The Curriculum Integration of a course of 'Introduction to Programming Logic' with a Serious Game - Colobot

Jos é R. M. Alves and Patrick Letouze

Abstract— **Programming logic is a knowledge and skill that find applications in daily lives, since nowadays most products, processes and systems are intrinsically technological. Therefore, it is reasonable to argue in favour of teaching it to all students, especially high school students. An engaging medium to accomplish that would be the use of a serious game to facilitate the teaching of programming logic. It is desirable that the learning process happens in a structured way respecting the traditional approach to teaching programming logic, which should avoid the resistance of instructors. If it is taken into consideration that, the Curriculum Integration is a success factor for educational games acceptance by teachers and its subsequent adoption in class, then it provides the desired structured way to teach programming logic with a serious game. Hence, this work presents a Curriculum integration of the serious game Colobot Gold Edition with the discipline of 'Introduction to Programming Logic' for the level of technician. This is obtained through the combination of the Interdisciplinary Research Project Management strategy with the Input-Process-Output model, keeping focus on Curriculum integration. The developed curriculum integrated missions provide an alternative for educating all high school students in programming logic with an engaging game, besides that a translation to Portuguese of Colobot was produced allowing Portuguese speaking students to use their native language to learn programming logic in a "fun" way.**

Index Terms—**Serious games, programming logic, Colobot, education.**

I. INTRODUCTION

Contreras and Siu in [1] verified the importance of tools that facilitate the teaching of programming logic. They argued that programming should be taught to students of all areas, since most products and processes are currently technological in nature, and students should not only know how to use technology, but also how to create technology. Introductory teaching of programming languages often encounters major barriers to student learning, and engagement is one such barrier [2]. A good alternative to solve this problem is the use of games [3], and particularly, they have already been used satisfactorily in teaching programming logic [4].

An example of a serious game for learning programming is the game Colobot: Gold Edition [5]. In 2009, a study conducted by Muratet et al.[6] about serious game for programming remarked that the game Colobot was not free and there existed some difficulties in building new exercises by teachers. However, in 2014, its source code became open, and many members of the community made it available, including new game modes [5]. Therefore, the problems highlighted by Muratet et al. in [6] were solved, and Colobot has improved as an option to improve students' programming skills.

Colobot is a game that focuses on the colonization of planets with the help of robots that the player can program to perform various tasks, and for that, it is using a programming language called CBOT, very similar to C++. It also allows its users to create and to custom missions divided into chapters and levels.

The main purpose of our work is to present the Curriculum integration of the serious game Colobot Gold Edition with the discipline of "Introduction to Programming Logic" for the level of technician to be ministered at the Instituto Federal de Educa ção, Ci ência e Tecnologia do Tocantins (IFTO - Federal Institute of Tocantins), campus Lagoa da Confus ão, Brazil. The secondary purpose is to report that we have translated the game Colobot to Portuguese to be available in the next edition [7], which is a requirement for its use in Brazilian classrooms.

For the organization of this paper, we chose the following structure: Introduction, Material and Methods, Results and Discussion, Conclusions. In the Material and Methods section, we presented Colobot Gold Edition as the material used in our work; the Interdisciplinary Research Project Management (IRPM) as our research strategy method [8]; the method for Curriculum integration that was based on the experience of Shapi'i and Ghulam in [9]; and we used the Input-Process-Output model [10] for the development of Missions in Colobot. Then we present the results and we discuss them in Section III. Finally, we present our conclusion in Section IV.

II. ATERIAL AND METHODS

A. Material: the Serious Game Colobot

A serious game is "an application with three components: experience, entertainment, and multimedia" [11], where through experience the player acquires knowledge, skills and other contents; entertainment is the immersion factor; and multimedia such as text, graphics, animations, and so on provides learning opportunities.

The game Colobot is a serious game for learning programming language in its conception. Colobot provides a

Manuscript received July 18, 2018; revised October 12, 2018.

