EXPLORING GAME BASED LEARNING: A SYSTEMATIC LITERATURE REVIEW (2021-2025)

Siti Sakinah Mohd Yusof, Haslinda Hashim and Erni Marlina Saari

Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris Malaysia

ABSTRACT

This systematic literature review (SLR) investigates how game-based learning (GBL) that involves game design, creation, and play supports programming education and the development of computational thinking (CT) skills. Searches were conducted in Scopus and ERIC in October 2025, applying strict inclusion criteria to focus on fully published journal articles from 2021 to 2025 written in English. The search strategy yielded 285 initial records, with 26 studies meeting the final eligibility requirements. All included studies are situated in the Computer Science and computing education domains and involve students ranging from primary to higher education. The results show that GBL approaches integrating either game play, game design, or a combination of both are consistently associated with improved problem-solving, algorithmic reasoning, motivation, and engagement in programming instruction. Game design and mixed activity models particularly contribute to higher-order CT development by encouraging creativity, abstraction, and meta-cognitive thinking. Immersive technologies, including virtual reality and hybrid interactive platforms, enhance learners' ability to visualize and interact with programming concepts, thereby supporting deeper engagement and understanding in educational contexts. Despite positive outcomes, key challenges persist, including variability in instructional alignment, limited collaboration mechanics, and technological constraints. This review highlights a growing trend toward learning experiences that blend constructionist and playful elements to better support novice programmers. Future research should deepen experimental rigor, explore diverse learner contexts, and establish more unified frameworks for evaluating CT development through game-based programming education.

KEYWORDS

Game-Based Learning, Computational Thinking, Programming Education

1. INTRODUCTION

Programming has become a core literacy in modern education, yet many novice learners struggle with abstract concepts such as algorithmic thinking, debugging, and problem decomposition when taught through traditional approaches [1][2]. Game-based learning (GBL) has proven to be an effective approach for addressing these challenges, as it provides dynamic, interactive environments where students can engage in active exploration, receive instant feedback, and develop problem-solving skills in a playful context [3][4]. By situating learning within meaningful game contexts, students demonstrate improved motivation, creativity, and persistence when tackling programming tasks [5][6].

GBL appears in the literature primarily through two pedagogical modalities: learning through playing games and learning through designing games. Gameplay supports conceptual learning and engagement by allowing students to test ideas, overcome challenges, and observe computational logic in action [7][8]. Game design aligns strongly with constructionist learning, enabling learners to express computational ideas by creating and refining game artifacts

DOI:10.5121/ijma.2025.17501

[9][10][11]. Evidence suggests that designing games can deepen meta-cognitive processes and debugging skills through iterative creation and troubleshooting [12][13][14].

Recently, scholars have begun combining design and play activities to leverage the strengths of both approaches. Learning sequences where students first play, then formulate problems as games, and finally create their own game systems may produce richer computational thinking outcomes [15]. Emerging technologies such as virtual reality and hybrid interactive platforms further expand the possibilities for engagement and learning transfer in programming education [6][16][17].

Despite these positive developments, current research remains fragmented across different educational levels, learning tools, and disciplinary perspectives. Key questions persist regarding which type of game activity yields the most effective CT development for novice programmers and how design-focused, play-focused, or mixed GBL approaches can be strategically integrated into curricula [18][19]. Scholars also emphasize the importance of aligning game mechanics with authentic learner needs and pedagogical goals to reduce dropout rates and ensure sustainable engagement [18][20].

To bridge these gaps, this Systematic Literature Review (SLR) analyzes recent journal publications from 2021 to 2025 that explore programming education through game design, game play, or mixed game design and game play approaches. The review focuses on research outcomes related to computational thinking, programming skills, and learner engagement and highlights trends, limitations, and implications for future instructional design.

Through this synthesis, the review contributes conceptual clarity on how GBL supports student learning in programming and provides practical guidance for leveraging playful and construction-focused experiences in computing education.

