# On the Applicability of Bloom's Taxonomy and Teacher Digital Competencies for Learning how to Code in Primary Schools

**Conference Paper** · December 2023



Maja Videnovik
NGO Dig-Ed Macedonia


Ana Madevska Bogdanova
Ss. Cyril and Methodius University in Skopje


Vladimir Trajkovik
Ss. Cyril and Methodius University in Skopje

# On the Applicability of Bloom's Taxonomy and Teacher Digital Competencies for Learning how to Code in Primary Schools

Maja Videnovik[1], Ana Madevska Bogdanova[2], Elena Vlahu Gjorgjievska[3] and Vladimir Trajkovik[2]

[1] *Center for Innovations and Digital Education Dig-Ed, Solunska glava 3, Skopje, North Macedonia;*

[2] *Faculty of Computer Science and Engineering, Ss Cyril and Methodius University in Skopje, Rugjer Boshkovikj 16, Skopje, North Macedonia*

[3] *School of Computing and Information Technology, University of Wollongong, Northfields Ave, Wollongong, NSW 2522, Australia*

*maja@dig-ed.org; ana.madevska.bogdanova@finki.ukim.mk; elenavg@uow.edu.au; vladimir.trajkovik@finki.ukim.mk;*

**Abstract:**

This paper investigates the effectiveness of game-based learning and gamification in teaching coding to primary school students. The paper also examines how teachers' implementation of coding subjects in low grades in primary schools can be supported. The study is motivated by the fact that most primary teachers lack computing backgrounds and may struggle to integrate coding effectively. To address this, a pilot study was conducted in six primary schools in North Macedonia to evaluate a proposed approach for teaching coding using game-based learning and gamification. In addition, the study investigates whether Bloom's taxonomy is suitable for evaluating the learning outcomes of a computational thinking course and which activities are most appropriate for teachers without formal coding training. The findings suggest that game-based learning and gamification can enhance critical thinking and problem-solving skills and help achieve educational goals for primary school students, regardless of teachers' coding skills.

**Keywords:**

coding, computational thinking, game-based learning, Bloom's taxonomy, teacher digital competences

## 1. Introduction

Computational thinking and understanding how computers execute programs are necessary 21st-century skills that can drastically influence the future labour market. Moreover, the reliance on computer code is not limited to the technology sector anymore, with an increasing number of businesses relying on it. Thus, understanding basic coding principles is becoming a necessary skill.

Educators face the challenge of defining the right learning objectives and strategies for implementing the subjects that develop students' critical thinking and problem-solving skills [1]. Since most students use the internet and mobile technologies daily, using familiar technology to learn coding seems natural. However, a logical framework for teaching computational thinking in primary education needs to be based on carefully selected tools and practices in teaching, learning, and instruction [2]. Furthermore, the subjects should be presented to address students' possible change of focus from general to specific knowledge [3].

In most countries, primary teachers who need to integrate basic coding skills in their classes have no computing background [4]. Additionally, there is little evidence of the problems teachers face and how they can integrate coding effectively [5]. Based on observations, it is crucial to support teachers in implementing the coding subject and provide different approaches for learning how to code. Furthermore, it is essential since educators generally lack training in different digital tools and regularly face the need to improve their digital literacy.

The main goal of this paper is to investigate which activities are most appropriate to be implemented in coding subjects for teachers without formal coding training. Furthermore, the paper investigates whether the educational goals' complexity concerning coding can be treated similarly to other subjects.

The test framework used in this paper builds on the approach published in [6] and includes cooperative and competitive cycles spanning several school years. Its initial cycle introduces game-based tools that lead the students to develop their computational thinking. Then, the students are acquainted with more challenging problems that require algorithm-based problem-solving skills using block-based coding. In the final phase, the students have already obtained the prerequisite knowledge and can start to learn and use different programming languages connecting coding with real-life situations.

The teacher's (educator) role in these processes varies from the facilitator in the initial phase to the mentor in the final phase of the methodology. However, the approach corresponds with teacher qualifications since the initial phase covers most (if not all) the students at a younger age, while the final phase is for some students in higher classes. Therefore, the teachers acting as facilitators do not need specific programming skills, while the teachers acting as mentors need suitable qualifications for teaching coding. That is the case with higher classes coding teachers.

