A Systematic Mapping of the Use of Games for Teaching Programming

Maurício Lima
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia, Brasil
k20x@outlook.com

Deller Ferreira
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia, Brasil
deller@ufg.br

Elisângela Silva Dias
Instituto de Informática (INF)
Universidade Federal de Goiás (UFG)
Goiânia, Brasil
elisangelasd@ufg.br

***Abstract*—The integration of digital games in programming education is an approach aimed at overcoming challenges such as high dropout rates and learning difficulties in this subject. This systematic mapping examines digital games applied to programming learning through the analysis of 19 selected studies from a total of 1143. The results highlight that digital games not only significantly increase student engagement, but also improve their understanding of fundamental programming concepts. Games offer a playful and interactive form of learning, enabling a more engaging and effective educational experience. This study contributes to the field of technology education by demonstrating how digital games can be effectively used to teach programming, suggesting a potential reduction in dropout rates and an improvement in the quality of learning in technical and engineering courses. The review also identifies a significant gap in the literature on the potential disadvantages of using games in educational environments, highlighting the need for future research in this area.**

***Keywords*—Games; Teaching; Programming.**

I. Introduction

The integration of digital games into programming education emerges as a response to the challenges faced, often marked by high dropout rates and learning difficulties. This raises questions about whether this integration offers a new perspective for teaching programming concepts, enhancing student engagement and understanding. Fundamental programming concepts, such as iteration and specific language constructs, can be difficult to grasp, affecting problem-solving in programming tasks [1]. Beginners often struggle to decompose tasks into manageable parts and map them to programming constructs. This difficulty arises from an uncertain understanding of both the problem and the language constructs [2]. Additionally, they may find the syntax and semantics of the programming language challenging. This is exacerbated by the variety of programming contexts, which require different skills and competencies [3]. Debugging requires the coordination of multiple information sources and is a complex skill that students often find challenging [2].

II. Theoretical Background

Digital games combine rules, challenges, and rewards to engage participants in cycles of trial and error that involve decision-making and scoring systems [4]. When designed for instructional purposes, they are referred to as educational games or serious games, understood as interactive tools that stimulate curiosity, fantasy, and challenge—classic elements with educational value [4]. The study by [5] emphasizes that such games reinforce intrinsic motivation through scenarios that evoke a sense of control, imagination, and achievement.

The study by [6] distinguishes three approaches: playing for entertainment, educational games (with explicit curricular objectives), and gamification (inserting game elements without the need to win or lose). In the scope of this research, we focus primarily on educational games and serious games, while recognizing gamification as a complementary strategy.

Furthermore, [4] points out that immediate visual feedback, gradual challenges, and fantasy elements help maintain the student in a “flow” state, promoting deep learning. Fogg’s Behavior Model (Motivation, Ability, and Triggers) supports the idea that triggers embedded in game systems can drive desirable behavioral changes in educational contexts. These perspectives align with constructivist learning, as students build meaning through active interaction with the playful environment.

In introductory teaching, the approach proposed by [7] suggests starting with 2D game creation to capture attention before addressing syntactic or paradigm-related issues. For specific concepts, serious games have proven effective. One example is a game focused on the concept of variables, which employs semantic visualization and interaction to foster understanding of assignment and data types [4].

Commercial platforms such as the one discussed in [6] apply gamification to more advanced topics (object-oriented programming, recursion), increasing motivation and engagement among high school students. Comparative analyses also indicate two categories of programming games: titles focused on specific units (e.g., variables, stacks, sorting algorithms) and games that cover multiple curricular topics. Most published studies focus on the first category and lack extensive empirical validation.

III. Related Work

Several systematic reviews and meta-analyses investigate the use of digital games in computing education, but their scopes differ substantially from the present mapping, which specifically focuses on games for teaching programming between 2018 and 2024.

