Humanities & Social Sciences Communications

REVIEW ARTICLE

https://doi.org/10.1057/s41599-025-04502-x OPEN

Check for updates

Integrating computational thinking, game design, and design thinking: a scoping review on trends, applications, and implications for education

Chih-Hung Wu[1], Yu-Cheng Chien[2], Mei-Tzu Chou[1] & Yueh-Min Huang[3]✉

This study integrated the fields of computational thinking (CT), game design, and design thinking through a scoping review, exploring their research development trends, as well as their applications and implications for education. A scoping review of game-design-based learning in CT literature was conducted, with a focus on technology-enhanced learning and high-impact factor journals published in the ScienceDirect (SDOS) and ISI Web of Science (ISI) databases. The study analyzed various factors, including different levels of education, development tools, game development task difficulty, length of learning tasks, cognitive and emotional measurement constructs in game design, research trends, and the status of CT skills. This study combined the concept of design-thinking process, integrating CT and game-based learning topics with design thinking, to propose a human-centered, innovative educational design process for CT and game-based learning. Future studies could further explore the effects of emotional aspects and differences in cognitive levels, as well as the emotional effects of various subject concepts or prior knowledge/experience, gender, national, or cultural differences. This study offers a comprehensive understanding of the trends and dynamics in the integration of CT, game design, and design thinking to facilitate continued research and advancements in the field of game-based learning for CT education.

Introduction

Computational thinking (CT) is a fundamental 21st-century skill (Voogt et al., 2015). Furthermore, it has become increasingly common in K–12 education (Chevalier et al., 2020; Hsu et al., 2018; Lee et al., 2020; Lye and Koh 2014). Programming can effectively cultivate CT and help K–12 students enhance their problem-solving abilities (Lye and Koh 2014). Previous scholars (Chevalier et al., 2020) also pointed out that CT can serve as a problem-solving model and even be extended to younger or older students. Additionally, CT is not limited to the field of computer science but can also be applied to K–12 STEM disciplines (science, technology, engineering, and mathematics) (Lee et al., 2020). Therefore, CT has increasingly attracted attention in K–12 education to address different disciplines and problems.

[1] Department of Digital Content and Technology, National Taichung University of Education, Taichung, Taiwan. [2] Department of Information Management, National Chin-Yi University of Technology, Taichung, Taiwan. [3] Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan. ✉email: huang@mail.ncku.edu.tw

Researchers have classified CT teaching activities into unplugged and plugged course activities (del Olmo-Muñoz et al., 2020; Saxena et al., 2020; Tsarava et al., 2017). Unplugged courses use physical props such as blocks, gears, board games, and in-class worksheets to teach students the basic concepts of CT. Plugged CT courses mostly involve digital games such as Scratch (Resnick et al., 2009) and Code.org (Kalelioğlu, 2015), where students learn programming by dragging and assembling code blocks to play games, making learning simpler and easier to understand. Through training with a graphical programming interface, students apply the learned concepts to solve problems, thereby improving their computational thinking skills (Barradas et al., 2020; Garneli and Chorianopoulos 2017; Pila et al., 2019; Rodríguez-Martínez et al., 2020; Wong and Jiang 2018). Generally, CT learning activities, courses, and games are combined to cultivate CT. Examples include combining English classes with board games (Hsu and Liang 2021) and math classes with Scratch (Rodríguez-Martínez et al., 2020). The essence of educational games is to allow learners to acquire knowledge. Many studies have also confirmed that cultivating CT through games not only improves elementary students' learning motivation, willingness, and efficiency but also makes learning enjoyable because of the fun nature of games (Hooshyar et al., 2020; Sharma et al., 2019; Turchi et al., 2019).

CT is a creative process (Wing, 2019), and its relationship with design thinking (DT) and critical thinking is important, as it is a critical dual process for solving design problems (Boom et al., 2022; Kelly and Gero 2021; Li et al., 2023b; Taimur et al., 2022; Wang, 2024). CT education utilizes programming to transfer learners' target tasks and associations to other fields, and learners can acquire skills and knowledge by creating their own programs and games. Therefore, CT education can create a learning environment by actively using design-based learning (DBL) rather than passive teaching (Jun et al., 2017). Design thinking is a creative way to solve problems, providing structured and systematic methods for problem-solving. This dual strategy of computational and design thinking requires learners to design and create games themselves, rather than learning from existing learning environments, which can more effectively stimulate their creativity and encourage their interest. Participation in game design involves playing games, as well as modifying or attempting to improve and change them (Melander Bowden, 2019). Students have the opportunity to receive formal and spontaneous critique, which can provide feedback for the creator and an understanding of future design work (Tucker-Raymond et al., 2019). Simultaneously, effective improvements in self-efficacy, learning attitudes, and participation can be achieved (Tsai et al., 2021). Game design thinking as a learning process can be used to encourage learners to engage deeply in learning activities (Choi and Noh 2021; Kelly and Gero 2021; Zhang et al., 2022).

CT games include problem-solving, building algorithms, debugging, simulation, and socializing (Boom et al., 2022; Kazimoglu et al., 2012; Kite et al., 2021). Therefore, the complexity of the game and the age range of the players should be considered when designing games. Several studies have conducted repeated testing on students of different ages (Leonard et al., 2018; Ostovar-Namaghi et al., 2023; Papavlasopoulou et al., 2019; Papavlasopoulou et al., 2020). In addition, the practical process of designing CT games involves not only cognitive learning but also emotions and engagement (Apiquian et al., 2020; Ostovar-Namaghi et al., 2023; Zainuddin, 2023). Spontaneous flow and emotional experiences are the basis of game complexity, and enjoyment and high engagement in the game experience are correlated with the game experience itself (Melander Bowden, 2019).

With the growing popularity of technology education, Computational Thinking (CT) has become an essential skill across all educational levels. Integrating CT into learning activities through game design enables students to engage with programming and problem-solving in a more interactive and enjoyable way. However, students at different educational stages face unique challenges in understanding and applying CT concepts. Therefore, it is crucial for educators to design suitable learning activities and develop effective evaluation methods. This study aims to explore how to design and assess CT learning activities based on game design, focusing on the following two research questions:

How can CT-based game design activities improve the learning outcomes of K-12 students? This research question seeks to analyze how CT teaching activities enhance students' learning outcomes, addressing the following sub-questions: (1) CT development tools for different educational levels: analyzing tools suitable for students from K-12 to higher education. (2) Educational levels and learning difficulties analysis: exploring challenges students encounter across different stages of learning. (3) Analysis of learning task duration: examining how the design of task duration impacts learning effectiveness. (4) Cognitive and affective analysis based on game design: investigating the emotional and cognitive development of students participating in game design activities. (5) Application of CT skills in game design: analyzing how students develop and apply CT skills within game design activities.

How can CT and design thinking (DT) be integrated to enhance students' problem-solving abilities? Beyond the analysis mentioned above, this study also aims to explore ways to integrate Design Thinking with CT and investigate how these two frameworks can complement each other to foster creative problem-solving in learning environments. By combining CT with DT, the research will further examine how to develop effective teaching strategies to enhance student performance across different disciplines. This question will prompt researchers to explore the best practices for integrating computational thinking and design thinking, as well as how to design a learning environment that encourages students to deeply engage in learning activities, how to evaluate the effectiveness of such a learning environment, and how to adjust it according to the needs of students in different age groups.

Computational thinking consists of four key components: decomposition, pattern recognition, abstraction, and algorithm design. CT skills enhance learners' logical reasoning (Wu and Richards 2011) and cognitive abilities (Wu and Richards 2011). CT emphasizes a logical and systematic approach to problem-solving, which, when applied to game design, helps designers develop programming logic and structure the flow of game levels (Wu and Richards 2011).

Design Thinking is a user-centered creative process that includes the iterative steps of empathizing, defining, ideating, prototyping, and testing. With DT, game designers not only address the technical aspects of game development but also create games that meet user needs and are engaging (Liu, 2024).

