Methodological changes in teaching algorithms in the early years of the Computer Engineering course

Rita Carolina Alamino Borges da Costa*, André Vargas‡, Cléo Billa§
Rafael Santos†, Regina Barwaldt ‖ and Silvia Botelho¶
Centro de Ciências Computacionais, Universidade Federal do Rio Grande - Rio Grande, Brasil
Email: *ritaalamino@gmail.com, §cleo.billa@gmail.com, ‡prisco.c3@gmail.com
†rapennas@gmail.com, ‖reginabarwaldt@furg.br, ¶silviacb.botelho@gmail.com
,

***Abstract*—This Innovative Practice of a Full Paper presents new ways of teaching classes. The teaching methodology of the algorithm disciplines of the first years of Computer Engineering courses directly affects student performance throughout the course. The current education system is in transition, taking slow steps in new ways of learning. By promoting self-reflection and critical thinking, students can develop problem-solving skills and observe an application of the tools learned. A proposed methodology includes new structuring of the discipline "Algorithms and Data Structures I" at the Universidade Federal do Rio Grande (FURG), with the learning of a modern programming language and the transformation of the classic classroom model to new methods and dynamics. Computational thinking is fundamental at the beginning of the course. From this, all students receive a introduction course at the beginning of the college year to immerse in dynamics to practice logic. Classes began, students received an extension to the basic learning of the discipline (the fundamentals of algorithms and Python as a modern programming language) and at the same time they received an assignment, a problem resolution, where they had space and time to develop a game with the theme of their choice. The activity requires teamwork and creativity. As lectures run as workshops, allowing collaboration and project creation, with the help of teachers and older students as tutors. To complete the program, an event made available by the university was used as an environment, for visitors, scholars as well for high school juniors interested in software, to attend and learn about. As a preliminary result of the new implementation methodology, after 4 years, there was an increase in the approval and average grade of learners.**

***Index Terms*—education, algorithms, computing, teaching methodology**

I. Introduction

As technology evolves, increasingly automating processes and decreasing human interference in repetitive tasks, the need to adapt and innovate such technologies arises exponentially. The methodologies used in the Brazilian educational system have been becoming obsolete due to the inevitability of innovation. It is fundamental to improve these teaching methodologies and to to guide students on the path of creativity and innovation, especially on problem-solving skills.

One of the factors that influence students' dropout is the assessment methods and didactic-pedagogical deficiency of teachers [1]. The expected is the methodology change in teaching on a subject as important as Algorithms, motivates the classes to reduce dropouts, essentially seeking to avoid the lack of motivation created by unpreparedness and discouragement, that happens when there is a belief of an extremely hard obstacle as observed [2].

By joining a Computer Engineering degree, students are dedicated to comprehend programming logic in their first year. "Algorithms and Data Structures I" is a fundamental discipline at the beginning of the course that refers to the programming introduction. The initial goal is to explain the principles that form the institution so they can have a light to follow [3] and to train people so that they have consistency in computational thinking and the necessary attributions to advance in graduation [4].

These initial encounters for balance general knowledge are fundamental. Most of them have never had contact with any programming language, saving those who come from a technical school related to computing. Even with the technology advancement, exists a lack of knowledge in Brazil's high school about computer science. Most students from Brazilian public universities are low income or come from public school [5], of these less than half have a computer at home and more than a third do not have access to the internet [6]. No theory can be understood outside the complexity of social, structural and super structural relations, which constitute the set of conditions of production. [7]

With the basics of algorithms taught, scholars must develop problem-solving skills [8]. The project named "Gaming" has the purpose to allow students to practice what was being learned through the development of a game. The most common reaction when games are involved is joy and pleasure for the activity developed as mentioned [9], games can be efficient tools because they have fun while motivating, facilitating or learning and increase the ability to retention of what is taught, exercising as mental and intellectual functions of the player. Dolmans have noted that "problem-based learning stimulates students towards constructive and collaborative processes which influence learning positively" [10, p. 736].

The classes followed a specific structure. Class time was divided in two, first teachers demonstrated concepts and structures and then an workshop model class where they could create freely and in groups. Teachers and scholarship students was available for orientate the whole practice.