Computational Thinking in Primary Education. Giannakoulas and Xinogalos [8] analyzed 61 studies (2010 to 2022) on educational games aimed at cultivating computational thinking in primary school students. Although some of the studies involve programming, the focus is on dimensions of computational thinking, such as abstraction and decomposition, rather than the systematic learning of programming languages or concepts. This study, in contrast, covers all age groups and considers outcomes directly related to the acquisition of programming skills.

Gamification in Programming Courses. The study by [9] conducted a meta-analysis of 21 experiments on gamification in programming education (2010 to 2020). The authors conclude that gamification primarily increases motivation and academic performance. Unlike the present work, their scope is limited to gamification elements (points, badges, competitions), whereas this study investigates complete games (serious games, game-based learning) regardless of whether they include such elements.

Game-Based Learning in Artificial Intelligence Education. The systematic review presented in [10] mapped 125 studies that use games to teach AI concepts (computer vision, machine learning, robotics, etc.). The focus is AI education, which only tangentially addresses basic programming. In contrast, the current mapping centers on games whose primary goal is to develop programming skills.

Games and the Development of Computational Thinking. Furthermore, the study by [11] synthesized 22 studies (2010 to 2020) showing an overall positive effect of games on the development of computational thinking. However, the analysis mixes unplugged activities, board games, and block-based environments, without distinguishing formal programming content. This contrasts with the approach taken in the present mapping.

In summary, previous works address computational thinking in contexts not necessarily focused on formal programming, isolated gamification strategies in programming courses, AI education, or specific aspects and adaptations of games. This mapping broadens the scope by systematically organizing recent literature that uses games (digital or hybrid) explicitly to teach programming, comparing methodologies, target audiences, evaluation metrics, and research gaps.

IV. Methodology

This work presents a Systematic Mapping, characterized by a methodical and replicable methodology, involving a comprehensive search to locate all published and unpublished works on a given topic; a systematic integration of the results of this search; and a critical analysis of the scope, nature, and quality of the evidence concerning a specific research question. They synthesize studies to draw broad theoretical conclusions about the significance of a literature, connecting theory and evidence, as well as evidence and theory [12].

The PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology in its 2020 version was adopted, which consists of a set of guidelines aimed at enhancing the transparency and quality of systematic review reports, including new guidance reflecting advances in methods for identifying, selecting, evaluating, and synthesizing studies [13]. The *Parsifal* tool was used for planning, execution, conduction, and documentation of the research. This tool allows researchers to define goals and objectives, import articles using BibTeX files, eliminate duplicates, establish selection criteria, and generate reports [14]. Specific inclusion and exclusion criteria were defined to select studies published since 2018 that discussed the use of digital games in programming education. The search was conducted across multiple databases, resulting in 19 studies included in the final analysis.

A. Research Question

This systematic mapping aims to answer a single research question:

[RQ1] What are the advantages and outcomes of applying digital games in programming education?

B. Inclusion and Exclusion Criteria

The following inclusion criteria were used: [IC1] The study involves the use of digital games for programming education; [IC2] The study's abstract or title relates to the research objective; [IC3] The study was conducted in 2018 or later. The applied exclusion criteria were: [EC1] The study does not mention digital games in programming education in its title or abstract; [EC2:] The study is a previous version of a previously selected study; [EC3:] The study is a duplicate; [EC4:] The study is prior to 2018; [EC5:] The study is not a primary research; [EC6:] The study is not accessible.

C. Search String

The search *string* used was:

(“*video games*” OR “digital games” OR “*computer games*” OR “educational *video games*”) AND (“*programming education*” OR “*computer science education*” OR “*coding education*” OR “*software development education*”)

D. Databases

Table I presents the databases used and their respective URLs.

Table I
DATABASES USED AND CORRESPONDING URLS.

Database	URL
ACM Digital Library	http://portal.acm.org
EI Compendex	http://www.engineeringvillage.com
IEEE Digital Library	http://ieeexplore.ieee.org
ISI Web of Science	http://www.isiknowledge.com
Science Direct	http://www.sciencedirect.com/
Scopus	http://www.scopus.com

E. Execution

During the identification phase, records were found from databases (n=6) and other sources (n=1,143), totaling 1,149 entries. Of these, 60 duplicate records, 640 ineligible records, and 18 secondary studies were removed before screening, resulting in 425 records for the next stage.