Table 1. Documents by type

Document Type	**Number**	**Description**
Journal Articles	**26**	Final included set (13 Scopus + 13 ERIC)

Analysis in Figure 1 shows that all 26 included documents are journal articles. This distribution reflects the eligibility criteria, which restricted the review to fully published peer-reviewed journals while excluding conference papers, reviews, books, and in-press manuscripts

Table 2. Documents by subject area

Subject Area	**Classification**	**Notes**
Computer Science	**100%**	Programming education, CT, algorithmic thinking
Education / Computing Education	Integrated focus	Extracted from ERIC + CS-education keywords

Figure 2 illustrates that 100% of the studies focus on Computer Science and Computing Education topics, emphasizing programming learning, computational thinking, and related problem-solving skills. This distribution directly results from the Computer Science filtering applied in Scopus and the focus on programming in the ERIC search strategy.

2. METHODOLOGY

2.1. Identification

Following PRISMA guidelines, a systematic search was conducted in October 2025 across the Scopus and ERIC databases. The search targeted peer-reviewed journal articles and conference papers published in English between January 2021 and October 2025. Relevant keywords were developed based on thesauri, dictionaries, academic databases, and prior research. The final search strategy focused on terms connected to game design in programming education, including “game creation,” “computational thinking,” and “novice programmers.” Tailored search strings were applied to each database, as presented in Table 3. The initial search retrieved 72 records from Scopus and 213 from ERIC, totaling 285 records for screening

Table 3. The search string

Database	**Search String**
Scopus Search Date: October 2025	TITLE-ABS-KEY (("game design" OR "game making" OR "game creation" OR "learning through game design" OR "constructionist game design") AND ("programming education" OR "computer programming" OR "coding education" OR "computing education" OR "computer science education") AND ("problem solving" OR "computational thinking" OR "algorithmic thinking") AND (student* OR learner* OR "novice programmer*")) AND PUBYEAR > 2020 AND PUBYEAR < 2026 AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (SRCTYPE , "p") OR LIMIT-TO (SRCTYPE , "j"))
ERIC Search Date: October 2025	(("educational games" OR "game based learning" OR "student developed materials" OR "game design" OR "learning through game design") AND ("programming" OR "computer science education" OR "coding" OR "computing education") AND ("problem solving" OR "computational thinking" OR "algorithmic thinking")) AND (student* OR learner*)

2.2. Screening

The screening process involved evaluating titles and abstracts for relevance to programming or computer science education through game-based or game design–oriented learning. In Scopus, the initial 72 records were reduced to 60 through restriction to the Computer Science subject area. Applying the publication year filter (2021–2025) yielded 17 studies, followed by a limitation to journal articles and conference papers resulting in 13 eligible studies. From the ERIC search, filtering for publication years 2021–2025 reduced the initial 213 records to 93 studies, and a further restriction to higher education contexts resulted in 19 studies progressing to eligibility assessment. Following screening procedures, 32 studies were deemed relevant for full-text evaluation.

2.3. Eligibility

The eligibility phase involved full-text screening to ensure a clear focus on programming education using game design–based strategies and related problem-solving constructs. In the ERIC dataset, five studies were excluded due to duplication or disciplinary misalignment, involving fields such as physics, mathematics, gas engineering, and chemistry. All 13 Scopus articles remained eligible. In total, 26 studies met all inclusion criteria and were included in the qualitative synthesis.

Table 4. The selection criterion is searching

Criterion	**Inclusion**	**Exclusion**
Language	English	Non-English
Timeline	2021 – 2025	< 2021
Literature type	Journal Article	Conference, Book, Review
Publication Stage	Final Published Articles	In Press

Table 5. The flow of the systematic searching process (Moher D, Liberati A, Tetzlaff J, 2009)

Identificationn	Record identify through Scopus searching **(n =72)**	Record identify through ERIC searching **(n =213)**
Screening	Records after screened Scopus **(n=13)**, ERIC **(n=19)** **(Total =32)** Duplicate record removed **(n = 5)**	Records excluded Follow the criterion; removed Non-English, < 2021, book chapter,conference proceeding,In Press. **(n = 253)**
Eligibility	Article access for eligibility **(n = 25)**	Full text excluded Due to the out of field **(n= 6)** Title not significantly **(n=227)** Abstract not related on the objective of the study **(n=6)**
Included	Studies included in qualitative analysis **(n =25)**	

