MALAYSIAN JOURNAL OF LEARNING AND INSTRUCTION
https://e-journal.uum.edu.my/index.php/mjli

How to cite this article:
Handayani, R. D., Prastowo, S. H. B., Jamalludin., Putra, P. D. A., & Thohir, M. A. (2026). Exploring the computational thinking skills of secondary school students who use supcath in the science classroom. *Malaysian Journal of Learning and Instruction, 23*(1), 118-138. https://doi.org/10.32890/mjli2026.23.1.6

EXPLORING THE COMPUTATIONAL THINKING SKILLS OF SECONDARY SCHOOL STUDENTS WHO USE SUPCATH IN THE SCIENCE CLASSROOM

[1]Rif'ati Dina Handayani, [2]Sri Handono Budi Prastowo, [3]Jamalludin, [4]Pramudya Dwi Aristya Putra & [5]Mohammad Anas Thohir

[1&2]Physics Education Department, University of Jember, Indonesia
[3]Magister Science Education Department, University of Jember, Indonesia
[4]Science Education Department, University of Jember, Indonesia
[5]Department of Elementary School Teacher Education, Universitas Negeri Malang, Indonesia

[1]Corresponding authors:rifati.fkip@unej.ac.id

Received: 2/11/2024 Revised: 3/6/2025 Accepted: 22/1/2026 Published: 31/1/2026

ABSTRACT

Purpose – This study has investigated the use of the SUPCATH educational game in science classrooms to explore students’ computational thinking skills. SUPCATH is an innovative educational game that combines elements of action and strategy.

Methodology – This study involved 21 seventh-grade students from a public secondary school in East Java, Indonesia. The study sample was selected through purposive sampling. The research employed a qualitative approach, with data collected through observations, semi-structured interviews, and gameplay recordings. Content analysis was used to examine students’ computational thinking processes as these were demonstrated during gameplay and reflected in their classroom interactions and verbal responses.

Findings – The study revealed that students could solve open-ended problems by applying structured and logical reasoning throughout their gameplay. Their engagement with tasks involving decomposition, abstraction, and algorithmic thinking indicated a growing capacity for systematic problem-solving. Students showed increased enthusiasm and persistence when iterating solutions, often verifying outcomes using data-driven approaches. These patterns suggest that the SUPCATH game effectively fosters analytical thinking and enhances core computational thinking skills.

Novelty – The novelty of this study lies in its detailed development of the SUPCATH educational game, an interactive and immersive learning tool which has been specifically designed to cultivate students' computational thinking skills through structured science content on substance pressure. Unlike existing educational games that emphasize general problem-solving or coding activities, SUPCATH directly integrates core elements of computational thinking into the context of the secondary science curricula. Its classroom implementation offers a subject-specific and pedagogically grounded approach that enhances students' engagement and cognitive development.

Significance – This study highlights the potential of game-based learning which could the integration of computational thinking in science education. The findings provide practical guidance for science teachers in designing classroom activities that promote structured thinking and problem-solving.

Keywords: Computational thinking, educational games, logical reasoning, science classroom learning, secondary students, systematic thinking.

INTRODUCTION

The dynamic era of the 21st-century requires every individual to think logically and solve problems systematically. Computational thinking is critical for K-12 education because it helps students understand how technology works and improves their ability to analyze complex issues (Yadav et al., 2016). Skills that promote the ability to engage in logical thinking and systematic problem solving are what all 21st-century pupils at all educational levels should develop, as these abilities will develop a problem-solving mindset (Angeli & Giannakos, 2020; Zhang & Nouri, 2019). Shute et al., (2017) defined computational thinking as a process that involves formulating issues and solutions, with alternative solutions implemented more effectively and efficiently. Computational thinking is a cognitive process involving rational, logical thinking in solving problems; it is through such a rational and logical mode of thinking that artifacts, methods, and systems are better understood (Kayhan et al., 2024). Computational thinking involves several high-level skills (Lee & Malyn-Smith, 2019; Sung & Black, 2020). That is why providing students with the set of required skills to solve higher-order thinking problems will help them deal with increasingly complex and dynamic developments in the future.

Implementing computational thinking is an exciting and challenging vision for future learning. This is because computational thinking involves multiple levels of cognition that can be applied across subjects and disciplines in science, technology, engineering, mathematics, social sciences, and literacy (Küçükaydın & Çite, 2024; Lee & Malyn-Smith, 2019). Computational thinking is an intellectual process involving the use of a set of skills (Wing, 2006). From a pedagogical perspective, computational thinking skills can deepen the learning of mathematics and science content (Sung & Black, 2020; Weintrop et al., 2016). Furthermore, computational thinking can provide a variety of intellectual and noncognitive advantages (Zhao & Shute, 2019). Students can significantly connect their thinking and problem-solving skills through contextual study. In the context of the secondary school, computational thinking skills are rarely applied because students perceive them as complicated and challenging to master (Handayani et al., 2022). They are unfamiliar with computational thinking skills, such as solving more straightforward problems and recognizing patterns (Handayani et al., 2023). One promising direction is using educational games which will align directly with the prevailing curriculum content, offering students an engaging, interactive, and cognitively stimulating environment for practicing computational thinking (Lu et al., 2022).

An educational game is characterized by interactive learning experiences in an exciting environment (Yaman et al., 2023). Games can boost pupils' attention, intellectual strength, and motivation. Wang and Zheng (2020) argued that educational games are more effective at achieving student outcomes than conventional learning. Weintrop et al., (2016) and Wu (2018) have pointed out that educational games provide new experiences for students to develop their computational thinking skills and encourage their interest in programming. The game is enjoyable, and fascinating for individual and collaborative efforts (Cai et al., 2025). Therefore, incorporating games into learning can be an alternative for students to practice computational thinking in a more enjoyable and engaging manner. However, despite the growing interest in game-based learning, most previous studies have focused on general CT skills in programming-heavy contexts, often overlooking how these skills are naturally enacted within domain-specific science topics. Furthermore, there is a lack of empirical evidence detailing the step-by-step cognitive process of students as they navigate science challenges through gameplay. The present study fills this gap by exploring how secondary students demonstrate specific computational thinking dimensions within a science classroom setting. The findings offer new insights into integrating CT into the science curriculum and will provide design implications for teachers seeking to use games as a tool for systematic problem-solving. Consequently, the research question posed is: How do students demonstrate computational thinking skills through playing educational games in their science classrooms?

LITERATURE REVIEW

Computational Thinking Skills in Learning

Computational thinking is a way of thinking and acting demonstrated through specific performance-based skills (Liu, 2024; Wang et al., 2018). It is a fundamental thinking skill involving logical reasoning to solve complex problems (Şen, 2023). Computational thinking is analytical and precise thinking that consists of implementing computational systems to clarify and solve problems (Chen & Chung, 2024). Alfayez and Lambert (2019) stated that computational thinking is an intellectual skill, a practice, and a set of methods for solving complex problems. Computational thinking skills are not limited to computers but are universal because they involve the mind in meaningful thinking processes. The scope of implementing computational thinking is comprehensive, including knowledge, attitudes, and general practices (Adanır et al., 2024). Most of the literature characterizes computational thinking as a multifaceted skill set comprising abstraction, algorithmic thinking, problem decomposition, and debugging (Turchi et al., 2019). Abstraction is the ability to think at several levels and model the main components of an issue by dismissing unnecessary information (Xing & Zeng, 2024). Algorithmic thinking is the capacity to develop procedures as ordered steps to accomplish solutions (Bayeck, 2024). Problem decomposition breaks complex concerns into manageable components and units (Pan et al., 2024). Debugging focuses on identifying and correcting errors when algorithms fail to produce the intended solution (S.-J. Chen et al., 2025). These interrelated skills highlight the multifaceted nature of computational thinking as both a cognitive process and a practical approach to learning and problem-solving.