In the screening phase, the 425 records were analyzed, of which 347 were excluded for not meeting the established inclusion criteria. Thus, 78 records proceeded to the eligibility assessment. There was no need to retrieve additional reports, as all records were accessible.

In the eligibility phase, the 78 reports were evaluated based on their alignment with the review objectives. In this stage, 59 reports were excluded for not addressing games aimed at teaching programming. Finally, in the inclusion phase, 19 studies met all criteria and were incorporated into the systematic review. The process was developed according to the PRISMA flowchart in Figure 1.

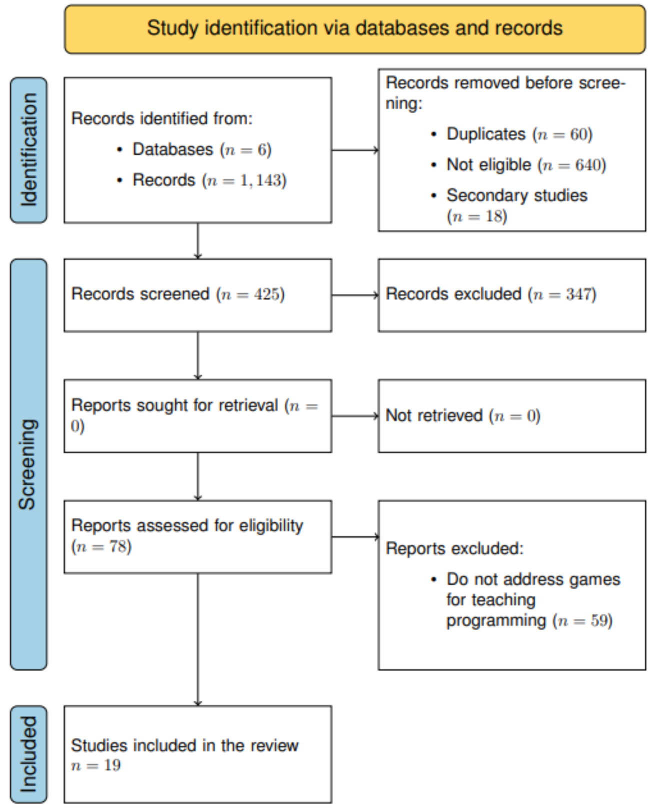

Figure 1. Study selection process

V. RESULTS

Through the search string, a total of 1,143 results were obtained, distributed as shown in Table II.

Table II
NUMBER OF RESULTS RETURNED BY DATABASE.

Database	Number of Results
ACM Digital Library	18
EI Compendex	278
IEEE Digital Library	147
ISI Web of Science	91
Science Direct	169
Scopus	440

Among the main objectives of the analyzed studies, the introduction of fundamental programming concepts for beginners stood out, as in [15], which evaluated the effectiveness of creative activities, such as character design, in learning basic concepts. Similarly, [16] explored the impact of digital games on students’ motivation and performance, obtaining positive results.

Other studies focused on specific strategies to teach complex concepts, such as recursion and functions, through games designed for this purpose, as described in [17]. Additionally, [18] showed that integrating digital games with traditional teaching significantly improved learning C programming.

Gamification was creatively used in [19], which developed the game "CodeToProtect" to teach C++ engagingly, and in [20], which introduced the concept of "cobotic games" to align programming and computational thinking with the game cycle. Furthermore, [21] developed an inclusive game aimed at attracting new audiences to programming, emphasizing diversity and creativity.

Approaches focused on specific contexts were explored in [22], which adapted commercial game rules for teaching programming, and [23], which used the Nintendo Switch development environment to teach virtual reality and game design.

Finally, studies like [24] and [25] highlighted project-based methodologies, such as using game development and parallel programming, to increase student engagement. Overall, the included studies suggest that game-based learning can enhance motivation, improve understanding, and make programming education more accessible and efficient for diverse audiences.

