Turing – Inter School Programming Contest: Pedagogical Innovation in Programming Teaching for Middle Schools

Rui Figueiredo
Agrupamento de Escolas Alcaides de Faria, Barcelos, Portugal
ruifigueiredo@aeaf.edu.pt

Bárbara Cleto
Escola Secundária Henrique Medina, Esposende, Portugal
550bcleto@eshm.edu.pt

José Manuel Cerqueira
Agrupamento de Escolas de Barcelos, Portugal
jose.cerqueira@aebarcelos.pt

Abstract

Turing is an interscholastic tournament that aims at promoting the teaching of informatics, particularly the learning of programming through gamification. It is a competition between secondary schools, organized by teachers of informatics, for their own students. Turing was developed due to the lack of tournaments and competitions organized by teachers in this level of education. By contrast, universities and polytechnic institutes regularly organize programming tournaments, aimed at students of both secondary schools and universities. Given that its Turing is a pilot project, the first edition of the tournament will take place in March 2020 and it will occur simultaneously in three secondary schools. The students who are (voluntarily) enrolled in Turing will have an hour and thirty minutes to solve a set of problems in C programming language via Web, through the E-Learning platform Moodle while using the plugin CodeRunner.

2012 ACM Subject Classification Computing methodologies → Logic programming and answer set programming

Keywords and phrases CodeRunner, Contest, Game, Gamification, Programming, Turing

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.9

1 Introduction

Nowadays, computer technology, computational thinking and programming languages are as important as other essential subjects in schools [15]. The relevance of these domains, particularly computational thinking is stated in PISA 2021 Mathematics Framework [1], which refers to

> "The increasing and evolving role of computers and computing tools in both day-to-day life and in mathematical literacy problem solving contexts is reflected in the recognition in the PISA 2021 framework that students should possess and be able to demonstrate computational thinking skills as they apply to mathematics as part of their problem-solving practice. Computational thinking skills include pattern recognition, designing and using abstraction, pattern decomposition, determining which (if any) computing tools could be employed in analysing or solving a problem, and defining algorithms as part of a detailed solution".

[1] http://www.oecd.org/pisa/pisaproducts/pisa-2021-mathematics-framework-draft.pdf

First International Computer Programming Education Conference (ICPEC 2020).
Editors: Ricardo Queirós, Filipe Portela, Mário Pinto, and Alberto Simões; Article No. 9; pp. 9:1–9:7
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Skills such as analysis, synthesis and evaluation are required and promoted while approaching these subjects. Algorithms and programmes interlink. Programming languages, redundancies of specifications/algorithms, allow the representation through encoding (creating programmes) of the (re)solution of problems. Therefore, when it comes to teaching skills that are related to programming languages, it is important to consider that programming is much more than the writing of a set of code lines in a given language [5] and that to teach to programme is essentially to teach how to think [3].

2 Background

The goal of teaching programming languages is to allow the students to develop skills such as observation, comprehension, analysis, reflection, logical thinking and autonomy, consequently acquiring basic tools and elementary knowledge that are necessary to develop programmes that are able to solve real problems [5]. However, the students reveal some difficulties in syntactic, conceptual and strategic knowledge [11]. These difficulties are related to several factors, namely the unfamiliarity of syntax, natural language, mathematical knowledge and imprecise mental models [11]. Furthermore, they reveal some difficulty in structuring the problems that are proposed to them, even when these are similar to decision-making situations that constitute the normal scenario of their daily routines [5, 14]. Thus, this problem is posed to every single intervenient of the teaching-learning process; teachers ask themselves what to teach and how to do it [14] and the students wonder how to achieve the necessary skills, given that programming implies thinking, solving, defining and formalizing [14]. It is also important that the students are motivated and involved in the process of learning how to programme, given that this area requires a considerable amount of effort in the first stages of learning [8]. The complexity and difficulty of the process results, at times, in demotivation and in the quitting from the programming courses by some students [16]. This factor is a matter of concern for those who are committed to teach subjects such as Introduction to Computer Programming, Programming Languages and other similar subjects. There are several strategies to address the problems that were identified [16, 12, 18, 5, 14], which encompass several types of computer systems that support the learning of programming, recurring to visual representations and animations of algorithms, programming languages based in icons, systems of intelligent tutors, and microworlds of learning [5], games, gamification, programming tournaments, among others.

2.1 Gamification

A new approach to learning [10] named Game Based Learning (GBL), looks at enhancing the motivation and the involvement of students in matters related to learning through educational games. These games, whose strengths are fantasy, curiosity, challenges and control are called Serious Games (SG) [20]. A more recent trend considers that students can also create their own games and, by doing so, develop solutions to problems and therefore, other skills in the areas of Computational Thinking and/or programming. As the research in this area is developed, new concepts are created, such as Edutainment, "Games for Change" [19], Playful Learning or Learning through play [13] e Gamification [4], in which elements of game design are used and included in contexts that are not necessarily related to the game. These concepts, particularly Gamification, were used as the starting point for the creation of a programming tournament which, along with the automatic evaluation of programming problems, would appeal to the students and motivate them when it comes to this particular area of study. Several studies have demonstrated that Gamification, when used in appropriate conditions, can create a favorable environment to the students and potentially increase the interest of the students in programming [9].