Integrating computational thinking into classroom instruction offers students a more authentic and contextualized learning experience that aligns with the demands of future careers (Araujo et al., 2019). Beyond technical skills, computational thinking supports cognitive and noncognitive abilities, such as analytical reasoning, mathematical thinking, and overall academic performance (Boucinha et al., 2019; Wu & Yang, 2022). As a foundational skill, computational thinking not only enhances students' problem-solving capacity, but also encourages deeper engagement with complex and meaningful content (Lee & Malyn-Smith, 2019; Sung & Black, 2020). When embedded in learning, it reshapes students' perspectives on knowledge and fosters thinking strategies essential for navigating future challenges.

Despite the recognized importance of these skills, there is a notable gap in practical pedagogical approaches that integrate computational thinking into science classrooms in ways that are engaging and accessible to students (Yadav et al., 2016). Traditional teaching methods tend to fail in motivating students or provide meaningful opportunities to develop computational thinking (Angeli & Giannakos, 2020). Many learning designs still focus on procedural knowledge and rote learning, offering limited space for exploration, problem-solving, and iterative thinking—core aspects of computational thinking (Hooshyar et al., 2020). As a result, students often struggle to transfer these abstract skills to real-world scientific problems, especially in content-heavy subjects.

Teaching Computational Thinking Skills through Educational Games

Educational games have increasingly gained recognition as a practical instructional approach to support the development of computational thinking (CT) in K12 learning environments (Liu, 2024). Unlike traditional instructional methods, educational games offer interactive, dynamic, and student-centered learning experiences that can transform how students engage with complex content. These games are not merely tools for entertainment but serve as structured pedagogical frameworks that align with curriculum goals and support teaching and learning processes (Li & Li, 2025).

Research has shown that educational games enhance student outcomes, including problem-solving abilities, motivation, engagement, and computational thinking skills (Stohlmann, 2023). For example, Wanglang et al., (2024) found that well-designed educational games significantly improved students' conceptual understanding and CT competencies. Similarly, Ayman et al., (2018) and Wang et al., (2023) emphasized that game-based learning encouraged students to actively participate in class activities, which is considered critical for fostering deep learning and sustained interest. One of the key strengths of educational games lies in their ability to simulate real-world problems and help visualize abstract concepts. This makes learning more contextual, practical, and meaningful for students (Hsu et al., 2018; Wang et al., 2020). When students are immersed in problem-solving scenarios through gameplay, they are more likely to apply skills such as decomposition, abstraction, and algorithmic thinking in authentic ways.

Moreover, the design elements embedded in educational games, such as clear goals, role-based challenges, real-time feedback, compelling storylines, opportunities for collaboration, and reward mechanisms, they play a vital role in increasing students' intrinsic motivation and engagement with computational thinking

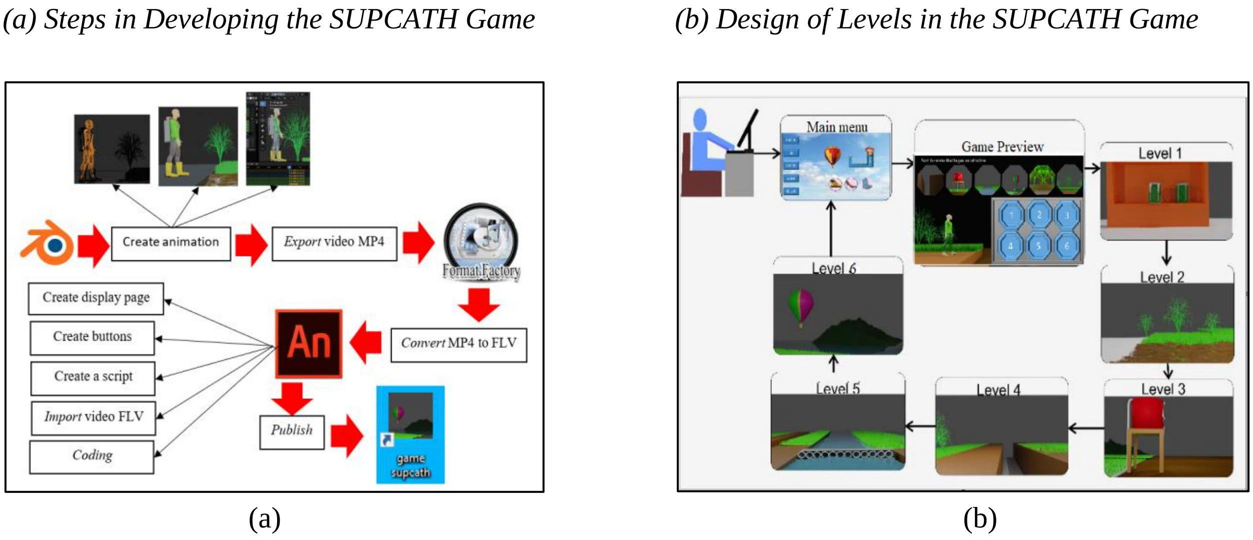
(Agbo et al., 2021; Alfaro-Ponce et al., 2023; Kim et al., 2022). These elements make learning enjoyable, foster persistence in tackling complex tasks, and create a safe space for experimentation and failure, which is essential for developing CT skills. In sum, educational games provide a unique blend of cognitive challenge and motivational support, making them a valuable strategy for embedding computational thinking in the science classroom.

Overview of the SUPCATH Educational Game

The term "SUPCATH" stands for substance-pressure computational thinking. This game has six levels and was developed to build students' fundamental computational thinking skills in the secondary school where students are aged between 12 to13 years old, in a more engaging way. Figure 1 presents the steps for creating and building a SUPCATH game. Three software tools, namely blender3d, FormatFactory, and Adobe Animate, were used in the production of the SUPCATH educational game. Each of these software tools have the following specific functions: blender3d software creates three-dimensional, colorized animated films and motion graphics; FormatFactory converts the MP4 video format to FLV, which is then used as input in Adobe Animate software; and Adobe Animate finishes the game's appearance and features with display pages, menus, animated videos in FLV format, coding pages, and buttons. The present study's researchers produced storyboards and drawings for every level of the SUPCATH game's development.

Furthermore, the SUPCATH game was developed by adopting indicators of computational thinking skills proposed by Shute et al., (2017), which included the following: decomposition, abstraction, algorithm design, and debugging. In devising these computational thinking indicators, Shute et al., (2017) sought to understand the cognitive processes and associated behaviors that can systematically develop students' competence in the science classroom. Table 1 details the computational thinking skills in the SUPCATH educational game.