Previous research has primarily focused on the integration of CT with game design or DT with game design. However, few studies explore how CT, DT, and game design can be integrated. Game design heavily relies on problem-solving skills, and both CT and DT play a crucial role in enhancing these abilities. Therefore, this study aims to explore the relationship between CT, DT, and game design. Through analysis, the study proposes how these three elements can complement each other in teaching

Fig. 1 Research goal. Revised from (Freepik, 2023b).

and learning design, fostering learners' creativity, problem-solving skills, and hands-on abilities.

This study integrated the fields of CT, game design, and design thinking. Research development trends in these three fields were presented through a scoping review. Their applications and implications for education are also presented as shown in Fig. 1. Thus, we identified trends in game design activities for CT based on the integration and analysis of relevant literature. We also analyzed these trends using the design-thinking process, which includes empathizing, defining, ideating, prototyping, testing, and implementing. This study demonstrates how the main research focuses on age groups, complexity, difficulty, game characteristics, and emotional aspects have changed since 2014. Based on a review of relevant research and our experiences, this study provides several key points that should be emphasized in CT education for game design activities to facilitate further research and communication.

Method

To review the relevant literature, we conducted a scoping review. Scoping reviews are useful when it comes to examining interesting features and concepts, that is, examining the substantial reports on CT and game design, clarifying the key concepts or definitions, and reviewing studies, rather than determining the feasibility of game design in CT. The instructional intention of game design in CT is to ask learners to design and develop their own games. Learners must adhere to design-thinking principles to complete their game projects. To match the appropriate learning level, learning tasks must be tailored to learners' age at different task levels, and the development tools required for making games should be provided. Learners also need to clarify game requirements considering the game's purpose to practice and create their games. Emotional factors also play an important role in this process. Learners' active involvement and positive participation will affect their teamwork and learning-by-doing, and in repeated testing and critical thinking, peer and teacher opinions will refine the game further.

Therefore, this study reviewed game design-based learning in CT in educational and learning papers published in the *ScienceDirect* (SDOS), *ISI Web of Science* (ISI), *Elsevier's Scopus* (Scopus), *Web of Science* (WOS), and *Education Resources Information Center* (ERIC) databases, mainly focusing on technology-based learning and high-impact factor journals such as *ACM Transactions on Computing Education*, *Computers & Education*, *Computers in Human Behavior*, *Educational Technology Research and Development*, *Interactive Learning Environments*, and others within our search scope. Three researchers experienced in game design-based learning in CT used the keywords "Game & CT," "Game-Design & CT," and "Design Game & CT" to search for journal articles in the Science Citation Index (Demartini et al., 2024) and Social Science Citation Index (SSCI) of the SDOS and ISI databases. The initial search found a total of 240 articles. To refine our search, three researchers read the list of articles and selected only those that included experimental activities that used game design for CT learning and were written in English as the final review list. Publications such as book reviews, letters, and editorials, as well as those that only use "game" for CT learning and those written in languages other than English, were excluded. Finally, 33 journal articles related to game design-based learning in CT education were obtained, as shown in Table 1.

Table 1 Number of published papers in journals.

Journal Title	SCI/SSCI	Papers
ACM Transactions on Computing Education	SCI	3
Asia-Pacific Education Researcher	SSCI	1
Behaviour & Information Technology	SSCI	1
Cognitive Systems Research	SSCI	1
Computers & Education	SSCI	2
Computers in Human Behavior	SSCI	3
Education and Information Technologies	SSCI	8
Educational Technology Research and Development	SSCI	3
IEEE Access	SCI	1
Interactive Learning Environments	SSCI	2
International Journal of Stem Education	SSCI	1
International Journal of Science Education	SSCI	1
International Journal of Technology and Design Education	SSCI	1
Journal of Research on Technology in Education	SSCI	1
Journal of Science Education and Technology	SSCI	1
Journal of Teacher Education	SSCI	1
Learning Media and Technology	SSCI	2
Total		33

Based on the data-collection procedure, we first identified journal articles in the education/learning field from 2010 to 2024 and reviewed 33 studies on game design-based learning in CT. Table 2 provides an overview of these studies in terms of the year, country, subject, education level, and type of development tool used, as shown in Table 2.

Results

CT development tools for different educational levels. This section focuses on examining student education levels as a measure. After reviewing the 33 studies, we first examined the development tools used in CT education based on game designs. As shown in Table 3, the development tools can be divided into graphical user interface (GUI), text-based language (TBL), visual-based language (VBL), and virtual reality (VR). Most studies use VBL as the main teaching and development tool in game design, as shown in Table 3 and Fig. 2. The GUI and TBL were used with the ARIS platform and Python as programming tools, respectively. Studies that use VBL as a development tool can be further categorized into AgentSheets, Alice, MIT App Inventor, MS Kodu, Scratch, and the hybrid use of AgentSheets/AgentCubes and Scratch/ModKit, as shown in Table 4 and Fig. 3.

Analysis of different educational levels and learning difficulties. Therefore, this study analyzed the development tools and difficulty levels of learning tasks by our collected data and categorized them as simple or advanced, as shown in Table 5. As

Table 2 Background information on reviewed articles.

Primary author (year of publication)	Country	Subject	Educational level	Building tools
(Werner et al., 2014)	USA	Maker Innovators	K7-K12	Alice 2.2 or Storytelling Alice
(Repenning et al., 2015)	USA	Workshop of creating an artifact	K7-K12	AgentSheets
(Leonard et al., 2016)	USA	CT and Climate science	K1-K12	LEGO® EV3 robotics & AgentSheets/AgentCubes (Scalable Game Design software)
(Jun et al., 2017)	South Korea	Arts	K1-K6	Scratch
(Leonard et al., 2018)	USA	Science and Arts	Adult	LEGO® EV3/NXT and AgentSheets/AgentCubes
(Munoz et al., 2018)	Chile	Civic issue (e.g., wildfires, animal adoption, pollution)	K1-K12	Scratch
(Melander Bowden, 2019)	Sweden	STEM	K1-K6	Scratch
(Cheng, 2019)	China	Programming workshop	K1-K6	MIT App Inventor
(Papavlasopoulou et al., 2019)	Norway	Computer Science	K1-K12	Scratch and Arduino
(Richard and Giri 2019)	USA	Programming curriculum	K7-K12	Makey Makey, Scratch, Lilypad Arduino and ModKit
(Tucker-Raymond et al., 2019)	USA	Informatics	K7-K12	Scratch
(Melander Bowden and Aarsand 2020)	Swedish	Computer Science	K1-K6	Scratch
(Hava et al., 2020)	Turkey	STEM+Arts	K1-K12	MS Kodu
(Litts et al., 2020)	USA	STEM	K1-K12	ARIS platform
(Newton et al., 2020)	USA	Climate	K1-K6	LEGO® EV3, AgentSheets
(Papavlasopoulou et al., 2020)	Norway	Workshop of creating an artifact	K1-K12	Arduino and Scratch
(Saritepeci, 2020)	Turkey	Maker Innovators	K7-K12	Scratch (and storyboardthat.com, Voki, audacity, Powtoon, Animaker, Adobe Spark, Moovly, or Wevideo… etc.)
(Wong and Cheung 2018)	Hong Kong	CT and Climate science	K1-K6	MS Kodu
(Seralidou and Douligeris 2021)	Greek	Arts	K7-K12	Scratch
(Agbo et al., 2021)	Nigeria	Science and Arts	Adult	Buliding Tools: VR game-based smart learning environment (SLE); OnlineTools: Zoom, WhatsApp
(Çakır et al., 2021)	Turkey	Civic issues (e.g., wildfires, animal adoption, pollution)	K1-K12	Python
(Tsai et al., 2021)	Taiwan	STEM	Adult	Scratch & Arduino
(Tucker-Raymond et al., 2021)	USA	Programming workshop	Adult	Scratch
(Agbo et al., 2021)	Nigeria	CT skill + OCD	Adult	SLEs for VR mini games
(Gentile and Lieto 2022)	Italy	Mathematical + STEM skills	Adult	Unity3D
(Othman et al., 2023)	Malaysia	CT	K4-K6	Scratch
(Agbo et al., 2023)	Nigeria/ Finland	Virtual Reality (VR), CT	K7-K9	Unity3D, iThinkSmart
(Jiang et al., 2023)	China	Game Design, CT	K7-K9	Computational Puzzle Design (CPD) Framework
(Tikva and Tambouris 2023)	Greece	Programming, CT	K7-K9	aMazeD
(Zhang et al., 2023)	Hong Kong/ China	Coding, CT	K4-K6	Coding Galaxy
(Fanchamps et al., 2024)	Netherlands	SmartGames, CT	K2-K6	LEGO® EV3, Bomberbot
(Cheng et al., 2023)	Taiwan	STEM	K4	Scratch
(Yeoh et al., 2024)	Taiwan	Scientific Inquiry, Collaborative Learning	Adult	TrainED

Table 3 Building tools in the reviewed articles.