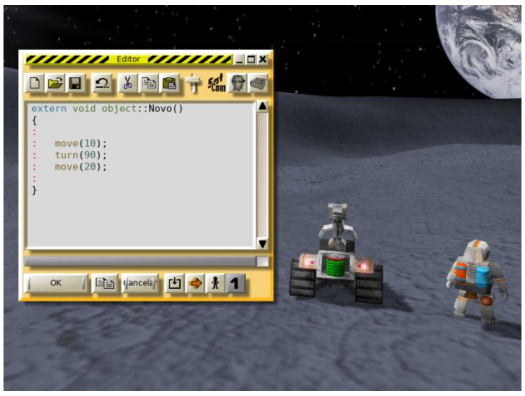
J. R. M. Alves is with the Federal University of Tocantins, Brazil (e-mail: jrma@uft.edu.br).

P. Letouze is with the Federal University of Tocantins, Brazil (e-mail: letouze@uft.edu.br).

doi: 10.18178/ijssh.2018.8.11.974

programming environment, as in Fig. 1. It has a large and accessible manual within that teaches basic concepts and presents in details all available functions. Its main story focuses on an astronaut looking for a habitable planet for humanity and with the help of programmable robots. Additionally, the game has several exercises to teach the player to program and understand their mechanics.

It should be emphasized that we do not recommend the exercises found in the game for introductory teaching of programming logic, because they do not follow a gradual increase of difficulty, nor do they contemplate all the bases of an introductory curriculum. Consequently, players with no programming experience may have some difficulties.



```
extern void object::Novo()
{

    move(10);
    turn(90);
    move(20);

}
```

Fig. 1. An example of the programming environment of Colobot.

B. Method: the Interdisciplinary Research Project Management

The Interdisciplinary Research Project Management (IRPM) [8] is a methodology for developing interdisciplinary research based on project management [12] that intends to maximize academic results as shown in Fig. 2.

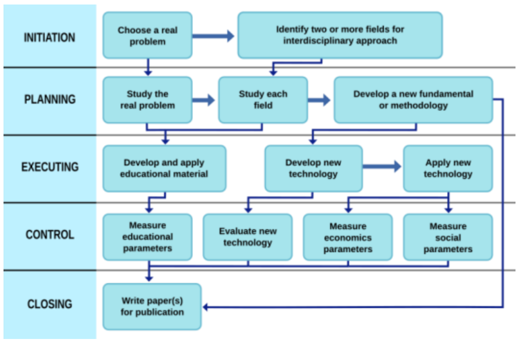

INITIATION
PLANNING
EXECUTING
CONTROL
CLOSING
Choose a real problem
Identify two or more fields for interdisciplinary approach
Study the real problem
Study each field
Develop a new fundamental or methodology
Develop and apply educational material
Develop new technology
Apply new technology
Measure educational parameters
Evaluate new technology
Measure economics parameters
Measure social parameters
Write paper(s) for publication

Fig. 2. Interdisciplinary Research Project Management.

We present the application of IRPM in our work in Fig. 3. We begin by choosing the problem of "enhancing students' engagement in studying programming logic", and we choose two fields, "serious games" and "teaching programming logic", ending the Initiation phase of IRPM. Following that, we have the Planning phase, where we start with analyzing the curriculum of the discipline "Introduction of Programming Logic" of the technician course of the advanced campus of Lagoa da Confusão of IFTO, Brazil. Then we studied "serious games" and "teaching programming logic", resulting in the choice of Colobot (Gold Edition) as the serious game. Also at this moment, we verified that Colobot did not have a version in Portuguese, a requirement for Brazilian students, because they do not have enough knowledge of other languages, not even English. In order to soften this drawback, we decided to translate Colobot to Portuguese, with the exception of the programming language commands, since the use of English commands will familiarize them with future programming languages.

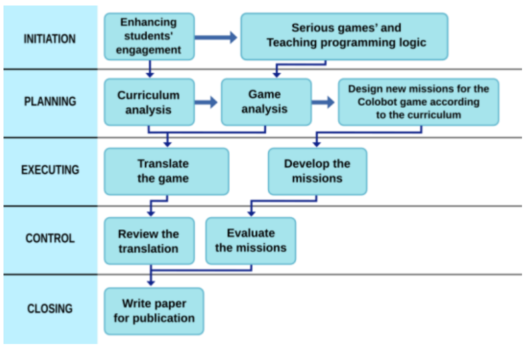

INITIATION
PLANNING
EXECUTING
CONTROL
CLOSING
Enhancing students' engagement
Serious games' and Teaching programming logic
Curriculum analysis
Game analysis
Design new missions for the Colobot game according to the curriculum
Translate the game
Develop the missions
Review the translation
Evaluate the missions
Write paper for publication

Fig. 3. The application of IRPM to Curriculum integration of 'Introduction of Programming Logic'.

