

A Contextualized Spiral Approach for Teaching Programming in IT Vocational Secondary Education

Original Title: Uma Abordagem em Espiral e Contextualizada para o Ensino de Programação na Educação Profissional em Informática

Luis Gustavo J. Araujo¹, Roberto A. Bittencourt¹, David Moisés B. Santos¹

¹ State University of Feira de Santana (UEFS) – Feira de Santana, BA – Brazil

ARTICLE INFO

Article history:
Received 15 March 2018
Accepted 25 October 2018
Available online July 1st 2018

Keywords:
Contextualized learning
Computational thinking
Programming
Spiral learning

ISSN: 2595-9077

DOI: JCThink.2018.v2.n1.p.81

ABSTRACT

INTRODUCTION: Brazilian vocational high school education in Computing faces problems such as course failure and dropout, plus dropout from the programs, factors that directly contribute to the deficit of such professionals. **OBJECTIVE:** This paper reports an experience of a contextualized and spiral approach carried out with students of the technical program in Informatics. **METHOD:** This work uses contextualized programming with tools suitable to novices. We used different contexts in three different modules and applied a survey after each module to measure motivations and learning with 12th grade students. **RESULTS:** Learned lessons suggest the adequacy of Python for students, and of contextualized activities, context reuse and challenges as factors contributing to motivation and learning. **CONCLUSION:** The use of contexts, languages and environments in sequential and spiral modules fosters a more active posture of students regarding learning activities. Results from the survey show that this approach is partially suitable and improves student motivation.

RESUMO

INTRODUÇÃO: A modalidade do Ensino Profissional na área de Informática enfrenta diversos problemas tais como reprovação e evasão em disciplinas, além de abandono dos cursos, fatores que impactam diretamente no déficit de profissionais nesta área. **OBJETIVO:** Este artigo relata uma experiência de uma abordagem contextualizada e em espiral realizada com estudantes do curso técnico em Informática na modalidade Integrada. **MÉTODO:** Este trabalho utiliza programação contextualizada com ferramentas adequadas aos novatos. Nós utilizamos diferentes contextos em três diferentes módulos e aplicamos questionários após cada módulo para avaliar a motivação e aprendizado dos estudantes do terceiro ano do ensino médio. **RESULTADOS:** As lições aprendidas sugerem a adequação de Python como linguagem para iniciantes, de atividades contextualizadas, de reuso de contextos e desafios como fatores que contribuem para a motivação e a aprendizagem. **CONCLUSÃO:** O uso de contextos, linguagens ou ambientes em blocos sequenciais e em espiral possibilita uma postura mais ativa dos estudantes em relação às atividades de aprendizagem.

1. Introduction

Vocational secondary education in Brazil faces several problems related to student permanence as well as technical workforce retention in information technology (IT) (SOFTEX, 2012). The SOFTEX report points out that students attending vocational education integrated with high school continue their studies in different fields, instead of working as IT technicians. This ultimately leads to a shortage of professionals in the field of IT.

Besides the lack of interest after training, vocational programs suffer from high levels of dropout and failure in subjects related to mathematics and programming (Cravo, 2012; Machado, Moreira, & Priscila, 2006). Regarding programming, various issues relate to the difficulties to learn coding by using a traditional teaching approach where content is presented in lectures and students must solve drills. Those drills are

usually dissociated from the students' lives, which leads them to consider programming as something tedious and overly technical (Guzdial, 2003).

What is usually missing is considering students' profiles and bringing contexts that interest them into class. Vocational secondary students increasingly interact in social networking platforms, immerse themselves in complex games and use image and music applications. As Prensky states, those students have already been born in the midst of a technological whirlwind, and interact daily with those media (Prensky, 2001).

Researchers have developed several tools and languages to allow adopting those contexts in the teaching of programming and, at the same time, to meet the needs of the novice programmers. Examples of these tools are: Scratch, Logo, Kturtle, App Inventor, Lego Mindstorms, Jython Environment for Students (JES) and PPlay (Bordini et al., 2016; Guzdial, 2003; IC-UFF, 2017). Together with those tools, various studies have used animations and stories, simulations, robotics, games (Marques, Costa, Silva, & Rebouças, 2011; Rebouças, Marques, Costa, & Silva, 2010) and media (Araujo, Bittencourt, & Santos, 2017; Santana, Figuerêdo, & Bittencourt, 2017) to motivate students and reduce those difficulties. Such actions potentially contribute to student learning, increase the number of graduates and encourage students to continue in their field of training.

Despite the variety of initiatives, little has been done on the use of media in Brazilian vocational secondary education. In this scenario, we realized the opportunity of exploring the media context in association with an educational approach aimed at reducing cognitive overload and presenting concepts in a phased way, such as in the spiral curriculum (Bruner, 1966).

This work reports a teaching and learning experience of programming developed in a 12th grade class in an Informatics vocational program in the modality integrated with high school. To improve students' motivation and learning, the approach is based on the contextualization of activities, the use of suitable environments for novices and a spiral approach. In this approach, we present content more than once, at increasingly complex levels, through contextualized modules that associate a context with a tool.

2. Related Work

The use of specific contexts in programming teaching is prevalent in the literature. Several papers present the use of games, robotics, automation and simulation, among others (Bordini et al., 2016). In this work, we use two contexts in particular: games and media.

