A Systematic Mapping of Games for Teaching Programming

Sávio L. Vieira[1], **Rafael C. Fonseca**[1], **Maurício R. A. Souza** [1]

[1]Programa de Pos-Graduação em Ciência da Computação
Instituto de Ciencias Exatas e Tecnológicas – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – CEP 37203-202 – Lavras/MG – Brazil

savioluizv@gmail.com, rafael.fonseca@educacao.mg.gov.br, mauricio.ronny@ufla.br

Abstract. *The use of games as a teaching tool is not new. In 2022, it became compulsory to teach computer science in elementary and high schools in Brazil. Serious games or educational games are a great approach to diversify teaching. The use of games is proving to be a viable methodology to complement teaching strategies. This systematic mapping aims to identify, analyze, and synthesize existing games for teaching programming. We identified 94 primary studies from 2016 to 2023. They were then classified by type of game and the learning domain and learning objective were mapped, related to categories within the programming discipline.*

1. Introduction

The challenges in learning algorithms, which involve nested instructions with multiple layers, arise due to the lack of pedagogical methods and theories that transform abstract concepts into tangible learning materials resulting in difficulties for students to understand algorithms, which can impact their analytical, creative, and problem-solving skills as adults. Researchers have identified that the difficulty lies in understanding the logic of programming itself, rather than learning programming languages. To address these challenges, authors propose the use of gamification-based tools that incorporate elements related to games, such as thinking, strategies, and mechanics, outside the actual gaming environment, to encourage specific actions, whether in problem-solving or learning. Combined with programming strategies, these tools can make learning more engaging and attractive [Rapkiewicz et al. 2007, Arimoto and Oliveira 2019, Werbach et al. 2012].

Game-based learning uses game principles and applies them in other contexts to improve engagement in a fun and dynamic way, where it is not simply creating an educational game or just a game with instructional elements, but proposing and planning pedagogical activities with clear objectives that are intended to be achieved through a tool, which in this case is the game [Tobias et al. 2014].

In the article by [Mendes et al. 2020] and [Garner et al. 2016] the authors address whether there is a need to make a new mapping, update an existing mapping, or whether there is no need to update (the SLM is current). [Garner et al. 2016] proposes an assessment based on three steps, to check whether or not there is a need to update the SLM, or make a new one. After answering the questions proposed by the authors, the need to update the SLM of [Battistella and Wangenhein 2016] was confirmed, given that it was conducted in 2015 and identified 107 games for teaching computing, mostly in the areas of software engineering, programming, networks, and algorithms. This SLM expands on the work carried out by [Battistella and Wangenhein 2016] by conducting a survey comprising studies from 2016 to May 2023 focusing exclusively on games for teaching programming.

By analyzing previous studies, this mapping seeks to identify the following research questions (RQ): RQ1 - Which games are used to teach programming?; RQ2 - What are the learning objectives of games for teaching programming?; and RQ3 - What types of games are used?

This study presents the results of a Systematic Literature Mapping (SLM), a research methodology aiming to identify, analyze, and synthesize available evidence in a specific area of interest [Petersen et al. 2008], to investigate the use of games for teaching computing. This chapter is organized as follows. Section 2 defines the scope and purpose of this SLM. Section 3 defines how the automated search strategy was carried out and the strings used. Section 4 describes the inclusion and exclusion criteria for selecting studies. Section 5 defines the data extraction and analysis strategy. Section 6 describes the results obtained from the data analysis. Section 7 presents a discussion of the main findings of the research. 8 discusses the main threats to the validity of this mapping. 9 brings the chapter to its conclusion.

2. Defining the scope

This SLM aims to identify, analyze, and synthesize existing games for teaching programming. The SLM development process was carried out in three steps: Planning, Execution, and Documentation. Step 1, planning, consists of: Elaborating the research question; Defining research questions; Defining the search String and keywords; Selecting research bases; Defining exclusion and inclusion criteria. Step 2, execution, consists of: Selecting studies; Extracting data; Analyzing data. Step 3, documentation, is Document SLM and report.