Still in Planning phase, we have the game analysis for designing new missions in Colobot with Curriculum integration in mind. The translation of the game to Portuguese is part of the Execution phase as much as the development of missions in Colobot. Control phase is responsible for the translation review and for mission's evaluation. In the Closing phase, we attempt to publish our results.

C. Method: Curriculum Integration

Our approach to Curriculum integration is based on the experience of Shapi'i and Ghulam [9],who used as theoretical foundation the work of Freitas in [13].Shapi'i and Ghulam state that a successful Curriculum integration connects curriculum components in a meaningful way to students [9], and for that, they provide three steps:

1) To design a course based on components of the curriculum that may benefit from game-enhanced approaches;
2) To generate an environment of flexible learning;
3) To start with topics that can be efficiently measured for latter to expand them.

The teaching of programming logic revolves around the creation of algorithms for problem solving, with the incremental introduction of new concepts for the construction of increasingly complex algorithms. Instructors use the practice of code writing as the main medium to increase learning rates. Colobot preserves this practice, which is a desirable feature for instructor's engagement and familiarity. Hence, in this case, the curriculum integration does not change the teaching culture of instructors. Even the game design provides the necessary flexibility for incremental introduction of new concepts. That is, the culture of teaching programming logic is completely preserved, while increases students' engagement in the learning process.

D. Method: the Missions

An algorithm is an ordered list of actions to solve a problem and may be described by the Input-Process-Output model, Fig.

4, also known as IPO model [10]: Input - is the data obtained from the keyboard, files or other form of data entry; Processing - is the input processing in order to generate a result; Output - is the result of processing, may be printed on the screen, stored in a file or otherwise output.

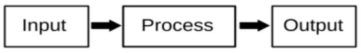


Fig. 4. IPO model.

Normally, in the introductory teaching of programming logic, the input is captured from the keyboard so that the algorithm can process this data and generate the output. In the Colobot game there is no way for the robot to read data from the user's keyboard, so in order to get an input process, information exchange posts will be used, as in Fig. 5. Its purpose is to store information and provide it to the robot when requested. All developed missions use information requests from an exchange post as 'input', the logic developed by the student as 'process' and the actions performed by the robot as 'output' are presented in Table I.

TABLE I: CATEGORIES OF ACTIONS FOR ROBOTS IN THE GAME COLOBOT

Action	Objective
Movement	Go from one place to another
Detection	Detect nearby objects
Transport	Object transportation
Fight	Destroy objects or defend itself
Construction	Construct robots or buildings
Communication	Receive or send messages

Hence, the process of creating missions begins with the choice of an item of the curriculum. After that, we define the purpose for the mission that will consequently determine the output, and then we define the inputs that will be provided and the conditions of victory or failure. We describe this process with a simple mission in Table II and in Fig. 5 we present the process code, then in Fig. 6 we present a picture of the mission.

TABLE II: MISSION 'INFORMATION GATHERING'

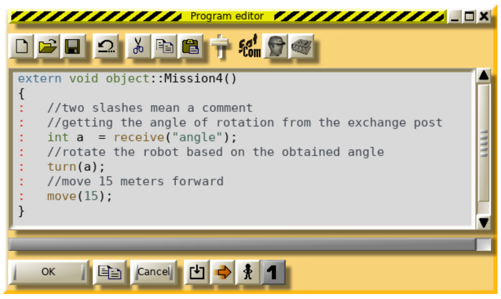
Property	Details
Mission name	Information gathering
Objective	Take the robot to the correct waypoint
Input	Rotation angle
Process	Obtain the angle of rotation from the information exchange post, rotate the robot based on the obtained angle, move 15 meters forward
Output	The robot moves toward the waypoint
Victory condition	The robot is on top of the correct waypoint
Fail condition	The robot is on top of an incorrect waypoint

E. Method: the Translation

A game is software and its production involves several different areas, from story building, character design, art development, sound creation and code programming. Thus, to prepare the translation of software, one must keep in mind three factors:

1) Localization;
2) Source code;
3) Syntax.