Game development is a context used in a course for teaching programming (Rebouças et al., 2010). The authors note that the Python language, combined with the pyGame library – a library for game development, offers an effective combination for the use of games to teach programming.

Marques et al. present the execution of a workshop in high school with the development of games as a motivational element to attract students and increase interest in computing (Marques et al., 2011). The authors conclude that the workshop contributed to increase the interest of high school students in computing and that participants were satisfied with the approach. It is worth mentioning that the workshop was motivating even for students who previously did not have direct contact with the computer. Such previous work is relevant but lack a more rigorous approach for evaluation, such as we did with the CIS survey, which is based on a motivation model and motivation theories.

In Brazil, there are very few papers reporting the use of media (i.e., manipulating images and sounds) for teaching programming in vocational secondary

education. However, we can cite previous work developed in other modalities, such as an experience the authors of this paper carried out in basic education (Araujo et al., 2017). This is a case study with the ninth grade, middle school students, and all programming novices. The study uses a spiral approach, divided into two contextualized modules, using geometric figures with the Turtle library and images with the JES environment, both with the Python language. The results show high motivation among students, especially in the dimensions of attention and confidence. The work presents significant changes in students' perceptions of how programs work, what programmers do, and the fun character of computing and image manipulation.

Media are also used in an approach in higher education, in a Civil Engineering program, in the Introduction to Computer Science course (CS1) (Santana et al., 2017). This case study investigates how an approach with appropriate environments for novices and contexts close to them can influence motivation. The approach is split into three units, respectively using Scratch, Turtle (Python) and JES in each unit. Although those works assessed motivation with both IMMS and CIS surveys, they are used in educational stages different from vocational technical education.

In addition, other works outside Brazil have used media in teaching programming for both Computing major (Simon, Kinnunen, Porter, & Zazkis, 2010) and non-major undergraduate students (Guzdial, 2003, 2013). Authors notice that the use of the Python language, besides offering suitable support to the novices, allows the adoption of diverse contexts, like games and media, which are relevant to students' lives. Those are useful experience reports, but, in general, do not use more than one context with contextualized programming, neither measure motivation levels.

3. Methodology

This section describes the used language and tools, participants, intervention planning, and the procedures for data collection and analysis.

3.1. Language and Tools

Python is a language suited to the needs of novice programmers, especially in vocational education, because it is not only well suited to teaching but it also popular and adopted in industry. According to Grandell et al., Python is a high-level language, originally created to facilitate learning, and initially suggested as an alternative to teaching programming by its creator, Guido Van Rossum (Grandell, Peltomäki, Back, & Salakoski, 2006). Python has various features to make it suitable for programming education, such as clear and reduced syntax, dynamic types, expressive semantics, immediate feedback, structure reinforcement, and relevant open-source projects.

JES – Jython Environment for Students – is an environment developed by Georgia Tech for teaching programming through media manipulation (Guzdial, 2003). JES has modules for manipulating images, sounds, videos, websites and a Turtle module, all in Python. Its functions are simple, to allow manipulating those media with a few lines of code, which allows novice students the development of relevant activities such as images effects, figure creation, sound creation, sound effects, video edition, and development of web sites.

PPlay is a framework developed by Universidade Federal Fluminense for teaching programming in Python through game development (IC-UFF, 2017). PPlay uses pyGame as basis. However, unlike this one, PPlay has a focus on learning and, for this reason, has simpler commands, aiming to minimize the cognitive overload during the learning process.