2.2 Supporting tools for the learning of programming (with Learning Management System – LMS)

Programming implies a way of studying which is considerably different from other subject areas, by demanding intensive practicing and a solid background of mathematical knowledge (such as number theory, logic, and others) and problem-solving. Attending classes and studying with books is not enough; programming demands intensive work done outside the classroom [5]. The use of an LMS system that might be applied in any other place, beyond the curricular timetable as part of autonomous studying can potentiate the learning of programming [5]. The positive reinforcement in the answers, the fast interaction, the (appealing) Web fact, the ‘awards’ as a reward of excellent work, and the ability to work anywhere, anytime, are all advantages in the use of a LMS system as a complement of an introductory subject. An LMS system appropriately adjusted and equipped presents the following strengths when it comes to learning and teaching programming:

- Orienting the correct solving of problems, forcing the student to an intensive practice, in which s/he has to follow every stage that encompasses comprehending, characterizing, representing, solving and reflection on the obtained solution [14];
- Allowing the students to create and simulate their own algorithms and the analysis of the results, while also correcting certain aspects that might have been less successfully developed [5];
- Verifying if the algorithms written by the students are behaving correctly with tests designated by the teacher (entry data and expected results), in an autonomous manner and without the concerns of being evaluated [5];
- Allowing the student to self-evaluate his/her knowledge though the simulation and the testing of their resolutions, giving the student an elevated degree of confidence in the system, as well as in his/her own abilities [5].

Finally, another aspect which is more global from a pedagogical point of view is the great advantage of giving prompt feedback, as well as feedback adjusted in time (formative evaluation).

2.3 Automatic evaluation of programming problems

The relevance of the evaluation of the code of a programme written by a student is related to the supply of data to the teachers, which allows them to assess the level of development of the student and therefore guarantee that the learning outcomes are achieved in order to correct courses [17], i.e., to guide teachers in making decisions that direct and determine the educational processes that are developed with their students. To give students the possibility to solve programming exercises by themselves is essential to develop their capacities. In order to fulfil this, the students need feedback, in an immediate and continuous manner, on their progress, as well as on their difficulties. However, providing individual and timely feedback is highly demanding for a teacher [12]. A likely solution for this problem, one that has been effective and a motivational factor, encompasses the use of on-line systems that enable the automatic evaluation of programming problems [18].

3 Related work

There are several initiatives that aim at promoting digital literacy in computer sciences, some more directed at computational thinking, others specifically related to programming activities, in which the programming contests are included. When it comes to computational thinking, endeavors such as Hour of Code[2] and Code Week[3] are normally destined to the general public and allow any person, anywhere, to organize or take part in coding activities. Hour of Code integrates the event "Education Week in Computer Sciences", which takes place in December for students around the world. This event consists in promoting one-hour long programming activities in a worldwide scale. In Europe, Code Week organizes the European Week of Programming every October. It is funded by the European Commission and it aims at taking programming and digital literacy to everyone in an engaging and playful way. Furthermore, Bebras[4]is an international initiative that aims at promoting Informatics, particularly Computer Sciences and Computational Thinking among students of all ages. The first edition of Bebras in Portugal was hosted by the Department of Computer Sciences of the Faculty of Sciences of the University of Porto (Departamento de Ciência de Computadores – DCC/FCUP, Faculdade de Ciências, Universidade do Porto) Regarding contests that are specifically dedicated to programming, the ones that already exist are promoted and held by universities or polytechnic institutes and are exclusively designed to university students, such as: i) Student Tournament of Multilanguages of Aveiro (TECLA[5] – Torneio Estudantil de Computação multiLinguagem de Aveiro), promoted and developed by the Águeda School of Technology and Management (Escola Superior de Tecnologia e Gestão – ESTGA) of the University of Aveiro; ii) Topas[6], a tournament of programming designed for secondary education students, organized by the Department of Computer Sciences of the Faculty of Sciences of the University of Porto or iii) the Informatics Olympics (Olimpíadas da Informática[7]), hosted by the Association for the Promotion and Development of the Information Society (Associação para a Promoção e Desenvolvimento da Sociedade da Informação (APDSI) in collaboration with the Department of Computer Sciences of the Faculty of Sciences of the University of Porto. For the younger public, there is the National Contest of Programming in Scratch: Creating with Scratch (Concurso Nacional de Programação em Scratch: A Criar Com Scratch[8], promoted by the Centre of ICT Skills of the School of Education of the Polytechnic Institute of Setúbal (Centro de Competência TIC, Escola Superior de Educação, Instituto Politécnico de Setúbal), as part of the project EDUSCRATCH, in partnership with the Directorate General for Education (Direção-Geral da Educação – DGE) of the Ministry of Education and Science (Ministério da Educação e Ciência) and the Committee of Protection of Children and Youth (Comissão de Proteção de Crianças e Jovens – CPCJ) of Setúbal. All these tournaments are organized by universities or polytechnic institutes with the purpose of appealing to students for the study of these specific areas. These contests intend to provide the students with an opportunity to show and improve their knowledge and skills in the solving of problems, by looking at programming not just as the activity of writing programmes, but as the act of developing software, implying that it must be perceived as a team activity [6].