	K1-K6	K7-K12	K1-K12	Adult	Total
GUI	1	1	1		3
TBL			1		1
VBL	9	7	5	4	25
VR		2		2	4
Total	10	10	7	6	33

shown in Fig. 4, the game design tasks were designed to be more complex as the learners' educational level increases, and conversely, simpler as the educational level decreases.

Advanced game design tasks can be further divided into three categories: robotics, teaching strategies, and complex tasks, as listed in Table 6. We found seven studies that used robotics as a game task, where students used Arduino or LEGO® EV3 to design games. The teaching strategy category included four studies that incorporated creativity, critique, and interactive pages as teaching strategies for learning tasks. The sophisticated game design category included four studies that used complex calculations to implement game designs or games designed for specific purposes.

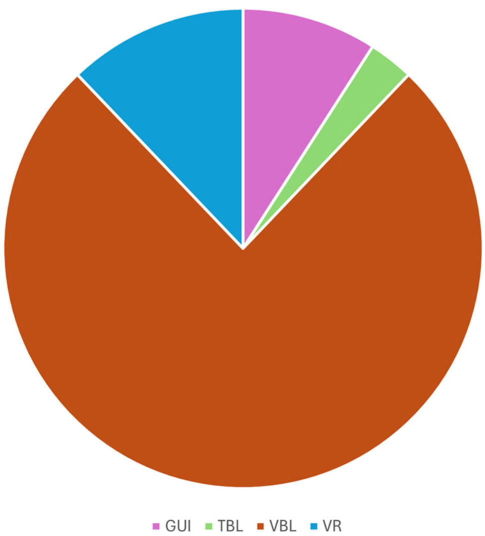

Fig. 2 Building tools.

Analysis of learning task duration. In terms of the length of the game design tasks, this study classified the task duration into three categories: one-time experimental activities with a duration of less than a month, middle- to long-term experimental activities with a duration between 1 month and 1 semester, and long-term experimental activities with a duration exceeding 1 semester. Tables 7 and 8 show that as game tasks in CT become more difficult, more time is required because an adequate game length is necessary to design complex game tasks. As shown in Fig. 5, advanced game design tasks require more time than simpler tasks.

Table 4 VBL in the reviewed articles.

	K1-K6	K7-K12	K1-K12	Adult	Total
AgentSheets	1	1			2
AgentSheets/AgentCubes			1	1	2
Alice		1			1
MIT App Inventor	1				1
MS Kodu	1		1		2
Scratch	3	3	3	2	11
Scratch/ModKit		1			1
Total	6	6	5	3	20

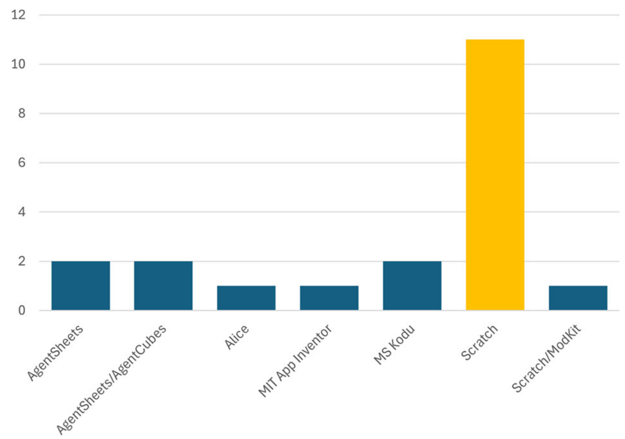

Fig. 3 VBL tools.

CT in cognitive and affective analysis based on game design. To examine the degree to which cognitive and affective aspects are emphasized in CT-based game designs, we examined the dimensions measured and analyzed in our collected data, which can be divided into cognitive, affective, and combined cognitive and affective analyses. The results are presented in Tables 9 and 10, respectively. The cognitive aspect was considered important, regardless of the educational level or task difficulty, whereas the current research trend considers both cognitive and affective aspects as important constructs, with less emphasis on effectiveness alone. Overall, both the effective and cognitive dimensions are more important at an advanced level.

As mentioned, the affective dimension has gradually received more attention in CT education. Therefore, this study investigated further the emotional constructs used in our collected data, and the results are shown in Table 11 and Fig. 6. The analysis shows that most studies consider attitude as an important indicator of the CT impact on game design. Therefore, nine studies measured learners' attitude dimensions. Self-efficacy is the next most important dimension.

Analysis of CT skills in game design. In terms of the CT skill evaluation indicators, this study referred to the CT game characteristics defined by Kazimoglu et al. (2012), which include problem-solving, building algorithms, debugging, simulation, and socializing. Based on the above content, this study separately counted and scored the game characteristics according to our collected data. Each characteristic fulfilled was given 1 point. If the score exceeded 3, it was considered CT-high, and vice versa, it was considered as CT-low. As shown in Table 12, the higher the age group of the participants, the more game characteristics that need to be met in the game design, and the higher the score of CT skills. Conversely, the lower the age group, the lower the CT skill score. In addition, we found that when game tasks became more

Table 5 The education level and task degree of game design-based learning.

	K1-K6	K7-K12	K1-K12	Adult	Total
Simple	7	3	3	2	15
Advance	3	6	4	5	18
Total	10	9	7	7	33

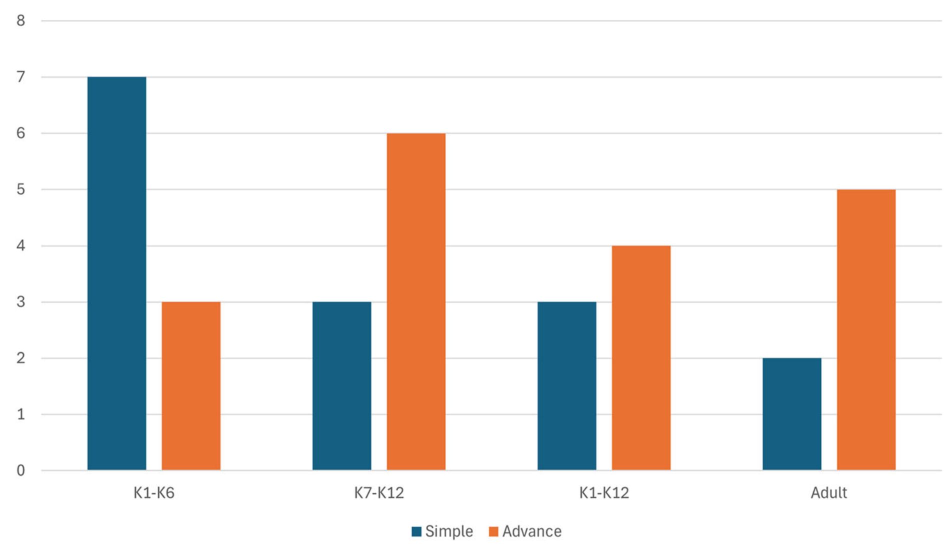

Fig. 4 Education level and task degree of game design-based learning.

Table 6 The education level and advanced task of game design-based learning.

Category	Detail	K1-K6	K7-K12	K1-K12	Adult	Total
Robotics (7)	Arduino		1	2	1	4
	LEGO® EV3	3		1	1	5
Teaching Strategy (4)	Creativity	1		1		2
	Critique		1			1
	Interactive page		1			1
	Multidimensional scaffolding				1	1
Sophisticated Game Design (4)	Computationally sophisticated: Patterns and Mechanics		1			1
	Science Technology Engineering & Mathematics simulations		1			1
	Serious Games				2	2
	Skyscrapers			1		1
	VR-iThinkSmart		1			1
Total		2	6	5	4	20

Table 7 Analysis of game design task length among various building tools.