3. Search steps

The search was carried out using an automated strategy for articles, conferences, and journals. The sources used were the following digital databases: ACM Digital Library[1], IEEE Xplore[2], and Scopus[3]. The search was carried out using the following keywords: **TEACH, LEARN, EDUCATION, GAME, EDUTAINMAINT e PROGRAMMING**. The resulting search string is:"**(teach OR learn OR education) AND (game OR Edutainment) AND programming AND NOT gamification AND NOT framework**". The search was adapted for each database, and limited to the meta-data fields. Studies containing the keywords frameworks or gamification for teaching programming and its variations were not selected, given the problem in the search for the terms game and gamification presenting false positives in the results, there was a need to place the terms gamification and frameworks as items not searched for in the search bases or exclusion filters.

4. Primary Study Selection

The selection of studies was made using only primary studies in English, looking for games for teaching programming, whether digital games or unplugged (non-digital) games.

[1]https://dl.acm.org/
[2]https://ieeexplore.ieee.org/
[3]https://www.scopus.com/

To select the primary studies for this SLM, after the initial search, the exclusion criteria were first applied by reading the title, abstract, and metadata to eliminate studies that were irrelevant to the research. The inclusion criteria were then applied after reading the full text of the primary studies resulting from this stage.

The exclusion criteria are:

- Language other than English;
- Secondary study;
- Duplicate;
- Teaching programming based on game development;
- Gamified teaching, gamification or framework;
- Format other than conference or journal article;
- Publication date before 2016.

The inclusion criteria is: The study proposes, evaluates, or reports on the use of games to teach programming.

5. Data extraction and analysis strategy

By analyzing and reading the selected articles, we were able to extract answers to each research question.

For RQ1, the name of the game was extracted as described in the study; in cases where it was not possible to identify the study, it was filled in with ”name not found”. A code was then assigned starting with G01 to G94 and associated with the respective bibliographic reference in APA (American Psychological Association) format.

For RQ2, according to the description of each game, it was listed and categorized by: Category, learning objectives, target audience, and Learning Domain. In cases where this information was not described, it was inferred from the reading and contextualization of the study. The following aspects were considered for the categories mentioned:

- Category: Categories within the programming subject according to learning objective, 9 categories were used.
- Learning objectives: Formulated based on educational concepts, these are specific objectives aimed at acquiring a certain concept to be learned at the end of the activity.
- Target audience: The audience for which the study is intended, was separated by age groups: Child age group less or equal to 12yo (years old), adolescent age greater then 12yo and age less or equal to 18yo , and the others as adults.
- Learning Domain (LD): According to Bloom’s Taxonomy, LDs can be classified as: cognitive, psychomotor, and affective. Cognitive deals with intellectual development and the acquisition of knowledge. Psychomotor focuses on the development of skills. Affective deals with issues of emotional development. [Bloom 1968, Krathwohl 1973].

For RQ3, two aspects were considered: the type of game and whether it was an unplugged (non-digital) game. The following categories were used to classify the type of game: Action, Drag and Drop, Adventure, Cards, Mixed (More than one category), Other (None of the other categories present in the study), Puzzle, Quiz, Augmented Reality, RPG (Role Playing Game), Simulation and Board Game. The categories used for unplugged games were ”yes”(non-digital game), ”both”(available digitally and non-digitally), and ”no”(only available digitally).

Unplugged games are games that don't depend on electronic equipment, such as computers, cell phones, tablets, or video game consoles. These games use cards, pencils, paper, pens, or other non-electronic materials, i.e. without the need to know how to use a computer or electronic device.

To produce, analyze, and extract the systematic mapping data, the Rayyan [Ouzzani et al. 2016] tool was used, available at https://rayyan.ai, as well as the Microsoft Excel application.

6. Results

The search resulted in 5,628 articles, 1,008 from ACM DL, 1,093 from IEEE Xplore, and 3,527 articles from Scopus, before the filters and exclusions. Of this total, 137 articles were initially selected and 5,493 articles were excluded, with 43 duplicates. The total number of articles selected for this study was 94, which met the inclusion and exclusion criteria.