[2] https://hourofcode.com/pt/
[3] https://codeweek.eu/
[4] http://bebras.dcc.fc.up.pt/index.html
[5] http://tecla.estga.ua.pt/
[6] https://topas.dcc.fc.up.pt/
[7] http://oni.dcc.fc.up.pt/2019/
[8] http://projectos.ese.ips.pt/eduscratch/index.php/42-videos/526-a-criar-com-scratch-2020

4 Turing – Inter School Programming Contest

Turing is a tournament promoted and organized by a group of teachers that work in several school groups: Agrupamento de Escolas Alcaides de Faria, Agrupamento de Escolas de Barcelos and Escola Secundária Henrique Medina. Every student that attends these schools can individually participate in the tournament. More information can be found https://turing.pt. The tournament takes place simultaneously in the headquarters of every school group that the students attend and consists of a test that lasts for an hour and thirty minutes, in which a set of problems is intended to be solved using the C programming language. The online platform that is meant to be used in the tournament combines Moodle[9],vastly used in schools, and CodeRunner[10], a plug-in.

4.1 Moodle

A vast majority of Secondary Schools and Middle Schools has Web servers with Moodle as a support platform for learning and teaching. Moodle is a system for the management of learning. It is a platform with a virtual desktop, in which it is possible to create courses, add resources such as videos, images, documents, databases, forums, among others [8] and that allows, with the version 2.5., to grant medals or badges to the students [8, 1], enabling the teacher to implement Gamification. There are three distinct ways of giving badges: i) they can be given manually (the medals are created and the teacher can choose when and to whom s/he will be giving them to) ii) after finishing a course or iii) after finishing a certain activity as part of a course, with the medal being automatically added to the student's profile. The badges have a name, a description, an image and the name of the person who assigned it [8, 1]. Another technique that helps in the use of Gamification in Moodle is "level up!" [8, 2]. Thereby, by taking advantage of that experience that was already acquired, a model that could integrate Moodle in an easy and clear manner was searched for. The chosen model was CodeRunner, a module that incorporates a question-type plugin, which can be easily integrated in one of the common questionnaires of Moodle, with the advantage of using its inherent abilities of integration and adaptability. Moreover, the positive experience with the use of CodeRunner by the authors of this article when teaching introductory courses of programming was also a determining factor in the choice of the model. The use of Virtual Programming Lab[11] (VPL) was also considered but this model works as an activity of Moodle that includes several features that are considerably complex, such as revision and verifying plagiarism, tools that transcend the needs of the introductory courses of computer programming in both Secondary Education and 3rd Cycle levels (7th to 9th grade).

4.2 CodeRunner

CodeRunner is an open-source module, free for Moodle and developed in the University of Canterbury in New Zealand, which allows teachers to define questions to include in the questionnaires of Moodle, whose answers are encoded in a particular programming language, such as C, C++, C sharp, Java, JavaScript (NodeJS), Python, PHP, Octave (Matlab), among others. On the other hand, the students develop and test their code using a normal environment of development. When they consider that their answer is correct, they submit

[9] https://moodle.org/?lang=pt
[10] https://coderunner.org.nz/
[11] https://vpl.dis.ulpgc.es/

the code, using a Web browser, through the editor that is provided by CodeRunner, which is associated with the given question. A fundamental pre-condition that lies behind CodeRunner is that the questionnaire in which the questions are inserted is executed in the adaptable mode of Moodle, giving immediate feedback to the students on the accuracy of their answers and allowing them to re-send a corrected answer, with the downside of suffering a certain penalty [7].

4.3 Turning Tournament

Turing, by being organized by a group of teachers, is intended to not only encourage the interest for programming within a younger public, but also to be a method of advertising and promoting innovative pedagogical methodologies, as well as good practices in teaching subjects such as Introduction to Programming (and other similar ones) in Secondary Education and in the 3rd Cycle Schools, by showing that it is possible, with the limited resources that schools own to establish ground-breaking systems which complement the processes of teaching and learning that are up-to-date and motivational for students. Turning is also intended to be distinct from other contests organized by Higher Education institutions, by answering directly to the national demands that are part of the Exit Profile of Students Leaving Compulsory Education (Perfil dos Alunos à Saída da Escolaridade Obrigatória), particularly to the Essential Outcomes (knowledge, abilities and attitudes) regarding skills in Computer Programming.