Figure 1



(a) Steps in Developing the SUPCATH Game

(b) Design of Levels in the SUPCATH Game

(a)

(b)

Table 1

Aspects of Computational Thinking Skills in the SUPCATH Educational Game

Computational Thinking Skills	Description	Features in the SUPCATH educational game
Decomposition	Solving problems becomes simple: break them down into small parts.	Break down the pressure problem into manageable units and classify the pressure of a substance as solid, liquid, or gas.
Abstraction	Find keywords and relevant information, filter out irrelevant data, and identify patterns.	Find and select important information about games and tools related to solids, liquids, and gas pressure at each game level.
Algorithm design	Create a solution step and a design step solution.	Create a solution to the sequence of steps for solving the pressure problem at each game level.
Debugging	Conduct an evaluation, correcting mistakes and errors.	Test and retest the solution in the game, and evaluate its strengths and weaknesses.

The SUPCATH game was designed and developed through a structured process grounded in a needs assessment that analyzed students' characteristics, available learning resources, and the science content on substance pressure. The game underwent an expert validation phase to ensure its pedagogical validity and suitability for fostering computational thinking (CT) skills. Five professionals, namely two science education experts, one educational technology specialist, and two instructional designers, were invited to review the game. They evaluated it based on criteria such as content relevance, clarity of instructions, alignment with CT indicators, user interface design, and overall educational effectiveness. Revisions were made to improve clarity, functionality, and learning outcomes before the game's classroom implementation.

METHODOLOGY

Research Design

This study employs a descriptive qualitative research design to explore in depth students' computational thinking processes within a Game-Based Learning environment. A qualitative approach was deemed most appropriate for this investigation as it enables researchers to examine complex phenomena in their natural setting, focusing on the how and why of student learning, rather than merely measuring outcomes (Creswell & Creswell, 2023). This methodological choice prioritizes the educational context, enabling a comprehensive examination of the dynamic interplay between the digital tool and students' reasoning abilities in real time. The design facilitates the observation of pedagogical effectiveness not just in terms of final success, but through the lens of struggle, debugging, and iterative problem-solving.

Participants

The study was conducted at a public secondary school in East Java, Indonesia. Participants were selected using purposive sampling. Purposive sampling is often used to select individuals with particular attributes, experiences, or contextual insights that align with the study's objectives (Naeem et al., 2023). The 21 participants (aged 12–13 years) were in seventh grade, including 17 females (81%) and four males (19%) with diverse academic abilities ranging from low to high achievement. Participants were selected based on their willingness to engage in game-based activities and their ability to share their learning experiences during interviews. Before the selection process, a brief survey was conducted to assess their basic computer skills, including in using word processing software, searching online, and browsing the internet. The selection criteria were determined based on these basic skills to ensure that all participants could participate in the study's CT classroom activities. No prior experience in programming or computational thinking was required. The data used to determine the selection criteria came from the study's survey and interviews with potential participants. Ethical considerations were strictly observed. Approval was obtained from the school's ethics committee, and all participants and their guardians had provided informed consent. The study ensured confidentiality, anonymity, and transparency throughout the research process. Additionally, participation in the study did not affect students' academic grading or final evaluations.

Data Collection

The study collected data from observation, interviews, and gameplay recordings. The observation method was the primary source used in answering the study's problem formulation and achieving its research objectives (Creswell, 2023). Observation is a critical method for evaluating instructional effectiveness and uncovering pedagogical strengths and areas for improvement. It also offers nuanced insights into student engagement that are often inaccessible through interviews (Tarusha & Bushi, 2024). Observational data were collected during three science learning classroom sessions, with each class lasting 80 minutes. These sessions were part of a planned instructional sequence integrating the SUPCATH educational game into substance-pressure material. A team of five trained observers assisted in documenting student activities throughout the study. Before data collection, observers were briefed by the study's researchers and they were provided with a structured observation protocol. The protocol included guidance on observer positioning, ethical conduct, for example, confidentiality and non-intervention, and standardized field notes. The observation checklist was developed based on key components of computational thinking adapted from widely recognized CT frameworks. Indicators included observable behaviors reflecting abstraction, as in identifying relevant variables; decomposition, as in breaking problems into smaller parts; algorithmic thinking, as in proposing steps or strategies to solve problems; and debugging, as in revising decisions or identifying errors during gameplay and problem-solving.

Furthermore, semi-structured interviews were used in the present study. The semi-structured interview design was selected because it is considered an interactive data-gathering method that will provide a deeper insight into how and why certain things occur, as well as the opinions, interests, and thoughts of the people involved (Jain, 2021). Semi-structured interviews help researchers gather more detailed and in-depth information about a topic (King et al., 2019). The interview protocol of the present study was developed to guide the semi-structured interviews, ensure consistency across sessions, and allow open-ended exploration. The protocol consisted of guiding questions designed to elicit students' experiences, perceptions, and reflections on their computational thinking processes while engaged in the use of the SUPCATH educational game. The interviews were conducted with all the 21 student participants to obtain more in-depth information. Each interview lasted approximately 15–20 minutes and was conducted individually in a quiet setting to allow focused discussions.

The interview guide included prompts as follows: *"What did you enjoy most about learning through educational games?"*; *"Which parts of the game were the most challenging?"*; *"In what ways did the game help you think more critically or solve problems?"*; and *"Can you describe how you approached solving a problem in the game?"*. Interviews were audio-recorded with participants' consent and transcribed verbatim. Last, the qualitative data source was recorded data from the classroom sessions in which students were engaged in the SUPCATH educational game. The computer system will record every item and level students perform and complete while playing the game.

Data Analysis

Observations, interviews, and gameplay recordings were analyzed using the method of content analysis. Content analysis is a method for systematically studying human communication in various forms, such as written texts, images, and audio (Baxter, 2020). Content analysis offers several benefits, including flexibility, applicability, and the ability to study sociocognitive concepts without intrusive methods (Reger & Kincaid, 2021). The analysis began with a close reading of the transcripts, followed by categorizing and coding the data using a coding scheme derived from established computational thinking frameworks. The core categories included abstraction, decomposition, algorithmic thinking, and debugging. Coding was conducted through a content analysis approach based on established computational thinking indicators. Data were coded independently by the research team members. To ensure the credibility and dependability of the findings, the researchers employed a peer debriefing strategy, in which initial codes were cross-checked and discussed during regular sessions. Any discrepancies in interpretation were resolved through in-depth discussions until a complete consensus was reached on the final themes. This process ensured that the categories consistently reflected the students' computational thinking behaviors across different data sources. Each category was then described in terms of the depth and variation of students' responses or behaviors related to the specific indicators of computational thinking. To improve the trustworthiness of the findings, triangulation was conducted across the following three data sources: classroom observations, interview transcripts, and game interaction recordings. The same indicators of computational thinking were used across all data types to ensure consistency. The findings were further validated by linking them to relevant theories and prior studies, thereby strengthening analytical rigor and grounding them in empirical evidence and conceptual frameworks.

RESULTS

Secondary Students' Computational Thinking Skills

Computational thinking is the process of logical reasoning and problem-solving that is ordered and structured. Introducing computational thinking to secondary school students should receive more attention, as they are developing relationships with specific academic disciplines and establishing their first career orientations. The SUPCATH educational game was created in the context of science learning to train students in the basic computational thinking skills of decomposition, abstraction, algorithm design, and debugging. The games were designed to allow learners to access more exciting parts of the computing subject matter. Based on the data analysis, the results for the categories and codes are as presented in Table 2. These findings have helped to address study's objective to explore how students demonstrate algorithmic thinking in a game-based science learning context.