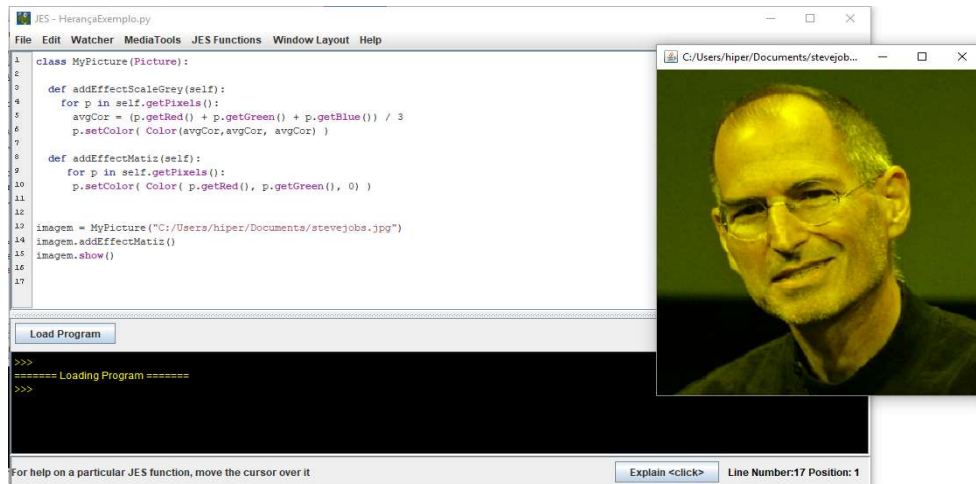


Figure 1. JES Interface

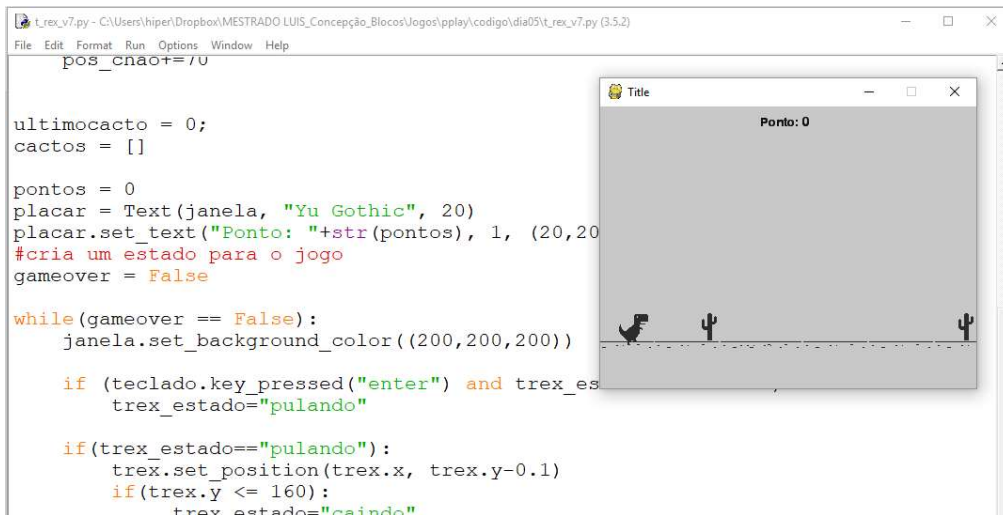


Figure 2. Python IDLE Interface running the T-Rex game made using PPlay

3.2. Participants

The study presented here was carried out in a vocational education school in the city of Feira de Santana, Bahia, in a class of the technical program in Informatics. The course was offered to 12th grade students, in their third year in vocational education, within the Programming Language 2 (PL2) course. Since those students had not previously learned programming in their previous course of Programming Language 1 (PL1) for issues beyond our knowledge, we consider them as novice students. This course intends to teach object oriented programming (OOP) concepts and skills. There were 10 students in class: 6 male and 4 female, with a mean age of 16.5 years (standard deviation of 0.9), and with previous experience of programming. The first author of this paper taught the course, aided by an undergraduate assistant, while the regular teacher followed the classes.

3.3. Class Planning

The proposed approach aims to teach programming concepts, seeking to facilitate learning and increase student motivation regarding programming. Therefore, we use tools and languages appropriate to these students and a spiral learning approach (Bruner, 1966). We planned to use the following tools: JES, for editing images, and

PPlay, for creating games, both using the Python language. We split the planning into three modules. Table 1 gives an overview of the approach.

The three modules that make up this study present concepts of OOP and programming repeatedly, at deeper and more complex levels. For instance, a simple selection structure (if) can be presented at first; then, a chained selection structure (if-else, if-elif-else); and, later, a nested selection structure and more complex forms combining relational and logical operators.

Place	Jair Santos Silva State School, Feira de Santana, Bahia, Brazil
Participants	10 Students
Modules	Images I – Image effects using OOP concepts
	Images II – Image effect deepening OOP concepts
	Games – Game Development
Content	Function, Parameters, Select and Repeat Structures, Variables, Arithmetic and Relation Operators, Vectors, Matrices, Classes, Objects, Fields, Methods, Inheritance, Composition, Method Overload, Method Overwriting
Tools	Python language, JES and PPlay
Duration	57 hours (3h per week)
Period	From April 11, 2017 until August 29, 2017

Table 1. Planning Synthesis

The first module presented general programming concepts (arrays, selection and loop structures) and OOP concepts (classes, objects, fields and methods). In the second module, we reintroduced the OOP concepts to enable not only the use of classes and objects, but also the creation of classes, as well as the use of inheritance, composition, constructors and method overloading and overwriting. Tables 1, 2 and 3 present the modules associated with their goals, content and activities for each class. The materials used in the approach are available on the support website¹.

3.4. Data Collection and Analysis

In parallel to this intervention, an exploratory research was carried out. Their results are beyond the scope of this experience report, but the qualitative and quantitative data collection and analysis instruments used in the research supported the writing of this report.

The procedures for collecting quantitative data were: i) pre-intervention survey; ii) post- module surveys; iii) post-intervention survey. Thus, at the end of each module, we surveyed students in order to measure changes in their concepts and motivation. After the whole intervention, we used a post-intervention survey. The quantitative results were analyzed through descriptive and inferential statistics. We also used data

¹ http://luisaraujo.github.io/programacao_com_midias/poo

from the written exam and the source code developed in classroom. All 10 students answered the surveys.

To collect qualitative data, we used: i) semi-structured interviews with students; ii) semi-structured interview with the teacher; iii) logbooks and iv) in-class observations. We performed qualitative analysis by using content analysis, starting with open coding, followed by axial coding and, finally, writing analytical memos to facilitate interpretation of the results.