	Once	Middle	Long	Total
GUI	4			4
TBL	1			1
VBL	9	12	4	25
VR	2	1		3
Total	16	13	4	33

Table 9 Analysis of cognitive and affective domain measurement among various education levels.

	Cognitive	Affective	Affective & Cognitive	Total
K1-K6	5	1	4	10
K7-K12	4		5	9
K1-K12	3		4	7
Adult	3		4	7
Total	15	1	17	33

Table 8 Analysis of game design task length (simple vs. advanced).

	Once	Middle	Long	Total
Simple	7	8	0	15
Advance	9	5	4	18
Total	16	13	4	33

Table 10 Analysis of cognitive and affective domain measurement among simple and advanced tasks.

	Cognitive	Affective	Affective & Cognitive	Total
Simple	9	1	5	15
Advance	6		12	18
Total	15	1	17	33

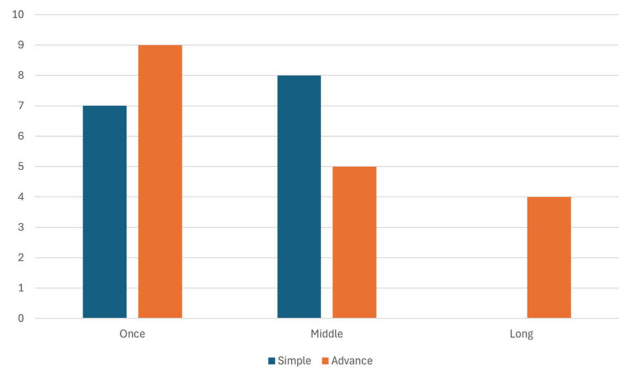

Fig. 5 Analysis of game design task length.

Table 11 Analysis of measurement variables in the affective domain.

Affective	Number
Attitude	11
Self-efficacy	6
Engagement	3
Self-interest	3
Social	2
Cognitive effort	2
Outcome expectancy	1
External assistance & external encouragement	1
Workshop experience	1

difficult, the CT skill score for game design increased. If the task was relatively simple, the CT skill score would be lower, as shown in Table 13.

In contrast, from the analysis of the cognitive and affective aspects of game design, this study found that most studies still tend to assess learners' affective and cognitive levels simultaneously, despite the number of CT game characteristics. However, when game design is more focused on the cognitive level, fewer game characteristics are used in learning, and their CT skill scores are relatively lower, as shown in Table 14.

Discussion

Integration of CT, DT, and game design. Design thinking comprises of three main stages and six steps. The first stage (Understand) included 1. Empathize and 2. Define. The second stage (Exploration) consists of 3. Ideate and 4. Prototype. The third stage (Materialize) encompasses 5. Test and 6. Implement. The Integration of Integration of CT and design thinking in Education Game design is illustrated in Fig. 7.

Design thinking is a human-centered problem-solving methodology and has been proven high correlation with DT and CT (Li et al., 2023a). Integration DT of practice and application could enhance learners' interdisciplinary collaboration and co-creation

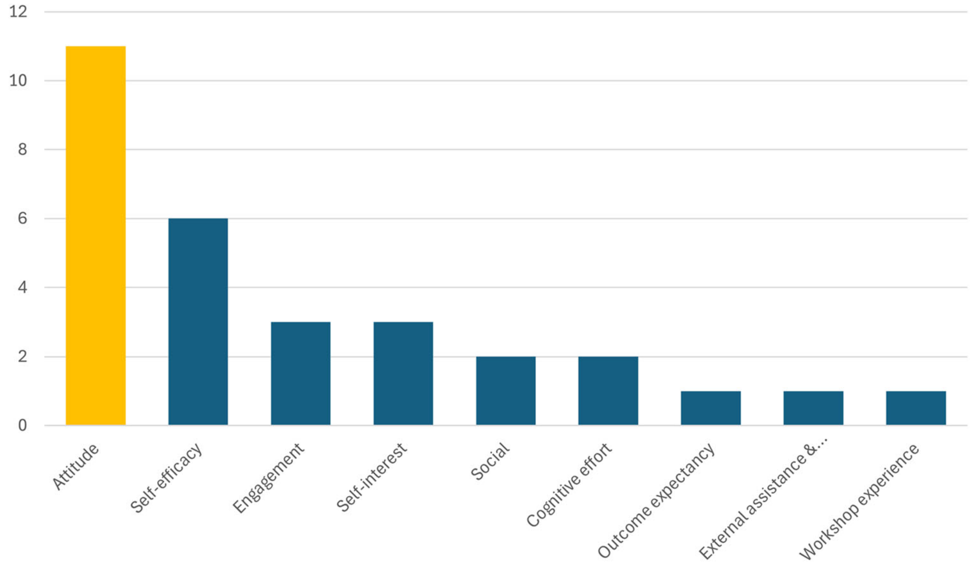

Fig. 6 Analysis of measurement variables in the affective domain.

Table 12 Analysis of CT skills among various education levels.

	CT-Low	CT-High	Total
K1-K6	6	4	10
K7-K12	5	4	11
K1-K12	5	2	5
Adult	3	4	7
Total	19	14	33

Table 13 Analysis of CT skill between simple and advanced tasks.

	CT-Low	CT-High	Total
Simple	13	2	15
Advance	6	12	18

(Constantinou and Rybska 2024; Wang, 2024). By focusing on human needs, it seeks innovative solutions to various issues and creates more possibilities. Therefore, the research results combine the concepts of the design-thinking process to integrate CT and game-based learning topics with design thinking, proposing a human-centered innovative educational design process for CT and game-based learning. The explanation for this is as follows in Table 15.

CT assessment framework. Wu and Huang (2020) proposed the CT assessment framework-LUPDA theory (learn, use, practice, design, apply, analyze) as the set of relevant assessment principles for STEM and CT concepts in game design (Wu and Huang 2020). The study provided an illustration of STEAM teaching activity that develops an artificial intelligent (AI) webcam game about detection of hand gestures (scissors, stones, or cloth). The game integrates deep learning techniques to develop a rock-paper-scissors game that can be executed through Micro:bit To reduce the development difficulty for learners, the study uses Micro:bit's block-based graphic programming language, combined with built-in deep learning modules, allowing learners to get started quickly. Another LUPDA study adopted and reviewed

Table 14 Analysis of CT skills among cognitive and affective domains.

	CT-Low	CT-High	Total
Cognitive	9	6	15
Affective	1		1
Affective & Cognitive	9	8	17
Total	19	14	33

LUPDA theory through the STEAM learning activities of the Micro:bit-Bostacle Avoidence Car (Chien et al., 2020). These results align with the analysis results of this study. Due to the complexity of game development, the study proposes several educational recommendations:

(1) Select tools based on learner ability and task difficulty.

For different learner levels and task complexities, appropriate game development tools should be chosen. For beginners, block-based graphical tools like Scratch or LOGO are more suitable. For advanced tasks, text-based programming languages like Python or C# can be used. Research results indicate that simpler tasks often use graphical tools, while advanced tasks are more suited to text-based tools, handling more complex features and functionalities in educational game development (Cafarella and Vasconcelos 2024).

(2) Integrating CT and DT in game development

Based on different learning tasks and the learner's prior knowledge, selecting the appropriate development tools and task duration (long or short), and combining the design concepts of CT and DT can effectively help students successfully complete game development with educational significance (Cafarella and Vasconcelos 2024). This approach not only enhances learners' sense of achievement and satisfaction but also improves their motivation and engagement, which in turn enhances learning outcomes (Tran, 2019).