Table 2

Categories and Code Data of Students' Computational Thinking Skills

Categories	Code	Examples of students' activities
Abstraction	Abs	1. Search for the main game characters in each level 2. Distinguishing between relevant physics variables 3. Collecting data related to the pressure of the subject topic
Decomposition	Dcm	1. Categorizing pressure types 2. Sort and separate necessary tools by function and type in the game levels
Algorithm design	Ald	1. Complete the puzzle following the structured solution 2. Create a sequence for each level based on a similar pattern
Debugging	Dbg	1. Testing and rearranging the solution for game completion 2. Check and recheck the steps for completing the game

Abstraction

Abstraction is the process of solving a problem by concentrating on its most essential components. Abstraction involves a student's ability to simplify a complex issue by focusing on the relevant elements and ignoring unnecessary details. Research findings indicated that all 21 student participants did well in demonstrating abstraction skills. This proficiency was evidenced by their ability to filter out non-essential game elements. For instance, F1's statement about ignoring background decorations and focusing only on the pressure tools (see Interview, F1) reflects the cognitive process of reduction, in which a student simplifies a complex problem by focusing on the variables that affect the physical outcome. Abstractions are evident from the beginning of the game's activities. When playing games, students need to understand the game's patterns and rules, ignore irrelevant information, and develop strategies. This involves the ability to recognize the main components of the game, for example, the end goal, obstacles, or critical rules, so that they can focus on the information relevant to achieving their goal. Students carefully read the guidance instructions for the SUPCATH game and solved the problem by looking for learning material about their study topic. They looked for information about the main game characters, the levels, and what must be completed in each level. In this activity, students also collected the necessary tools by function and type, such as bags, reservoirs, and shoes, which are essential for completing the game.

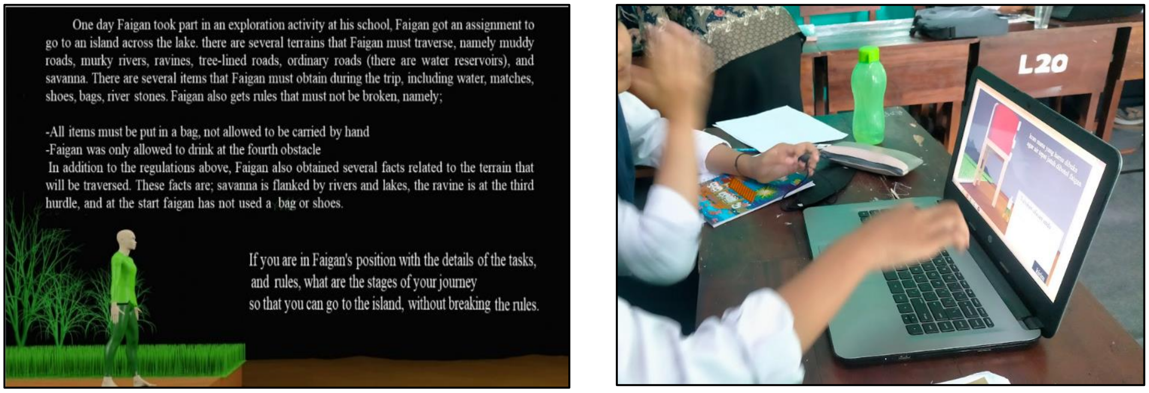
> *"There was much information at the beginning, but I tried to find what was important. I focused only on the goal and the tools needed to achieve it. I didn't pay much attention to parts that didn't help solve the problem." (Interview, F1)*

> *"I ignored the background decorations and only looked at the items and instructions. I knew that the game wanted me to solve a pressure problem, so I just searched for what was related to that topic." (Interview, M3)*
>
> *"I read the instructions first and tried to understand the main problem. I didn't try everything; I picked the tools that made sense. For example, if it's about gas pressure, I immediately think of the balloon and leave out other stuff." (Interview, F4)*

The ability to find critical information and keywords, and sort and separate, assists students in finding solutions to problems more efficiently. In simple terms, abstraction in the SUPCATH educational game trains students to focus on the most critical data and evidence, thereby deepening the scope of the problem in the game and helping them understand the concept and provide the right solution. Figure 2 shows an example of students' activity in developing the skill of abstraction in the SUPCATH game.

Figure 2

An Example of a Student's Activity in the Abstraction Process When Playing the SUPCATH Educational Game



One day Faigan took part in an exploration activity at his school, Faigan got an assignment to go to an island across the lake. there are several terrains that Faigan must traverse, namely muddy roads, murky rivers, ravines, tree-lined roads, ordinary roads (there are water reservoirs), and savanna. There are several items that Faigan must obtain during the trip, including water, matches, shoes, bags, river stones. Faigan also gets rules that must not be broken, namely;

-All items must be put in a bag, not allowed to be carried by hand
-Faigan was only allowed to drink at the fourth obstacle
In addition to the regulations above, Faigan also obtained several facts related to the terrain that will be traversed. These facts are; savanna is flanked by rivers and lakes, the ravine is at the third hurdle, and at the start faigan has not used a bag or shoes.

If you are in Faigan's position with the details of the tasks, and rules, what are the stages of your journey so that you can go to the island, without breaking the rules.

Decomposition

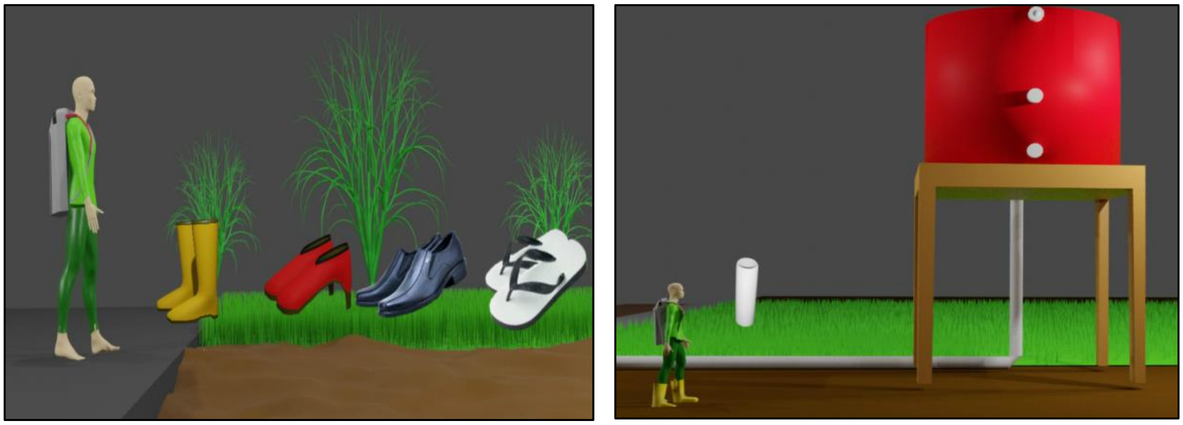
Decomposition involves breaking a complex problem into smaller, more manageable sections. In the context of the SUPCATH game, decomposition helps students identify key components and specific steps required to overcome scientific challenges. Rather than relying on trial-and-error, students who apply decomposition can develop strategic solutions that involve resource planning and action prioritization. Data analysis showed that 18 of the student participants demonstrated competent decomposition skills. These students did not merely group items; they systematically partitioned a complex scientific domain—substance pressure—into specific conceptual sub-units, such as hydrostatic, solid, and gas pressure. This logical breakdown was crucial for identifying the appropriate physical principles required for each game level. Students demonstrated these skills by breaking down game tasks into the following simple steps: identifying patterns, arranging objects in a functional order, and determining the priority of actions. For instance, students categorized the issues faced by game characters by grouping data related to the specific pressure concepts at each level—linking reservoirs to hydrostatic pressure, shoes to solid pressure, and balloons to gas pressure. This process of conceptual partitioning enabled them to apply targeted physical laws to solve the specific problem at hand. Evidence of this systematic decomposition can be seen in the students' strategic planning during gameplay as follows:

> *"When I first saw the problem in the game, I didn't understand what to do. But then I looked at the available tools and grouped them by the kind of pressure they relate to. For example, the shoes must be linked with solid pressure, and the reservoir with water pressure. That made it easier to plan what I should do first." (Interview, M1)*
>
> *"I didn't try everything at once. I separated the steps. First, I checked the character's needs, decided what items could help, and tested the action. If it failed, I tried another order. It was like making a plan, step by step." (Interview, F2)*
>
> *"I always look at what each level is asking—some are about gas, others about water. I make small notes in my mind: gas = balloon, water = reservoir. That way, I know what is important in each level." (Interview, F6)*

These excerpts illustrate that students used decomposition to move beyond trial-and-error. By categorizing objects, students reduced the cognitive load required to solve the puzzles. This grouping process allowed them to prioritize actions and design effective strategies. Figure 3 illustrates this decomposition process, in which students separated tasks into stages, ensuring that each physical variable was systematically addressed before moving on to the next challenge. By practicing these skills, students improved their ability to manage complex scientific issues through a structured, step-by-step approach.

Figure 3

The Display of a Decomposition Process in the SUPCATH Educational Game



Algorithm Design

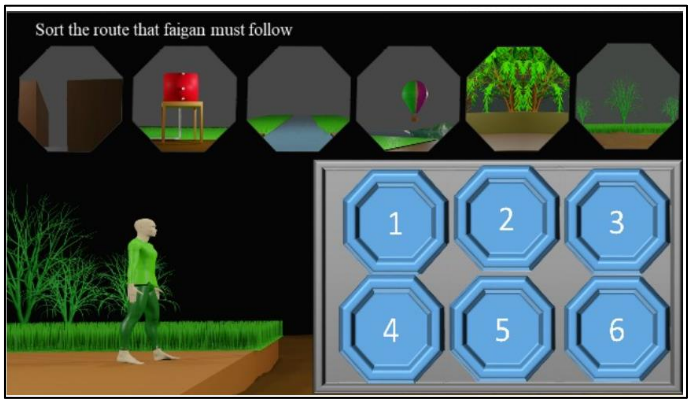
Algorithm design is a complex cognitive process in which students use predictive abilities and logic to construct a series of systematic procedures for problem-solving. In the present study, algorithm design skills emerged as students developed structured sequences to solve game-based science challenges. The 17 student participants who precisely designed algorithms displayed conditional logic (if-then thinking). As noted by F7, planning moves before execution suggests that students were mentally simulating the outcomes of their actions. This process is an indication of learners' use of computational modeling, in which learners create a mental simulation of steps to predict scientific results within the game environment. The emergence of algorithm design was evident when students formulated step-by-step strategies to reach goals or avoid obstacles (see Figure 4). Students constructed specific sequences, such as selecting items and moving objects in a precise order, to advance through the levels. This involves the ability to design instructions that can be followed repeatedly with consistent results. The following excerpts illustrate how students had internalized these logical sequences:

> *"I always planned my moves before I started. First, I looked at the steps I needed to do, then I followed them one by one to make sure I didn't miss anything. Sometimes, if I made a mistake, I would try again but change the order to see what worked better." (Interview, F7)*
>
> *"When playing the game, I arranged the actions carefully, like picking the right items first, then using them in order. I repeated the steps until I could solve the puzzle without errors." (Interview, F5)*
>
> *"At first, I wasn't sure about the steps, so I tried different sequences a few times. After some tries, I figured out the correct order to finish the level faster." (Interview, M1)*

These responses suggest that students were not merely clicking randomly; they were developing replicable solutions. By carefully arranging actions, students transitioned from trial-and-error to a more structured, algorithmic approach. This practice allows students to develop logical and efficient solutions that can be applied across various game contexts. In short, implementing these rational algorithms trains students to think in a structured way, enabling them to predict outcomes and refine their procedures based on instant feedback from the game.

Figure 4

The Display of a Puzzle Level in the SUPCATH Educational Game



Sort the route that faigan must follow

Debugging

Debugging is an evaluative process that involves identifying, analyzing, and correcting errors within a strategy. In the present study, debugging skills emerged when students recognized discrepancies between their expected outcomes and the actual game results. This ability is essential in science-based games, where students must pinpoint specific procedural failures when their hypotheses or actions do not yield the desired results. Data analysis indicated that all 21 student participants engaged in debugging by iteratively testing and rearranging their solutions until they could successfully continue to play the game to the next level. Students demonstrated debugging by systematically re-examining their steps after they had failed to overcome an obstacle. Rather than repeating the same mistakes, they looked closer at the game's feedback to identify which part of their logic needed adjustment. This process of check-and-recheck signifies a transition from impulsive action to reflective problem-solving. The following interview excerpts illustrate this iterative process:

> *"Whenever I couldn't pass a level, I would carefully check what I did wrong. Sometimes I rearranged the steps or tried a different approach until the game worked." (Interview, M3)*

> *"I often repeated the parts that failed and looked closely at each action to find the mistake. It helped me fix the problem and continue playing." (Interview, F8)*

> *"If the game didn't respond as I expected, I tested the sequence again and changed the order. This way, I learned from my errors and improved my strategy." (Interview, F1)*

These responses highlight that debugging in the SUPCATH game fosters persistence and a growth mindset. By treating failure as a source of feedback rather than a final result without further deliberation, students developed a willingness to iterate on their responses. This systematic error-correction process trains students to think critically about their own mental models. Through these practices, students' debugging skills evolved from a simple trial-and-error process to a more structured evaluation of their scientific reasoning, which is a vital component of advanced computational thinking.