The act of producing knowledge is not the work of a singular conscience, but one of the forms of social practice. [7] The university built collaborative spaces [11] throughout the campuses that were used to provide monitoring for students who needed help with their studies and for those who wanted to develop any kind of work or exchange experiences with their colleagues.

To extend knowledge to society [12], the outcomes were presented at a university exhibition-fair that happens inside the university campus.

> "It is noticed that university extension as a form to establish a relationship between higher education and society it is essential to form citizens committed to the social reality" [14]

Scholars had their first-time contact with a "user" in their first year of graduation. The game presentation was also used for teachers as an evaluation method.

As Freire says in [7], it is very important to always ask students and understand their reality, in modern times data is used to have this answer in a way that approximates reality.

This strategy was developed to run as a cycle, developed in six steps, III-A Features and Approaches III-B Prepare for Training III-C Teach and Share III-D Focus and Create III-E Share with Community III-F Feedback and Learn. The purpose was to create a healthy environment for students to learn, connect with the community and all involved see it thrive. It evolves not just computational software interconnected with teaching, but a whole perspective-changing.

This paper describes a methodology that changes algorithms' ways of teaching, with an emphasis on critical thinking and practical theory. Compare grades and approbation parameters before and after the modifications and relate to the communities' and students' evaluation.

It is appropriate for college staff who aspire to collaborate to increase the motivation level of students, intend to decrease the number of dropouts and failed students, also to encourage critical thinking and get better results in teaching problem-solving skills. When allowing the community to participate and integrate, it works as a scientific divulgation.

II. Motivation Background

Learning in any area tends to focus on two types of knowledge, the declarative, which involves concepts and principles, and the procedural that refers to the use strategy, how to solve problems. Knowing a programming language is crucial to develop more complex reasoning, and can be used to develop problem-solving skills as argued in [15].

Linn and Dalbey [16] shows that there is an archetype chain of cognitive achievements to be achieved by teaching a programming language. Three principles make up the archetype: First an adequate training of single language resources through expository classes and workshops for development. Second, experience in project development through the receipt of a basic problem. Third is the acquirement of general problem-solving skills. They also show a better result in learning through practical activities of developing simulators and didactic visual tools for the representation of abstract concepts.

Group formation consists of learning to learn, as redefinition of learning models in which we were configured, group didactic recover for learning the social character of knowledge production. It allows the exchange of information, of vital experiences, the confrontation of learning styles. Information and experiences and styles can be processed and articulated in an enriching group synthesis for everyone and for each member. [17, p. 25]

The importance of teaching programming as seen in [9] reinforces the use of pedagogic strategies alongside with available technologies to instruct algorithms. The use of creating a game as a problem is due to the motivation factor.

This motivation factor, according to the systematic literature review made by [18], teaching-learning programming through digital games at colleges published in the last few five years, in relevant events and magazines in the area demonstrate how important it is to use games in teaching. Most of the articles analyzed shows that the games collaborates to improve student performance in programming disciplines, and the games have general acceptance from students.

In [19], it was investigated ways to increase students' interest in informatics and programming and make them able to develop simple games. It shows that in computing graduation courses, game developing is a factor that can greatly increase students' motivation and contribute to their training.

To make object-oriented programming introduction less abstract and theoretical and more motivating, [20] also using a literature review observed that games are a powerful tool to optimize the individual's cognitive development. Learning algorithms can be fun, and this methodology uses this motivating elements in different aspects.

Due to its simple syntax compared to other languages, [21] showed that using Python for teaching algorithms makes it easier to present concepts, being coherent with the initial strategies for algorithms teaching, making it fundamental to make this methodology more appropriate to the new way of teachings.

III. Methodology

This teaching methodology is divided into six steps. These steps create a cycle with the purpose to renew the environment and become self-sustainable.

A. Features and approaches

Teachers, students and visitors expose different points of view about the new routine at college in the first meeting among all new students. The expectations are balanced before they have training in computational thinking and programming logic.

The meeting must show students the institute principles, so they would direct their attention to the energy and relationships required to achieve the desired outcome, as [22] describes, when each person is trusted to work freely with those principles, to interpret them, learn from them, talk about them, a pattern of ethical behavior emerges.

