



Studying the effects of educational games on cultivating computational thinking skills to primary school students: a systematic literature review

Andreas Giannakoulas¹ · Stelios Xinogalos¹

Received: 22 July 2023 / Revised: 2 October 2023 / Accepted: 4 October 2023
© The Author(s) 2023

Abstract

This article presents a systematic literature review (SLR) on the effects of serious games, or more specifically educational games that aim to teach Computational Thinking (CT) skills to primary school students. Sixty one studies from various data sources were evaluated based on the CT skills and programming concepts addressed, the evaluation instruments used, the target audience, the learning outcomes and their results. The findings of the studies on the efficiency or impact of educational games on the acquisition of the proposed topics were positive, indicating that educational programming games can help primary school students develop CT skills or understand fundamental programming concepts. Additionally, the results suggest a general positive attitude towards the use of an educational game for learning purposes, while students perceive games as a great motivator for engaging in CT activities. Finally, the research discusses research gaps and shortages, as well as methodological limitations and recommendations for future work in the relevant domain.

Keywords Computational Thinking · Programming · Serious games · Educational games · Effectiveness · Primary education · Systematic literature review (SLR)

Introduction

All around the world, K–12 curriculum already includes Computational Thinking (CT), which is regarded as an essential competency for the twenty-first century (Cutumisu et al., 2019). Although Papert initially proposed CT in 1980 and again in

✉ Stelios Xinogalos
stelios@uom.edu.gr

Andreas Giannakoulas
agiannakoulas@uom.edu.gr

¹ Department of Applied Informatics, School of Information Sciences, University of Macedonia, 156 Egnatia Street, 54636 Thessaloniki, Greece

1996 (Papert, 1980, 1996), Wing popularized CT (2006). She stated that CT encompasses a mix of abilities, approaches, procedures, and mindsets that facilitate resolving a diverse set of problems, beyond just those associated with computing (Wing, 2006). Moreover, she proposed that CT should be considered an important foundational skill for all children just like reading or writing (Zhao & Shute, 2019). Despite the fact that there are numerous research studies on CT (Berland & Wilensky, 2015; García-Penalvo & Mendes, 2018; Hsu et al., 2018; Taslibeyaz et al., 2020; Bati, 2022), there is still no agreement among scientists regarding its definition and the specific skills that it comprises (Lye & Koh, 2014; García-Penalvo & Mendes, 2018; Zhao & Shute, 2019; Guenaga et al., 2021). As a result, many issues remain (Guenaga et al., 2021) and various definitions appear alongside Wing's (2006), highlighting the need for systematically reviewing existing literature and drawing conclusions on the current state of the field.

Brennan and Resnick (2012) put forward three aspects of CT, namely computational concepts, computational practices, and computational perspectives (Lye & Koh, 2014; Tang et al., 2020). Atmazidou and Demetriadis (2016) introduced a CT model that incorporates a collection of key CT skills, such as abstraction, decomposition, generalization, algorithmic thinking, and modularity. Shute et al. (2017) classified CT into six primary dimensions, which include decomposition, abstraction, algorithms, debugging, iteration, and generalization. Furthermore, in their work, Grover and Pea (2018) suggest that the essential components of CT can be categorized into concepts and practices. Specifically they state that CT concepts encompass logic and logical thinking, algorithms and algorithmic thinking, patterns and pattern recognition, abstraction and generalization, evaluation, and automation, while CT practices, on the other hand, involve problem decomposition, the creation of computational artifacts, testing and debugging, iterative refinement, as well as collaboration and creativity (Grover & Pea, 2018). In their study, Tang et al. (2020) classified the definitions of CT found in the literature into two distinct categories. The first category encompasses definitions that link CT to programming and computing concepts, while the second category comprises definitions that regard CT as a collection of competences necessitating students to enhance their domain-specific knowledge and problem-solving skills (Tang et al., 2020).

