Difficulties of learning Programming

Chaker Abid[1], Hedia Mhiri Sellami[2], and Lamjed Ben Said[3]

***Abstract*— Programming learning has become a 21st century skill, several countries have integrated it into their school curricula. Numerous studies have addressed the teaching of computational thinking and programming in primary schools and even nursery schools. These studies have shown that learning programming presents many difficulties for learners, especially beginners. This paper proposes a new classification of the difficulties encountered in learning programming for beginners, in particular for secondary school students. We also aim to study the impact of gender and level of learners in less developed regions on programming learning. We carried out two questionnaires with 79 students in a final year science class at a secondary school in Tunisia. The study showed that the majority of learners encountered difficulties of several types and with several concepts. The most difficult concepts were variables, operator priority rules and loops. The study also showed that gender had no effect on programming.**

Keywords : secondary school, learning programming, challenges.

I. Introduction

This Learning programming has become an important skill in the 21st century. According to [1], it begins in some countries at primary school, or even in pre-school. [2] have shown that learning programming presents many difficulties for learners, especially beginners, and even for teachers, who have to find pedagogical alternatives to facilitate the learning of difficult concepts. [3], [4] have shown that these difficulties have led to very high failure rates among learners. Again, [5] have shown that learners' levels of engagement in learning programming courses are low. This high failure rate highlights the need to address these difficulties in order to help learners overcome them.

The literature has mentioned several studies that have addressed programming learning difficulties. Most of these studies have been carried out in higher education and mainly for students in computer science courses, who generally have a basic knowledge of programming.

As education systems evolve, interest in programming continues to grow. New generations have become accustomed to programming from an early age, mainly through games. Further studies are needed to determine whether learning programming still presents difficulties for learners.

In this study, we propose a new classification of the difficulties encountered by beginners in disciplines where programming is not a major subject in secondary schools, based on the Tunisian curriculum.

In Tunisia, learning computational thinking starts in college at the age of 12. Learning programming starts in secondary school using project-based teaching. We have a section called computer science in which we teach programming and dynamic web design. In the other sections, programming is not a core subject in which we teach simple, conditional and iterative structures, procedural programming and sorting algorithms. We also aim to address the effect of gender on programming learning.

We aim to answer the following questions:

Does learning to program still present difficulties for learners?

Does gender has an effect on learning to program?

The remainder of this paper is detailed as follows. The next section describes the related work, section 3 presents the methodology used, while section 4 analysis of the results. Finally, we finish with a conclusion.

II. Related work

In a world where technology is advancing exponentially, learning to program has always been a major concern for researchers. The literature cites several studies that have identified difficulties in learning to program. In this section, we will address these difficulties by proposing a new classification. This classification is necessary, as it is the first step to identify solutions.

A. Difficulties related to problem-solving skills

[6] have shown that a lack of problem-solving skills prevents beginners from learning programming. In addition, they showed that learners who lack problem-solving skills have difficulty solving a task quickly, so they abandon the problem. [2] have shown that the lack of these skills can be explained by two causes. Firstly, learners do not understand the problem, either because they have misinterpreted the problem statement, or because they immediately start writing a solution before they have fully understood the problem [7]. Secondly, learners have difficulty using their prior knowledge. They do not correctly make analogies between solutions to previous problems and the problem to resolve. They are unable to use the knowledge they have acquired to solve a new problem. Studies [3], [8], [9], [10] have reported a lack of knowledge consolidation.