The technology center can be seen as a dynamic system. Consider that there are individuals and groups working for multiples purposes which follow a few key principles that express the system's overall identity.

If we recognize that organizations are live systems, possessing the same capacity to adapt and grow that is common to all life [22] and as it is not possible to control every individual of the system, principles can be used as guides, guaranteeing freedom to express their creativity and following organizations goals.

B. Prepare for Training

These primary subjects such as computational thinking and programming logic were conducted as a workshop [16] that happens before the year's planned classes begin. The whole workshop organization was managed by students and had at least three tutors (volunteers) per class. A participation certificate for the students involved with the organization was guaranteed. The available resources in the class were developed to be appealing for all students, through dynamics and playful activities such as the use of games and challenges during activities. It is optional for the use of games like Scratch or block programming software.

C. Teach and Share

During the year, among the early month's classes, was illustrated the basic structures of algorithms using the workshop model [16] in the classroom. A modern and popular programming language was lectured. [21]

Laboratories with good computers were available, where classes happened. Students sat side by side in their computers. The class time was divided in two: The first part, teachers demonstrated concepts and structures with live examples so they could test in their machines. Exercises were given at the other half of the class, they were free to do exercises in groups, search on the internet, see videos or help each other. Three teachers and scholarship students was available for orientate the whole practice, answering questions and making critical thinking happen.

Collaborative spaces were created throughout the whole institute, provided with students that had already approved in the subject, to support those who prefer study groups, having tutors available at the times specified in those spaces. [7]

To employ students in practicing problem-solving skills online judges [13] were used to solve problems in real-time, as homework.

D. Focus and Create

As soon as the professors noticed that the students were more confident about the content learned, the problem [8], [16] was designated and had to be solved through the next months. The development was a game where the theme was a students' choice, it needed to be done in four months of class plus one month of university holiday. This time limit was necessary so the game could be ready for the final event.

Groups were formed of two or three students. All needed preview knowledge for the development of the game was demonstrated in class. The class after the release of the problem for game development followed the preview structure, reducing the time of presenting concepts: The encounter time was composed of one-third of the class, the professors presented concepts and structures, and the other part, the class was made as an workshop [16] composed by the professors and of scholarship students orienting the practice.

While creating the games, students were allowed to use their creativity choosing their themes. Professors closely followed learning, ensuring that the tools previously learned had been used.

E. Share with Community

By the end of the period of the task, the games were presented in an environment that is intermediate between high school students and citizens interested in knowing university affairs.

The students who developed the game, work on a schedule during the fair. They were oriented to present their games to visitors, explain what it is like to study at the university, to deliver a pamphlet talking about the courses.

Buses were available to bring young audiences (13 to 17 years old) still at school to see, learn and play. Scholars had their first-time contact with a user in their first year of graduation. Before leaving, visitors complete a survey about the games and the course. The game presentation was also used for teachers as an evaluation method.

F. Feedback and Learn

Survey was available to collect valuable data, that helped to understand better the needs and activity evaluation. The forms were sent in the year after the activity happened to students that participated, so they could evaluate better. Course coordination has to evaluate and analyze the metrics, compare with grades and assiduity, adapting actions as needed.

This cycle has a year duration, according with the following schedule the step III-A has a one-day duration following the step III-B with three or four days varying as needed and college calendar, starting classes by step III-C lasting a whole two months varying between step III-D in which the next four months follows getting done to III-E where the event happens allowing step III-F that can happen until the end of the year before the new cycle begins.

IV. Results

In a first moment, data were extracted from the college system, made available by course coordinators, from 2014 to 2019. Table 1 shows the number of newcomers students.

It was considered an analysis divided into two phases: before and after 2016 where the changes started. Another

TABLE I
COMPUTER ENGINEERING STUDENTS' NUMBER

Year					
2014	*2015*	*2016*	*2017*	*2018*	*2019*
45	52	48	47	50	47

consideration is 2018, due to students with a very low grade at SISU (Unified Selection System, a program that democratizes access to university), increased the first-year dropouts as expected [23].