Day	Goals	Content	Activities
1	Building knowledge on digital images, objects, classes and applying them in the JES environment.	RGB Standard, Digital Images, Classes, Objects, JES functions (get/setRed, get/setGreen, get/setBlue, pickAFile, getPixel, setColor), JES classes (Color, Picture, FileChooser).	Creating geometric figures (square, triangle and circle).
2	Building knowledge on arrays and repeat structures.	Decomposition, code function, arrays, loops for, functions in, range and JES functions (write).	Creating geometric figures (square, triangle and circle) and gradient effect with loops.
3	Applying knowledge of arrays and repeat structures to manipulate images.	Arrays, loops for, Arithmetic Operators, already presented JES functions and getPixels.	Creating image effects: hue, negative and greyscale.
4	Building knowledge on selection structures and applying them to manipulate images.	Select structure (if), Relational Operators and already presented JES functions, getWidth and getHeight.	Creating image effects: half image and black and white.
5	Applying knowledge built during the course of the module.	All previously presented content.	Creating Chromakey effect with green background.
6	Applying knowledge built during the course of the module.	All previously presented content.	Written exam.

Table 2. Planning for LP2 Classes – Module I

Day	Goals	Content	Activities
1	Deepening knowledge on objects and classes.	OOP concepts , Classes and Objects.	Creating a calculator using classes and objects, Creating classes for image effects.
2	Building knowledge on constructors, parameters and deepening knowledge on select structures.	Constructors, Parameters, Methods e Select Structures.	Remodeling image classes using parameters and methods to create effects using select structures.

3	Building knowledge on composition and applying them to manipulate images.	OOP concepts (Composition), Select and Repeat Structures, and already presented JES Functions.	Using class composition to create image effects.
4	Building knowledge on inheritance and applying them to manipulate images.	OOP concepts (Inheritance), Select and Repeat Structures, and already presented JES Functions.	Creating a class that inherits methods and fields from Picture class.
5	Building knowledge on input and output and applying them to manipulate images.	Inheritance, JES Functions (addText), Repeat Structures, I/O instructions.	Creating a class to create Memes.
6	Applying knowledge built during the course of the module.	All previously presented content.	Written exam.

Table 3. Planning for LP2 Classes – Module II

Day	Goals	Content	Activities
1	Deepening knowledge on objects, classes and methods and applying them in PPlay.	Objects, Classes, Methods and PPlay Functions (Sprite, Window), Game Loop with while.	First part of the Pong game, creating ball and bar.
2	Applying knowledge on select structures and use of variables in games.	Simple and Nested Select Structures, Relational Operators and Variables.	Adding score, win condition, and exhibiting score.
3	Deepening knowledge on repeat structures and use of simple data structures.	Repeat structures, Arrays, Relational Operators and PPlay Functions (Animation).	First part of the T-Rex game.
4	Applying and deepening knowledge on OOP	Class Creation with Fields, Methods and Constructors.	Refactoring commands, creating classes, fields and methods in the T-Rex game.
5	Refactoring code, using inheritance and composition.	Inheritance and Composition.	Organizing classes aiming code reuse.
6	Applying knowledge built in projects of free choice.	All previously presented content.	Creating a game of free choice.
7	Applying knowledge built during the course of the module.	All previously presented content.	Written exam.

Table 4. Planning for LP2 Classes – Module III

4. Results

In the first contextualized module, the teacher presented the JES environment and how to edit images through programming. First, he presented the main items of the JES GUI and presented how to use the initial functions (to get an image and display it). This initial moment is quite expositive and students repeated the commands used by the teacher.

Although images are close to the students' lives, the teacher needed to present concepts related to image representation in the computer. Thus, concepts about RGB and the image as an array of pixels have been addressed. Students showed ease with these elements and with the initial functions.

As a first activity, students were asked to modify some pixels of an image. Then they were asked to create some geometric figures like squares, triangles and circles. In this lesson, some students, who had already used Python, used repetition structures to generate the images, even though this was a subject of the following class.

On the second day, students used the code from the first lesson to decompose it and add repeat structures, and created gradient shapes using mathematical formulae. Students had no difficulty in using simple mathematical concepts such as sum or subtraction within a repetition structure. This activity allowed students greater freedom to modify their codes, generating several gradient effects. Two students added a random function, generating a gradient with points.

On the third and fourth days, students went from creating images on an initially white screen to manipulating previously constructed images. An image of Steve Jobs was used for manipulation – the lab does not have access to the internet, which made it impossible to use images chosen by the students themselves. Students presented more difficulties to create effects that demanded the use of mathematical formulae such as average – for the Grayscale effect – or subtracting a value from 255 (maximum RGB value) to generate the Negative effect. As effects were presented in theory, sometimes students would gather to find their way to produce the effect. One notices that these activities arose more student satisfaction than easier activities. The challenge has somehow driven students to solve activities in class. The activities also awakened a collaborative environment. Students who finished them early helped their colleagues, when requested.

In the fifth day, students created the chroma key effect. The teacher initially explained the effect, omitting details on how to produce it on the computer. Then they were asked to implement it. Most of the students were able to finish the activity within classtime.

The second contextualized module continued with image manipulation. However, the teacher sought to deepen OOP concepts. Thus, on the first day, students were introduced to the possibility of class creation. Since creating image-related classes requires knowledge of other OOP concepts, we started with an example of a calculator as a class and the four arithmetic operations as methods. After this activity, the teacher presented how to create classes for each of the effects already seen.

In the second class, students grouped classes into one class and used conditionals and parameters to define the effect to be applied. Students had no difficulties to implement these initial activities. However, the case-sensitive text in Python was a catch, generating errors by using uppercase rather than lowercase letters and vice-versa.