A total of 285 records were initially retrieved from database searching, consisting of 72 records from Scopus and 213 records from ERIC. After applying the inclusion criteria, including publication year (2021–2025), English language, final publication status, and relevance to programming and computational thinking within game design contexts, the number of Scopus records was reduced to 13, while ERIC records were filtered down to 19, followed by the removal of five non-computer science and duplicate articles, resulting in 12 eligible ERIC records. The study selection process followed a multi-stage filtering approach based on predefined inclusion and exclusion criteria. Initially, all search results were restricted to the subject areas of Computer Science or Computing Education. This was followed by a time filter, which retained only studies published between January 2021 and October 2025. Next, the publication type was narrowed to include only fully published peer-reviewed journal articles, excluding conference papers, book chapters, reviews, and preprints. Titles and abstracts were then screened for relevance to programming education through game-based or game design strategies involving computational thinking (CT) or problem-solving skill development. Finally, a full-text review was conducted to confirm alignment with the scope of the review, ensuring that selected studies reported interventions or findings specifically related to programming education through GBL design or play.

In the eligibility stage, all 26 remaining articles (Scopus = 13; ERIC = 12) underwent full-text review against the defined scope of computational thinking and programming within student game design. No further exclusions occurred during this stage. Finally, 25 studies were confirmed as meeting all criteria and were included in the qualitative synthesis of this systematic review.The review method adopted in this study offers several distinct advantages. By focusing exclusively on recent (2021–2025) peer-reviewed journal articles, the study ensures that the synthesis reflects current trends in both game-based learning (GBL) and computational thinking (CT) education. It prioritizes studies involving game design and mixed GBL models, thus providing deeper insight into constructionist learning approaches—a gap in previous reviews that predominantly emphasized game play. The use of thematic synthesis allows for the integration of diverse research designs (qualitative, quantitative, and mixed-methods), resulting in richer and more meaningful interpretations of how GBL supports programming education. Furthermore, this review identifies instructional, technological, and pedagogical challenges often overlooked in intervention-specific studies, thereby contributing to more comprehensive instructional design guidance.

2.4. Data Abstraction and Synthesis

Data extraction followed a structured approach, capturing publication year, education context, research design, intervention characteristics, and primary outcomes related to programming skills and computational thinking development. A thematic synthesis approach was employed to integrate findings across qualitative, quantitative, and mixed-method studies. Three(3) researchers independently coded and validated themes to ensure accuracy, consistency, and methodological rigor. Analytical decisions were documented throughout the process to promote transparency and reliability.

A thematic synthesis approach was employed to systematically interpret the findings of the included studies. The process began with open coding, where researchers independently reviewed the extracted data from each study—such as outcomes, sample details, and game-based learning (GBL) activity types—and coded meaningful units of text. These codes were then clustered to develop descriptive themes, including "Game Design,""Game Play," and "Problem-solving via Game Construction". To ensure reliability, the coding framework and thematic mappings were peer-reviewed for consistency and accuracy, with any discrepancies resolved through discussion and cross-examination of sources, thereby enhancing the credibility of the synthesis.

3. Results and Findings

A total of 25 studies were included in the synthesis: 7 Design, 16 Play, and 2 Mixed. These studies collectively demonstrate strong evidence that game-based strategies enhance computational thinking (CT), problem-solving skills, motivation, and attitudes toward programming.

3.1. Game Design Studies (n = 7)

Studies focusing on design activities show that building games supports deeper engagement with programming logic. Young learners meaningfully applied AI and pathfinding concepts while designing Pac-Man-style games [9]. Progressive enhancement tasks in visual programming boosted CT skills and interest in coding [10]. Tools such as Alice enabled novice learners to visualize object-oriented programming and improve algorithmic thinking [11].

Participatory Design approaches strengthened debugging and creative problem-solving [12]. Physical computing projects involving Scratch and Arduino/micro:bit improved preservice teachers' CT knowledge and teaching readiness [13][14]. These results suggest design-driven learning enables students to act as creators, leading to higher cognitive engagement and transferable skills.