CT represents a versatile and widely applicable problem-solving approach (Grover & Pea, 2013) that spans various domains (Grover & Pea, 2018; Wing, 2008) and furthermore, it is acknowledged as a vital skill not only for computer scientists but for individuals of all backgrounds, highlighting the need for its early education and acquisition (Hooshyar et al., 2021b). Taking this into account CT has been applied to a variety of different topics including Mathematics or Biology (Hsu et al., 2018). However, current research suggests that programming activities can serve as an effective vehicle for CT development (Grover & Pea, 2018) from an early age (Moreno-León et al., 2018; Zhao & Shute, 2019). As a result, many curriculum developers that want to integrate CT into classroom settings use various programming tools and environments to familiarize kids to coding (Theodoropoulos & Lepouras, 2020). Moreover, notable personalities in the field of information technology are conducting various initiatives worldwide to encourage learning computer programming. Educational events on an international level, such as the “Hour of

Code” (Hourofcode, 2023) or the “Bebras” competition (Bebras, 2023), are held on an annual basis for kids of all ages with the goal of increasing their experience with computer programming and CT.

Learning programming principles has always been challenging for students, particularly for inexperienced programmers (Zaharija et al., 2013; Shahid et al., 2019). Novice programmers’ insufficient enthusiasm and engagement can be attributed to the conventional teaching approach and the challenges they encounter in comprehending abstract programming concepts (Brusilovsky et al., 1997; Shahid et al., 2019). Several strategies have been proposed (Xinogalos & Satratzemi, 2004), such as programming microworlds (Xinogalos et al., 2006), instructional robotics tasks, and educational games (Vahldick et al., 2014), to address these issues and make learning programming easier and more interesting for students.

LOGO (Papert, 1980) is widely recognized as the earliest and most renowned microworld designed for programming education (Djelil et al., 2016). Microworlds for programming are constructed using metaphors, with the purpose of bridging the gap between abstract programming concepts and the tangible real world, thus facilitating comprehension for novice learners (Djelil et al., 2016). This endeavor ultimately narrows the disparity between students’ mental models and the programming language itself (Xinogalos et al., 2006), leading to a more effective learning experience.

Educational Robotics (ER) is widely regarded as a potent, versatile, and groundbreaking educational instrument that presents possibilities for constructing and commanding robots by utilizing a diverse range of sensors, motors, and other components (Atmatzidou & Demetriadis, 2017; Papadakis, 2020) and has found its way into classrooms from kindergarten to high school (Constantinou & Ioannou, 2018; Papadakis, 2020). Studies involving younger children have revealed favorable results, indicating that even kids aged 4–6 can successfully construct basic robotics projects, getting opportunities to become familiar with important concepts in STEM (Science, Technology, Engineering and Mathematics) education or computer programming (Atmatzidou & Demetriadis, 2016; Papadakis, 2020) and develop CT skills (Atmatzidou & Demetriadis, 2016; Constantinou & Ioannou, 2018). Educational Robotics encourages a pleasant (Atmatzidou & Demetriadis, 2016) and age-appropriate approach to learning technology (Taylor & Baek, 2018), effectively increasing engagement among both students and teachers (Budiyanto et al., 2021), and furthermore, it fosters collaboration and motivation among students (Atmatzidou & Demetriadis, 2016; Budiyanto et al., 2021).

Educational games are a subcategory of serious games that prioritize teaching over entertainment and actively facilitate a “learning by doing” educational approach (Malliarakis et al., 2017). Incorporating educational games into the teaching process brings forth a multitude of advantages. These games provide students with a unique learning experience where they can actively engage in an interactive environment that sparks their motivation. Simultaneously, they receive relevant feedback, encounter challenges, and receive appropriate support to aid their learning process (Kazimoglu et al., 2012; Laporte & Zaman, 2016). Similarly, when it comes to primary school students, the significance of using educational games to introduce programming concepts and promote CT is further amplified. As emphasized by

Zaharija et al. (2013), games can assume a crucial role in instructing programming to young students, as the traditional teaching methods are inadequate for their age group and do not effectively address challenges such as their limited attention span. One significant benefit of educational games that supports their integration into the educational process is their lack of cost compared to the expenses associated with purchasing robotic kits. The high cost of robotic kits acts as a hindrance to their widespread adoption in education (Fessakis et al., 2019), making educational games a more accessible alternative. Moreover, educational games, just like programming microworlds, bridge the gap between abstract programming concepts and the tangible real world (Djelil et al., 2016), but in a much more familiar and engaging way since students today are “native speakers” of the digital language of video games as Prensky (2001) highlighted.