When writing software, the programmer can use resources to make it multi-language, a process known as localization [14], wich can be achieved by separating files for each language, external or internal to the source code.



```
extern void object::Mission4()
{
:   //two slashes mean a comment
:   //getting the angle of rotation from the exchange post
:   int a  = receive("angle");
:   //rotate the robot based on the obtained angle
:   turn(a);
:   //move 15 meters forward
:   move(15);
}
```

Fig. 5. Process code to complete mission 'Information gathering'.

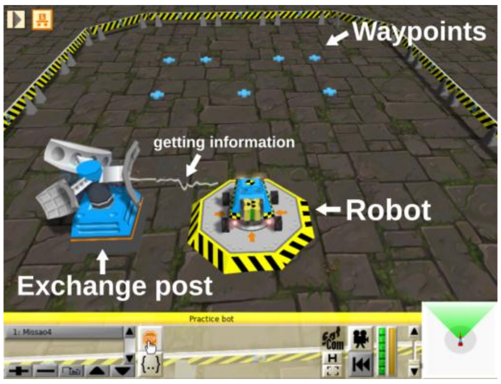


Fig. 6. Mission 'Information gathering'.

The source code is the code written by the programmers, ie the software itself, it can be closed (restricted access) or open (available to all). Commercial games are examples of closed source software, while free games may have their code open.

The syntax refers to expressions used in translation for formatting or structuring the text, usually they are expressions in English that follow a pattern and should not be translated as they would lose their effect.

Taking into account the localization and the source code, Table III was created to demonstrate an analysis of the "translatability" of a game, that is, the condition of a possible translation.

TABLE III: TRANSLATABILITY OF A GAME

Localization	Source code	Translation
Not	Closed	Not possible
Not	Open	Very difficult
Yes	Closed	Possible
Yes	Open	Easy

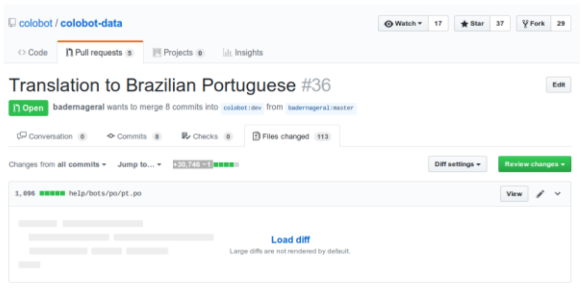
If the software does not implement any localization process and its source code is closed, it is said that the translation is not possible because there are no language files or access to the code for modification.

The fact that the software does not use any localization process indicates that it does not accept multiple languages, so the only way to translate would be to change the source code and replace the original language with the target language or implement a localization process. This would be an arduous process, requiring great programming skills.

When the software implements some localization process, one should investigate how this process was implemented. Some software use external files, while others use internal files. Internal files are inside the source code, so if the source code is closed, there is no way to access these files. Typically closed source software uses external files to enable translation without source code access, while open source software uses both approaches.

The game Colobot implements a localization process using internal files and has open source code [15]. The first step in translating was to get the source code for the game and the documentation on the syntax used in the translation files that the project team makes available on their site. Note that in this case, a basic programming knowledge is required, since it is necessary to change the source code following the syntax obtained to perform the translation.

The Colobot translation involved the addition of approximately 30,000 new lines to the project source code (see Fig. 7), where at least 30% of these lines correspond to sentences in Portuguese, with the rest of the lines representing line breaks or identifiers. Fig. 8 represents a translated sentence, where the first line contains the identifier for the phrase, that is, the original message in English and the second the translation.

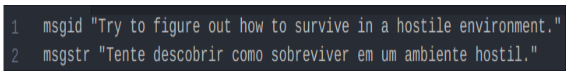


Translation to Brazilian Portuguese #36

badernageral wants to merge 8 commits into colobot:dev from badernageral:master

1,096 help/bots/po/pt.po

Fig. 7. The translation of Colobot Gold Edition to Portuguese.



```
msgid "Try to figure out how to survive in a hostile environment."
msgstr "Tente descobrir como sobreviver em um ambiente hostil."
```

Fig. 8. Translation example.

III. Results and Discussion

The curriculum of the discipline was analyzed and divided into five clusters of components that group similar topics. Since Colobot allows the creation of hierarchy between chapters and missions, each cluster is represented by a chapter and each chapter has several specialized missions, which are presented in Table IV.