In the third class, students learned about composition by modifying the class they had previously coded and using a JES Picture type field. Thus, they used the field both inside and outside of the class, through an object.

In the fourth class, students learned about inheritance and used this concept in a new class that inherited Picture attributes and methods. This class, called MyPicture, had methods for applying effects. Thus, students implemented the necessary changes for inheritance. Students had difficulties using the *self* operator, but they were able to carry out the activity with the help of the teacher or the assistant.

In the fifth class, students continued to use inheritance and created a Meme class, where text is added to an image when instantiated. To do so, they needed to create a constructor, call the constructor of the parent class, use the *self* operator, parameters and mathematical operations for positioning the text. This activity was the most complex and the one with less instruction, similarly to the chroma key. During the activities, some students explored the tool, discovered JES features, or added commands they already knew, such as *random*. Although JES material is predominantly in English, some students, all of them speaking Portuguese as their native tongue, have found tutorials on the internet and coded their solutions from them.

Despite students' engagement in class, they did not usually study at home or did homework. However, the activities were satisfactory both in terms of relevance as well as capable of holding students' attention. Challenge solving, facilitated by Python and JES, provided students with satisfaction and confidence. The spiral approach has shown as an element that positively affects students' learning for not frustrating them, by introducing content gradually and at increasing levels of complexity. This makes it possible to learn concepts and accomplish the proposed challenges and activities.

4.1. Post-Intervention Survey on Motivation

We performed a post-intervention survey with students using a validated instrument for measuring motivation on classroom education. This instrument, called Course Interest Survey (CIS), measures motivation in terms of four categories: Attention, Relevance, Confidence and Satisfaction (Keller, 2010). Here we present the results of the survey in detail. When reading results, we group partial and full agreement simply as agreement, and partial and full disagreement simply as disagreement.

Figure 3 describes student attention in class. Results show that 77% of the students agree that the teacher knows how to enthuse the students (A1). 88% agree that the teacher uses a variety of teaching techniques (A6). 66% of students assert that the teacher does something unusual and amazing (A5). Only 22% claim to daydream in class (A7). However, only 22% agree that curiosity was stimulated in class.

Figure 4 describes the relevance of the course for the students. We notice that 77% of students agree that the teacher makes the subject matter seem important (R2). 66% disagree that they did not see a relationship between content and something they already knew (R3). 55% of students confirm that content relates to their expectations and goals (R5) and that students actively participate in class (R6).

Figure 5 shows the level of confidence students achieved with the course. We notice that almost all the students (88%) state that the course subject is difficult (C4) and that it is difficult to predict the grade the teacher will give (C5). However, the same number of students (88%) states that success in the course depends on them (C3) and 55% say they can succeed if they try hard enough (C6).

Finally, Figure 6 describes student satisfaction with the course. Results show that 66% of students agree that they need to work hard to succeed in the course (S1), that they are satisfied with the instructor's evaluations (S5) and that they receive enough recognition of their work (S8). 88% of students disagree of being disappointed with the course (S7). However, 55% of the students disagree to be satisfied with the course (S6).