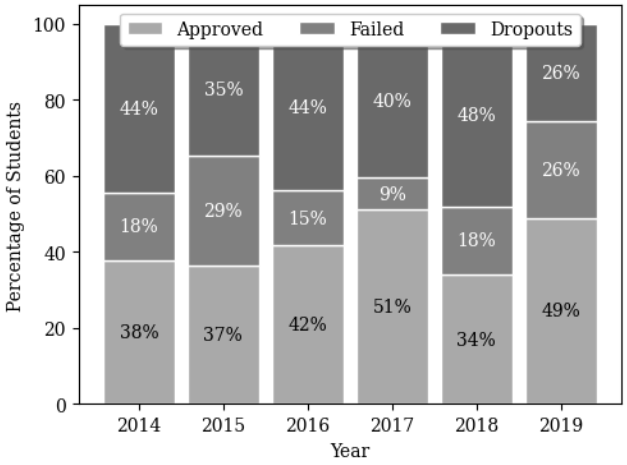

Fig. 1. Percentage of students approved, failed and dropouts from each class.

Analyzing the dropouts numbers in Figure 1, they did not have significant variation until 2018 when increased to 48% for the first time and showed a big drop to 26% in the next year. This shift happened only due to the implementation of the so-called "collaborative spaces" and close monitoring of students' performance that was implemented only in 2019. These 26% represents 12 dropouts from 2019, where 6 of these came from subsequent calls (when a student is not approved on the first call from SISU, he can be approved on subsequent calls), which lessens the likelihood of continuing on the course. [23]

The approvals number seen in Figure 1 shows a significant increase in the percentage, raising from 37% in 2015 to 42% in 2016 going to 51% in 2017 and dropping to 34% in 2018. Increasing again to 49% in 2019.

To illustrate the difficulty and probable inaccuracy of pointing out a single cause for the student's dropout [24], reproduces the phrase of T. E. Corts, former president of Samford Unviersity:

> "As there are certainly 50 ways to end a love affair, according to a popular song, there are also 50 ways and 50 reasons for a student to end his "love affair" with a college. Short-term campaigns to accommodate students meet momentary emergencies, but do not build long-term commitments. Some research indicates that students do not abandon colleges for big reasons, but because of the accumulation of small reasons that destroy their justifications for choosing an institution."

It is noted that there are countless factors that influence the student's decision to leave the institution [14], [24], some of the factors that are reached with these changes are:

- Irritation with the precariousness of the services offered by the institution.
- Disappointment with the little motivation and attention of teachers.

According to [24], there are points to decrease evasion, which were based on some successful examples released internationally:

I Establish a working group to reduce evasion.
II Assess evasion statistics.
III Determining the causes of evasion.
IV Stimulate the vision of the institution, focused on the student.
V Create conditions that meet the objectives that attract the students.
VI Making the environment at the institution pleasant to students.
VII Create student advice and guidance program.

Considering these points, it is clear that through the changes in the methodology, creating this new cycle, several actions and stages are mixed with the successful examples. Trying to understand the reasons for evasion is a complex activity, but taking into account the teachings of Paulo Freire where it is necessary to consider the entire context where students and teachers are involved, and comparing with modern applications, we see an optimistic advance in relation to numbers approved which grows in a non-linear manner.

In the creation of a new ecosystem, the first years are of structuring and trust in the system, the data of the works already carried out make the new methodology a favorable possibility for institutions and coordinators that intend to take risks with new changes.

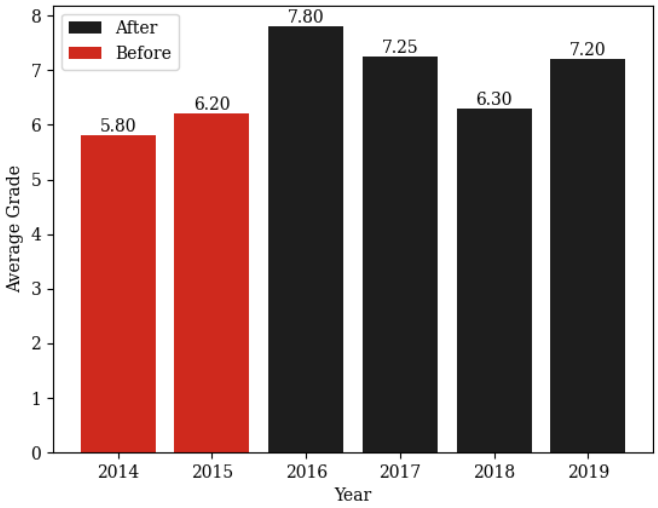

Fig. 2. Average of students grades from each class.