The Figure 2 shows that 42% of the 19 selected articles were indexed exclusively in Scopus, while EI Compendex accounts for 26%. The ISI Web of Science and IEEE Digital Library each contribute 16%. This imbalance was expected, as Scopus covers a broader range of conferences and journals in Computer Education. However, it also indicates a slight coverage bias that may exclude works not indexed in commercial databases or open repositories, such as the ACM Digital Library. For future expansions of the mapping, it is recommended to include additional databases in order to reduce the risk of selection bias.

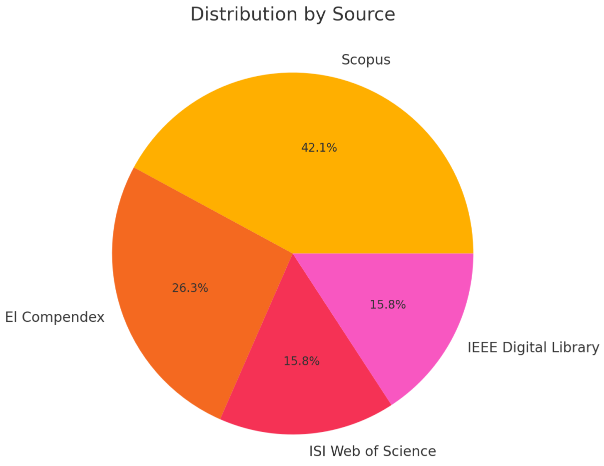

Figure 2. Selected studies by source

The temporal evolution of the studies, shown in Figure 3, reveals an initial peak in 2018 (5 articles), followed by a sharp drop in 2020 (only 1 article). This decline coincides with the first year of the pandemic, a period during which many research initiatives faced logistical delays. Starting in 2021, there is a recovery and steady growth, culminating in 2023 with 4 publications. This suggests that interest in games for programming education has regained momentum.

Three patterns can be observed: relatively stable output of approximately 3 ± 0.5 articles per year after the pandemic, no prolonged gaps that would suggest a lack of interest in the field, and a recent increase that may indicate the consolidation of the topic as a research trend. This historical overview supports the relevance and timeliness of the present mapping.

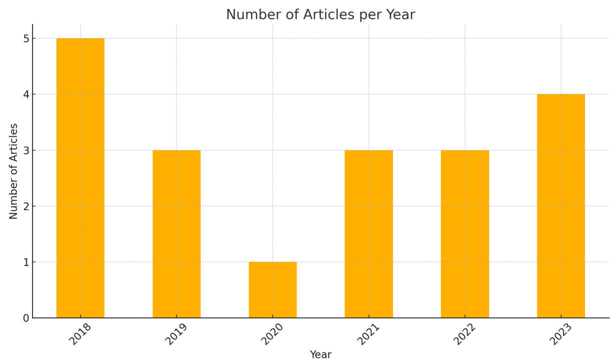

Figure 3. Selected studies by year

VI. Responding to RQ1

According to [26], digital games are a technology with which today's students are already familiar and comfortable. This familiarity is reflected in the thematic focus of this study, and Figure 4 shows that the category *Engagement and Motivation* dominates the analyzed publications, accounting for more than half of the citations.

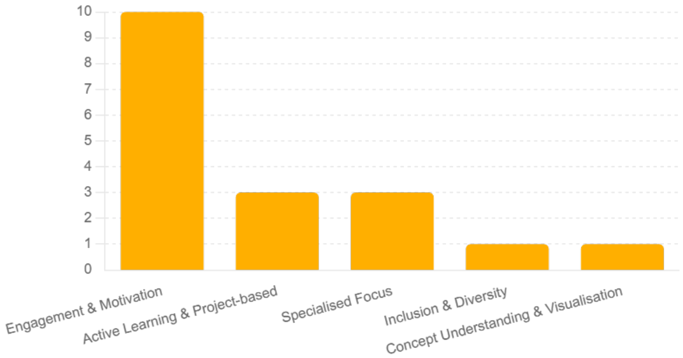

Figure 4. Distribution of Themes Covered in the Publications

[25] reinforces this idea by showing that games focused on parallel programming increase student engagement by offering a practical and enjoyable application. A similar result was observed in introductory C programming courses, where the integration of games into the traditional learning environment led to improved performance [18].