TABLE IV: Curriculum Components and Their Chapters

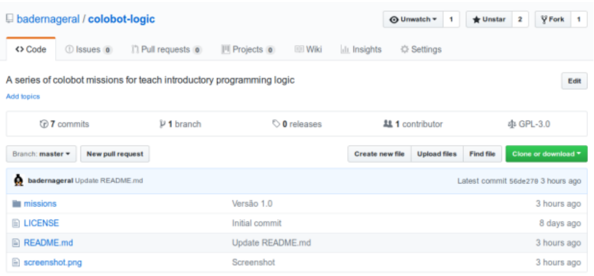
Curriculum components	Chapters	Number of missions
Variables	I - First steps	9
Decision structures	II – Making decisions	5
Repetition structures	III -Time to repeat	7
Arrays	IV - Grouping	5
Functions	V - My instructions	4

The first chapter (I) deals with the introduction of the programming language through the use of variables and varied exercises that introduce the actions of a robot in order to familiarize the student with the programming process. The second chapter (II) adds to the decision making, the third chapter (III) the concept of repetition, the fourth chapter (IV) the ability to group information and the last chapter (V) to create custom functions (see Table V).

TABLE V: Curriculum Integrated Missions Objectives

Chapter	Mission(s)	Objective
I	1, 2 and 3	Introduce the robot and mechanics of the game, and the concept of process and output
I	4	Introduce the concept of input
I	5 and 6	Working with coordinates (Cartesian plane)
I	7	Use of radar, object detection
I	8	To learn how to carry objects
I	9	To learn how to fire, destruction of objects
II	1 and 2	Making decisions, working with cardinal directions
II	3	Decisions with intercardinal directions
II	4	Decisions with multiple conditions
II	5	Chaining of decisions
III	1	To learn how to repeat
III	2	To work with infinite loops
III	3 and 4	To work with loops and object detection
III	5, 6 and 7	To work with loops and destruction of objects
IV	1 and 2	To learn how to work with arrays and object detection
IV	3 and 4	To work with arrays and destruction of objects
IV	5	To work with multiple-dimensional arrays
V	1	To learn how to make functions
V	2	To make functions of destruction of objects
V	3	To make functions for work with cardinal directions

The missions are available in a GitHub repository [7] (see Fig. 9) so that anyone can download and use them in the game. The secondary objective of this work was to provide a version of Colobot in Portuguese; the translation of the game into Brazilian Portuguese has been successfully completed, and may be obtained from the official game repository [7]. Hence, we provide the missions in English and Brazilian Portuguese, the language of the game will define the language of the missions.



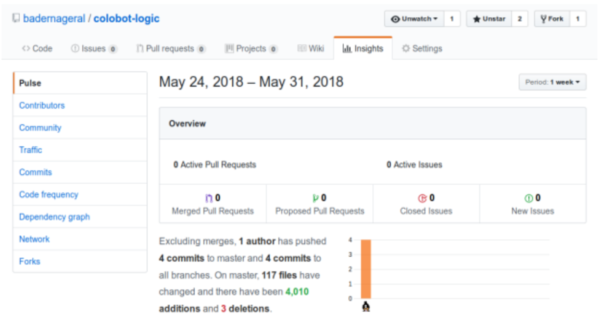
A series of colobot missions for teach introductory programming logic

badernageral Update README.md — Latest commit 56de278 3 hours ago

missions	Versão 1.0	3 hours ago
LICENSE	Initial commit	8 days ago
README.md	Update README.md	3 hours ago
screenshot.png	Screenshot	3 hours ago

Fig. 9. Repository of missions in GitHub.

The missions were designed with an approach of incremental difficulty, that is, complexity increased little by little, in order to provide the time needed for the student to adapt. In order to demonstrate the amount of work necessary to design and develop the missions that are both in English and Portuguese, we present the missions status in the GitHub in Fig. 10. We have 4,010 code lines divided in 117 files. In addition to that, in Fig. 11, we have the Colobot screen in Portuguese of the missions.