Student's achievement of the educational goals while implementing the proposed methodological framework and defining the development of coding and computational thinking skills determines the framework's suitability in coding subjects. According to Bloom's taxonomy, teachers define these cognitive and educational goals, carefully adjusting learning outcomes to different complexity levels.

Bloom's taxonomy is a widely used framework for assessing students' depth of knowledge. It establishes a hierarchy of six levels (remembering, understanding, applying, analyzing, evaluating, and creating) for increasing the degree of cognitive expertise of the student. Every level assumes the student's capability for the lower levels. Research in implementing Bloom's taxonomy in teaching programming fundamentals reveals different interpretations and revisions of Bloom's taxonomy [7]. In this paper, we aim to discuss the relevancy of Bloom's taxonomy for defining educational goals concerning the development of computational thinking and coding skills.

## 2. Methodology

Implementing innovative practices in elementary education should prioritize student-centred approaches facilitated by Information and Communication Technology (ICT) with a coaching role for teachers. The curriculum for ICT subjects and topics integrated into various subjects should adhere to principles outlined in the European Union Digital Competence Framework [8]. The suggested framework aims to provide fundamental skills and knowledge to a large student population while enabling higher programming skills for advanced students. The spiral curriculum enables revisiting specific topics in higher classes to enhance previous knowledge, increasing students' complexity levels and knowledge throughout their schooling. This progression applies to coding, with students starting with simple programs and advancing to more complex coding in subsequent years. This approach allows for minimum coding competencies in large classes while providing specialized classes with more advanced coding competencies. The proposed methodology fosters teamwork and individual learning by using various tools and educational paradigms.

Following the constructivism theory [9], a methodological approach is suggested for teaching coding by incorporating students' experiences and reflections [10]. Game-based learning [11] and project-based learning [12], and gamification elements [13] are used to enhance students' computational thinking skills by encouraging them to explore, practice, and collaborate. The aim is to increase motivation, persistence, and engagement in learning while enabling collaboration among students with varying knowledge backgrounds. The flipped classroom approach [14] allows students to utilize additional knowledge to achieve learning outcomes related to coding, with new knowledge built upon reflecting and applying previously acquired knowledge. By using collaboration, game-based, and project-based learning, student-centred learning is established, and the teacher's primary role is to facilitate this collaboration and encourage students' reflection on their learning. After introducing coding concepts through a collaborative approach, students master the learning topics individually.

The methodology for developing students' coding skills consists of three mandatory cycles and one optional cycle, with students repeating practice on coding principles using different tools in each cycle. This graduate approach enables students to progress according to their interests. Many games, programming platforms, and tools can be utilized in this approach, with "Scottie Go!" [15], "Scratch" [16], "Micro:Bit" [17], and "Python" [18] being the specific tools selected in this instance (Figure 1). However, the framework focuses on skills and learning approaches rather than tools.