The search period ranged from 2016 to 2023. Analyzing the studies, there was no upward or downward movement in publications during the period analyzed. However, it is notable that in the years 2017, 2019, and 2021 there was a greater flow of publications. The year 2023 had only one publication, given that the search for this systematic mapping was carried out in March/2023.

6.1. RQ1 - Which games are used to teach programming?

After reading 137 primary studies in full, 94 games for teaching programming were selected. The games are classified by type of game. To answer this research question, a Table (Appendix A) was created, available at https://doi.org/10.5281/zenodo.10891250 containing the game code, game name, and references in APA (American Psychological Association) format.

6.2. RQ2 - What are the learning objectives of games for teaching programming?

To answer RQ2, Table 1 was made and is where the Learning Domain and learning objective of the studies were extracted. The categories within the programming discipline were then related to the learning objectives. They were extracted and summarized in 9 categories described below:

- **Basic programming concepts**: Involves fundamental concepts such as algorithms, variables and constants, functions, and classes, among other concepts that are the basis for teaching programming.
- **Software Engineering**: Encompasses the development, maintenance, management, and planning of software systems, applications, and programs. It also includes aspects such as software architecture, project management, systems analysis, and optimization.
- **Data structures**: Covers the study and implementation of organizational structures for storing and manipulating data such as children, stacks, trees, and graphs, among others.

- **Programming language**: Contains the study of formal languages used to write computer programs, including syntax, semantics, and their practical applications.
- **Programming logic**: Encompasses how to think and solve problems using a logical sequence of commands and instructions.
- **Programming paradigm**: A programming paradigm is a model, pattern, or style. Paradigms include structured, distributed, sequential, and object-oriented.
- **Computational thinking**: Involves a set of cognitive skills, such as logic, data analysis, breaking down a problem into smaller parts, analysis, and problem-solving.
- **Advanced Programming**: Works on concepts such as debugging, competitive programming, and dynamic programming, among others.
- **Basic Programming**: Involves basic programming knowledge such as control structures, functions, and basic theoretical concepts for programming.

Among the categories for teaching programming, the following stand out: basic programming concepts with 30% of the studies, basic programming and programming language with 15% of the studies each, and programming paradigm with 11% of the studies. The categories with the fewest studies were advanced programming and programming logic with 4% of the studies each, and data structures and software engineering with 6% of the studies each. Computational thinking was among the categories with the most studies and the categories with the least studies had 9% of the studies.

It is noticeable that the number of articles aimed at the cognitive learning domain (LD) of Bloom's taxonomy is the highest, and this category is the most sought after among authors, with 89% of the studies, followed by the psychomotor LD, with 11% of the studies, while the affective LD did not have any work developed in the search carried out for the teaching of programming in the period from 2016 to 2023.

6.3. RQ3 - What types of games are used?

Among the types of games listed, simulation games (31 PS) and puzzle (22 PS) are the types with the most studies, while the other types with 7 or fewer studies are adventure games (7 PS), role-playing games (6 PS), action games (5 PS), card games (4 PS) and board games (4 PS), augmented reality games (3 PS), other games (3 PS), drag and drop games (2 PS) and mixed games (2 PS).

The target audience for most studies was children and/or adolescents with 44 published studies, followed by only adolescents with 18 publications, thirdly adolescents and/or adults with 17 publications, fourthly only children with 13 publications, and lastly only adults with 2 studies. Some studies mention their target audience as "undergraduates", and this group was placed next to the "Adolescents/Adults"group. The studies are mapped in Table 2.