Student Response

Games that are interesting and challenge computational thinking skills will trigger students' enthusiasm and active involvement. The interview analysis showed that most students are interested in and satisfied with learning via the SUPCATH educational game. All participants agreed that learning through digital educational games made the science class exciting and enjoyable. The game has made students happy and enthusiastic in science lessons. They considered learning science through games more entertaining, less stressful, and less tiring. Students feel challenged to solve problems, try new strategies, or find specific patterns in the game. This challenge encourages them to try different approaches to achieve their goals. The SUPCATH educational game trains them to simplify substance problems, find critical information, engage in algorithmic thinking, and evaluate their decision-making process. In short, students agreed that implementing game-based learning helps them explore their skills in more fun ways. Below is an example of students' positive remarks in using SUPCATH to engage in computational thinking:

> *"Learning through games makes me happy since it is less monotonous and boring…. The display and challenge at all levels of the SUPCATH educational games are satisfactory, and the language is clear" (Interview_F8).*
>
> *"Learning through games teaches us to persist and play more carefully…. Playing games trains us to solve problems in greater detail because if we follow the wrong pattern, the game will not complete" (Interview_F6)*
>
> *"Learning science through games is entertaining. The game improves our scientific knowledge, reasoning, and motivation in the science classroom" (Interview_M9)*
> *"Educational games kept us engaged because each level was tied to contextual issues" (Interview_M1)*

Furthermore, students were also satisfied with the appearance and levels of the game because the language was easy to understand, the software was simple to run and user-friendly, and the game was attractive and entertaining. They were happy because science learning has become more fun and less repetitive than regular learning. The game has allowed students to learn through hands-on experience and provided them instant feedback on their decisions. This direct interaction made the students feel more engaged and connected to the activity. They could try different solutions, see the results, and learn from their mistakes, which have made the learning process more enjoyable. Students felt that the educational games used in their learning have challenged them to complete the games correctly and sparked their curiosity. The reason given was that they had to create a level sequence first so that all levels in the game could be opened and the game could be completed.

DISCUSSION

Computational thinking is an intellectual skill, practice, and procedure used for problem-solving. Implementing the SUPCATH educational game aims to introduce computational thinking to students in science learning in a more engaging way. Acquah & Katz (2020) posit that educational games can potentially engage students' interests and enhance their enthusiasm for studying. Educational games also help students build computational thinking skills and basic computer programming knowledge and serve as motivators (Giannakoulas & Xinogalos, 2023). The findings from the present research have illustrated that students are skilled at engaging in games using computational thinking skills, namely abstraction, decomposition, algorithm design, and debugging. Students became fascinated and would attempt to learn more about the SUPCATH game. The SUPCATH games provide an exciting and enjoyable environment for pupils to choose important information, simplify processes, and solve problems carefully.

In this study, decomposition and abstraction served as forms of cognitive scaffolding, helping students define complex issues and extract critical scientific information. As noted by Ehsan et al. (2020), decomposition and abstraction train students to identify a problem's boundaries and constraints, restate their goals, and familiarize themselves with the task's requirements. Specifically, when students decomposed pressure into hydrostatic, solid, and gas categories, they were not merely playing; they were mapping scientific concepts onto computational structures. This finding suggests that CT is not just a computing skill, but a vehicle for deeper scientific inquiry, a perspective that is increasingly relevant in the global discourse on STEM integration (Lu et al., 2023).

Furthermore, the findings revealed that educational games foster students' analytical thinking and logical reasoning during problem-solving. Rational thinking about a scientific problem facilitates the production of accurate solutions through iterative evaluation. This aligns with the findings of Agbo et al., (2021), who have found that analyzing and verifying facts through a clear, coherent, and accurate mindset constituted a systematic thinking process. This process enables students to draw upon their internal mental models to make and verify predictions. In the use of SUPCATH, this was evident as students transitioned from impulsive trial-and-error to structured, algorithmic reasoning, actively predicting potential outcomes before proceeding to execution.

Debugging is an evaluation process that involves identifying and correcting errors while testing solutions to ensure functionality (Sun et al., 2024). In this study, students engaged in continuous cycles of checking and rechecking their strategies to ensure the science challenges were completed adequately. Rather than passive repetition, students verified, tested, and rearranged their tactical solutions to resolve obstacles within the game levels. This iterative behavior suggests that the SUPCATH game provides an effective environment for fostering students' debugging skills. Such checking and rechecking behaviors, even when involving initial trial-and-error, are core characteristics of scientific thinking (Veenman et al., 2022). Furthermore, students who successfully resolved algorithmic problems demonstrated the ability to identify and systematically improve their solutions. In this context, debugging serves as a metacognitive tool, helping students reflect on their decisions and enhance their computational thinking through structured refinement. As noted by Sun et al. (2024), this process constitutes reflection-on-action, in which students evaluate the accuracy and appropriateness of their scientific reasoning during problem-solving.

In addition, the findings of this study have shown that there was a highly positive response toward the use of educational games. Participants found the SUPCATH experience engaging and novel, noting that the contextual levels, which are closely related to daily life, help to alleviate the boredom often associated with traditional science instruction. This heightened engagement has led to increased active involvement, helped to drive curiosity and increase the desire for immediate gameplay. Such findings align with the standpoint of Tavares (2022), who argued that games could provide immersive experiences that would significantly boost student involvement. Similarly, Ranieri et al., (2018) suggested that enjoyable learning environments has enabled students to grasp complex topics more effectively and achieve learning objectives more easily.

In this study, the enjoyment derived from gameplay catalyzed analytical, systematic, and logical problem-solving. As students became acclimatized to challenges demanding decomposition, abstraction, and algorithmic design, their thought patterns would become more structured. The advantages of students applying CT included fostering in them the ability to brainstorm and generate practical solutions to scientific problems, a finding corroborated in the studies by Mao et al. (2022) and Adipat et al. (2021). By analyzing and verifying facts through careful reasoning, students used logical reasoning to navigate the game's complexities. Moreover, the game-based environment encouraged students to make decisive actions, test hypotheses, and still persist eventhough encountering failure. This persistence is vital; by considering problems from various perspectives and evaluating the strengths of their solutions, students developed a resilient approach to challenging scientific issues.

While the study context is local, the implications of the present study's findings can be effectively applied to other different international educational environments. The findings from the present study will therefore, have extended their applicability and achieved cumulative impact for the field. The design principles of SUPCATH, which incorporated CT dimensions, have mapped these directly onto science-specific variables. The teaching of CT using SUPCATH can be adopted by teachers globally to bridge the gap between abstract science concepts and systematic problem-solving. This aligns with international frameworks, such as PISA or NGSS, that emphasize the integration of computational modeling in science curricula. By situating CT within a domain-specific context, such as substance pressure, this research provides a replicable model for fostering 21st-century skills without isolating them from the core subject matter.

Despite overall positive demonstrations of students' computational thinking skills, specific challenges can still emerge, particularly in algorithm design. Several students struggled to devise precise, step-by-step procedures to solve complex game levels. This difficulty may be attributed to limited previous exposure to logical structuring or procedural problem-solving within their formal science instruction. Furthermore, the tendency of some students to repeat unsuccessful trials without modifying their strategies indicates a lack of metacognitive awareness. These findings suggest that future game-based interventions should integrate metacognitive prompts, worked examples, or peer collaboration strategies to scaffold students' understanding. This is important to facilitate the transition from intuitive actions to structured algorithmic reasoning in problem-solving contexts.

Limitations of the Study

This study acknowledges several limitations. Given its qualitative and exploratory nature, the findings are based on a specific group of secondary students and a particular science game context focusing on the topic of substance pressure. Therefore, the results are not intended for broad generalization. Instead, the study aims at transferability, providing a thick description that allows other researchers to apply these insights to similar educational settings. Furthermore, because the analysis focused on the immediate enactment of CT skills during the intervention, it may not capture the long-term retention of these abilities. Future research should consider a larger, more diverse sample and incorporate control groups to better measure the impact of educational games such as SUPCATH. Additionally, further game development across various science topics is recommended to examine the consistency of CT integration across disciplinary contexts.