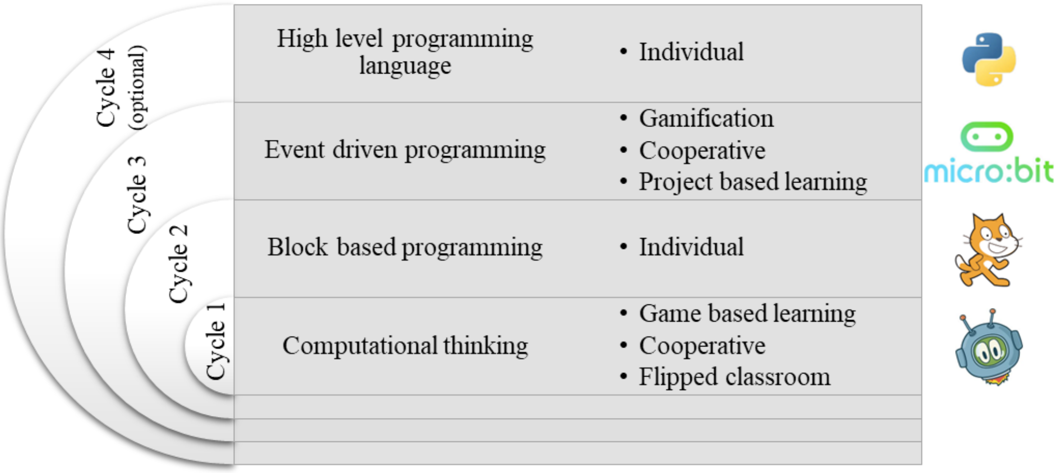


Figure 1. A methodological framework for developing students' coding skills

The proposed approach aims to teach coding using games and gamification techniques to engage students in learning [19, 20]. Through this process, students develop problem-solving skills, computational thinking, knowledge of algorithms and programming concepts, and the ability to create programs in various programming environments. However, the level of acquired coding knowledge and skills according to the educational goal complexity is the topic of further discussion. Furthermore, obstacles such as teachers' lack of digital competencies [21] can prevent the successful integration of the proposed methodology. To overcome these challenges, the approach is designed to progress students from understanding basic programs to upgrading their skills by creating solutions and then mastering their knowledge by demonstrating the practical use of their programming skills in new situations. However, the primary goal of the approach is to teach children how to think differently and experiment creatively to solve problems, which builds their confidence in a fun and exciting way.

Therefore, the research questions raised by this approach are whether the Bloom taxonomy is relevant for describing educational goals for coding skills and whether this approach can minimize the level of required coding skills among teachers.

## 2.1. Participants

A mixed-method pilot study was conducted in six primary schools in North Macedonia, using interviews with the teachers and a survey for the students. The study evaluated a proposed approach for teaching coding using game-based learning and gamification. The approach was implemented as part of the compulsory coding subject for students from the third grade (8 years old).

The study used "Scottie Go!" and "Micro:bit" as tools for learning to code. 121 students participated in the survey about "Scottie Go!" with 52% male and 48% female students, 76% from urban and 24% from rural areas. The "Micro:bit" survey was shared among 78 students, with 50.6% male and 49.4% female, as well as 66.2% from urban and 33.8% from rural areas. The survey was conducted in four well-equipped schools in urban areas and two not-well-equipped schools in rural areas.

Six teachers teaching coding classes in the surveyed schools were interviewed. The evaluation of the methodological framework aimed to investigate the relevance of Bloom's taxonomy for describing educational goals for coding skills and whether this approach minimizes the level of needed coding skills among the teacher population.

## 2.2. Procedure and Instruments

The study used two tools: the “Scottie Go!” board game and the BBC “Micro:bit” platform, along with the popular coding environments “Scratch” and “Python”. The focus is put on “Scottie Go!” and “Micro:bit” as tools that promote collaborative learning environments through game-based learning and gamification.

The “Scottie Go!” game combines physical cardboard tiles with a mobile app that sets coding tasks and scans proposed code solutions. Students work in groups to create coding instructions and solve tasks, using their previous experience playing mobile and board games to engage with the learning process. The teacher acts as a facilitator, helping students share solutions and develop reflection skills without requiring additional teacher training in coding.

The “Micro:bit” tool is presented to the students as a tool needed to develop an interdisciplinary project. Students work in teams, with a small competition for the best project. In this case, the teacher also acts as a facilitator, although some understanding of “Scratch” is required since “Micro:bit” can be programmed using blocks, similar to “Scratch”. The study found no need for additional teacher training in coding, especially because “Scratch” was used in a previous methodological framework cycle.

The teaching materials used in the study cover the rules (syntax), environment, and problems that need to be solved. Students develop computational thinking skills during the play, demonstrating decomposition of the problem, logical thinking, abstraction, finding patterns, creating algorithms, and evaluating the created program. The students' learning tasks are typical problem-solving assignments, such as how to most efficiently get from one point to another using a predefined set of rules and avoiding obstacles.