3.2. Game Play Studies (n = 16)

Game play studies revealed strong outcomes in motivation and CT development. Learners showed improved procedural thinking, perseverance, and problem-solving when interacting with educational games that integrated puzzles, challenges, and rewards. Gameplay interventions were especially effective in reducing cognitive load and anxiety among novice programmers.

Across these studies, playing games acted as a gateway to programming by supporting curiosity and easing conceptual difficulties before learners transitioned into writing code.

3.3. Mixed Design and Play Studies (n = 2)

Hybrid approaches that alternate between creativity and challenge demonstrated particularly strong learning effects. A tier-based model combining gameplay, problem formulation, and game creation enhanced metacognitive awareness and teamwork [15]. Game-based robotics design in Besiege elevated performance and creativity beyond traditional robotics instruction [19].

These studies emphasize that toggling between experiencing and building game mechanics helps students solidify their CT understanding.

Table 6. Game Design Studies

No.	Author (s) & Year	Purpose / Objective	Type of Game Activity	Programming / CT Focus	Key Findings
1	Repenning (2024)	Enhance young learners' programming by adding AI blocks in RULER.game for pathfinding game creation	Students created Pac-Man–like games with AI features	AI logic, pathfinding, CT	Students built fully functional games with improved creativity and programming complexity
2	Tian et al. (2024)	Evaluate Puyo-Puyo-based design tool for improving programming engagement in elementary learners	Game enhancement workshops	Basic programming, CT, problem solving	Increased engagement, better understanding of logic and CT through progressive game modification
3	Al-Tahat (2021)	Review educational uses of Alice to improve programming and problem-solving	Use of 3D storytelling for novice learners	OOP basics, algorithmic thinking, game design	Alice supports motivation, visualization, and CT growth in introductory programming
4	Humble et al. (2023)	Identify CT game design needs in higher education	Concept development based on teacher feedback	CT components: decomposition, abstraction, algorithmic thinking	Produced design guidelines linking game features to student needs and motivation
5	Theodoropoulos (2022)	Use participatory methods to improve CT through game coding and debugging	6-week workshops developing mobile games	Debugging, problem solving, iterative CT	Active debugging improved CT skills and creativity in game creation
6	Tsai et al. (2022)	Improve preservice teachers' CT and attitudes via Scratch + Arduino game projects	Two-player games with physical controllers	CT, energy knowledge, fundamental programming	Significant CT and knowledge gains; high satisfaction and feasible for teacher training
7	Tsai (2023)	Prepare preservice teachers to teach CT using Scratch + micro:bit games	Motion sensor educational games	CT concepts, visual programming, teaching readiness	Improved CT, energy knowledge, and teaching confidence; positive attitudes

Table 6 synthesizes seven studies centered on constructionist game design approaches for programming and computational thinking development. Across these works, learners engaged in creating or enhancing games using tools such as Scratch with Arduino, micro:bit, and AI-enabled design environments like RULER.game. Students were positioned as creators who actively integrated logic, debugging, and physical computing elements into game mechanics. Findings consistently show improvements in core CT competencies including decomposition, abstraction, and iterative problem-solving, alongside heightened creativity and motivation to learn programming. Teacher-preparation studies particularly emphasized increased instructional readiness and confidence among preservice teachers. Collectively, game design interventions provide rich opportunities for deep learning through experimentation, reflection, and the application of computational principles in personalized contexts.