CONCLUSION

This study concludes that the educational games SUPCATH serves as an effective pedagogical tool for engaging secondary students in computational thinking within the science curriculum. The findings demonstrate that students successfully enacted core CT dimensions, specifically abstraction, decomposition, and debugging, by systematically partitioning complex scientific concepts, for example, substance pressure, into manageable tasks. While the game-based environment fostered logical reasoning and structured problem-solving with high student enthusiasm, the challenges identified in algorithm design highlight that students still require targeted scaffolding to master complex procedural logic. In addition, this research has shown that integrating CT into domain-specific science topics does not isolate the skill, but rather deepens scientific inquiry through iterative evaluation and reflection-on-action. In the final analysis, these findings encourage teachers to be more proactive in integrating computational thinking into their teaching of science subjects, creating a learning environment that is both cognitively rigorous and emotionally engaging.

ACKNOWLEDGMENT

The University of Jember funded this research with a grant, number 3631/UN25.3.1/LT/2023.

REFERENCES

Acquah, E. O., & Katz, H. T. (2020). Digital game-based L2 learning outcomes for primary through high-school students: A systematic literature review. *Computers & Education*, *143*, 103667. https://doi.org/10.1016/J.COMPEDU.2019.103667

Adanır, G. A., Delen, I., & Gulbahar, Y. (2024). Research trends in K-5 computational thinking education: A bibliometric analysis and ideas to move forward. *Education and Information Technologies*, *29*(3), 3589–3614. https://doi.org/10.1007/S10639-023-11974-4/METRICS

Adipat, S., Laksana, K., Busayanon, K., Ausawasowan, A., & Adipat, B. (2021). Engaging students in the learning process with game-based learning: The fundamental concepts. *International Journal of Technology in Education*, *4*(3), 542–552. https://doi.org/10.46328/ijte.169

Agbo, F. J., Oyelere, S. S., Suhonen, J., & Laine, T. H. (2021). Co-design of mini games for learning computational thinking in an online environment. *Education and Information Technologies*, *26*(5), 5815–5849. https://doi.org/10.1007/S10639-021-10515-1/FIGURES/9

Alfaro-Ponce, B., Patiño, A., & Sanabria-Z, J. (2023). Components of computational thinking in citizen science games and its contribution to reasoning for complexity through digital game-based learning: A framework proposal. *Cogent Education*, *10*(1). https://doi.org/10.1080/2331186X.2023.2191751/ASSET/6352DEBC-C662-4EF3-B0C4-1D1671F75C09/ASSETS/GRAPHIC/OAED_A_2191751_F0003_OC.JPG

Alfayez, A. A., & Lambert, J. (2019). Exploring Saudi computer science teachers' conceptual mastery level of computational thinking skills. *Computers in the Schools*, *36*(3), 143–166. https://doi.org/10.1080/07380569.2019.1639593

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*, 106185. https://doi.org/10.1016/J.CHB.2019.106185

Araujo, R. C., Araujo, R. C., Floyd, L., & Gadanidis, G. (2019). Teacher candidates' key understandings about computational thinking. *Journal of Computers in Mathematics and Science Teaching*, *38*(3), 205–229.

Ayman, R., Sharaf, N., Ahmed, G., & Abdennadher, S. (2018). Minicolon; Teaching kids computational thinking using an interactive serious game. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11243 LNCS*, 79–90. https://doi.org/10.1007/978-3-030-02762-9_9/COVER

Baxter, J. (2020). Content analysis. *International Encyclopedia of Human Geography, Second Edition*, *2*, 391–396. https://doi.org/10.1016/B978-0-08-102295-5.10805-4

Bayeck, R. Y. (2024). Understanding computational thinking in the gameplay of the African Songo board game. *British Journal of Educational Technology, 55*(1), 259–276. https://doi.org/10.1111/ bjet.13353

Boucinha, R. M., Barone, D. A. C., Reichert, J. T., Brackmann, C. P., & Schneider, A. M. (2019). Relationship between the learning of computational thinking and the development of reasoning. *International Journal of Advanced Engineering Research and Science*, *6*(6), 623–631. https://doi.org/10.22161/ijaers.6.6.71

Cai et al. (2025). Effect of collaboration in digital game-based learning: The roles of flow experience and intrinsic motivation. *Journal of Computer Assisted Learning*, *41*(4), e70065. https://doi.org/10.1111/jcal.70065

Chen, C. H., & Chung, H. Y. (2024). Fostering computational thinking and problem-solving in programming: Integrating concept maps into robot block-based programming. *Journal of Educational Computing Research*, *62*(1), 406–427. https://doi.org/10.1177/07356331231205052

Chen, S.-J., Shan, X., Liu, Z.-M., & Chen, C.-Q. (2025). Employing large language models to enhance K-12 students' programming debugging skills, computational thinking, and self-efficacy. *Educational Technology & Society*, *28*(2), 259–278. https://doi.org/10.30191/ETS

Creswell, J. W., & Creswell, J. D. (2023). *Research Design: Qualitative, quantitative and mixed methods approaches*. SAGE Publications. https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf

Ehsan, H., Rehmat, A. P., & Cardella, M. E. (2020). Computational thinking embedded in engineering design: Capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education,31*(3), 441–464. https://doi.org/10. 1007/S10798-020-09562-5

Giannakoulas, A., & Xinogalos, S. (2023). Studying the effects of educational games on cultivating computational thinking skills to primary school students: A systematic literature review. *Journal of Computers in Education*, 1–43. https://doi.org/10.1007/S40692-023-00300-Z/TABLES/3

Handayani, R. D., Lesmono, A. D., Prastowo, S. B., Supriadi, B., & Dewi, N. M. (2022). *Bringing computational thinking skills into physics classroom through project-based learning.* 8th International Conference on Education and Technology (ICET), 76–80. https://doi.org/10.1109/ ICET56879.2022.9990631

Handayani, R. D., Prastowo, S. B., Jamalludin, Triyanto, & Thohir, M. A. (2023). *Students' engagement in computational thinking skills using a game based-learning.* Proceedings - International Conference on Education and Technology, ICET, 148–151. https://doi.org/10.1109/ICET59790. 2023.10435201

Hooshyar et al. (2020). From gaming to computational thinking: An adaptive educational computer game-based learning approach. *Journal of Educational Computing Research*, *59*(3), 383–409. https://doi.org/10.1177/ 0735633120965919

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. https://doi.org/10.1016/J.COMPEDU.2018.07.004

Jain, N. (2021). Survey versus interviews: Comparing data collection tools for exploratory research. *Qualitative Report*, *26*(2), 541–554. https://doi.org/10.46743/2160-3715/2021.4492

Kayhan, O., Korkmaz, Ö., & Çakır, R. (2024). How do computational thinking and logical and math thinking skills predict programming self-efficacy? *Computers in the Schools*, *41*(1), 51–73. https://doi.org/10.1080/07380569.2023.2220696

Kim, Y. J., Valiente, J. A. R., Ifenthaler, D., Harpstead, E., & Rowe, E. (2022). Analytics for game-based learning. *Journal of Learning Analytics*, *9*(3), 8–10. https://doi.org/10.18608/jla.2022.7929

King, N., Horrocks, C., & Brooks, J. M. (2019). *Interviews in qualitative research*. SAGE.