5 Conclusion and Future Work

In this article, several important aspects related to the learning and teaching of programming were discussed. Taking into account the relevance that programming languages have been gaining in schools and societies, a tournament of computer programming, organized by secondary education teachers and targeting students of that same level of education, with some amount of knowledge of programming was also presented in this paper. The tournament named Turing is still at its early stages of creation and it will only take place in March 2020. Given the results that may be achieved in its first edition and after evaluating its impact, some additional research will be carried out regarding the technical (used tools) and pedagogical (strategies and methods involved in learning) levels of the project. In the future, new schools may be included within the same framework in which Turing was developed. There is also the hypothesis of including an exclusive edition of Turing adapted to 3rd Cycle (7th to 9th grade) students. By adding new plug-ins, such as level-up, it will also be possible to use it as a method of evaluating the students who enrol in the tournament.

References

1 Marcelo Claro. Badges no Moodle - Gamification - Moodle Livre, 2014. URL: `https://www.moodlelivre.com.br/tutoriais-e-dicas/potal/tutoriais-e-dicas-moodle/badges-no-moodle-gamification`.

2 Marcelo Claro. Configurar Condicionais no Moodle - Moodle Livre, 2016. URL: `https://www.moodlelivre.com.br/tutoriais-e-dicas/1606-configurar-condicionaisno-moodle`.

3 Ole-Johan Dahl, Edsger Wybe Dijkstra, and Charles Antony Richard Hoare. *Structured programming*. Academic Press Ltd., 1972.

4 Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15, 2011.

5 Anabela Gomes, Joana Henriques, and António Mendes. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias-ISSN 1646-933X*, 1(1):93–103, 2008.

6 Pedro Guerreiro. A mesma velha questão: como ensinar programação? *Mexico City: UNAM*, 1986.

7 Richard Lobb and Jenny Harlow. Coderunner: A tool for assessing computer programming skills. *ACM Inroads*, 7(1):47–51, 2016.

8 Jorge Adolfo David Castro Monteiro. Ludificação do ensino da programação: Um caso de estudo, 2017.

9 Stamatios Papadakis and Michail Kalogiannakis. Using gamification for supporting an introductory programming course. the case of classcraft in a secondary education classroom. In *Interactivity, Game Creation, Design, Learning, and Innovation*, pages 366–375. Springer, 2017.

10 Marc Prensky. Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1):21–21, 2003.

11 Yizhou Qian and James Lehman. Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–24, 2017.

12 Ricardo Queirós and José Paulo Leal. Ensemble: An innovative approach to practice computer programming. In *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming*, pages 173–201. IGI Global, 2015.

13 Mitchel Resnick. Edutainment? no thanks. i prefer playful learning. *Associazione Civita Report on Edutainment*, 14:1–4, 2004.

14 Sónia Rolland Sobral and Pedro Cravo Pimenta. O ensino da programação: exercitar a distância para combate às dificuldades, 2009.

15 Seyyed Meisam Taheri, Minoru Sasaki, Jiangcheng Oliver Chu, and Harrison Thuku Ngetha. A study of teaching problem solving and programming to children by introducing a new programming language. *The international journal of e-learning and educational technologies in the digital media (IJEETDM)*, 2(1):31–36, 2016.

16 Paula Tavares, Elsa Gomes, Pedro Henriques, and Maria João Pereira. Técnicas para aumentar o envolvimento dos alunos na aprendizagem da programação. In *VII Congresso Mundial de Estilos de Aprendizagem, CMEA'2016*, pages 1565–1577. Instituto Politécnico de Bragança, 2016.

17 Zahid Ullah, Adidah Lajis, Mona Jamjoom, Abdulrahman Altalhi, Abdullah Al-Ghamdi, and Farrukh Saleem. The effect of automatic assessment on novice programming: Strengths and limitations of existing systems. *Computer Applications in Engineering Education*, 26(6):2328–2341, 2018.

18 Elena Verdú, Luisa M Regueras, María J Verdú, José P Leal, Juan P de Castro, and Ricardo Queirós. A distributed system for learning programming on-line. *Computers & Education*, 58(1):1–10, 2012.

19 Nelson Zagalo and Dionisia Laranjeiro. Brinquedos e jogos digitais para o jardim de infância. *4. o Encontro sobre Jogos e Mobile Learning*, 2018.

20 Michael Zyda. From visual simulation to virtual reality to games. *Computer*, 38(9):25–32, 2005.