May 24, 2018 – May 31, 2018

Overview

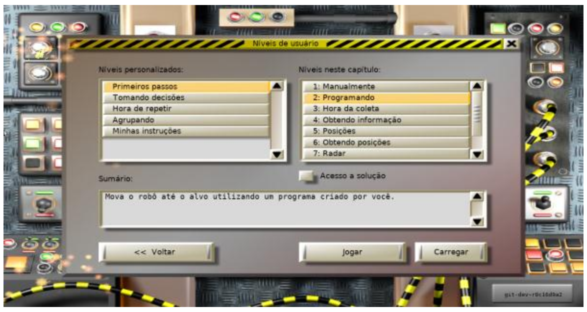
0 Active Pull Requests

0 Active Issues

0	0	0	0
Merged Pull Requests	Proposed Pull Requests	Closed Issues	New Issues

Excluding merges, **1 author** has pushed **4 commits** to master and **4 commits** to all branches. On master, **117 files** have changed and there have been **4,010 additions** and **3 deletions**.

Fig. 10. Missions repository status.

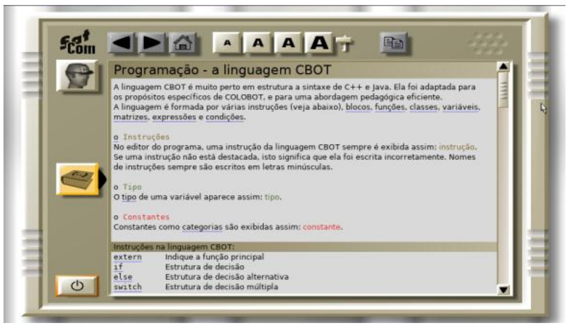


Sumário: Mova o robô até o alvo utilizando um programa criado por você.

Fig. 11. Missions in Portuguese in Colobot.

We encountered a problem when attempting to create more dynamic missions, the impossibility of creating random values, such as positions of objects, or conditions of victory, which could be solved only by modifying the way the missions are created. Unfortunately, the modification of the mechanics of the game was a task not performed in this work, due to its complexity and because of the scope of this work.

About the translation, the most significant contribution is the translation of the game manual (see Fig. 12), which will make a big difference when the game is used in classrooms that use Portuguese as the main language.



Programação - a linguagem CBOT

A linguagem CBOT é muito perto em estrutura a sintaxe de C++ e Java. Ela foi adaptada para os propósitos específicos de COLOBOT, e para uma abordagem pedagógica eficiente.
A linguagem é formada por várias instruções (veja abaixo), blocos, funções, classes, variáveis, matrizes, expressões e condições.

o Instruções
No editor do programa, uma instrução da linguagem CBOT sempre é exibida assim: instrução.
Se uma instrução não está destacada, isto significa que ela foi escrita incorretamente. Nomes de instruções sempre são escritos em letras minúsculas.

o Tipo
O tipo de uma variável aparece assim: tipo.

o Constantes
Constantes como categorias são exibidas assim: constante.

Instruções na linguagem CBOT:

extern	Indique a função principal
if	Estrutura de decisão
else	Estrutura de decisão alternativa
switch	Estrutura de decisão múltipla

Fig. 12. Colobot's manual in Portuguese.

IV. Conclusions

We developed a series of missions in Colobot (Gold Edition) that may be used for a course of 'Introduction to Programming Logic', both in English and Portuguese, with the translation of the game to Portuguese [5,7,15].We have accomplished that with a Curriculum integration based on the experience of Shapi'i and Ghulam in [9], the IPO Model [10] and the Interdisciplinary Research Project Management strategy [8].

We are convinced that the game Colobot shows great potential for teaching programming logic based on an informal assessment with students. The curricular integration made from the creation of the missions opens a range of possibilities, since it allows, in addition to classroom use, that other instructors use them as a basis to produce their own missions. For instance, these missions may be used as a basis for the development of interdisciplinary missions such as presented by Batista et al. in [16] (An Interdisciplinary Approach to Teaching Geometry with Serious Game 3D Colobot), which already used IRPM.

We suggest an analysis based on the study of Freitas and Oliver [17] as future work, because we did not make a formal analysis of the effectiveness of these missions for students' engagement and learning. Our main objective was to provide a guide for future mission development and to incentive others to translate the game to other languages. Hence, we expect that with the dissemination of this work, more people will join the cause and contribute to the development of the game, or see the possibilities of using this type of tool in the classroom, and that, the developed curriculum integrated missions provide an alternative for educating all high school students in programming logic with an engaging game.