[1] Chaker Abid is now with the Laboratory of Strategies for Modeling and ARtificial intelligence, Institut Supérieur de Gestion de Tunis, Université de Tunis Cité Bouchoucha 2000 Le Bardo,Tunis, Tunisia; abidchaker@yahoo.fr

[2] Hedia Mhiri Sellami is now with the Laboratory of Strategies for Modeling and ARtificial intelligence, Institut Supérieur de Gestion de Tunis, Université de Tunis Cité Bouchoucha 2000 Le Bardo,Tunis, Tunisia; hedia.mhiri@isg.rnu.tn

[3] Lamjed Ben Said is now with the Laboratory of Strategies for Modeling and ARtificial intelligence, Institut Supérieur de Gestion de Tunis, Université de Tunis Cité Bouchoucha 2000 Le Bardo,Tunis, Tunisia; bensaid_lamjed@yahoo.fr

According to [11], non-computer science students have poor logical, creative and critical thinking skills that prevent them from solving problems.

[12] have showed that novice students often take a line-by-line approach to programming rather than dealing with the whole program.

B. Problems related to learners' conceptions

Learners sometimes have misconceptions that affect their learning, such as the choice of loops [13]. They are unable to use the appropriate loop to solve a problem. [14] have shown that these misconceptions prevent them from assimilating the automatic change of the counter in the loop and the index of an array. Similarly, [15] have shown that learners have difficulty assimilating a number of concepts such as memory, list, loops and pointers. They are unable to understand the changes in values that occur for these concepts during program execution.

According to [16], these misconceptions are a demotivating factor. They prevent learners from assimilating many programming concepts and inhibit their learning progress.

C. Problems of teaching methods

According to [13], programming consists of two phases: problem solving and code implementation, and each phase requires specific skills. Learners are faced with two complicated tasks. The problem-solving phase is essential and forms the basis for the second phase. Teachers should therefore emphasise this phase. Teachers often focus on teaching the programming language and its syntax instead of focusing on the problem-solving process. Learners must master the syntax of the implementation language. [17] have identified some common mistakes made by beginners, such as the use of semicolons, braces and program design. According to [12], learners are unable to identify syntax and logic errors. When an error message is displayed during compilation, they are unable to identify its meaning and make the right corrections. Furthermore [18], showed that beginners make mistakes in their code such as using variables, boolean expressions, assignment operators and comparison operators. Also, [6] have showed that programming should be taught in a personalised way, rather than using traditional teaching methods. The teacher should monitor each learner individually and help him or her in the problem-solving phase. This supervision is not always possible due to time constraints and the content to be taught. Learners learn at different speeds and in different styles, which sometimes do not correspond to the teacher's strategy. The teacher must therefore adopt the best teaching strategy according to a number of criteria, such as the target audience.

D. Problems related to the subject

Programming is a subject that requires a high level of skills such as abstraction, generalisation and critical thinking [2]. In addition, programming languages contain a relatively complex syntax that is difficult to memorise. Learners are asked to perform two difficult tasks: they must focus on developing the algorithm and, at the same time, master the syntactic rules of the programming language. These two tasks can sometimes complicate the learning process and problem solving.

[19] have shown that learning programming is done using a textbook, which is a static tool and is not appropriate for teaching programming, a subject that is constantly evolving.

[20] have showed that many teachers are unable to motivate learners because they lack the experience and sometimes the knowledge to convey information effectively to learners. They are also unfamiliar with different teaching methods and teaching tools.

E. Psychological problems

[2] have shown that programming has a bad reputation. It is seen as difficult to understand which reduces learner motivation. [21] have shown that motivation influences learner outcomes. Consequently, less motivated learners performed poorly in introductory programming courses. [22] has shown that there is a correlation between a positive attitude and success.

Once these difficulties have been identified, we conduct a classroom experiment with the learners to answer the questions posed in this study. The following section describes our methodology.

III. Methodology

In this study, we want to find out whether learning to program still presents difficulties for today's learners, a generation closely connected with technology. We address the difficulties encountered by secondary school learners in the Tunisian context in learning a number of basic programming concepts. We also want to study the impact of gender in learning programming. We carried out the experiment with 79 learners in a scientific final year class at a secondary school in Tunisia, including 52 boys and 27 girls. The learners are aged between 17 and 20, and are programming beginners. In this class, programming is not the most important subject.