Although it is well recognized that employing educational games in the instructional approach results in a range of beneficial outcomes and multiple advantages (Laporte, & Zaman, 2016; Malliarakis et al., 2014a), it is also widely acknowledged that the influence of educational games on CT cultivation has not been extensively studied (Zhao & Shute, 2019). As far as we know, there are no reviews investigating the effects of educational games for CT and programming on the development of CT abilities, particularly among young students.

This article aims to deal with the aforementioned research gap and disclose the findings of a Systematic Literature Review (SLR) that investigates the outcomes of educational games designed to teach CT abilities to primary school students. The remaining sections of the article are structured in the following manner. Sect. “[Related work](#)” provides a summary of relevant literature. Sect. “[Review method](#)” outlines the methodology used in our SLR along with the research questions (RQs) and choices made throughout the analysis of the studies. Sect. “[Results](#)” demonstrates and evaluates the results, whereas Sect. “[Discussion](#)” debates the findings concerning the research questions. At last, Sect. “[Conclusions](#)” summarizes the conclusions derived from the analysis of the studies and emphasizes the next steps in the research, as well as the constraints of this study.

Related work

The major purpose of this SLR is to identify the most recent research on the use of educational games to cultivate young students’ CT abilities and promote fundamental programming principles, as well as to investigate the influence of these games on the acquisition of these abilities. To accomplish this research, existing relevant research papers were examined and analyzed and their key findings are summarized in chronological order. Nine relevant systematic reviews were found in total, but only one of them (Theodoropoulos & Lepouras, 2020) dealt with the use of educational games to teach programming principles and foster CT abilities. Table 1 summarizes these nine relevant systematic investigations in relation to our study. The following databases were the most commonly utilized in the studies: ERIC, Science Direct, Scopus, Web of Science, ACM Digital library, IEEE Xplore, SpringerLink Digital library and Google Scholar.

Table 1 Systematic literature reviews on CT, programming and educational games

Study	Research type	Year	Quality assessment of primary studies	number of primary studies/number of data sources	Period	Context
Lye and Koh	Review on CT & P	2014	No quality assessment	27/2	2009–2013	Trends on cultivating CT through programming to K-12 students
Hsu et al.	Meta-review on CT	2018	Based on criteria/coding scheme developed by Hwang and Wu (2014)	120/1	2006–2017	CT development and implementation approaches in education
Shahid et al.	Review on Games & P	2019	No quality assessment	41/4	2015–2019	Factors that influence the effectiveness of gamification strategies
Cutumisu et al.	Scoping review on CT assessment	2019	Based on Arksey and O'Malley (2005) approach	39/7	2014–2018	CT assessments' features
Theodoropoulos and Lepouras	SLR on CT & Games & P	2020	Based on the revised guidelines of SLR by Kitchenham et al. (2015)	44/6	2009–2019	Impact of Programming Games in K-12 Education and factors affecting their use
Tang et al.	SLR on CT assessment	2020	Based on established criteria used in earlier reviews and on the procedures of a content analysis (Fraenkel, Wallen, & Hyun, 2015)	96/3	Until August 2019	Features of studies exploring CT assessment
Taslibeyaz et al.	SLR on CT	2020	Based on criteria by Gough, Thomas, & Oliver (2012)	39/2	Until 2020	CT development process in learning environments
Papadakis	SLR on CT & CF	2021	Based on the SLR guidelines by Kitchenham (2004)	21/16	2010–2020	Effectiveness of 4 coding apps in developing young children's CT and Computational Fluency (CF)

Table 1 (continued)

Study	Research type	Year	Quality assessment of primary studies	number of primary studies/number of data sources	Period	Context
Bati	SLR on CT & P	2022	Based on the SLR guidelines by Kitchenham (2004)	24/4	2008–2020	CT skills and Programming skills development in early childhood education
This study	SLR on CT & Games & P	2022	Based on the SLR guidelines by Kitchenham (2004)	61/5	2010–2022	Impact of educational games for promoting CT and Programming principles to primary school students

Lye and Koh (2014) conducted a review of empirical studies on cultivating CT skills through programming activities to K-12 students over a period of 5 years (2009–2013), noting a number of beneficial results. Authors concluded that in most cases a visual programming language was used and believe that it is necessary to carry out more empirical teaching interventions in K-12 education regarding the development of CT.