The study [22] aimed to evaluate the effectiveness of this methodology in achieving educational goals of varying complexity. The evaluation was conducted through a short test consisting of 10 multiple-choice questions concerning the linear structure in programming, which aimed to measure the level of retention of students' learning with a playful approach. According to Bloom's taxonomy, the questions in the test refer to different levels of achieving learning outcomes: remembering facts, comprehension, applying the knowledge in new situations and higher-thinking skills (analysis, evaluation and creating).

The interview with the teachers that implemented the proposed methodology in their schools was conducted to investigate the acceptability of the methodological approach and the level of teachers' competencies necessary for its implementation. Additionally, the teachers' opinions regarding the benefits of this approach for developing students' computational and coding skills were collected.

# 3. Results and discussion

The study results concerning the students' achieved learning outcomes are presented in Figure 2, showing the percentage of students' correct answers on questions with different complexity levels. The results revealed that students using "Scottie Go!" exhibited more significant achievement in more complex educational goals than "Micro:bit". Specifically, game-based learning and flipped classrooms (“Scottie Go!”) led to more significant achievement of more complex educational goals. These results are expectable because “Micro:bit” is proposed for use in the third cycle, which specifically emphasizes the comprehension of programming tools and requires prior coding knowledge and skills. In addition, all students achieved educational goals that required only the reproduction of factual information, and the percentage of correct answers decreased with the questions' complexity increment.

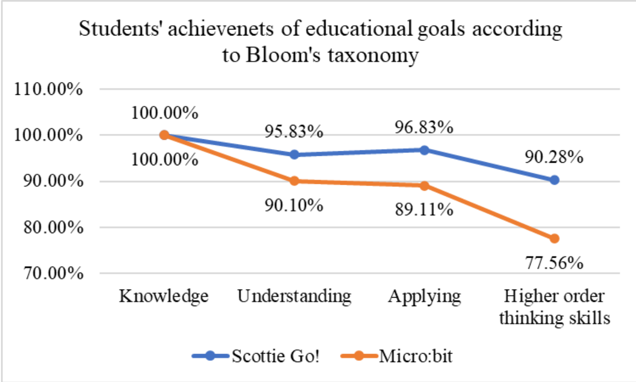


Figure 2. Students' achievement of educational goals with different complexity levels

The results indicate that students had slightly better outcomes in applying their knowledge in new scenarios than in comprehending coding concepts when learning with "Scottie Go!". This finding raises an important question regarding the suitability of Bloom's taxonomy, commonly used to evaluate students' cognitive knowledge in assessing the learning outcomes of a computational thinking course. Notably, experts in the field are divided on interpreting Bloom's taxonomy in computational thinking tasks [23]. For instance, the taxonomy considers creating as a more challenging task than understanding. However, it is debatable whether creating a basic project, such as moving a sprite from one point to another, is more cognitively demanding than fully comprehending the concept of concurrency [24]. One possible explanation for our results could be attributed to the complexity of the assigned tasks concerning applying gathered knowledge in new situations. Research indicates that in disciplines such as computer science, the new category "higher application" should be added at the top of Bloom's taxonomy as the highest level of gathered knowledge, where evaluation and creation are used during the process of applying knowledge in new situations (e.g. complex programs) [6]. However, this is inconclusive for coding using "Micro:bit", although students' answers do not differ much in these two categories (understanding and applying). This can be a result of the tasks given to students using "Micro:bit" at applying level, which require elements of creation and evaluation

The interviews were conducted with teachers to investigate their attitudes towards using the tools and teaching methods in the classroom.

The findings from the interviews with the teachers indicate that the educational tools under research were perceived to be user-friendly, and students did not encounter any difficulties using them in the classroom. All teachers who participated in the study strongly agreed that integrating the tools into their teaching practices positively impacted student motivation and engagement with learning. Furthermore, the teachers reported that the tools fostered a collaborative and stimulating learning environment and that working in teams and learning from mistakes helped students to persist in achieving their learning objectives. The teachers appreciated that using a flipped-classroom approach and gamification eliminated the need for additional training in coding. They also highlighted that their role as facilitators allowed them to emphasize problem-solving and critical thinking skills, in addition to developing coding skills, by focusing on learning outcomes. One of the respondents also reports that it is a useful example of using technology the students are more familiar with.