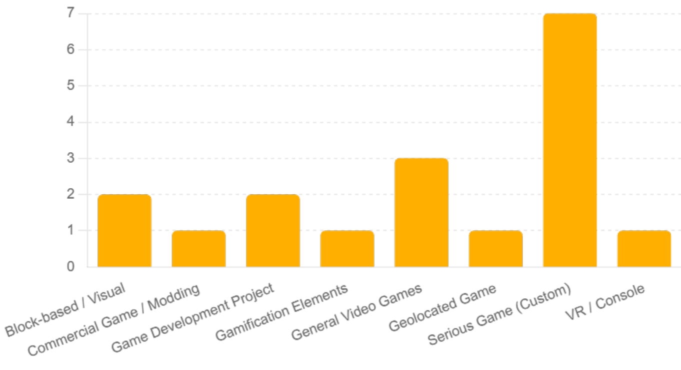

Figure 5. Most Commonly Used Game Platforms in the Studies

Figure 5 shows that most of the studies (n = 7) adopt custom-developed *serious games*, while game development projects and block-based solutions appear less frequently. This diversity of platforms illustrates the range of environments in which the playful approach can be applied—from targeted gamification efforts [27] to full-fledged game engines used in VR environments [23].

[20] and [19] observe that the playful context helps maintain students' focus and encourages experimentation with code. [24], in turn, shows that developing a game project stimulates the learning of object-oriented programming concepts. These findings align with the topic distribution shown in Figure 6, where studies focused on general programming introductions predominate, but specific themes such as parallel programming and VR are also present.

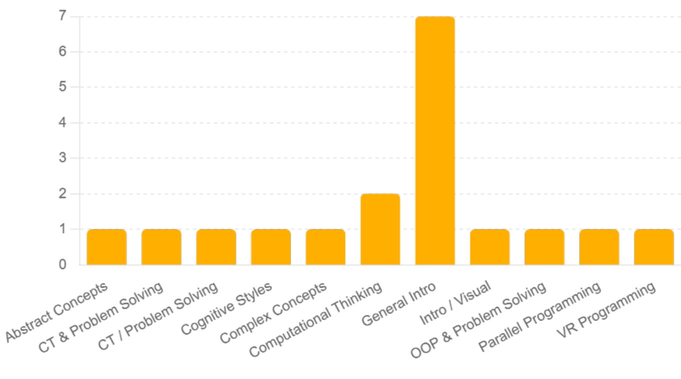

Figure 6. Conceptual Focus of the Analyzed Studies

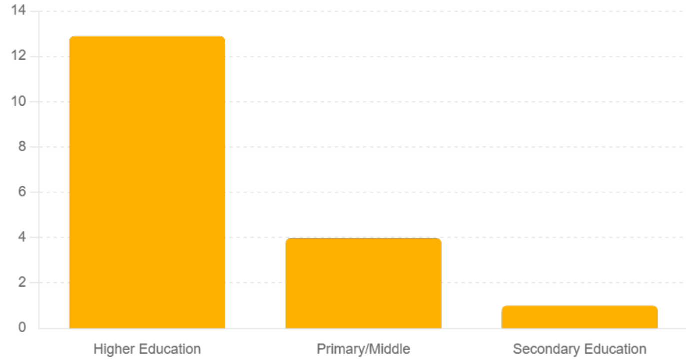

Figure 7. Educational Level of the Target Audiences

Although the literature demonstrates motivational gains across different age groups, Figure 7 indicates a strong concentration on higher education (13 publications). Even so, studies such as [21] highlight the inclusive potential of games to attract girls in elementary and secondary education, reinforcing the importance of expanding research to younger and more diverse audiences.