Hsu et al. (2018) presented a review of literature published over a 12 years period (2006–2017) relating to CT development and implementation approaches in education. According to the findings, there have been more CT studies in recent years than ever before, with a wider range of subjects, research questions, and teaching tools, and CT has mostly been used to computer science and programming activities, with only a small number of studies relevant to other topics. The necessity to evaluate students' learning progress and the requirement for teaching staff training to develop adequate CT interventions are the main points of recommendations for future research.

Shahid et al. (2019) conducted an SLR over a 5-years period (2015–2019) on the existing literature of serious games for programming, in order to examine the factors that influenced the effectiveness of gamification strategies. The SLR findings revealed two major aspects influencing negatively instructors' adoption of a game in teaching programming: (1) a lack of highly attractive features in games; and (2) a lack of effective ways for assessing game impact. As the authors state, the absence of visual appeal in games renders them monotonous and uninteresting, leading to a detrimental impact on the student's engagement. In order for educators to integrate games into their teaching methods, they must possess knowledge about aspects like students' gaming behavior, their learning process, and the tools available for teachers to understand the impact of educational games in the classroom. Understanding the ways in which students engage with, learn from, and enhance their skills through games is a crucial element in promoting broader acceptance of educational gaming (Giannakoulas et al., 2020). Learning analytics can be a valuable tool for teachers to monitor students' progress, analyze user behavior during gameplay (Eguiluz et al., 2018), and assess the impact of games on learning, ultimately aiding in the evaluation of students' understanding of the taught concepts (Giannakoulas & Xinogalos, 2020; Malliarakis et al., 2014b).

To determine and categorize the distinctive features of current CT assessments, Cutumisu et al. (2019), reviewed 39 empirical studies that were published over a 5-years period (2014–2018). The findings of this study reveal a wide range of approaches used by scientists to evaluate CT, but which are typically not verified enough to support their adoption by educational researchers and incorporation in the learning process. Future research ideas focus on developing new methods for assessing CT and aiding academics in selecting suitable CT evaluations.

Theodoropoulos and Lepouras (2020) conducted an SLR providing a full summary of efforts made to include into P12 education, digital games aiming to CT cultivation or to introduce fundamental programming principles, examining 44 studies, published throughout an 11-years period (2009–2019). The findings showed that while using digital games in CT education presents favorable outcomes, these are highly dependent on both the learning environment and the individuals involved.

Moreover, authors suggest that while building and deploying digital games, we should consider numerous aspects such as personality qualities, the element of feedback or the games' genre, and that further investigation is necessary in this area.

Tang et al. (2020) presented an SLR which examined 96 empirical studies related on CT assessments, published before August 2019, aiming to examine the studies from four dimensions: the educational context, the CT skills measured, the measurement tools used and the CT evaluations' accuracy proof. The SLR findings show that more frequently CT evaluations are carried out in young students and usually focus on measuring the CT skills or the programming abilities of students.

Taslibeyaz et al. (2020) conducted an SLR analyzing 29 empirical studies focused on CT development, without limiting the period of publication, aiming to investigate the common learning process followed in the current environments and determine the CT measurements tools used. Findings of the SLR showed that the development of CT skills is mostly evaluated utilizing robotics and programming environments like Scratch, while for college students, studies on the development of CT are also conducted in other content areas outside programming.

Papadakis (2021) conducted an SLR on twenty-one empirical studies over an eleven-year period (2010–2020), attempting to assess the effectiveness of four coding applications in developing young children's CT abilities and Computational Fluency (CF). According to the study findings, all applications have a good impact on the growth of children's CT skills, whereas none of them can eventually foster the growth of CF.

Bati (2022) conducted an SLR on twenty-four empirical studies that prioritize the growth of CT skills and the enhancement of preschool pupils' programming abilities, published over a thirteen-year period (2008–2020). The studies were examined based on three criteria: intervention technique (plugged-in or unplugged), participant age, and student gender. The results showed that age, rather than gender, were a crucial factor in the development of CT in young children. Unplugged techniques also appeared to be more efficient than plugged-in ones.

