey, PA: Information Science.

Ramos, S., 2013. Motivação Académica dos Alunos do Ensino Superior. *Psicologia.pt, O portal dos Psicólogos.*

Ryan, R., Deci, E., 2000. Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *American Psychologist, Vol. 55, pp. 68-78.*

Santos, R., Costa, H., 2006. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. *INFOCOMP – Journal of Computer Science*, Lavras/MG – Brasil, volume 5, n. 1, pp. 41-50.

Silva, T., Mascarenhas, I., Medeiros, C., Sousa, E., 2014. A Motivação no Ensino Superior: Um Estudo com Alunos dos Cursos de Administração e Direito. *Journal of Management Analysis, Volume 3, pp.104-113.*Fortaleza.

Smith, J., 1998. *The book*, The publishing company. London, 2nd edition.

Tavares, P., Henriques, P., Gomes, E., 2015a. Animation and Automatic Evaluation to support Programming Teaching. *10ª Conferência Ibérica de Sistemas e Tecnologias de Informação, Vol. II, Simpósio Doutoral* (CISTI 2015), Portugal.

Tavares, P., Henriques, P., Gomes, E., 2015b. Animation and Automatic Evaluation to Support Programming Teaching. *7th International Conference on Computer Supported Education – Doctoral Consortium* (CSEDU 2015). Lisboa, Portugal.

Tavares, P., Henriques, P., Gomes, E., 2016a. Computer-Supported Techniques to Increase Students Engagement in Programming. *8th International Conference on Computer Supported Education* (CSEDU 2016). Rome, Italy.

Tavares, P., Henriques, P., Gomes, E., Pereira, MJ., 2016b. Técnicas para aumentar o Envolvimento dos Alunos na Aprendizagem da Programação. *VII Congresso Mundial de Estilos de Aprendizagem* (CMEA2016), Julho 2016, Portugal.

Verdú, E., Regueras, L., Verdú, M., Leal, L., Castro, J., Queirós, Q., 2011. A distributed system for learning programming online. Computers & Education 58, pp. 1–10.

Williams, K., Williams, C., 2011. Five Key Ingredients for Improving Student Motivation. *Research in Higher Education Journal*, pp. 104-122.