(3) Using the LUPDA framework for CT assessment

For the assessment of CT in different learning tasks, the LUPDA framework can be used to evaluate learners' familiarity at different stages of learning. LUPDA provides an assessment structure that covers each step of CT and different aspects of STEAM, evaluating learners across six stages: learn, use, practice, design, apply, and analyze. LUPDA proposes a detailed

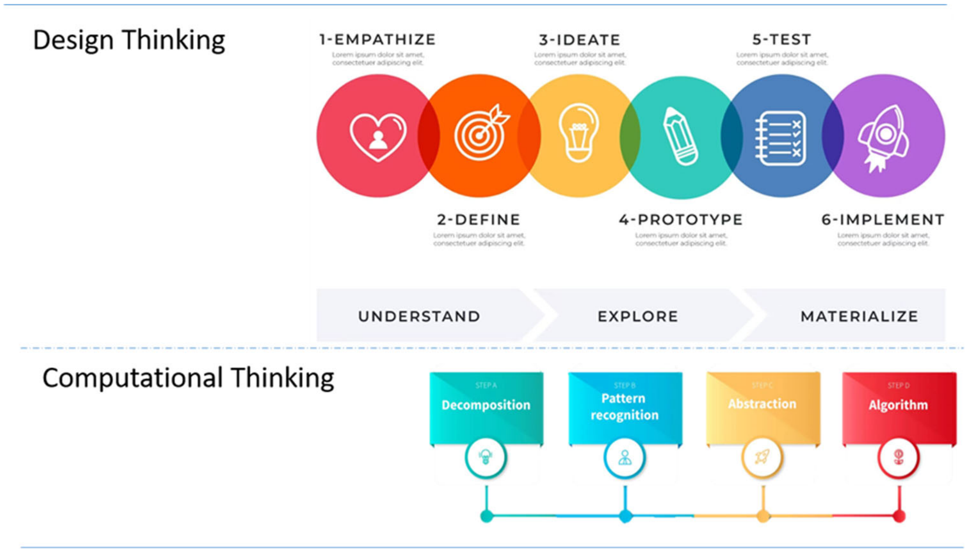

Fig. 7 Integration of CT and design thinking in Education Game design. Revised from (Freepik, 2023a).

Table 15 Integration of CT and design thinking in Education Game design.

Design Thinking	Implications and applications
Empathize	The first step in design thinking is to establish empathy. By approaching from the user's perspective, the true issues that users experience can be identified. Therefore, based on the conclusions in Tables 5 and 7, this study suggests that during teaching and game development, the learners' educational levels and prior knowledge should be considered. The choice of development tools and game task design should be divided into simple and advanced levels, providing different degrees of difficulty and learning task durations for learners of different grades or varying knowledge levels in teaching tasks and development tools.
Define	The second step in design thinking is defining needs (Define). Through empathy planning from the previous step, an appropriate needs definition is established. Based on the conclusions of Table 6, this study suggests that educators should plan suitable game design goals according to the appropriate need definition and plan suitable tools, hardware, and software choices, as well as select the best teaching strategies, such as adopting interactive or sophisticated game design approaches.
Ideate	The third step in design thinking is creative ideation (Ideate). The results of the needs definition from the previous step are used to generate numerous possible solutions through creative ideation. This study suggests integrating the steps of CT through Decomposition, Pattern recognition, Abstraction, and Algorithm to analyze problems and ultimately propose the best possible solution. At this stage, problems can be decomposed through CT to break down complex needs definition issues into smaller, more understandable, and more manageable subproblems. Then, through Pattern recognition, similarities between problems can be identified, assisting in the contemplation of the best possible solutions.
Prototype	The fourth step in thinking is Prototype. This step involves creating a prototype to establish a tangible outcome. At this stage, Abstraction and Algorithm from CT can be applied to consider which development tools to choose for building the prototype. Based on the conclusions of Tables 3 and 4, learners with different educational levels and prior knowledge could select different development tools. For example, younger learners might opt for GUI-based development tools (such as Kodu) or block-based development tools such as App Inventor and Scratch. More experienced learners in higher grades can choose traditional text-based development tools, such as Python or C#.
Test	The fifth step in design thinking is Test & Implication. Testing involves using the prototype from the previous stage to interact with learners, allowing them to test whether their output is executed correctly and to observe their learning effectiveness and difficulties. At this stage, Abstraction and Algorithm from CT can be applied, enabling learners to review their abstraction and algorithm for feedback and revision, thereby reinforcing their learning. Simultaneously, teachers can use the learners' game development outcomes and user reactions during this stage to assess whether the specified games can be developed correctly and quickly, to redefine and revise requirements or improve teaching strategies and gain deeper insights into the learning status.After reviewing the literature in this study and based on the analysis of Tables 9 and 10, understanding the learning status should consider not only the cognitive but also the affective domain. In the field of game design, measuring the affective domain is becoming an increasingly important topic. Currently, few studies consider both cognitive and affective domains simultaneously. Future research should evaluate both domains together.
Implication	In the implementation stage of game design, according to the analysis in Tables 12, 13, and 14 regarding CT skills, the level of CT skills is closely correlated with the educational level and the simplicity or difficulty of game design tasks. The higher the age group and the more complex the learning tasks, the greater the number of required CT skills. Conversely, the lower the age group and the simpler the learning tasks, the fewer CT skills are needed.

assessment framework and indicators, which are helpful for researchers and educators in assessing learners' familiarity with CT and their progress at various learning stages, assigning corresponding scores. This framework offers concrete indicators and scoring mechanisms (Chien et al., 2020) from 1-3 points (1 point: minimal, 2 points: solid evidence, and 3 points: exceeds expectations) for teachers and researchers to evaluate learners' performance at different stages of CT (Chien et al., 2020; Wu and Huang 2020). The Guidelines for educators in LUPDA are shown in Fig. 8, and detailed indicators are provided in Fig. 9.

STEAM	Main purposes	Guidelines for educators
Science	Learn	The STEAM activities and courses in science part could focus on teaching students how to study and understand science content.
Technology	Use	The STEAM activities and courses in technology part could focus on training students how to properly use technology, tools, and coding.
Engineering	Practice	The STEAM activities and courses in engineering part could focus on training students how to properly use tools to assemble various materials in their work. Educators can design several activities to help students practice how to use these tools.
Art	Design	The STEAM activities and courses in art part could focus on training students how to design beautiful unique works based on art theories. The teacher uses the designed activity to help students enhance their imagination and creativity.
Math	Apply /Analyze	The STEAM activities and courses in math part could focus on training students how to use math to apply or analyze data and to discover trends in data.

Fig. 8 Guideline for educators of LUPDA theory with STEAM game activity (Wu and Huang 2020).

STEAM	LUPDA theory	Computational thinking	Score	Assessment principle
Science	Learn	Decomposition	+1	Being able to divide the science problem into several sub problems.
			+2	Being able to understand every useful science concepts or theories or algorithms for sub problems.
			+3	Being able to choose/use science correctly according to specifications in each sub problems.
		Pattern recognition	+1	The ability to figure out the pattern for each sub problems.
			+2	The ability to select the most appropriate science concept in each patterns.
			+3	Using science correctly in each patterns.
		Abstraction	+1	The ability to design/plan several science sub programs to solve the problem.
			+2	Can choose the most appropriate science concepts/theories/algorithms and apply it in different sub problem and patterns.
			+3	Knowing how to build a good system or model (e.g. AI model).
		Algorithm	+1	Being able to design the flowchart that includes system/model/experimental hypotheses and then verify them.
			+2	The ability to collect data and analyze the experimental/simulation results.
			+3	Making generalizations and reasonable inferences based on the results of analysis
Teaching activity: Building and training AI models				

Fig. 9 Framework of assessment principles for the STEAM science dimension (An illustration of science dimension) (Wu and Huang 2020).

Future researchers can use this framework to evaluate learning outcomes or the attitude and assessment questionnaire of CT in computer science classes that proposed by (Cafarella and Vasconcelos 2024). They can also adjust the evaluation criteria at each stage according to different learning tasks and the learners' backgrounds, thereby developing more suitable assessment tools for learning.

Conclusion

This study reviewed research trends of game design in CT learning. The results showed a significant increase in the number of papers published recently, with major contributions from researchers in the United States. Empirical studies on game design in CT learning have mainly focused on K–12 education as the primary research sample, which explains the importance of CT in K–12 education and the mainstream trend in targeting K–12 students. In terms of development tools, we found that visual programming based on Scratch is the main learning and development tool used in CT education, and most studies have used Scratch as a game development tool. Furthermore, some interdisciplinary strategies, such as educational robotics, creativity, and critical thinking, have been used in recent game design studies on CT. Based on these results, we believe that the current research trends have directionality and will provide benefits for researchers and educators in their future work.