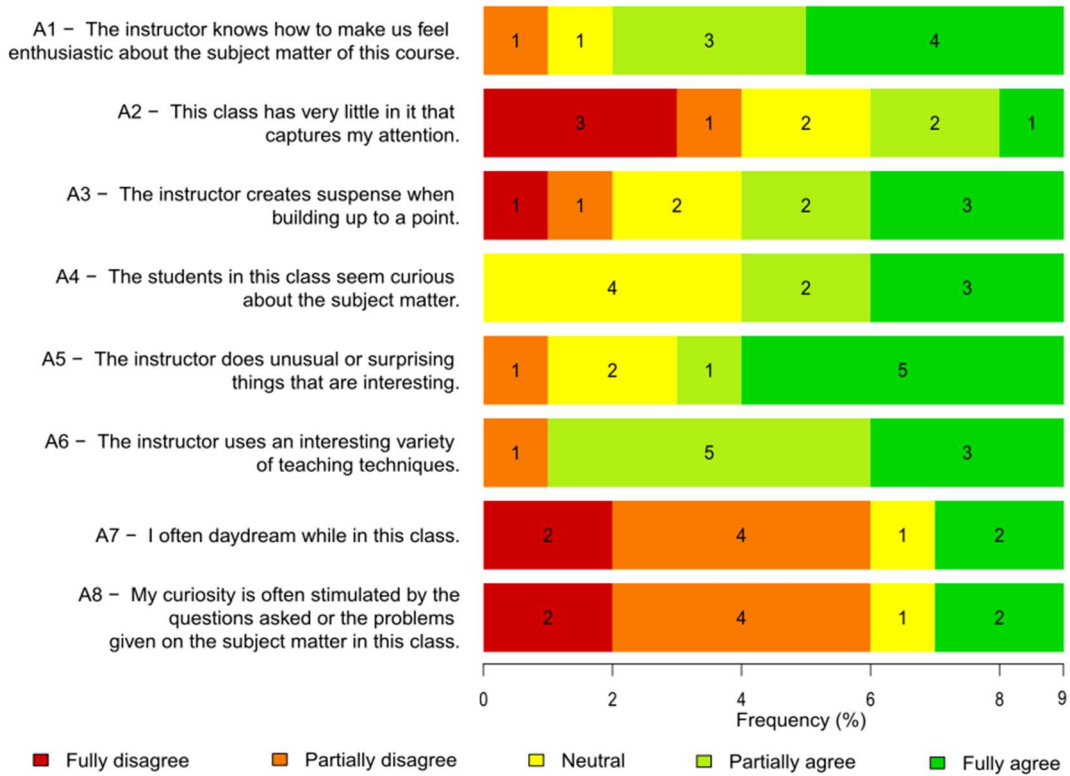


Figure 3. CIS Results on Attention

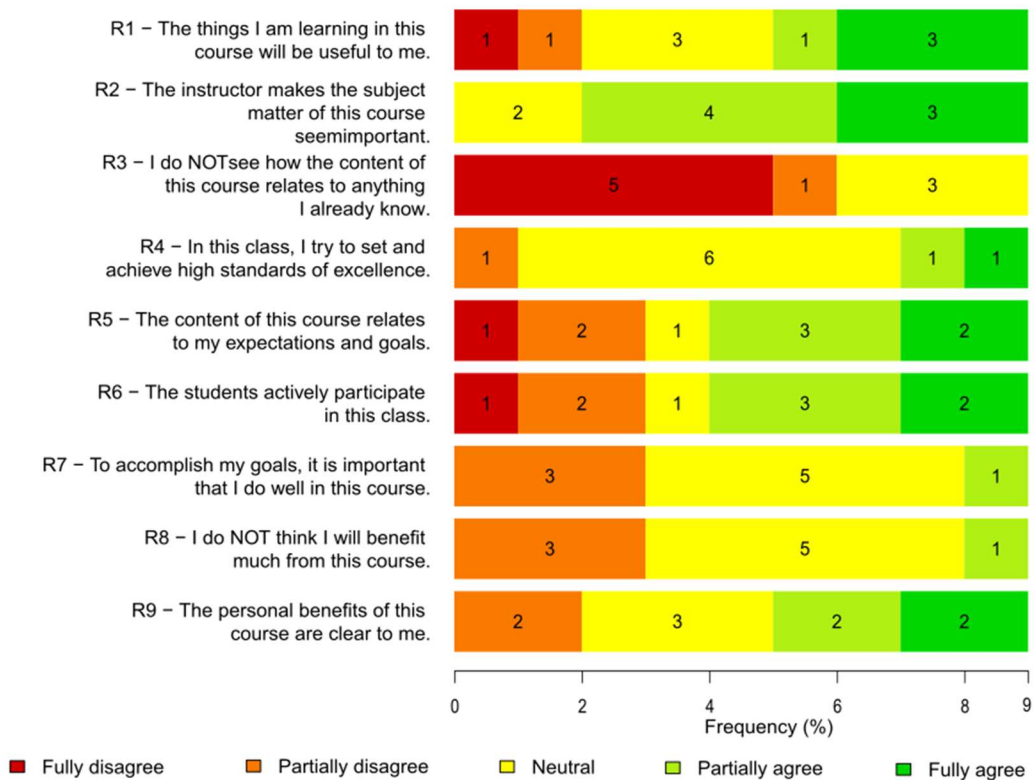


Figure 4. CIS Results on Relevance

5. Learned Lessons

Here we present the main lessons we learned during the course of our intervention. These lessons are derived from both the qualitative and quantitative data in the exploratory research, as stated in Section 3.4.

Python is adequate to novice students. One notices that, although there are initial issues to be mastered such as indentation and case-sensitive text, the language offers easiness on programming and OOP concepts. This is due to Python's easily understandable syntax and the concision, which implies less chance of making errors. This also motivates novice students who, in their first lines of code, are already able to build real applications.

Contextualized activities are important to arouse attention and relevance. Young and teen students use social networking apps, apply filters on pictures, listen to music, and watch videos. Hence, editing images is a context close to the students, which makes the activities relevant to their interests, becoming an additional positive factor for motivation.