Table 7. Game Play Studies

No.	Author(s) & Year	Purpose / Objective	Type of Game Activity	Programming / CT Focus	Key Findings
1	Chen, B. (2025)	Improve programming learning using balanced game design principles	Students played WannaBone	Loops, conditionals, basic CT	CT and motivation improved through engaging gameplay
2	Adharsh, C., et al. (2024)	Teach sorting algorithms using a gamified constructivist game	Sorting challenges through game commands	Sorting, data structures, CT	Better engagement and understanding of algorithm concepts
3	Akkaya, A., & Akpinar, Y. (2022)	Evaluate serious game impact on OOP and CT skills	61 non-engineering students played OOP learning game	OOP fundamentals, abstraction, creative problem solving	Significant CT and OOP gains regardless of prior skills
4	Villegas-Ch, W., et al. (2024)	Use VR for algorithm and data structure learning	VR algorithm challenges	Problem-solving and teamwork	Improved accuracy, speed, and collaboration
5	Grizioti, M., & Kynigos, C. (2024)	Link CT and data science via classification gameplay	SorBET gameplay and modding	Data modeling, interpretation, CT	Enhanced CT and deeper critical engagement through modding
6	Adam, K., et al. (2023)	Explore analog GBL for CT development	3D tic-tac-toe then digital programming	Abstraction, decomposition, debugging	Analog + digital tasks effectively supported CT learning
7	Soleymani, A., et al. (2025)	Assess gamified CS learning network effectiveness	Gamified professional learning platform	Engagement, problem solving	Increased applied learning value; collaboration remained low
8	Yesengazyevna, S. A., et al. (2022)	Study perceptions of game-based programming learning	University students used educational games	Motivation, problem solving	High motivation and retention; tech infrastructure issues noted
9	Zhuang, Y., et al. (2024)	Improve CT visibility via hybrid paper-digital games	Board game with digital feedback	Logic, CT visualization	Better CT performance and thinking process clarity
10	Agbo, F.	Support CT with	VR	Decomposition,	CT skills and motivation

No.	Author(s) & Year	Purpose / Objective	Type of Game Activity	Programming / CT Focus	Key Findings
	J., et al. (2023)	immersive VR mini-games	experiment vs. control group	abstraction, algorithmic thinking	increased through VR immersion
11	Gronseth, S., Itani,et al. (2025)	Teach AI/ML through digital escape room	Collaborative escape room puzzles	AI/ML literacy, CT, teamwork	Improved teamwork and engagement with AI/ML concepts
12	Wang, Y.-H. (2021)	Enhance programming via modified WebQuest	Debugging and inquiry-based tasks	Debugging, problem solving	Improved reasoning and debugging vs. traditional teaching
13	De Santo, A., et al. (2022)	Gamify notebooks for CT in non-CS learners	115 students used gamified notebooks	CT skills and active learning	Stronger engagement; active learning predicted CT outcomes
14	Ng, D. T. K., et al. (2023)	Review AI literacy pedagogies	Play elements in reviewed studies	AI literacy, CT, robotics, game-based learning	Shift toward accessible, gamified AI learning for K–12
15	Kleinman, E., et al. (2022)	Make problem-solving behaviors interpretable	Parallel learning game	CT behavior, decision-making	Identified behavior profiles enabling more personalized instruction
16	Borrás-Gené, O., et al. (2024)	Increase programming motivation via escape room	157 preservice teachers	Programming concepts, problem solving	Higher motivation and perceived relevance of programming

Table 7 presents findings from sixteen studies using gameplay as the primary learning activity. These studies showcase a diverse set of platforms, including VR environments, gamified coding tasks, escape rooms, and hybrid physical–digital games. Results demonstrate that gameplay offers a highly motivational pathway for developing foundational coding skills, data structures, artificial intelligence concepts, and CT strategies. Enhanced engagement, improved problem-solving efficiency, and stronger conceptual retention were widely reported, particularly when collaborative elements were present. However, some studies highlighted persistent challenges such as limited teamwork effectiveness or technological barriers. Overall, gameplay-based interventions remain an effective mechanism for lowering cognitive anxiety, increasing participation, and enabling iterative learning through challenge-driven interaction.