We created a fifteen-question quiz covering the concepts included in the official curriculum for this level. We focused on variables, operator priority rules, loops, conditional structures, functions and arrays. The first three questions deal with variables (naming a variable, initialization and use of variables), two with operator priority, three with loops, two with conditional structures, two with functions and three with arrays. This quiz is carried out after class sessions at the end of the semester, its aim being to assess the students' learning. Moreover, we used a second anonymous quiz containing 10 questions based on the likert scale to measure learners' attitudes towards learning programming. The results of the learners of this experiment are presented in the next section.

IV. resultats And Discussion

This study has highlighted the difficulties of learning to program. It has shown that these difficulties have different causes and some of them are introduced in the section 2. We focused on basic programming concepts, since the learners are programming beginners. This section presented the results of learners in two quizzes. These results showed that these concepts, in particular variables, loops and operator precedence rules, caused difficulties for learners. Table 1 and

Figure 1 describe the results of student responses by concept. This results show that 54,43% of students gave incorrect answers to questions about the variables. Analysis of the responses showed that some learners did not attach any importance to the naming of variables and even that they did not master the rules for assigning names to variables. They also have difficulties using variables (initialisation, change of value). For example, to calculate the n terms of a sequence of order 1, they use several variables as in mathematics, rather than a single variable that takes the value of one term in the sequence each time. Also, as in mathematics, they consider that the equality x=3 is the same as 3=x, even though this is incorrect in programming.

Learners' results to the questions on loops showed that 64.14% of students gave incorrect answers. The study of the answers showed that learners don't know the difference between for and while loops, and their uses. Learners who don't know variables can't assimilate the counter in loops.

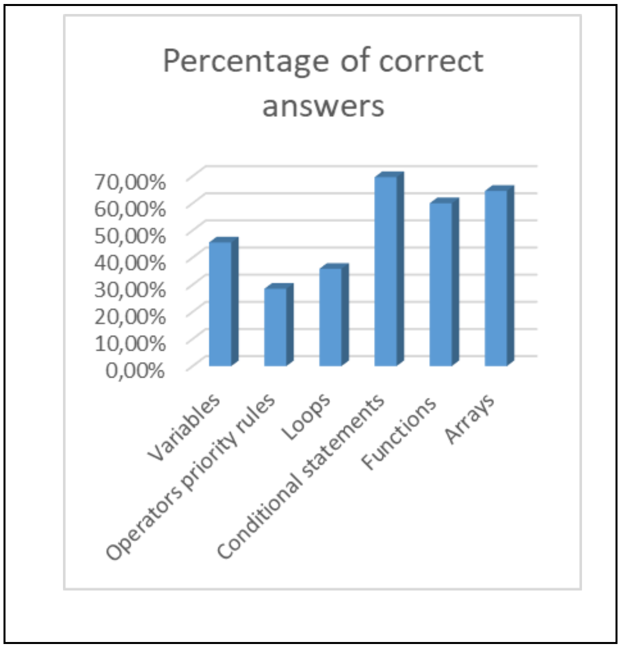
The study of the answers to questions relating to operator priority rules showed that 71,52% of the answers were invalid. Learners make mistakes when they evaluate an expression using operators with different priorities. They always apply the same rules as in mathematics, whereas in some languages the evaluation of the priority of operators is always done from left to right, independently of the operators.

Analysis of the rest of the results shows that learners have difficulty choosing the array index. They think that the counter can be of type float and that, on the other hand, it cannot be of type char. Some learners confused the if statement with loops such as while.

TABLE I. ANSWERS OF STUDENTS PER CONCEPT

	Correct answers	**Incorrect answers**	**Percentage of correct answers**	**Percentage of incorrect answers**
Variables	108	129	45,57%	54,43%
Operators priority rules	45	113	28,48%	71,52%
Loops	85	152	35,86%	64,14%
Conditional statements	110	48	69,62%	30,38%
Functions	111	47	70,25%	29,75%
Arrays	153	84	64,56%	35,44%

Figure 1. Percentage of correct answers per concept



After the first quiz, we carried out a direct interview with the students to identify the most difficult concept from the list of concepts studied. The majority of students considered that the variable was the most difficult concept. It causes a lot of confusion among learners. Moreover, mastery this concept is indispensable for assimilating the other concepts such as loops, arrays.