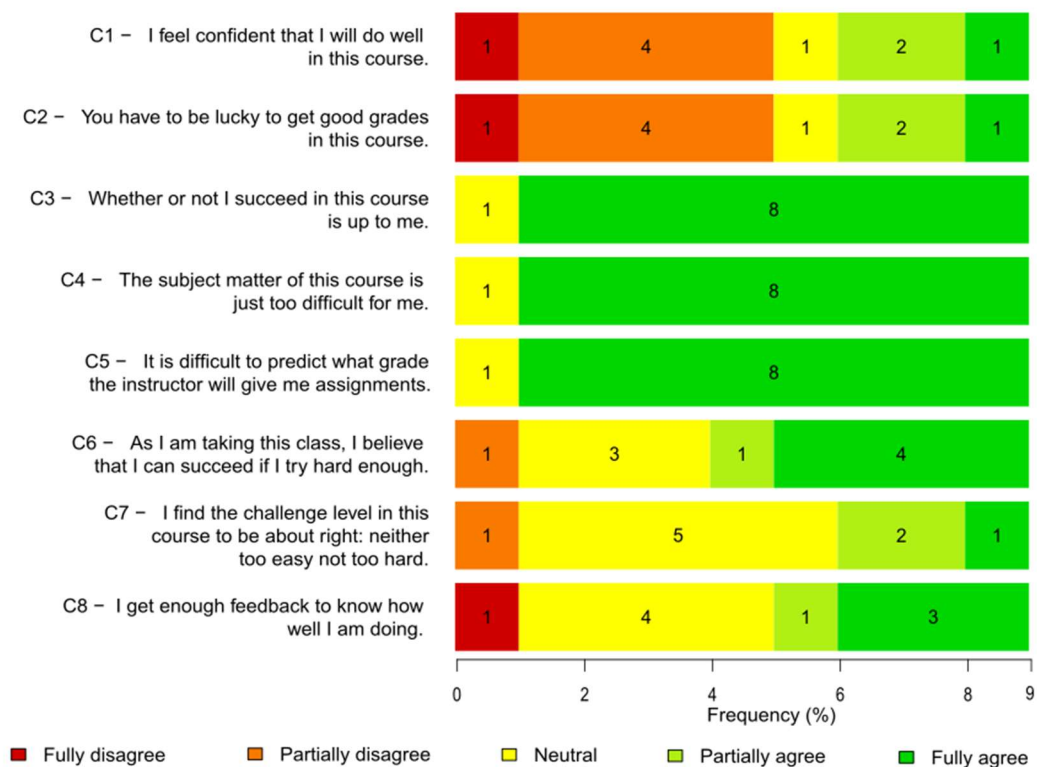


Figure 5. CIS Results on Confidence

Introduction to new content and environments is important and should be done in a way that students feel gradually comfortable to exercise their creativity. However, one notices that the initial activities that aim to present the concepts are less active, therefore, one needs to look for more active ways of presenting these concepts.

Use of similar contexts is a relevant feature, since it minimizes the problem of environment introduction, allowing focusing on concepts related to programming, paradigms and patterns, among others. This reduces cognitive overload and enables more effective learning by revisiting and deepening the concepts. This factor also provides a more active posture of students, since they already know the basics of the context.

Students do not study at home. This is a challenging issue. We know that homework should be encouraged, by offering tutorials, links to home environment installation, and reference materials (e.g., classroom slides). Requesting activities to be returned in the following classes is an alternative that leads students to study at home. However, the particular profile of Brazilian public school students may not guarantee that all students have computers at home.

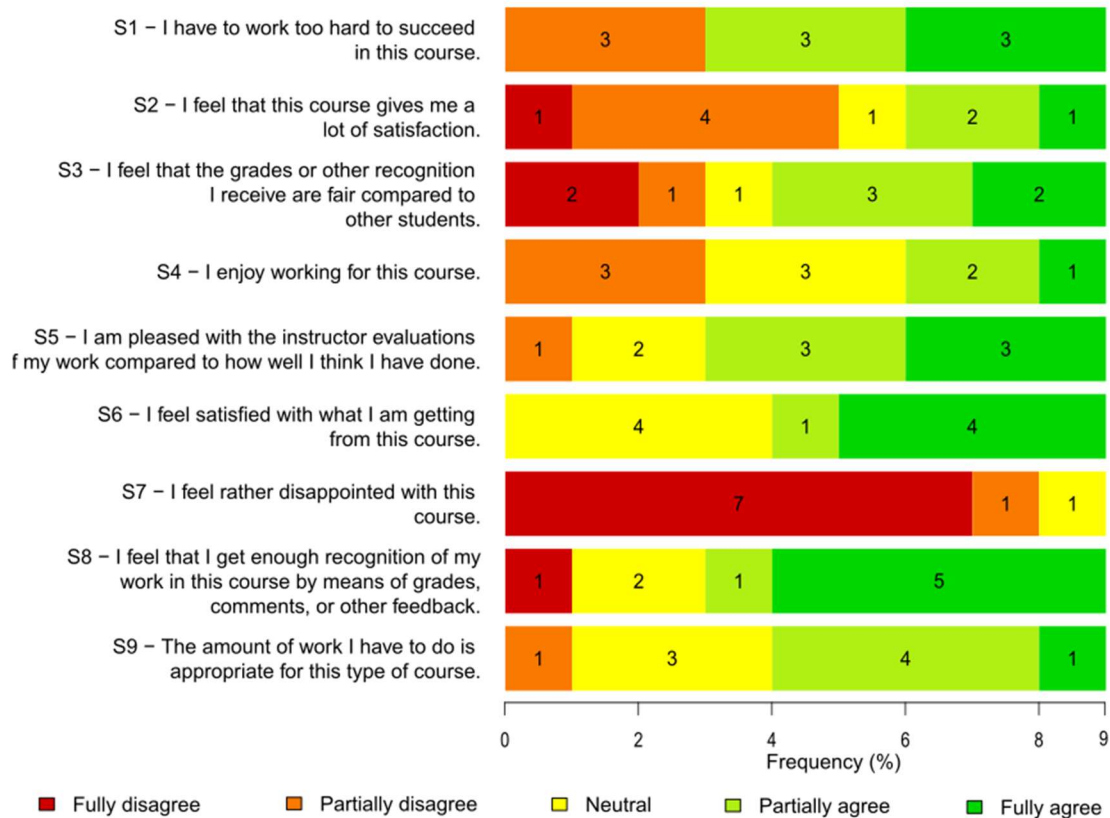


Figure 6. CIS Results on Satisfaction

Challenges are positive. For students, challenges are relevant because they motivate them. The idea of achieving something that the teacher judged difficult or that the student himself, through attempts, realized that it is difficult is something relevant that affects student confidence and satisfaction. This can be awakened by the teaching approach, when proposing a challenge, or by the very activity that, given its complexity, generates a challenging state. Therefore, one should plan activities beyond what has been taught, so that students are motivated, exercise their creativity and explore the tools.