Table 8. Game Design and Game Play Studies

No.	Author(s) & Year	Purpose / Objective	Type of Game Activity	Programming / CT Focus	Key Findings
1	Tanimoto, S. L. (2022)	Evaluate a three-tier gamification model integrating game play, game formulation, and game creation	College freshmen played, reframed problems as games, then designed games	Problem solving, CT, meta-cognition, agile development	Model increased motivation, strategy development, teamwork, and creative CT application
2	Kunovic, I., et al. (2024)	Examine physics-based virtual robotics (Besiege) for design and robotics education	Design and play in Besiege vs. physical robots	Robotics design, control logic, CT, problem solving	Higher engagement, creativity, and understanding in game-based group; strong support for design-centered CT learning

Table 8 includes two studies that blended both design and play elements, positioning learners as both players and creators. These hybrid models emphasize active meaning-making through cycles of playing games, reframing problems as game mechanics, and finally producing functional game artifacts. Findings indicate clear benefits to student collaboration, meta-cognitive thinking, and applied problem-solving, particularly in areas like robotics design and agile software development. Both studies noted strong motivational growth and adaptability of CT concepts to real-world technical scenarios. Mixed models appear especially promising for higher education learners who require deeper engagement with programming logic and system thinking, offering a balanced pathway that combines structured learning through play with the open-ended creativity of design.

4. DISCUSSION

Table 8 includes two studies that blended both design and play elements, positioning learners as both players and creators. These hybrid models emphasize active meaning-making through cycles of playing games, reframing problems as game mechanics, and finally producing functional game artifacts. Findings indicate clear benefits to student collaboration, meta-cognitive thinking, and applied problem-solving, particularly in areas like robotics design and agile software development. Both studies noted strong motivational growth and adaptability of CT concepts to real-world technical scenarios. Mixed models appear especially promising for higher education learners who require deeper engagement with programming logic and system thinking, offering a balanced pathway that combines structured learning through play with the open-ended creativity of design\

5. CONCLUSION

This systematic review synthesized 26 studies published between 2021 and 2025 that employed game design, game play, or mixed game-based strategies to enhance programming and computational thinking (CT) education. Across the studies, strong evidence demonstrates that game-based learning environments contribute to significant improvements in CT competencies, programming performance, and problem-solving effectiveness. Learners reported higher motivation, enjoyment, and reduced anxiety toward coding tasks when immersed in playful and interactive learning formats.Studies that positioned learners as creators, rather than solely players, further strengthened creativity, debugging skills, and iterative thinking by requiring students to design or refine games and game interactions. Mixed approaches that combine both design and gameplay fostered deep engagement with real-world problem framing and computational modeling.

Collectively, these findings show that allowing learners to act as game makers broadens their computational empowerment and strengthens transferable digital skills relevant to future academic and career pathways. Even with the strong consensus on benefits, several limitations persist. Many studies involved short-term interventions or small cohorts, limiting scalability and long-term insight into learning retention. Geographic representation remains uneven, with the majority of research stemming from Asia, Europe, and the United States. Future research should prioritize longitudinal tracking, multi-site experimentation, and more diverse learner populations to improve generalizability and address persistent barriers in integrating game-based programming pedagogy within formal curricula.Overall, this review affirms that game-based learning—whether through design, play, or hybrid modalities—presents a powerful approach to developing confident, motivated, and computationally literate learners who can innovate and solve problems creatively in a digitally evolving world.

While several prior systematic reviews have examined game-based learning (GBL) in general educational contexts, few have focused explicitly on game design as a learning modality for programming education. Earlier reviews tended to emphasize game play environments such as Scratch or Code.org as tools for motivating learners, whereas this review extends the inquiry to include game design and mixed game-making models, highlighting the constructionist benefits of having learners actively create games [5], [6]. Unlike earlier reviews that grouped all GBL studies together, this study distinguishes between game play, game design, and hybrid approaches to analyze their unique contributions to higher-order computational thinking (CT) skills such as abstraction and debugging [12], [13]. Additionally, this review incorporates more recent studies that integrate advanced technologies like virtual reality, hybrid interactive platforms, and game-based robotics, offering insights into emerging pedagogies and tools in programming education not typically captured in pre-2021 analyses [16], [17].

ACKNOWLEDGEMENTS

The authors extend their sincere appreciation to Universiti Pendidikan Sultan Idris for the academic guidance, research support, and scholarly environment that made this work possible. Gratitude is also warmly expressed to Kolej Matrikulasi Perak for their continued encouragement and assistance throughout the research process. Their collective contributions have been invaluable in bringing this study to completion.