We also examined the effect of gender on coding learning difficulties. The results by gender are detailed in Table 2. These results show that the percentage of correct answers for girls is 24.31% of the total answers, compared to 27.33% for boys. We can conclude that these results are almost similar, so gender has no effect on learning to program and both have difficulties.

TABLE II. PERCENTAGE OF ANSWERS PER GENRE.

	Percentage of correct answers	Percentage of incorrect answers
boys	27,33%	22,91%
girls	24,31%	25,45%

We also studied learners' attitudes to learning programming in a second quiz. The results are summarized in Table 3. Analysis of learners' attitudes showed that 70.89% of them had a negative impact on learning programming. Most of them said that programming was a difficult and unattractive subject. 10% of learners expressed an interest in studying it in higher education. We could not observe any relationship between gender and learners' attitudes towards the learning program, as the attitudes of girls and boys were almost similar.

TABLE III. ATTITUDE TOWARDS LEARN PROGRAM PER GENRE

	Boys	**Girls**	**Total of learner**
Positive attitude	12	5	17
Negative attitude	38	18	56
Neutral attitude	2	4	6

This study has shown that learning to program still presents difficulties for learners, particularly beginners and those for whom learning to program is not a subject. Section 2 shows that there are different types of these difficulties. The literature has shown that these difficulties have reduced learner motivation. For this reason, it is essential to use attractive and motivating tools to assist programming learning, such as the use of serious games. Numerous studies have shown that serious games enhance learning and motivation. As learners have different skills and learning styles, the game needs to take these different learner profiles into account, in order to offer them tasks tailored to their skills and maintain their motivation and commitment. This adaptation is achieved using artificial intelligence techniques.

V. CONCLUSION

In this paper, we have studied the difficulties of learning programming for learners, particularly beginners. We have proposed a new classification of these difficulties. We carried out an experiment with a class where learning programming is not a core subject. The study covered the basic concepts of programming. Learners have shown an interest in learning programming despite declaring that it is difficult. The results showed that variables, operator priority rules and loops are the most difficult concepts. The study showed that gender has no effect on learning to program.

In order to help students and teachers overcome these difficulties, we propose the use of interactive and attractive tools such as serious games. However, the use of a static serious game for all learners is ineffective, so it is necessary for the game to take into account the profile of each learner. In our future work, we plan to design, develop and apply an adaptive serious game for learning programming, using the technique of intelligent learning.

REFERENCES

[1] P. Toukiloglou et S. Xinogalos, « Adaptive Support With Working Examples in Serious Games About Programming », Journal of Educational Computing Research, vol. 61, no 4, p. 766-789, juill. 2023, doi: 10.1177/07356331231151393.

[2] S. A. Houssein et Y. Peter, « Outils d'assistance et les difficultés d'enseignement / apprentissage de la programmation, quelle aide ? », p. 8, 2017.

[3] Simon et al., « Pass Rates in Introductory Programming and in other STEM Disciplines », in Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, Aberdeen Scotland Uk: ACM, déc. 2019, p. 53-71. doi: 10.1145/3344429.3372502.

[4] C. Watson et F. W. B. Li, « Failure rates in introductory programming revisited », in Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14, Uppsala, Sweden: ACM Press, 2014, p. 39-44. doi: 10.1145/2591708.2591749.

[5] M. Butler, J. Sinclair, M. Morgan, et S. Kalvala, « Comparing international indicators of student engagement for computer science », in Proceedings of the Australasian Computer Science Week Multiconference, 2016, p. 1-10.