Figure 2 represents students' final average grade over the years, it does not include the dropouts. Before the changes its seen an average grade of 5.8 with deviation 2.66 in 2014 and 6.20 points with deviation of 3.24 in 2015. After the changes the average was 7.8 with standart deviation of 2.46 in 2016, the average of 7.25 with standart deviation 2.42 in 2017, 6.30 and deviation of 2.8 in 2018 and 7.20 with deviation of 2.59 in 2019.

The determinants of student grades vary from study time, context, skill, motivation, organization. Even if it were tried, it would be complex, and by the time we succeed, it may have become obsolete. So how to adapt the student evaluation in this dynamic reality?

Considering that [25] reports that one of the main influences on the student's grade, the study time, this new methodology through the use of adapted classroom model, using current subjects that interest students and making it fun together with the development of something practical that the result can be observed by them in the short term, increasing motivation levels (as can be seen in table III), it can be related to the non-linear increase in grades.

The new methodology is very recent, and it is fundamental a persistence in its application during a long period of time so that its real effects are noticed. Until then, this slight optimistic non-linear growth encourages continuity.

TABLE II
COMMUNITY ASSESSMENT

Question Number	**Option**			
	1	***2***	***3***	***4***
Q1	2.9%	8.9%	47.6%	40.4%
Q2	12.9%	37.1%	43.6%	6.4%
Q3	29.6%	2.2%	27.8%	40.4%

Reviewing the community rating, 285 visitors passed through the workshop over these four years (2016-2019), responding to a quick survey. The questions were (Q1) "What did you think of the games?" [4 Loved It — 3 Liked it — 2 Nothing big — 1 Not the expected] (Q2) "Do you understand the difference between creating a game and playing it?" [4 Yes, and I know how to program — 3 Yes — 2 I think so — 1 I really don't know] (Q3) "Did the open week make you consider doing Computer Engineering?" [4 Began to consider — 3 Already considered — 2 Used to consider, but now not anymore — 1 Did not want nor changed my opinion]

TABLE III
EVALUATION BY STUDENTS

Question Number	**Option**				
	1	***2***	***3***	***4***	***5***
Q1	-	7.5%	18.9%	26.4%	47.2%
Q2	3.7%	5.6%	25.9%	27.8%	37%
Q3*	18.5%	13%	31.5%	25.9%	11.1%
Q4*	9.3%	5.6%	20.4%	29.6%	35.2%
Q5	3.7%	7.4%	14.8%	22.2%	51.9%

Analyzing the students' assessment of the activity, questions were asked that could be answered on a scale of 1 to 5. Where 1 would be [Terrible / It didn't help at all*] and 5 would be [Great / It helped a lot*]. The questions were. (Q1) "How do you describe the experience of creating games in your first year?" (Q2) "How do you describe the experience of having contact with the customer and presenting your created product?" (Q3) "Analyzing this experience in general with Gaming, did it help you to get a better idea of the area of work of a computer engineer?" (Q4) "Did this activity help to increase your level of motivation in any way?" and (Q5) "How do you evaluate the methodology of learning the python language in the first year of the course? (Independent of didactics) ".

Table 3 shows most of the students had a positive experience participating in the creation of a game (73,6%), using Python in their first year of college (74,1%) and presenting it to people who actually played it (64,8%). It also shows to be inconclusive whether students find developing games in their first year to be what a computer engineer does in their career, which is expected because they are in their first year. When talking about motivation more than a half found the activity increased their level of motivation in some way (64,8%).

Table 2 shows when speaking about community most visitors had a positive experience testing the games (88,3%) furthermore when asked if they understood the difference between creating a game and playing it, 50% claimed to know the difference. About whether the visit made them consider to join a computer engineer degree, 40,4% of visitors began to consider, 27,8% already considered it.