REFERENCES

[1] Wang, Y.-H. (2021) “Integrating modified WebQuest activities for programming learning”, *Journal of Computer Assisted Learning*, Vol. 37, No. 4, pp. 978-993. https://doi.org/10.1111/jcal.12534

[2] Adam, K., Lightfoot, R., Chowdhury, M., and Phillip Donaldson, J. (2023) “Cultivating computational thinking through game-based learning”, *Proceedings of the Frontiers in Education Conference (FIE 2023)*, IEEE. https://doi.org/10.1109/FIE58773.2023.10342931

[3] Chen, B. (2025) “Programming games for fundamental computational concepts: A technical design framework”, *Proceedings of the 10th International Conference on Communication and Information Processing (ICCIP 2024),* Association for Computing Machinery.https://doi.org/10.1145/3708657.3708753

[4] Adharsh, C. S., Angeline Benitta, D., Sherin Eliyas, Balamurugan, B., et al. (2024) “Gamification: The effective way for better learning”, *IEEE International Conference on Emerging Electronics and Communication Technologies (ICEECT 2024),* pp. 1-5. https://doi.org/10.1109/ICEECT61758.2024.10739152.

[5] Akkaya, A. and Akpinar, Y. (2022) “Experiential serious-game design for development of knowledge of object-oriented programming and computational thinking skills”, *Computer Science Education*, Vol. 32, No. 4, pp. 421-444. https://doi.org/10.1080/08993408.2022.2044673.

[6] Agbo, F. J., Oyelere, S. S., Suhonen, J., and Tukiainen, M. (2023) “Design, development, and evaluation of a virtual reality game-based application to support computational thinking”, *Educational Technology Research and Development,* Vol. 71, No. 2, pp. 505-537. https://doi.org/10.1007/s11423-022-10188-6

[7] Villegas-Ch, W., Govea, J., Naranjo Godoy, L., and Mera-Navarrete, A. (2024) “Virtual reality simulations for skills training: Improving learning through immersive experiences in educational environments”, *IEEE Access*, Vol. 12, pp. 134921-134935. https://doi.org/10.1109/ACCESS.2024.3456628

[8] Borrás-Gené, O., Hijón-Neira, R., Paredes-Barragán, P., and Serrano-Luján, L. (2024) “A hybrid escape room to foster motivation and programming education for pre-service teachers”, *International Journal of Game-Based Learning*, Vol. 14, No. 1, pp. 1-17.

[9] Repenning, A. (2024) “Escaping the Turing tar-pit with AI programming blocks”, *ACM International Conference Proceeding Series (ICCPS 2024)*. https://doi.org/10.1145/3677619.3677646

[10] Tian, R., Saito, D., Washizaki, H., Fukazawa, Y., and Kobayashi, H. (2024) "Enhancing programming education through game-based learning: Design and implementation of Puyo Puyo-inspired teaching tool", *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE 2024)*.https://doi.org/10.1145/3626253.3635477

[11] Al-Tahat, K. (2021) "Alice adventures in computingland: A review", *Proceedings of ACIT 2021*, IEEE. https://doi.org/10.1109/ACIT53391.2021.9677413

[12] Theodoropoulos, A. (2022) "Participatory design and participatory debugging: Listening to students to improve computational thinking by creating games", *International Journal of Child-Computer Interaction*, Vol. 33, 100525. https://doi.org/10.1016/j.ijcci.2022.100525

[13] Tsai, F.-H., Hsiao, H.-S., Yu, K.-C., and Lin, K.-Y. (2022) "Development and effectiveness evaluation of a STEM-based game-design project for preservice primary teacher education", *International Journal of Technology and Design Education*, Vol. 32, No. 5, pp. 2403-2424. https://doi.org/10.1007/s10798-021-09725-4

[14] Tsai, F.-H. (2023) "Using a physical computing project to prepare preservice primary teachers for teaching programming", *SAGE Open*, Vol. 13, No. 4. https://doi.org/10.1177/21582440231201436

[15] Tanimoto, S. L. (2022) "Three tiers of gamification in a college course on problem-solving for global challenges", *Interaction Design and Architecture(s) Journal*, Vol. 53, pp. 101-127.