Table 1. Category and Learning Objetives

Category	Learning objetive	Primary studies
Advanced programming (4)	Competitive programming	G41, G69
	Debbuging	G60
	Dynamic programming	G46
Basic programming (14)	Algorithms	G03, G11, G36, G42, G78, G87
	Basic programming	G13, G67, G68
	Introductory programming	G20, G29, G34, G51, G81
Computational thinking (8)	Computational thinking	G04, G15, G32, G47
	Problem-solving	G77, G89, G93
	Planning	G64
Data Structures (6)	Arrays and linked lists	G92
	Linked lists	G23, G55, G92
	Pointers	G94
	Stack, queue	G38
Basic programming Concepts (28)	Advanced Concepts	G07
	Maps and filters	G22
	Parallel programming	G39, G43
	Programming Concepts	G01, G08, G09, G10, G12, G26, G28, G30, G31, G37, G44, G49, G50, G53, G59, G62, G66, G72, G74, G75, G83, G85, G86, G88
Programming Language (14)	C	G84, G91
	C++	G06
	Java	G17, G71
	JavaScript	G90
	Language semantics	G35
	Python	G16, G18, G19, G33, G40, G48, G82
Programming Logic (4)	Boolean operations	G02
	Logic	G80
	Programming logic	G61
	Truth tables	G21
Programming Paradigm (10)	Object-oriented programming	G05, G14, G27, G54, G56, G58, G65, G70, G73
	Sequential programming	G79
Software Engineering (6)	Code prediction	G63
	Code review	G57
	Code smells	G45, G52, G76
	Refactoring	G24

Table 2. Mapping articles by target audience

Target Audience	#	%	Map
Children / teenagers	44	47%	G04, G06, G08, G09, G11, G15, G16, G18, G19, G26, G27, G29, G30, G32, G36, G37, G38, G39, G40, G41, G42, G43, G44, G48, G49, G50, G54, G58, G61, G63, G65, G66, G70, G72, G74, G75, G80, G83, G84, G85, G87, G88, G92, G93.
Teenagers	18	19%	G02, G03, G07, G12, G13, G20, G21, G22, G23, G25, G31, G45, G46, G68, G69, G71, G76, G77.
Children	13	14%	G17, G33, G35, G47, G52, G56, G59, G67, G73, G78, G79, G86, G89.
Teenagers / Adults	17	18%	G01, G05, G10, G24, G28, G51, G53, G55 G57, G60, G62, G64, G81, G82 G90, G91, G94.
Adults	2	2%	G14, G34.

We separated categories with more than two studies among the target audiences and found that the most used game categories for children were puzzle games (4) and simulation games (6). For children/adolescents, simulation games (14 PS) and puzzle games (13 PS) also stood out, followed by adventure games (4 games), board games, and role-playing games (3 games each). Among the games aimed at teenagers, simulation games stood out with 5 games, followed by card games with 4 games, and the action, adventure, and puzzle categories were tied with 2 games each. Among the Adolescents/Adults group, simulation games for teaching programming stood out.

Table 3. Most used games by target audience

Children	Puzzle	4
	Simulation	6
Teenagers / Adults	Action	2
	Puzzle	2
	Simulation	5
Children / Teenagers	Adventure	4
	Tabletop	3
	Puzzle	13
	RPG	3
	Simulation	14
Teenagers	Action	2
	Adventure	2
	Cards	4
	Puzzle	2
	Simulation	5

About unplugged games, we mapped G18, G22, G24, and G30, which are unplugged games, while G19 and G83 are games that use both digital and unplugged approaches. Among the 94 games selected, 6 games use unplugged resources, which shows the great difference in proportion and lack of availability of this type of resource.

7. Discussion

7.1. Use of games to teach programming

The use of games to teach programming is not new. Game-based teaching is an approach that diversifies teaching and learning. This approach does not replace other traditional teaching methodologies and approaches; on the contrary, it complements them, especially when it comes to achieving specific objectives. When knowledge is transmitted in an engaging, fun, and harmonious way, it increases the student's intellectual development and creates an affinity with what is being learned, making the student part of the process [Barros et al. 2019].

The article [HOSSEINI et al. 2009] presents a Game-Based Learning (GBL) approach. The study conducted by the authors explores the effectiveness and existing potential of games to promote the learning of programming concepts and the development of computational skills. In the article, the authors discuss the importance of GBL for teaching computing and highlight the importance and ability of games to motivate and engage students to learn. Many games presented in this mapping use the GBL methodology. The authors argue that games can help students develop computational skills such as logical reasoning and problem-solving, as well as promote creativity.