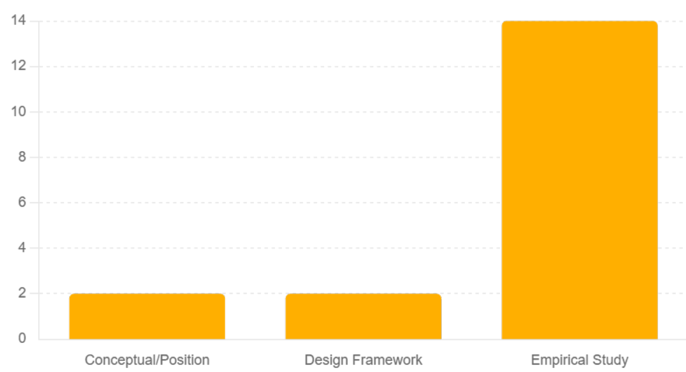

Figure 8. Types of Study (Empirical, Conceptual, Frameworks)

Figure 8 clearly shows that the majority of publications present empirical evidence (14 out of 18), while design frameworks or conceptual discussions remain a minority. Studies such as [28] and [29] suggest that hands-on game development projects can foster active learning of complex programming concepts, reinforcing the need for more long-term experimental research.

VII. Conclusion

Digital games emerge as an intriguing approach in programming education, demonstrating a substantially positive impact on student engagement and learning, as evidenced by this systematic literature review. This method captures students' attention, facilitating intuitive understanding and practical application of programming concepts, suggesting an effective path to reduce dropout rates and improve the quality of programming learning.

The results suggest that digital games have a positive impact on students' engagement and understanding of programming concepts. This systematic review highlights improvements in motivation, concentration, and comprehension of programming concepts. However, as no research was found addressing the disadvantages of using games for teaching programming, it suggests that the drawbacks of using games in programming education are an underexplored aspect. It is inferred, therefore, that the emergence of research aimed at identifying, quantifying, and understanding the potential disadvantages of using games in programming education—both qualitative studies exploring perceptions of students and educators, as well as quantitative analyses investigating the impact of these teaching methods on specific learning variables—would be a positive gain for the field of study, given its scarcity.

Despite these positive points, current research is still insufficient for a comprehensive understanding of the limitations and potential disadvantages of this methodology. It is important for future studies to investigate not only the contexts and conditions under which digital games maximize learning but also the circumstances in which they may not be effective or even counterproductive. Future research may include in-depth qualitative analyses of the experiences of students and educators, as well as longitudinal studies that monitor the long-term impact of this educational approach.

Of the articles used in the systematic literature review, none addressed the disadvantages of using games for teaching programming. This identifies a significant gap in the current literature: the absence of discussion about the disadvantages or limitations of using digital games in programming education. This lack of investigation highlights a possible area for future research, as a complete understanding of any educational methodology should consider both its benefits and its potential flaws or challenges.

Additionally, it would be of considerable interest to develop pedagogical guidelines based on solid evidence for the effective implementation of games in programming education. These guidelines would help educators integrate digital games more effectively, ensuring they complement and enrich traditional teaching methods.

Therefore, digital games represent a promising field for pedagogical innovation in computer science. However, it is essential for the academic community to continue critically exploring both the potentials and the limits of this approach, aiming for more inclusive, engaging, and effective programming education.

References

[1] M. Rouhani, M. Lillebo, V. Farshchian, and M. Divitini, "Learning to program: an in-service teachers' perspective," *2022 IEEE Global Engineering Education Conference (EDUCON)*, pp. 123–132, 2022.

[2] D. Sleeman, "The challenges of teaching computer programming," *Commun. ACM*, vol. 29, pp. 840–841, 1986.

[3] M. Fojcik, M. Fojcik, S.-O. Høyland, and J. Hoem, "Challenges in teaching programming," *Education and New Developments 2022 – Volume I*, 2022.

[4] M. Zapušek and J. Rugelj, "Learning programming with serious games," *EAI Endorsed Transactions on Game Based Learning*, vol. 13, no. 1, pp. 1–8, 2013.