V. CONCLUSIONS

Learning algorithms at 21th century is very important, as having critical thinking and problem-solving skills. These results confirm an optimistic scenario change due to a cycle that has been created, indicating that the new methodology, together with constant monitoring of students and specific actions (such as using the monitors and retaking tests if needed) gives indications that it is possible to reduce evasion.

As focusing on achievement and learning, it is elementary to make available institutional resources, bring staff together to analyze data, have feedback and learn, search for new perspectives, expand the web and support more local efforts and innovations. [3]

This presented methodology is an idea of how it can be done, results may vary, but when seeking evolution, long-term growth is certain. Especially in institutions that may be experiencing obstacles in their development. We believe in a world where education is accessible to everyone. Students are the future of this world we hold. If we can inspire them, we can create an infinite cycle of prosperity.

In practical means, course coordination needs to be willing and determined to generate new changes regardless of the variation in results. Always looking forward to creating reasonable changes.

For scalability to happen, it is essential that a well-paid scholarship candidates that guarantees the occurrence of technological and cultural integration events, creates campaigns

so that all differences are respected, continuously. We cannot count on the likelihood that the ecosystem will willingly depend on it to continue functioning with stability and balance.

REFERENCES

[1] C. Matta, S. Lebrão and M. Heleno, *"Adaptação, rendimento, evasão e vivências acadêmicas no ensino superior: revisão da literatura"/*, Psicologia Escolar e Educacional, vol. 21, no. 3, pp. 583-591, 2017. Available: https://www.scielo.br/scielo.php?script=sci_arttextpid=S1413-85572017000300583lng=pttlng=pt. [Accessed 17 May 2020].

[2] R. Santos and H. Costa, "Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática", *Semanticscholar.org*, 2006. [Online]. Available: https://www.semanticscholar.org/paper/An%C3%A1lise-de-Metodologias-e-Ambientes-de-Ensino-para-Santos-Costa/0dcb8872c1d8793e347a298ea45d876febcf33bd. [Accessed: 04- Apr- 2020].

[3] M. Wheatley, *"Leadership and the new science."* Readhowyouwant.com Ltd, 2011.

[4] J. Wing, *"Computational Thinking"*, Cs.cmu.edu, 2006. [Online]. Available: https://www.cs.cmu.edu/ 15110-s13/Wing06-ct.pdf. [Accessed: 17- May- 2020].

[5] "Maioria dos alunos das universidades federais tem renda baixa – Andifes", *Andifes.org.br*, 2020. [Online]. Available: http://www.andifes.org.br/maioria-dos-alunos-das-universidades-federais-tem-renda-baixa-e-parda-ou-preta-e-vem-de-escola-publica/. [Accessed: 11- Apr- 2020].

[6] "Survey on the Use of Information and Communication Technologies in Brazilian Households", *Cetic.br*, 2018. [Online]. Available: https://www.cetic.br/media/docs/publicacoes/2/12225320191028-tic_dom_2018_livro_eletronico.pdf. [Accessed: 11- Apr- 2020].

[7] P. Freire and P. Riviere, *"O processo educativo"*, Acervo.paulofreire.org, 2020. [Online]. Available: http://www.acervo.paulofreire.org:8080/jspui/bitstream/7891/1622/3/FPF_PTPF_12_009.pdf. [Accessed: 19- May- 2020].

[8] D. Wood, *"ABC of learning and teaching in medicine: Problem based learning"*, BMJ, vol. 326, no. 7384, pp. 328-330, 2003. Available: 10.1136/bmj.326.7384.328 [Accessed 25 May 2020].

[9] C. Rapkiewicz, G. Falkembach, L. Seixas, N. Rosa, V. Cunha and M. Klemann, "Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais.", *RENOTE*, vol. 4, no. 2, 2006. Available: 10.22456/1679-1916.14284.

[10] D. Dolmans, W. De Grave, I. Wolfhagen and C. van der Vleuten, "Problem-based learning: future challenges for educational practice and research", *Medical Education*vol. 39, no. 7, pp. 732-741, 2005. Available: 10.1111/j.1365-2929.2005.02205.x.