7.2. Trends and shortcomings of studies

It is common for researchers to look for solutions to real problems and thus develop games aimed at the areas where learners have the most difficulty, focusing on the most common problems and difficulties for the majority of students, but this approach leaves out some knowledge that can also be assisted by this approach to games that enriches teaching. During our studies, it was possible to see that most of the games for teaching programming are in the Basic programming concepts category with 28 studies (30%), as can be seen in Table 1, followed by Basic programming and Programming language with 14 studies (15% of the total) each. There is a big difference between the categories, where the first has twice as many publications as the second and third, and three times as many studies as the fourth and fifth, Programming Paradigm and Computational Thinking, which each have an average of 10% of the studies in this mapping, and 700% more than the Advanced Programming and Programming Logic areas. It is noticeable that that the categories Data Structures, Software Engineering, Advanced Programming, and Programming Logic need more games to teach them, because although the categories Data Structures and Programming Logic are considered to be basic categories for teaching other categories, they have respectively 6% and 4% of the studies, while the categories Software Engineering and Advanced Programming have respectively the same percentages and are not basic categories for teaching other categories, but because they are advanced categories they may have as many difficulties as learners who are at initial levels.

Another factor that was observed is the absence of unplugged games. Of the 94 primary studies, only 6 [G18, G19, G22, G24, G30, G83] are unplugged games or both (digital and unplugged). Bearing in mind that in 2022 it became compulsory to teach computer science in Brazil, it is necessary to adapt curricula and professionals to teach this knowledge [BRASIL 2022]. In Brazil, like many countries in the southern hemisphere, there is a notable inequality in access to information and communication technologies, as well as a lack of access to the internet and the absence of computers in the homes of a large part of the population. Many schools still don't have computer labs or internet access to carry out basic educational activities [PERINE and ROWSELL 2019, Stevanim 2020]. Considering [Mazuim and Gomes 2019] the advantages of using approaches such as active methodologies and game-based learning, which according to [Tobias et al. 2014] uses game principles and applies them in other contexts to improve engagement in a fun and dynamic way. In this way, limiting digital games to Brazilian education excludes and limits a large part of the population that doesn't have access to electronic devices and the internet in schools. This mapping highlights the need for games, techniques, and unplugged tools for more inclusive and comprehensive teaching.

This SLM analyzed games for teaching programming, using primary studies in the English language, and possibly a study in national databases that brings games developed for teaching programming, which can add options and knowledge to this SLM. There is also a need to explore games for other areas of computer education such as computer networks, databases, and other computer knowledge.

8. Threats to validity

The two main threats to the validity of this mapping are biases both in the selection of articles and in the extraction of information from them, for example in the categorization of games, inference of target audience, and inference of type of game when they did not contain such information or it was not explicitly informed. There is also the risk of possible inaccuracies in the extraction of data from the articles. To mitigate these risks, this study conducted the SLM systematically and strategically, following a protocol to ensure that these errors do not occur.

9. Final considerations

This paper presents the results of an SLM that looked at articles published internationally from 2016 to March 2023 on games used as a tool for teaching and learning programming. After analyzing the data extracted from the mapping, it was possible to observe that these games prove to be a viable methodology to complement other teaching strategies, mainly because they arouse student interest and engagement, and also approach the proposed topic in different ways. The research questions aimed to find out which studies use games to teach programming, their learning objectives, and the types of games that are used.

The RQ1 maps all the studies in this SLM. Thought the RQ2 our results show that the most used language for teaching programming is Python. The prominence of Python for teaching programming demonstrates how this language has been more widely used for teaching programming, probably due to its simplicity, abstraction of concepts, and objectivity. Among the learning objectives, it was noted that the three main learning objectives most sought after were Programming Concepts with 24 studies, followed by Object-Oriented Programming with 9 studies, and Python (Programming Language) with 7 studies. The RQ3 shows that the type of game with the most studies were puzzle and simulation, for all target audiences.