6. Conclusions

This work reported an experience of a teaching and learning approach in the course of Programming Language 2 with 12th grade, secondary students of a vocational program in Informatics integrated with high school. Our approach combines the use of environments (JES and PPlay) and language (Python) suitable for programming novices, contexts relevant to these students (games and images) and presentation of content in gradual levels of complexity, based on spiral learning.

We realized, when formulating the didactic modules, that contexts that use new tools tend to have a more expositive initial character and less conducive to students exercising their creativity. Thus, the reuse of contexts, tools and languages in the different sequential modules allows a more active posture of the students regarding their activities.

The present study suggests that Python is a language suitable for students, because it allows the accomplishment of activities, minimizing syntactic problems. In addition, one notices that activities that use contexts present in students' lives are motivational elements that awaken attention and perceptions of relevance. The combination of JES and Python builds confidence in students. On the other hand, the challenging activities, when completed, increase the degree of satisfaction.

We intend, in future work, to perform a detailed analysis of the approach presented here, aiming to understand more in depth, through qualitative and quantitative data, how the configuration of the approach impacts on motivation and student learning in the modality of vocational secondary education.

Referências

- Araujo, L. G. J., Bittencourt, R. A., & Santos, D. M. B. (2017). Ensino de Programação na Educação Básica através da Manipulação de Mídias. In *COBENGE 2017 – XLV Congresso Brasileiro de Educação em Engenharia*.
- Bordini, A., Avila, C. M. O., Weissshahn, Y., da Cunha, M. M., Cavalheiro, S. A. C., Foss, L., ... Reiser, R. H. S. (2016). Computação na Educação Básica no Brasil: O Estado da Arte. *Revista de Informática Teórica E Aplicada*, 23(2), 210–238.
- Bruner, J. S. (1966). *Toward a theory of instruction*. Toward a theory of instruction.
- Cravo, A. C. (2012). Análise Das Causas Da Evasão Escolar do Curso Técnico de Informática em uma Faculdade de Tecnologia de Florianópolis. *Revista Gestão Universitária Na América Latina - Revista GUAL*, 238–250. <https://doi.org/10.5007/1983-4535.2012v5n2p238>
- Grandell, L., Peltomäki, M., Back, R. J., & Salakoski, T. (2006). Why Complicate Things? Introducing Programming in High School using Python. In *ACE '06 Proceedings of the 8th Australasian Conference on Computing Education* (Vol. 52, pp. 71–80).
- Guzdial, M. (2003). A Media Computation Course for Non-Majors. *ACM SIGCSE Bulletin*, 35(3), 104–108.
- Guzdial, M. (2013). Exploring hypotheses about media computation. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research - ICER '13*, 19. <https://doi.org/10.1145/2493394.2493397>
- IC-UFF. (2017). PPLay: Aprenda a Programar Desenvolvendo Jogos. Retrieved from <http://www2.ic.uff.br/pplay/>
- Keller, J. M. (2010). *Motivational Design for Learning and Performance: The ARCS Model Approach*. Springer US. Retrieved from <https://books.google.com.br/books?id=y289SAAACAAJ>
- Machado, M. R. L., Moreira, R. L., & Priscila, R. (2006). Educação Profissional no Brasil, Evasão Escolar e Transição para o Mundo do Trabalho. In *I Colóquio Internacional sobre Educação Profissional e Evasão Escolar*.
- Marques, D. L., Costa, L. F. S., Silva, M. A. de A., & Rebouças, A. D. D. S. (2011). Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução a Programação utilizando Jogos e Python. *Anais Do Workshop de Informática Na Escola*, 1(1), 1138–1147. <https://doi.org/10.5753/CBIE.WIE.2011.1138-1147>

- Prensky, M. (2001). Digital Natives, Digital Immigrants Part 1. *On the Horizon*, 9(5), 1–6.
- Rebouças, A. D. D. S., Marques, D. L., Costa, L. F. S., & Silva, M. A. de A. (2010). Aprendendo a Ensinar Programação Combinando Jogos e Python. In *XXI Simpósio Brasileiro de Informática na Educação* (pp. 1–10).
- Santana, B. L., Figuerêdo, J. S. L., & Bittencourt, R. A. (2017). Motivação de Estudantes Non-Majors em uma Disciplina de Programação. In *WEI 2017 – XXV Workshop sobre Educação em Computação*.
- Simon, B., Kinnunen, P., Porter, L., & Zazkis, D. (2010). *Experience report: CSI for majors with media computation. Proceedings of the 15th ACM SIGPLAN international conference on Functional programming - ICFP '10* (Vol. 45). <https://doi.org/10.1145/1863543.1863595>
- SOFTEX. (2012). Software e Serviços de TI--A Indústria Brasileira em Perspectiva. SOFTEX. Retrieved from http://www.mct.gov.br/upd_blob/0222/222489.pdf