[11] P. Lippman, *"Designing collaborative spaces for schools"*, T.H.E. Journal, vol. 2015, pp. 39-41, 2015. Available: http://thejournal.com/articles. [Accessed 21 May 2020].

[12] O. Silva, *"O que é extensão universitária?"*, Ecientificocultural.com, 1997. [Online]. Available: https://www.ecientificocultural.com/ECC3/oberdan9.htm. [Accessed: 23- May- 2020].

[13] J. L. Bez, N. A. Tonin and P. R. Rodegheri, *"URI Online Judge Academic: A tool for algorithms and programming classes"*, 2014 9th International Conference on Computer Science Education, Vancouver, BC, 2014, pp. 149-152, doi: 10.1109/ICCSE.2014.6926445.

[14] A. Nunes and M. Silva, *"A extensão universitária no ensino superior e a sociedade"*, Mal-Estar e Sociedade, vol., no. 7, p. 130, 2020. [Accessed 25 May 2020].

[15] D. Palumbo, "Programming Language/Problem-Solving Research: A Review of Relevant Issues", *Review of Educational Research*, vol. 60, no. 1, pp. 65-89, 1990. Available: 10.3102/00346543060001065.

[16] Linn and J. Dalbey, "Cognitive consequences of Programming Instruction: Instruction, Access, and Ability", *Educational Psychologist*, vol. 20, no. 4, pp. 191-206, 1985. Available: 10.1207/s15326985ep2004_4.

[17] M. Gadotti, "A escola eo professor: Paulo Freire e a paixão de ensinar", Acervo.paulofreire.org, 2020. [Online]. Available: http://acervo.paulofreire.org:8080/jspui/bitstream/7891/2773/1/FPF_PTPF_12_026.pdf. [Accessed: 18- May- 2020].

[18] T. Jesus Medeiros, T. Reis da Silva and E. Henrique da Silva Aranha, "Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura", *RENOTE*, vol. 11, no. 3, 2014. Available: https://seer.ufrgs.br/renote/article/view/44363. [Accessed 4 April 2020].

[19] A. Rebouças, D. Marques, L. Costa and M. Silva, "Aprendendo a Ensinar Programação Combinando Jogos e Python", *Semanticscholar.org*, 2010. [Online]. Available: https://www.semanticscholar.org/paper/Aprendendo-a-Ensinar-Programa%C3%A7%C3%A3o-Combinando-Jogos-e-Rebou%C3%A7as-Marques/c21636b6b203ba558d4c11bb7b2556f74df0c7d4. [Accessed: 04- Apr- 2020].

[20] P. Moratori, "Por que utilizar jogos educativos no processo de ensino aprendizagem?", *Nce.ufrj.br*, 2003. [Online]. Available: http://www.nce.ufrj.br/GINAPE/publicacoes/trabalhos/t_2003/t_2003_patrick_barbosa_moratori.pdf. [Accessed: 04- Apr- 2020].

[21] L. Begosso, "An approach for teaching algorithms and computer programming using Greenfoot and Python", *Frontiers In Education* 2013. Available: 10.1109/FIE.2012.6462307 [Accessed 04 April 2020].

[22] M. Wheatley, "Margaret J. Wheatley: How Large-Scale Change Really Happens: Working with Emergence", *Margaretwheatley.com*2007. [Online]. Available: https://www.margaretwheatley.com/articles/largescalechange.html. [Accessed: 08- Apr- 2020].

[23] C. Machado and C. Szerman, "Centralized Admission and the Student-College Match", *SSRN Electronic Journal*, 2016. Available: 10.2139/ssrn.2844131.

[24] M. B. C. M. Lobo, *"Panorama da evasão no ensino superior brasileiro: aspectos gerais das causas e soluções.*", Associação Brasileira de Mantenedoras de Ensino Superior, Cadernos 25, 2012. [Accessed 25 May 2020].

[25] J. Michaels and T. Miethe, *"Academic Effort and College Grades"*, Social Forces, vol. 68, no. 1, p. 309, 1989. Available: 10.2307/2579230.