References

[1] G. J. Contreras and K. W. M. Siu, "Computer programming for all: A case-study in product design education," *Procedia - Social and Behavioral Sciences*, vol. 182, pp. 388-394, 2015.

[2] M. Rahmat, S. Shahrani, R. Latih *et al.*, "Major problems in basic programming that influence student performance," *Procedia - Social and Behavioral Sciences*, vol. 59, pp. 287-296, 2012.

[3] L. A. Annetta, J. Minogue, S. Y. Holmes, and M. Cheng, "Investigating the impact of video games on high school students' engagement and learning about genetics," *Computers and Education*, vol. 53, Issue 1, pp. 74-85, 2009.

[4] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, "A serious game for developing computational thinking and learning introductory computer programming," *Procedia - Social and Behavioral Sciences*, vol. 47, pp. 1991-1999, 2012.

[5] (2014). "3D real time game of strategy and adventure for learning programming". *Colobot: Gold Edition*. [Online]. Available: http://colobot.info.

[6] M. Muratet, P. Torguet, J. Jessel, and F. Viallet, "Towards a serious game to help students learn computer programming," *International Journal of Computer Games Technology*, vol. 2009, 2009.

[7] (2018). "A series of colobot missions for teach introductory programming logic", *Colobot-logic*. [Online]. Available: https://github.com/bader nageral/colobot-logic.

[8] P. Letouze, "Interdisciplinary research project management," *International Proceedings of Economics Development and Research*, vol. 14, pp. 338-342, 2011.

[9] A. Shapi'i and S. Ghulam, "Model for educational game using natural user interface," *International Journal of Computer Games Technology*, vol. 2016, 2016.

[10] S. Sentance, E. Barendsen, and C. Schulte, "Computer science education: Perspectives on teaching and learning in school," *Bloomsbury Publishing*, 2018.

[11] F. Laamarti, M. Eid, and A. E. Saddik, "An overview of serious games," *International Journal of Computer Games Technology*, 2014.

[12] K. Heldman, "PMP project management professional exam deluxe study guide: Updated for the 2015 Exam," John Wiley & Sons, 2015.

[13] S. D. Freitas, "New pedagogical approaches in game enhanced learning: Curriculum integration," Idea Group Inc (IGI) United States of America, 2013.

[14] A. C. Prudêncio, D. A. Valois, and J. E. Lucca, "Introdução à internacionalização e à localização de softwares," Cadernos de Tradução, vol. 2, no. 14, pp. 211-242, 2004.

[15] (2014). "Source code of open-source Colobot: Gold Edition project". Colobot: *Gold Edition*. [Online]. Available: https://github.com/colobot.

[16] R. R. Batista, E. R. Ferraz, and P. Letouze, "Uma Abordagem Interdisciplinar para o Ensino de Geometria com o Serious Game 3D Colobot" (An Interdisciplinary Approach to Teaching Geometry with Serious Game 3D Colobot), Proceedings of the XVI SBGAMES– Simpósio Brasileiro de Games (XVI Brazilian Games Symposium), 2017.

[17] S. de Freitas and M. Oliver, "How can exploratory learning with games and simulations within the curriculum be most effectively evaluated?", *Computers & Education*, v. 46, no. 3, pp 249-264, 2006.

José R. M. Alves in São José de Piranhas, Paraíba, Brazil, in 1989. Postgraduate in Systems Engineering at High Open School of Brazil in 2016 and graduate in Analysis and Systems Development at University North of Paraná, Brazil, in 2014.

He is a professor in Federal Institute of Tocantins, Brazil and researches the development and use of games in education, as well as other minor research about software development.

He is a student of the Graduate Program in Computational Modeling of Systems in the Federal University of Tocantins, Brazil.

Patrick Letouze Moreira has a bachelor degree in Control and Automation Engineering (Universidade Federal de Santa Catarina,); a Master Science degree and a Doctor in Science degree in Electrical Engineering.

He is a professor at the Universidade Federal do Tocantins (UFT). He is the vice-director of the Graduate Program of Computational Modeling of Systems, since its begining; and he was the director of the Graduate Program of Health and Science Education, since its creation until th first quarte of 2018. He is a fellow member of IEDRC and a member of IEEE.

.