Regarding the use of technology as a pedagogical tool, the technology was used to assess students' learning of applying theoretical concepts in practice and to detect where they have challenges, establishing guidelines for improvement. In cases where the students have difficulties using the tool, the role of the teacher is to combine the students with difficulties with students who work on similar projects but have no difficulties implementing them. This approach is an excellent way to promote student collaboration. However, from the teachers' perspective, the students are more motivated and inspired, and they focus more on learning outcomes by establishing needed functionalities with different tools. In parallel, students master the technological tools as well. The competitive cycle enhances the engagement and students' eagerness to do better.

Based on the interview results, the teachers believe that "Scottie Go!" and "Micro:bit" can effectively enhance critical thinking and problem-solving skills and achieve educational goals. Interestingly, some teachers who lacked prior coding skills expressed hesitancy in implementing these tools in the classroom. However, after seeing the students take the lead and collaborate to achieve the learning outcomes, they were satisfied with the results. The teachers could serve as facilitators, and their lack of coding skills did not hinder the attainment of learning objectives. Consequently, it can be inferred that the proposed framework reduces the coding skills required among teachers.

## 4. Conclusion

This paper explores the effectiveness of game-based learning and gamification in teaching coding skills to primary school students. The study investigated whether Bloom's taxonomy can be applied to describe educational goals for coding skills and to determine the most appropriate activities for teachers without formal coding training. The results indicated that students using more straightforward tools achieved significant educational goals, especially applying their knowledge to new scenarios. The findings suggest that Bloom's taxonomy may not be suitable for evaluating the learning outcomes of a computational thinking course.

Additionally, the study shows that teachers who lack coding skills could effectively facilitate the implementation of these tools in the classroom, and their lack of coding skills did not hinder the attainment of learning objectives. This outcome reduces the level of needed coding skills among the teachers' population. Overall, this study provides empirical evidence that game-based learning and gamification can enhance critical thinking and problem-solving skills and help achieve educational goals for primary school students, regardless of teachers' coding skills. This study contributes to understanding the effectiveness of game-based learning and gamification in teaching coding skills, which has practical implications for educators, policymakers, and researchers.

**References**:

[1] D. W. Johnson, R. T. Johnson, Cooperative, competitive, and individualistic learning, Journal of Research & Development in Education 12(1) (1978) 3–15.

[2] S. L. Mason, P. J. Rich, Preparing elementary school teachers to teach computing, coding, and computational thinking, Contemporary Issues in Technology and Teacher Education 19(4) (2019) 790-824.

[3] E. V. Soboleva, T. N. Suvorova, A. V. Grinshkun, M. I. Bocharov, Applying gamification in learning the basics of algorithmization and programming to improve the quality of students' educational results, European Journal of Contemporary Education 10(4) (2021) 987-1002. doi:10.13187/ejced.2021.4.987.

[4] P. J. Rich, B. Jones, O. Belikov, E. Yoshikawa, M. Perkins, Computing and engineering in elementary school: The effect of year-long training on elementary teacher self-efficacy and beliefs about teaching computing and engineering, International Journal of Computer Science Education in Schools 1(1) (2017) 1-20. doi:10.21585/ijcses.v1i1.6.

[5] European Schoolnet, Computing our future - Computer programming and coding Priorities, school curricula and initiatives across Europe, 2015. URL: http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0.

[6] E. Vlahu-Gjorgievska, M. Videnovik, V. Trajkovik, Computational thinking and coding subject in primary schools: Methodological approach based on alternative cooperative and individual learning cycles, 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Wollongong, NSW, Australia, IEEE, 2018, pp. 77-83. doi:10.1109/TALE.2018.8615334.

[7] S. R. Sobral, Bloom's Taxonomy to improve teaching-learning in introduction to programming, International Journal of Information and Education Technology 11(3) (2021) 148-153. doi:10.18178/ijiet.2021.11.3.1504.