[6] A. Gomes et A. J. Mendes, « Learning to program-difficulties and solutions », in International Conference on Engineering Education–ICEE, 2007.

[7] J. Oroma, H. Wanga, et F. Ngumbuke, « Challenges of teaching and learning computer programming in a developing country: lessons from Tanzania », in INTED2012 Proceedings, IATED, 2012, p. 3820-3826. Consulté le: 11 janvier 2024. [En ligne]. Disponible sur: https://library.iated.org/view/OROMA2012CHA2

[8] Y. Bosse et M. A. Gerosa, « Difficulties of programming learning from the point of view of students and instructors », IEEE Latin America Transactions, vol. 15, no 11, p. 2191-2199, 2017.

[9] Y. Bosse et M. A. Gerosa, « Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage », SIGSOFT Softw. Eng. Notes, vol. 41, no 6, p. 1-6, janv. 2017, doi: 10.1145/3011286.3011301.

[10] P. Piwek et S. Savage, « Challenges with Learning to Program and Problem Solve: An Analysis of Student Online Discussions », in Proceedings of the 51st ACM Technical Symposium on Computer Science Education, Portland OR USA: ACM, févr. 2020, p. 494-499. doi: 10.1145/3328778.3366838.

[11] R. Kadar, N. A. Wahab, J. Othman, M. Shamsuddin, et S. B. Mahlan, « A study of difficulties in teaching and learning programming: a systematic literature review », International Journal of Academic Research in Progressive Education and Development, vol. 10, no 3, p. 591-605, 2021.

[12] T. F. B. A. Rahman, N. Anuar, et R. F. M. Said, « How the nature of programming and learning materials affects novice learner's motivation and programming ability », in Proceedings of the 6th International Conference on Information and Education Technology, Osaka Japan: ACM, janv. 2018, p. 124-128. doi: 10.1145/3178158.3178184.

[13] C. S. Cheah, « Factors contributing to the difficulties in teaching and learning of computer programming: A literature review », Contemporary Educational Technology, vol. 12, no 2, p. ep272, 2020.

[14] A. V. Robins, « 12 novice programmers and introductory programming », The Cambridge handbook of computing education research, p. 327, 2019.

[15] M. M. Mhashi et A. Alakeel, « Difficulties facing students in learning computer programming skills at Tabuk University », Recent Advances in Modern Educational Technologies, p. 15-24, 2013.

[16] Y. Qian et J. Lehman, « Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review », ACM Trans. Comput. Educ., vol. 18, no 1, p. 1-24, déc. 2017, doi: 10.1145/3077618.

[17] S. Mohamed Shuhidan, M. Hamilton, et D. D'Souza, « Understanding novice programmer difficulties via guided learning », in Proceedings of the 16th annual joint conference on Innovation and technology in computer science education, Darmstadt Germany: ACM, juin 2011, p. 213-217. doi: 10.1145/1999747.1999808.

[18] Y. Qian et J. D. Lehman, « Using Targeted Feedback to Address Common Student Misconceptions in Introductory Programming: A Data-Driven Approach », SAGE Open, vol. 9, no 4, p. 215824401988513, juill. 2019, doi: 10.1177/2158244019885136.

[19] X. Zhang, C. Zhang, T. F. Stafford, et P. Zhang, « Teaching Introductory Programming to IS Students: The Impact of Teaching Approaches on Learning Performance », vol. 24, 2013.

[20] S. Xinogalos, « Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy », Educ Inf Technol, vol. 21, no 3, p. 559-588, mai 2016, doi: 10.1007/s10639-014-9341-9.

[21] A. J. Gomes, A. N. Santos, et A. J. Mendes, « A study on students' behaviours and attitudes towards learning to program », in Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education, Haifa Israel: ACM, juill. 2012, p. 132-137. doi: 10.1145/2325296.2325331.

[22] M. Baser, « Attitude, gender and achievement in computer programming. », Online Submission, vol. 14, no 2, p. 248-255, 2013.