Küçükaydın, M. A., & Çite, H. (2024). Computational thinking in primary school: Effects of student and school characteristics. *Education and Information Technologies*, *29*(5), 5631–5649. https://doi.org/10.1007/S10639-023-12052-5/METRICS

Lee, I., & Malyn-Smith, J. (2019). Computational thinking integration patterns along the framework defining computational thinking from a disciplinary perspective. *Journal of Science Education and Technology*, *29*(1), 9–18. https://doi.org/10.1007/S10956-019-09802-X

Li, X., & Li, R. (2025). How game-based learning supports the creativity of university students? The context of China. *Innovations in Education and Teaching International*, *62*(3), 823–837. https://doi.org/10.1080/14703297.2024.2351528

Liu, T. (2024). Assessing implicit computational thinking in game-based learning: A logical puzzle game study. *British Journal of Educational Technology*, *55*(5), 2357–2382. https://doi.org/10.1111/bjet.13443

Lu, Z., Chiu, M. M., Cui, Y., Mao, W., & Lei, H. (2022). Effects of game-based learning on students' computational thinking: A meta-analysis. *Journal of Educational Computing Research, 61*(1), 235–256. https://doi.org/10.1177/07356331221100740

Lu, Z., Chiu, M. M., Cui, Y., Mao, W., & Lei, H. (2023). Effects of game-based learning on students' computational thinking: A meta-analysis. *Journal of Educational Computing Research*, *61*(1), 235–256. https://doi.org/10.1177/07356331221100740

Mao, W., Cui, Y., Chiu, M. M., & Lei, H. (2022). Effects of game-based learning on students' critical thinking: A meta-analysis. *Journal of Educational Computing Research*, *59*(8), 1682–1708. https://doi.org/10.1177/07356331211007098

Naeem, M., Ozuem, W., Howell, K., & Ranfagni, S. (2023). A step-by-step process of thematic analysis to develop a conceptual model in qualitative research. *International Journal of Qualitative Methods*, *22*, 1–18. https://doi.org/10.1177/16094069231205789

Pan, Y., Adams, E. L., Ketterlin-Geller, L. R., Larson, E. C., & Clark, C. (2024). Enhancing middle school students' computational thinking competency through game-based learning. *Educational Technology Research and Development*, *72*(6), 3391–3419. https://doi.org/10.1007/s11423-024-10400-x

Ranieri, M., Raffaghelli, J. E., & Bruni, I. (2018). Game-based student response system: Revisiting its potentials and criticalities in large-size classes. *Active Learning in Higher Education*, *22*(2), 129–142. https://doi.org/10.1177/1469787418812667

Reger, R. K., & Kincaid, P. A. (2021). Content and text analysis methods for organizational research. *In Oxford Research Encyclopedia of Business and Management*. Oxford University Press. https://doi.org/10.1093/acrefore/9780190224851.013.336

Şen, Ş. (2023). Relations between preservice teachers' self-efficacy, computational thinking skills and metacognitive self-regulation. *European Journal of Psychology of Education*, *38*(3), 1251–1269. https://doi.org/10.1007/s10212-022-00651-8

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Stohlmann, M. S. (2023). Categorization of game-based learning in middle school mathematics - A review. *Middle School Journal*, *54*(5), 16–26. https://doi.org/10.1080/00940771.2023.2254175

Sun, C., Yang, S., & Becker, B. (2024). Debugging in computational thinking: A Meta-analysis on the effects of interventions on debugging skills. *Journal of Educational Computing Research*, *62*(4), 1087–1121. https://doi.org/10.1177/07356331241227793

Sung, W., & Black, J. B. (2020). Factors to consider when designing effective learning: Infusing computational thinking in mathematics to support thinking-doing. *Journal of Research on Technology in Education*, *53*(4), 404–426. https://doi.org/10.1080/15391523.2020.1784066

Tarusha, F., & Bushi, J. (2024). The role of classroom observation, its impact on improving teacher's teaching practices. *European Journal of Theoretical and Applied Sciences*, *2*(2), 718–723. https://doi.org/10.59324/EJTAS.2024.2(2).63

Tavares, N. (2022). The use and impact of game-based learning on the learning experience and knowledge retention of nursing undergraduate students: A systematic literature review. *Nurse Education Today*, *117*(October), 105484. https://doi.org/10.1016/j.nedt.2022.105484

Turchi, T., Fogli, D., & Malizia, A. (2019). Fostering computational thinking through collaborative game-based learning. *Multimedia Tools and Applications*, *78*(10), 13649–13673. https://doi.org/10.1007/S11042-019-7229-9

Veenman, K., Tolboom, J. L. J., & van Beekum, O. (2022). The relation between computational thinking and logical thinking in the context of robotics education. *Frontiers in Education*, *7*, 956901. https://doi.org/10.3389/FEDUC.2022.956901/BIBTEX

Wang, B., Wang, Q., Wei, Y. M., & Li, Z. P. (2018). Role of renewable energy in China's energy security and climate change mitigation: An index decomposition analysis. *Renewable and Sustainable Energy Reviews*, *90*, 187–194. https://doi.org/10.1016/j.rser.2018.03.012

Wang, M., & Zheng, X. (2020). Using game-based learning to support learning science : A study with middle school students. *The Asia-Pacific Education Researcher*. https://doi.org/10.1007/s40299-020-00523-z

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2020). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education and Development*, *32*(1), 26–48. https://doi.org/10.1080/10409289.2020.1759012

Wang, X., Cheng, M., & Li, X. (2023). Teaching and learning computational thinking through game-based learning: A systematic review. *Journal of Educational Computing Research*, *61*(7), 1505–1536. https://doi.org/10.1177/07356331231180951

Wanglang, C., Sraubon, K., & Piriyasurawong, P. (2024). Combining game-based learning with design thinking using block-based programming to enhance computational thinking and creative game for primary students. *Higher Education Studies*, *14*(2), 137. https://doi.org/10.5539/hes.v14n2p137

Weintrop et al. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/S10956-015-9581-5

Weintrop, D., Holbert, N., Horn, M. S., & Wilensky, U. (2016). Computational thinking in constructionist video games. *International Journal of Game-Based Learning*, *6*(1), 1–17. https://doi.org/10.4018/ijgbl.2016010101

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wu, S. Y. (2018). *The development and challenges of computational thinking board games.* Proceedings - 2018 1st International Cognitive Cities Conference, IC3 2018, 129–131. https://doi.org/10.1109/IC3.2018.00-45

Wu, W.-R., & Yang, K.-L. (2022). The relationships between computational and mathematical thinking: A review study on tasks. *Cogent Education*, *9*(1), 1–19. https://doi.org/10.1080/2331186X. 2022.2098929

Xing, D., & Zeng, Y. (2024). Exploring the effects of secondary school student's information and communication technology literacy on computational thinking skills in the smart classroom environment. *Education and Information Technologies*, *30*(7), 9069–9092. https://doi.org/10.1007/S10639-024-13179-9

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st Century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/S11528-016-0087-7

Yaman, H., Bekar, Ş. N., Yildiz, H., Koyun, S., & Nas, S. E. (2023). Secondary school students' cognitive structures regarding educational games. *Science Education International*, *34*(1), 15–24. https://doi.org/10.33828/sei.v34.i1.2

Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, *141*, 103633. https://doi.org/10.1016/j.compedu.2019.103633