[8] European Commission – EU Science Hub, The Digital Competence Framework 2.0, 2017. URL: https://ec.europa.eu/jrc/en/digcomp/digital-competence-framework.

[9] M. Muhajirah, Basic of learning theory: (behaviourism, cognitivism, constructivism, and humanism), International Journal of Asian Education 1 (2020) 37-42. doi:10.46966/ijae.v1i1.23.

[10] Y. Tran, Computational thinking equity in elementary classrooms: What third-grade students know and can do, Journal of Educational Computing Research 57(1) (2019) 3-31. doi:10.1177/0735633117743918.
[11] V. Trajkovik, T. Malinovski, T. Vasileva-Stojanovska, M. Vasileva, Traditional games in elementary school: Relationships of student’s personality traits, motivation and experience with learning outcomes, PloS one 13(8) (2018) e0202172. doi:10.1371/journal.pone.0202172.
[12] S. Safaruddin, N. Ibrahim, J. Juhaeni, H. Harmilawati, L. Qadrianti, The effect of project-based learning assisted by electronic media on learning motivation and science process skills, Journal of Innovation in Educational and Cultural Research 1.1 (2020): 22-29. doi:10.46843/jiecr.v1i1.5.
[13] M. Beatriz, J. Frez, J. Cruz-Lemus, M. Genero, An empirical investigation on the benefits of gamification in programming courses, ACM Transactions on Computing Education (TOCE) 19(1) (2018) 1-22. doi:10.1145/3231709.
[14] J. L. Bishop, M. A. Verleger, The flipped classroom: A survey of the research, in: Proceedings of the ASEE national conference, 30(9), 2013, pp.1-18. Doi:10.18260/1-2—22585.
[15] M. Videnovik, V. Trajkovik, Using Scottie Go! as Game-Based Learning Tool for Computational Thinking Course, in: ICERI2018 Proceedings, IATED, 2018, pp. 1337-1344. doi:10.21125/iceri.2018.1304.
[16] J. Fagerlund, P. Häkkinen, M. Vesisenaho, J. Viiri, Computational thinking in programming with Scratch in primary schools: A systematic review, Computer Applications in Engineering Education 29(1) (2021) 12-28. doi:10.1002/cae.22255
[17] M. Videnovik, E. Zdravevski, P. Lameski, V. Trajkovik, The BBC micro: bit in the international classroom: learning experiences and first impressions, in: 17th International Conference on Information Technology Based Higher Education and Training (ITHET), IEEE, 2018, pp. 1-5. doi:10.1109/ITHET.2018.8424786.
[18] N. Tabet, H. Gedawy, H. Alshikhabobakr, S. Razak, From Alice to Python. Introducing text-based programming in middle schools, in: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, 2016, pp. 124-129. doi:10.1145/2899415.2899462.
[19] M. Videnovik, V. Trajkovik, L. V. Kiønig, T. Vold, Increasing quality of learning experience using augmented reality educational games, Multimedia tools and applications 79(33-34) (2020) 23861-23885. doi:10.2196/32095
[20] T. Vasileva-Stojanovska, M. Vasileva, T. Malinovski, V. Trajkovik, The educational prospects of traditional games as learning activities of modern students, in: Proceedings of the European Conference on Games-Based Learning 2, 2014, pp. 746-759.
[21] E. J. Instefjord, E. Munthe, Educating digitally competent teachers: A study of integration of professional digital competence in teacher education, Teaching and teacher education 67 (2017) 37-45. doi:10.1016/j.tate.2017.05.016.
[22] M. Videnovik, E. Vlahu-Gjorgievska, V. Trajkovik, To code or not to code: introducing coding in primary schools, Computer Applications in Engineering Education 29(5) (2021) 1132-1145. doi:10.1002/cae.22369
[23] C. G. Johnson, U. Fuller, Is Bloom's taxonomy appropriate for computer science?, in: Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling ACM, New York, USA, 2006, pp. 120-123.
[24] O. Meerbaum-Salant, M. Armoni, M. Ben-Ari, Learning computer science concepts from scratch, Computer Science Education 23(3) (2013) 239-264. doi:10.1145/1839594.1839607