Although various methods for assessing CT have been addressed, recent research trends, including game design in CT learning, have moved beyond simple cognitive evaluation and have begun to incorporate affective considerations. For example, self-efficacy and attitude are the most used measures for assessing learners' engagement in game-making activities, and they combine cognitive and affective factors to comprehensively evaluate learning and understand learners' progress. In terms of game design analysis, we confirmed that most studies tend to concurrently evaluate learners' effective and cognitive levels. Therefore, future research could further investigate the impact of affective aspects and differences in cognitive levels and explore the affective effects of various disciplinary concepts or prior knowledge/experience, sex, and national or cultural differences.

This study examines the trends and current state of research in game design, focusing on different groups' learning stages, learning tasks, and the tools used for development through a scoping analysis. Research results explore the impact of game design with CT and DT on education and learning. The findings provide valuable insights for future research, particularly in the areas of equity, accessibility, and scalability.

The integration of CT, DT, and game design can be supported by constructionism theory (Harel and Papert 1991). Constructionism theory suggests that learners construct knowledge most effectively when they actively engage in creating meaningful products (Harel and Papert 1991; Kynigos, 2015). Through participation in game design, students use CT to develop and program game mechanics, while applying DT to enhance the user experience, thereby constructing knowledge through iterative design and computational tasks (Liu, 2024).

The relationship between CT, DT, and game design demonstrates how these three elements complement each other in teaching and learning contexts, fostering learners' creativity, problem-solving abilities, and hands-on skills. In integrated applications within game design, CT provides the logical and structural framework, while DT introduces creativity and improvements in user experience. Constructionism emphasizes that learners can develop new knowledge and deepen their conceptual understanding through hands-on activities and direct experience, such as game design. Research shows that combining CT and DT in game design not only enhances learners' creativity but also strengthens their logical thinking and problem-solving abilities (Cafarella and Vasconcelos 2024).

In addition, combining CT, DT, and game design can address significant educational issues such as equity, accessibility, and scalability in these directions.

(1) Equity

Incorporating CT and DT into game design courses helps create an interactive and easy-to-understand learning environment. For example, a previous study analyzed 200 elementary school students who participated in a 10-week coding program using engaging and accessible content with the Blocky programming language in code.org. This effectively enhanced the students' computational thinking skills. The study showed that applying CT to solve complex problems can attract students from different backgrounds and varying levels of prior knowledge. Using DT helps design suitable educational programs, allowing underrepresented groups to learn computational thinking in an enjoyable and accessible way, promoting educational equity (Tran, 2019).

(2) Accessibility

Combining game design and design thinking helps create a more inclusive learning environment. Design thinking emphasizes understanding user needs, which can assist teachers in designing personalized learning experiences that cater to diverse learning requirements. Integrating game design with CT and DT allows learning to be tailored to different needs, especially for learners with disabilities or those lacking access to traditional educational resources. For instance, using block-based programming languages in game development can lower the entry barriers for students unfamiliar with coding, making educational content more accessible to all learners. This is particularly helpful for students who might struggle or feel frustrated in traditional text-based programming courses (Cafarella and Vasconcelos 2024). The algorithm design component of computational thinking helps students grasp complex programming concepts more easily. By utilizing graphical game design tools like Scratch, students can overcome the technical challenges of coding, enhancing participation for learners with various disabilities (Cafarella and Vasconcelos 2024).

(3) Scalability

Game design with CT and DT can provide scalable solutions suitable for various educational environments. Through CT with DT concept to design scalable game design models, educators can implement these concepts across different settings, from middle school to graduate programs. For example, previous studies using scalable game design have demonstrated success in teaching computer science, from K-12 to graduate levels, highlighting its adaptability and scalability (Basawapatna et al., 2010). Therefore, future researchers, whether conducting studies with students of different grade levels or other learning groups, can benefit from the findings of this research. The results not only provide substantial contributions to educational practice but also serve as a foundation for future related studies, helping to develop new theories or apply them in different contexts.

In summary, by integrating CT, game design, and DT, educational systems can create more equitable, accessible, and scalable learning environments, addressing broader educational challenges (Basawapatna et al., 2010; Cafarella and Vasconcelos 2024; Tran, 2019).

(4) Educational implications

Game design is a complex and innovative process that requires the integration of computational thinking (CT) and design thinking (DT) to effectively address various challenges. Combining CT and DT can significantly enhance problem-solving abilities, thereby improving game design capabilities. The integration of these two thinking approaches helps designers create games that are both challenging and aligned with user needs, ultimately improving players' problem-solving skills. Furthermore, by integrating CT and DT into game development, learners can further foster CT and DT skills (Chen and Huang 2017; Wanglang et al., 2024; Wu and Richards 2011).

Computational thinking in game design. The core of CT includes decomposition, pattern recognition, abstraction, and algorithm design. Applying CT in game design learning helps designers analyze the game's architecture, processes, system development, and logic. Through decomposition, complex game elements (such as characters, animations, and code) can be broken down and analyzed to find effective solutions and workflows. Pattern recognition allows the analysis of player behavior and game dynamics, aiding in the modular design of smarter enemies and more challenging game scenarios. Using abstraction and algorithm design, developers can more efficiently handle programming, character development, and scene design.

A previous study (Wu et al., 2024) conducted a systematic review of 37 articles collected from the WOS database. The results reveal that computational thinking skills represent a new paradigm in problem-solving. The stages of computational thinking—including decomposition, pattern recognition, abstraction, and algorithm design—are closely related to enhancing problem-solving abilities.

Design thinking in game design. Design Thinking (DT) is a user-centered problem-solving approach that emphasizes understanding user needs, proposing innovative ideas, and rapidly iterating tests. When applied to learning game design, DT helps designers analyze user requirements and approach the design from the player's perspective. This includes designing educational game features and proposing innovative concepts, creating engaging and educationally meaningful games from the learner's point of view. By incorporating DT processes, game design can be continuously optimized. For example, using DT methods allows designers to prototype the game's core mechanics and quickly adjust based on user feedback, improving playability and ensuring the game has educational value. DT emphasizes human-centered innovation, aiming to understand user needs, propose creative solutions, and iterate through testing. In game design education, DT helps learners create more engaging and educationally meaningful games that meet user needs, allowing players to enjoy the game while achieving the design objectives (Wanglang et al., 2024).

In summary, CT provides techniques for handling complex problems, including decomposition, pattern recognition, and algorithm design. In game design, CT helps designers accurately analyze game logic and development mechanisms. Through algorithm design, they can develop game mechanics and player interaction systems. This is crucial for enhancing the completeness, playability, and educational significance of game development (Chen and Huang, 2017; Wu and Richards 2011). For example, a previous study (Wanglang et al., 2024) developed a combined game-based learning approach with design thinking using block-based programming to enhance computational thinking and creative games through platforms like SCRATCH or LOGO (Wu and Richards 2011). Design thinking is a creative, human-centered, participative, exploratory, and problem-solving process. A previous study proposed a curriculum framework for K-12 students using a block-based game tool, APP Inventor, to foster computational thinking (Chen and Huang 2017).

Overall, the combination of CT and DT makes game design more effective and focused on the educational goals of the players. This not only enhances learners' problem-solving and innovation skills during game design development but also improves the quality of the game (Wanglang et al., 2024).

Research limitation. Several research limitations are described as below. (1) Scope limitation: This study only selected literature involving actual game development and the use of games for experimental research. As a result, theoretical or conceptual studies were excluded. This may limit the broader understanding of the relationship between game design and computational thinking. We suggest that future researchers expand the scope to incorporate a broader sample size, potentially exploring additional research methods or timeframes to achieve greater inclusivity. Future studies could expand the database selection to include a wider range of sources, such as open-access journals or specialized repositories, to provide a more comprehensive view. (2) Language limitation: due to language constraints, this study only includes English-language literature for analysis, excluding non-English publications. This may lead to a biased understanding of the global landscape of the research field. Future studies could consider including multilingual literature to obtain a more comprehensive perspective. These limitations have been outlined in the research findings, and it is recommended that future studies expand the language range and types of research to achieve a more comprehensive analysis.