Our research shows that 4 games out of 94 were unplugged, which limits and restricts the learning of many worldwide students who do not have access to digital equipment. This demonstrates the urgency and the real need to develop more unplugged games to teach programming.

Overall, through the results presented, this study contributes by presenting an overview of the international panorama on games for teaching programming, serving as a starting point for new research, as well as facilitating the understanding of this topic by teachers interested in using them in their classes.

Acknowledgements

The authors would like to thank CAPES (grant 88887.808334/2023-00) for their financial support.

References

[Arimoto and Oliveira 2019] Arimoto, M. and Oliveira, W. (2019). Dificuldades no processo de aprendizagem de programaçao de computadores: um survey com estudantes de cursos da área de computaçao. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 244–254. SBC.

[Barros et al. 2019] Barros, M. G. F. B., Miranda, J. C., Costa, R. C., et al. (2019). Uso de jogos didáticos no processo ensino-aprendizagem.

[Battistella and Wangenhein 2016] Battistella, P. and Wangenhein, C. G. V. (2016). Games for teaching computing in higher education–a systematic review. *IEEE Technology and Engineering Education*.

[Bloom 1968] Bloom, B. S. (1968). *Taxonomy of Educational Objectives: The Classification of Educational Goals; Handbook. Cognitive Domain*. McKay.

[BRASIL 2022] BRASIL (2022). Computação: complemento à bncc.

[Garner et al. 2016] Garner, P., Hopewell, S., Chandler, J., MacLehose, H., Akl, E. A., Beyene, J., Chang, S., Churchill, R., Dearness, K., Guyatt, G., et al. (2016). When and how to update systematic reviews: consensus and checklist. *bmj*, 354.

[HOSSEINI et al. 2009] HOSSEINI, H., HARTT, M., and MOSTAFAPOUR, M. (2009). Learning is child's play: Game-based learning in computer science education.

[Krathwohl 1973] Krathwohl, D. R. (1973). Taxonomy of educational objectives. *Affective domain*.

[Mazuim and Gomes 2019] Mazuim, C. H. R. and Gomes, M. S. S. (2019). *Teorias e vivências de metodologias ativas*. Perse, 1ª edition.

[Mendes et al. 2020] Mendes, E., Wohlin, C., Felizardo, K., and Kalinowski, M. (2020). When to update systematic literature reviews in software engineering. *Journal of Systems and Software*, 167:110607.

[Ouzzani et al. 2016] Ouzzani, M., Hammady, H., Fedorowicz, Z., and Elmagarmid, A. (2016). Rayyan—a web and mobile app for systematic reviews. *Systematic Reviews*, 5(1):210.

[PERINE and ROWSELL 2019] PERINE, C. M. and ROWSELL, J. (2019). Making it work in the global south: Stories of digital divides in a brazilian context. *Stories from inequity to justice in literacy education*, pages 185–202.

[Petersen et al. 2008] Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, page 68–77, Swindon, GBR. BCS Learning & Development Ltd.

[Rapkiewicz et al. 2007] Rapkiewicz, C. E., Falkembach, G. A. M., Seixas, L. M. J. d., Santos, N. d. S. R. S. d., Cunha, V. V. d., and Klemann, M. (2007). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *RE-NOTE: revista novas tecnologias na educacao [recurso eletroˆnico]. Porto Alegre, RS*.

[Stevanim 2020] Stevanim, L. F. (2020). Exclusão nada remota: desigualdades sociais e digitais dificultam a garantia do direito à educação na pandemia. *ENSP - Artigos de Periódicos*, 2210.

[Tobias et al. 2014] Tobias, S., Fletcher, J. D., and Wind, A. P. (2014). *Game-Based Learning*, pages 485–503. Springer New York.

[Werbach et al. 2012] Werbach, K., Hunter, D., and Dixon, W. (2012). *For the win: How game thinking can revolutionize your business*, volume 1. Wharton digital press Philadelphia.