[5] C. Malliarakis, S. Xinogalos, and M. Satratzemi, "Educational games for teaching computer programming," in *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives*. Springer, 2012.

[6] O. Karram, "The role of computer games in teaching object-oriented programming in high schools – codecombat as a game approach," *WSEAS Transactions on Advances in Engineering Education*, vol. 18, pp. 37–45, 2021.

[7] S. Leutenegger and J. Edgington, "A games first approach to teaching introductory programming," in *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, 2004.

[8] A. Giannakoulas and S. Xinogalos, "Studying the effects of educational games on cultivating computational thinking skills to primary school students: a systematic literature review," *Journal of Computers in Education*, vol. 11, no. 4, pp. 1283–1325, 2024.

[9] Z. Zhan, L. He, Y. Tong, X. Liang, S. Guo, and X. Lan, "The effectiveness of gamification in programming education: Evidence from a meta-analysis," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100096, 2022.

[10] Z. Zhan, Y. Tong, X. Lan, and B. Zhong, "A systematic literature review of game-based learning in Artificial Intelligence education," *Interactive Learning Environments*, vol. 32, no. 3, pp. 1137–1158, 2024.

[11] L. Sun, Z. Guo, and L. Hu, "Educational games promote the development of students' computational thinking: a meta-analytic review," *Interactive Learning Environments*, vol. 31, no. 6, pp. 3476–3490, 2021.

[12] A. P. Siddaway, A. Wood, and L. Hedges, "How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses." *Annual review of psychology*, vol. 70, pp. 747–770, 2019.

[13] M. Page, J. McKenzie, P. Bossuyt, I. Boutron, T. Hoffmann, C. Mulrow, L. Shamseer, J. Tetzlaff, E. Akl, S. Brennan, R. Chou, J. M. Glanville, J. Grimshaw, A. Hrõbjartsson, M. Lalu, T. Li, E. Loder, E. Mayo-Wilson, S. McDonald, L. McGuinness, L. Stewart, J. Thomas, A. Tricco, V. Welch, P. Whiting, and D. Moher, "The prisma 2020 statement: An updated guideline for reporting systematic reviews." *Journal of clinical epidemiology*, 2021.

[14] A. Carrera-Rivera, W. Ochoa, F. Larrinaga, and G. Lasa, "How-to conduct a systematic literature review: A quick guide for computer science research," *MethodsX*, vol. 9, p. 101895, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215016122002746

[15] A. Basawapatna, A. Repenning, M. Savignano, J. Manera, N. Escherle, and L. Repenning, "Is drawing video game characters in an hour of code activity a waste of time?" in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. Larnaca Cyprus: ACM, Jul. 2018, pp. 93–98. [Online]. Available: https://dl.acm.org/doi/10.1145/3197091.3197136

[16] A. Alam, "A Digital Game based Learning Approach for Effective Curriculum Transaction for Teaching-Learning of Artificial Intelligence and Machine Learning," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. Erode, India: IEEE, Apr. 2022, pp. 69–74. [Online]. Available: https://ieeexplore.ieee.org/document/9760932/

[17] R. Law, "Introducing novice programmers to functions and recursion using computer games," in *Proceedings of the 12th European Conference on Game Based Learning*. Sophia Antipolis: Academic Conferences and Publishing International Limited, 2018, pp. 325–334.