Data availability

Not applicable. Data sharing is not applicable to this research as no data was generated or analyzed.

Received: 28 April 2024; Accepted: 27 January 2025;
Published online: 08 February 2025

References

Agbo FJ, Oyelere SS, Suhonen J, Laine TH (2021) Co-design of mini games for learning computational thinking in an online environment. Educ Inf Technol 26(5):5815–5849

Agbo FJ, Oyelere SS, Suhonen J, Tukiainen M (2023) Design, development, and evaluation of a virtual reality game-based application to support computational thinking. Educ Technol Res Dev 71(2):505–537

Apiquian R, Ulloa RE, Victoria G, Gómez-Tello MF, Morales E, García-Covarrubias L (2020) Standardization and validity of Chefmania, a video game designed as a cognitive screening test for children. Humanit Soc Sci Commun 7(1):51

Barradas R, Lencastre JA, Soares S, Valente A (2020) Developing computational thinking in early ages: a review of the code.org Platform

Basawapatna A, Koh KH, Repenning A (2010) Using Scalable Game Design to teach computer science from middle school to graduate school. https://doi.org/10.1145/1822090.1822154

Boom K-D, Bower M, Siemon J, Arguel A (2022) Relationships between computational thinking and the quality of computer programs. Educ Inf Technol 27(6):8289–8310

Cafarella L, Vasconcelos L (2024) Computational thinking with game design: an action research study with middle school students. Educ Inf Technol. https://doi.org/10.1007/s10639-024-13010-5

Çakır NA, Çakır MP, Lee FJ (2021) We game on skyscrapers: the effects of an equity-informed game design workshop on students' computational thinking skills and perceptions of computer science. Educ Technol Res Dev 69(5):2683–2703

Chen P, Huang R (2017) Design thinking in app inventor game design and development: a case study. 2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT), Timisoara, Romania, 139–141, https://doi.org/10.1109/ICALT.2017.161

Cheng G (2019) Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. Comput Hum Behav 92:361–372

Cheng Y-P, Lai C-F, Chen Y-T, Wang W-S, Huang Y-M, Wu T-T (2023) Enhancing student's computational thinking skills with student-generated questions strategy in a game-based learning platform. Comput Educ 200:104794

Chevalier M, Giang C, Piatti A, Mondada F (2020) Fostering computational thinking through educational robotics: a model for creative computational problem solving. Int J STEM Educ 7(1):1–18

Chien Y-C, Chang P-Y, Lee H-Y, Huang T-Y, Huang Y-M (2020) A LUPDA Assessment Model for Activities in STEAM Education. Innovative Technologies and Learning, Cham

Choi D-H, Noh G-Y (2021) The impact of presence on learning transfer intention in virtual reality simulation game. SAGE Open 11(3). https://doi.org/10.1177/21582440211032178

Constantinou CP, Rybska E (2024) Design principles for integrating science practices with conceptual understanding: an example from a digital learning environment on microbial resistance to antibiotics. Humanit Soc Sci Commun 11(1):501. https://doi.org/10.1057/s41599-024-03022-4

del Olmo-Muñoz J, Cózar-Gutiérrez R, González-Calero JA (2020) Computational thinking through unplugged activities in early years of Primary Education. Comput Educ 150

Demartini CG, Sciascia L, Bosso A, Manuri F (2024) Artificial Intelligence Bringing Improvements to Adaptive Learning in Education: A Case Study. Sustainability 16(3):1347

Fanchamps N, van Gool E, Slangen L, Hennissen P (2024) The effect on computational thinking and identified learning aspects: Comparing unplugged smartGames with SRA-Programming with tangible or On-screen output. Educ Inf Technol 29(3):2999–3024

Freepik (2023a) Design thinking process. https://www.freepik.com/free-vector/process-infographic-template_15441403.htm#query=design%20thinking%20process&position=0&from_view=keyword&track=ais

Freepik (2023b) Free vector gradient circular diagram infographic. https://www.freepik.com/free-vector/gradient-circular-diagram-infographic_14455222.htm#query=cycle&position=12&from_view=search&track=sph

Garneli V, Chorianopoulos K (2017) Programming video games and simulations in science education: exploring computational thinking through code analysis. Interact Learn Environ 26(3):386–401. https://doi.org/10.1080/10494820.2017.1337036

Gentile M, Lieto A (2022) The role of mental rotation in TetrisTM gameplay: An ACT-R computational cognitive model. Cogn Syst Res 73:1–11. https://doi.org/10.1016/j.cogsys.2021.12.005

Harel I, Papert S (1991) Constructionism. Ablex Publishing. https://psycnet.apa.org/record/1991-99006-000

Hava K, Guyer T, Cakir H (2020) Gifted students' learning experiences in systematic game development process in after-school activities. Educ Technol Res Dev 68(3):1439–1459

Hooshyar D, Pedaste M, Yang Y, Malva L, Hwang G-J, Wang M, Delev D (2020) From gaming to computational thinking: an adaptive educational computer game-based learning approach. J Educ Comput Res 59(3):383–409

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 Educ 126:296–310

Hsu T-C, Liang Y-S (2021) Simultaneously improving computational thinking and foreign language learning: interdisciplinary media with plugged and unplugged approaches. J Educ Comput Res 59(6):1184–1207. https://doi.org/10.1177/0735633121992480

Jiang X, Harteveld C, Yang Y, Fung A, Huang X, Chen S (2023) "If it's sunny, don't take an umbrella": a systematic evaluation of design principles for CT teaching games. Educ Technol Res Dev 71(4):1725–1763

Jun S, Han S, Kim S (2017) Effect of design-based learning on improving computational thinking. Behav Inf Technol 36(1):43–53

Kalelioğlu F (2015) A new way of teaching programming skills to K-12 students: Code.org. Comput Hum Behav 52:200–210

Kazimoglu C, Kiernan M, Bacon L, MacKinnon L (2012) Learning programming at the computational thinking level via digital game-play. Procedia Comput Sci 9:522–531

Kelly N, Gero JS (2021) Design thinking and computational thinking: a dual process model for addressing design problems. Design Science, 7. https://doi.org/10.1017/dsj.2021.7

Kite V, Park S, Wiebe E (2021) The code-centric nature of computational thinking education: a review of trends and issues in computational thinking education research. SAGE Open 11(2):21582440211016418

Kynigos C (2015) Constructionism: Theory of Learning or Theory of Design? In SJ Cho (Ed.), Selected Regular Lectures from the 12th International Congress on Mathematical Education (pp. 417-438). Springer International Publishing. https://doi.org/10.1007/978-3-319-17187-6_24

Lee I, Grover S, Martin F, Pillai S, Malyn-Smith J (2020) Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. J Sci Educ Technol 29(1):1–8

Leonard J, Buss A, Gamboa R, Mitchell M, Fashola OS, Hubert T, Almughyirah S (2016) Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. J Sci Educ Technol 25(6):860–876

Leonard J, Mitchell M, Barnes-Johnson J, Unertl A, Outka-Hill J, Robinson R, Hester-Croff C (2018) Preparing teachers to engage rural students in computational thinking through robotics, game design, and culturally responsive teaching. J Teach Educ 69(4):386–407

Li Q, Liu Z.-x, Wang P, Wang J.-j, Luo T (2023a) The influence of art programming courses on design thinking and computational thinking in college art and design students. Education and Information Technologies

Li W, Liu C-Y, Tseng, JCR (2023b) Effects of the interaction between metacognition teaching and students' learning achievement on students' computational thinking, critical thinking, and metacognition in collaborative programming learning. Education and Information Technologies

Litts BK, Lewis WE, Mortensen CK (2020) Engaging youth in computational thinking practices through designing place-based mobile games about local issues. Interact Learn Environ 28(3):302–315

Liu T (2024) Assessing implicit computational thinking in game-based learning: a logical puzzle game study. Br J Educ Technol 55(5):2357–2382

Lye SY, Koh JHL (2014) Review on teaching and learning of computational thinking through programming: What is next for K-12? Comput Hum Behav 41:51–61. https://doi.org/10.1016/j.chb.2014.09.012