[16] Zhuang, Y., Lin, Y.-H., Liyanawatta, M., Saputro, A. H., Utami, Y. D., and Wang, J.-H. (2023) "An interactive programming learning environment supporting paper computing and immediate evaluation for making thinking visible and traceable", *Interactive Learning Environments*, Vol. 32, No. 9, pp. 5253-5266. https://doi.org/10.1080/10494820.2023.2212709.

[17] Gronseth, S., Itani, A., Seastrand, K., Beech, B., Bruce, M., Solorio, T., and Kakadiaris, I. (2025) "Exploring a digital educational escape room game to support artificial intelligence literacy and team building in an online environment", *Journal of Interactive Learning Research*, Vol. 36, No. 1, pp. 29-44..

[18] Humble, N., Mozelius, P., and Sällvin, L. (2023) "Need-based game design: Developing a CT game based on educational needs", *European Conference on Games-Based Learning*, 17(1). https://doi.org/10.34190/ecgbl.17.1.1489.

[19] Kunovic, I., Filipovic, I., and Sovic Krzic, A. (2024) "Teaching robotics concepts with Besiege: A game-based approach for developing design and problem-solving skills", *47th ICT and Electronics Convention, MIPRO 2024*, IEEE. https://doi.org/10.1109/MIPRO60963.2024.10569275

[20] Soleymani, A., De Laat, M., and Specht, M. (2025) "Evaluating value creation, motivation, and personal experiences in a game-based professional learning network for computer science education", *Electronic Journal of e-Learning*, Vol. 23, No. 3, pp. 45-63.

[21] Grizioti, M. and Kynigos, C. (2024) "Integrating computational thinking and data science: The case of modding classification games", *Informatics in Education*. https://doi.org/10.15388/infedu.2024.03

[22] Ng, D. T. K., Lee, M., Tan, R. J. Y., Hu, X., Downie, J. S., and Chu, S. K. W. (2023) "A review of AI teaching and learning from 2000 to 2020", *Education and Information Technologies*, Vol. 28, No. 7, pp. 8445-8501. https://doi.org/10.1007/s10639-023-11632-2

[23] Kleinman, E., Shergadwala, M. N., Teng, Z., Villareale, J., Bryant, A., Zhu, J., and Seif El-Nasr, M. (2022) "Analyzing students' problem-solving sequences: A human-in-the-loop approach", *Journal of Learning Analytics*, Vol. 9, No. 2, pp. 138-160. https://doi.org/10.18608/jla.2022.7571

[24] Yesengazyevna, S. A., Niyetbayeva, N., Tassuov, B., Kalima, T., and Bekbulatovna, A. A. (2022) "Teaching students programming with the help of educational games in the conditions of additional education in computer science", *Cypriot Journal of Educational Sciences*, Vol. 17, No. 6, pp. 1943-1956.

[25] De Santo, A., Farah, J. C., Martinez, M. L., Moro, A., Bergram, K., Purohit, A. K., Felber, P., Gillet, D., and Holzer, A. (2022) "Promoting computational thinking skills in non-computer-science students: Gamifying computational notebooks to increase student engagement", *IEEE Transactions on Learning Technologies*, Vol. 15, No. 3, pp. 392-405. https://doi.org/10.1109/TLT.2021.3126906

AUTHORS

Siti Sakinah Mohd Yusof is a Computer Science lecturer at Kolej Matrikulasi Perak and a PhD candidate in Education (Information Technology) at Universiti Pendidikan Sultan Idris (UPSI), focusing on enhancing programming problem-solving skills through innovative learning approaches.

Dr Haslinda Hashimis a lecturer from the Faculty Of Computing And Meta-Technology, Universiti Pendidikan Sultan Idris (UPSI), Tanjung Malim, Perak, Malaysia. Specializes in Instructional Technology, Educational Technology, Computing Education & Teacher Training.

Erni Marlina Saari is a lecturer from the Faculty Of Computing AndMeta-Technology , Universiti Pendidikan Sultan Idris (UPSI), Tanjung Malim, Perak, Malaysia. Specializes in Multimedia in Education; ICT ; Game-Based Learning; Computational Thinking Skills.