[18] K. Daungcharone, "Effects of blending digital games into traditional lecture-based learning on university students' programming learning achievement," in *Effects of blending digital games into traditional lecture-based learning on university students' programming learning achievement*, O. R., B. M., and M. B., Eds., vol. 2019-November. Academic Conferences Limited, 2019, Conference paper, p. 151 – 159, cited by: 1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077504305&doi=10.34190%2fEEL.19.070&partnerID=40&md5=be9db398cbf36a2f0dd36867a62dba8e

[19] M. M. Ariffin, N. M. Aszemi, and M. S. Mazlan, "CodeToProtect©: C++ Programming Language Video Game for Teaching Higher Education Learners," *Journal of Physics: Conference Series*, vol. 1874, no. 1, p. 012064, May 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1874/1/012064

[20] R. Higashi, E. Harpstead, J. Solyst, J. Kemper, J. Odili Uchidiuno, and J. Hammer, "The Design of Co-Robotic Games for Computer Science Education," in *Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play*. Virtual Event Austria: ACM, Oct. 2021, pp. 111–116. [Online]. Available: https://dl.acm.org/doi/10.1145/3450337.3483472

[21] P. Mozelius, R. Pechuel, B. Fernández Manjón, T. Kreuzberg, N. Humble, and L. Sällvin, "Development of an Enjoyable Educational Game on Fundamental Programming: Designing for Inclusion and Learning Analytics," *European Conference on Games Based Learning*, vol. 17, no. 1, pp. 434–442, Sep. 2023. [Online]. Available: https://papers.academic-conferences.org/index.php/ecgbl/article/view/1909

[22] D. Saito, R. Tian, H. Washizaki, and Y. Fukazawa, "Programming Education for Young People using the Falling-Puzzle Game, "Puyo Puyo"," in *2023 IEEE Global Engineering Education Conference (EDUCON)*. Kuwait, Kuwait: IEEE, May 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10125210/

[23] A. Sirinterlikci, J. Mativo, and J. Pham, "Using Nintendo Switch Development Environment to Teach Computer Game Programming and Virtual Reality," in *2020 ASEE Virtual Annual Conference Content Access Proceedings*. Virtual On line: ASEE Conferences, Jun. 2020, p. 35467. [Online]. Available: http://peer.asee.org/35467

[24] M. Tomaiuolo, G. Angiani, A. Ferrari, M. Mordonini, and A. Poggi, "A Week of Playing with Code, the Object-Oriented Way," in *Methodologies and Intelligent Systems for Technology Enhanced Learning, 8th International Conference*, T. Di Mascio, P. Vittorini, R. Gennari, F. De La Prieta, S. Rodríguez, M. Temperini, R. Azambuja Silveira, E. Popescu, and L. Lancia, Eds. Cham: Springer International Publishing, 2019, vol. 804, pp. 62–69, series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-319-98872-6_8

[25] W. Feng and L. Davis-Wallace, "Parallel Programming with Pictures: Choosing Your Own Adventure," in *2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. St. Petersburg, FL, USA: IEEE, May 2023, pp. 256–261. [Online]. Available: https://ieeexplore.ieee.org/document/10196652/

[26] M. Wynn and D. Bouchard, "Hello code: An active programming video game," *European Conference on Games Based Learning*, 2018.

[27] I. Perenc, T. Jaworski, and P. Duch, "Teaching programming using dedicated Arduino Educational Board," *Computer Applications in Engineering Education*, vol. 27, no. 4, pp. 943–954, Jul. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/cae.22134

[28] M. Aedo Lopez and E. G. Castro Gutierrez, "Teaching Model of Object Oriented Programming based on Video Game Development and applying Software Project Management and Scrum," in *Proceedings of the 20th LACCEI International Multi-Conference for Engineering, Education and Technology: "Education, Research and Leadership in Post-pandemic Engineering: Resilient, Inclusive and Sustainable Actions"*. Latin American and Caribbean Consortium of Engineering Institutions, 2022. [Online]. Available: https://laccei.org/LACCEI2022-BocaRaton/meta/FP229.html

[29] M. A. López, V. Cornejo-Aparicio, and S. Flores-Silva, "Experience in the application of a geolocated video game to understand the fundamental concepts of the object oriented programming paradigm; [experiencia en la aplicación de un videojuego geolocalizado para comprender los conceptos fundamentales del paradigma de la programación orientada a objetos]," *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, vol. 2021, no. E46, p. 452 – 462, 2021, cited by: 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85148354008&partnerID=40&md5=2a2f6c0dade64346ad4c6e7ca74ad941