Melander Bowden H (2019) Problem-solving in collaborative game design practices: epistemic stance, affect, and engagement. Learn, Media Technol 44(2):124–143

Melander Bowden H, Aarsand P (2020) Designing and assessing digital games in a classroom: an emerging culture of critique. Learn Media Technol 45(4):376–394. https://doi.org/10.1080/17439884.2020.1727500

Munoz R, Villarroel R, Barcelos TS, Riquelme F, Quezada A, Bustos-Valenzuela P (2018) Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. IEEE Access 6:63880–63889

Newton KJ, Leonard J, Buss A, Wright CG, Barnes-Johnson J (2020) Informal STEM: learning with robotics and game design in an urban context. J Res Technol Educ 52(2):129–147

Ostovar-Namaghi SA, Morady Moghaddam M, Rad, E (2023) The effect of interactive games on English language learners' reading comprehension and attitudes. Asia Pacific Education Review
Othman MK, Jazlan S, Yamin FA, Aman S, Mohamad FS, Anuar NN, Saleh AY, Abdul Manaf AA (2023) Mapping computational thinking skills through digital games co-creation activity amongst Malaysian sub-urban children. J Educ Comput Res 61(2):355–389
Papavlasopoulou S, Giannakos MN, Jaccheri L (2019) Exploring children's learning experience in constructionism-based coding activities through design-based research. Comput Hum Behav 99:415–427
Papavlasopoulou S, Sharma K, Giannakos MN (2020) Coding activities for children: coupling eye-tracking with qualitative data to investigate gender differences. Comput Hum Behav 105:105939
Pila S, Aladé F, Sheehan KJ, Lauricella AR, Wartella EA (2019) Learning to code via tablet applications: an evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Comput Educ 128:52–62
Repenning A, Webb DC, Koh KH, Nickerson H, Miller SB, Brand C, Horses IHM, Basawapatna A, Gluck F, Grover R (2015) Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. ACM Trans Comput Educ (TOCE) 15(2):1–31
Resnick M, Maloney J, Monroy-Hernández A, Rusk N, Eastmond E, Brennan K, Silverman B (2009) Scratch: programming for all. Commun ACM 52(11):60–67
Richard GT, Giri S (2019) Digital and Physical Fabrication as Multimodal Learning: Understanding Youth Computational Thinking When Making Integrated Systems Through Bidirectionally Responsive Design. ACM Trans Comput. Educ 19(3):1–35. https://doi.org/10.1145/3243138
Rodríguez-Martínez JA, González-Calero JA, Sáez-López JM (2020) Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. Interact Learn Environ 28(3):316–327
Saritepeci M (2020) Developing computational thinking skills of high school students: Design-based learning activities and programming tasks. Asia-Pac Educ Res 29(1):35–54
Saxena A, Lo CK, Hew KF, Wong GKW (2020) Designing unplugged and plugged activities to cultivate computational thinking: an exploratory study in early childhood education. Asia-Pac Educ Res 29(1):55–66
Seralidou E, Douligeris C (2021) Learning programming by creating games through the use of structured activities in secondary education in Greece. Educ Inf Technol 26(1):859–898. https://doi.org/10.1007/s10639-020-10255-8
Sharma K, Papavlasopoulou S, Giannakos M (2019) Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes? Int J Child-Comput Interact 21:65–76
Taimur S, Onuki M, Mursaleen H (2022) Exploring the transformative potential of design thinking pedagogy in hybrid setting: a case study of field exercise course, Japan. Asia Pac Educ Rev 23(4):571–593
Tikva C, Tambouris E (2023) The effect of scaffolding programming games and attitudes towards programming on the development of Computational Thinking. Educ Inf Technol 28(6):6845–6867. https://doi.org/10.1007/s10639-022-11465-y
Tran Y (2019) Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. J Educ Comput Res 57(1):3–31
Tsai F-H, Hsiao H-S, Yu K-C, Lin K-Y (2021) Development and effectiveness evaluation of a STEM-based game-design project for preservice primary teacher education. Int J Technol Design Educ. 1–22
Tsarava K, Moeller K, Pinkwart N, Butz M, Trautwein U, Ninaus, M (2017) Training computational thinking: Game-based unplugged and plugged-in activities in primary school. Paper presented at the European conference on games based learning
Tucker-Raymond E, Cassidy M, Puttick G (2021) Science teachers can teach computational thinking through distributed expertise. Comput Educ 173:104284
Tucker-Raymond E, Puttick G, Cassidy M, Harteveld C, Troiano GM (2019) "I Broke Your Game!": critique among middle schoolers designing computer games about climate change. Int J STEM Educ 6(1):1–16
Turchi T, Fogli D, Malizia A (2019) Fostering computational thinking through collaborative game-based learning. Multimed Tools Appl 78(10):13649–13673
Voogt J, Fisser P, Good J, Mishra P, Yadav A (2015) Computational thinking in compulsory education: Towards an agenda for research and practice. Educ Inf Technol 20(4):715–728
Wang C-C (2024) Using design thinking for interdisciplinary curriculum design and teaching: a case study in higher education. Humanit Soc Sci Commun 11(1):307
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. High Educ Stud 14:137
Werner L, Denner J, Campe S (2014) Children programming games: a strategy for measuring computational learning. ACM Trans Comput Educ (TOCE) 14(4):1–22
Wing JM (2019) A conversation about computational thinking. Chapter 8:127–140
Wong GK-W, Cheung H-Y (2018) Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. Interact Learn Environ 28(4):438–450. https://doi.org/10.1080/10494820.2018.1534245
Wong GKW, Jiang, S (2018) Computational Thinking Education for Children: Algorithmic Thinking and Debugging. 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). 328–334 https://doi.org/10.1109/TALE.2018.8615232
Wu C-H, Huang Y-M (2020) Integration of LUPDA Theory and STEAM with Computational Thinking Concepts to Develop Assessment Principles for an AI Based STEAM Activity. Innovative Technologies and Learning, Cham. https://doi.org/10.1007/978-3-030-63885-6_31
Wu ML, Richards K (2011) Facilitating Computational Thinking through Game Design. In: Chang M, Hwang WY, Chen MP, Müller W (eds) Edutainment Technologies. Educational Games and Virtual Reality/Augmented Reality Applications. Edutainment 2011. Lecture Notes in Computer Science, vol 6872. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23456-9_39
Wu T-T, Asmara A, Huang Y-M, Permata Hapsari I (2024) Identification of problem-solving techniques in computational thinking studies: systematic literature review. SAGE Open 14(2):21582440241249897
Yeoh C-P, Li C-T, Hou H-T (2024) Game-based collaborative scientific inquiry learning using realistic context and inquiry process-based multidimensional scaffolding. Int J Sci Educ, 1–23
Zainuddin Z (2023) Integrating ease of use and affordable gamification-based instruction into a remote learning environment. Asia Pacific Educ Rev
Zhang B, Goodman L, Gu X (2022) Novel 3D contextual interactive games on a gamified virtual environment support cultural learning through collaboration among intercultural students. SAGE Open 12(2):21582440221096141
Zhang S, Wong GKW, Chan PCF (2023) Playing coding games to learn computational thinking: What motivates students to use this tool at home? Educ Inf Technol 28(1):193–216

Acknowledgements

The authors would like to thank the Ministry of Science and Technology Council, Taiwan, for financial support (NSTC 112-2410-H-142-027-MY3).

Author contributions

Chih-Hung Wu: Conceptualization, Investigation process, Methodology, Data curation, Visualization, Resources, Writing-Original draft preparation. Yu-Cheng Chien: Methodology, Data curation, Writing-Original draft preparation. Mei-Tzu Chou: Data curation, Data analysis, Writing-Original draft preparation. Yeuh-Min Huang: Conceptualization, Investigation process, Supervision, Validation, Writing-Reviewing and Editing.

Competing interests

The authors declare no competing interest.

Ethical approval

Ethical approval was not required as the study did not involve human participants.

Informed consent

This study does not involve human participants or their data.

Additional information

Correspondence and requests for materials should be addressed to Yueh-Min Huang.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.