Although there have been reviews exploring the latest advancements in CT assessments (Cutumisu et al., 2019; Tang et al., 2020), the different strategies employed to foster CT in recent years (Cutumisu et al., 2019; Hsu et al., 2018), the CT concepts covered in the studies (Shahid et al., 2019; Cutumisu et al., 2019; Theodoropoulos & Lepouras, 2020; Tang et al., 2020), the typical learning process followed by researchers (Lye & Koh, 2014; Cutumisu et al., 2019; Taslibeyaz et al., 2020; Bati, 2022), the context of application (Tang et al., 2020; Taslibeyaz et al., 2020), the diverse learning environments (Hsu et al., 2018; Papadakis, 2021) and the various tools used to measure the cultivation of CT among students (Shahid et al., 2019; Cutumisu et al., 2019; Theodoropoulos & Lepouras, 2020; Tang et al., 2020; Taslibeyaz et al., 2020), we were not able to find any review that investigates the effects of educational programming games on cultivating CT skills, particularly among young students. Through this research, our aim is to compile existing empirical studies conducted on the utilization of educational programming games in primary education. Our objective is to systematically analyze this research, shedding light on the effectiveness of these games in fostering the growth of CT skills. Furthermore, we aim to emphasize the potential advantages associated with incorporating these

games into the educational process, as well as the challenges and barriers that may impede their successful integration into the learning process.

Moreover, our intention is to address the concerns raised by researchers and educators who often express a lack of knowledge on effectively integrating games into the learning process (Giannakoulas et al., 2020). By showcasing the different ways in which games have been utilized within education, we aim to facilitate the acceptance of these games as valuable tools for learning among these stakeholders.

The SLR we conducted covers a 13-year period (2010–2022) and intends to highlight the existing research on the impact of educational games built for promoting CT and programming principles to primary school students. The studies found are reviewed using criteria such as CT skills, programming concepts covered, evaluation instruments, intended audience learning results, and evaluation conclusions.

Review method

Kitchenham's (2004) recommendations served as the foundation for the methodology employed to conduct this literature review.

Research questions

To accomplish the main goal highlighted in Sect. "Related work", the SLR was steered by the subsequent research questions, aligned with the research's primary objective:

RQ1: Which facets of CT and programming are addressed by the available studies?

RQ2: What evaluation methods are utilized to gauge the effectiveness of educational games produced to teach young students CT and programming?

RQ3: Do educational games for CT and programming targeted to young students have a positive effect on learning the underlying concepts?

Strategy

To find empirical studies regarding educational games designed for teaching CT and programming to young students, the subsequent search term was applied:

("serious game*" OR "educational game*") AND ("primary school children" OR "young children" OR "K-12" OR "primary education") AND ("teaching programming" OR "computational thinking" OR "learning programming" OR "algorithmic thinking")

The following digital research databases were searched initially between April 2021 and September 2022: Scopus, ACM Digital Library, SpringerLink, Science Direct and IEEE Xplore. Afterward, additional searches were carried out on Google Scholar to discover published and unpublished material that had not been indexed by

the databases. The process of retrieving data from the studies was performed using a data extraction form created in Microsoft Excel. Subsequently, the selection process consisted of three phases. During the first phase, we looked over each paper's title, abstract, and keywords. At this stage, we saved the potentially relevant papers and eliminated the duplicate or irrelevant studies. By means of this methodology, this study also excluded literature reviews. The second phase involved eliminating papers that were not published in English, papers with unavailable full texts, and publications that were not complete articles (such as extended abstracts).

Figure 1 depicts the full research selection procedure as well as the actual number of articles in each phase. The two researchers that carried out the whole study rigorously applied Kitchenham's (2004) recommendations for conducting SLRs and agreed on the research protocol and the paper selection process prior paper selection. Both researchers had experience in conducting reviews and SLRs. After retrieving the records returned by applying the search query, the two researchers independently carried out each one of the three phases of the selection process and compared the results after each phase in order to agree on a common pool of records for the subsequent phase.

Quality assessment

After completing the second phase, the papers were screened according to a set of inclusion and exclusion criteria based on the SLR methodology proposed by Kitchenham (2004), as presented in Table 2, to finalize the study selection process.

Studies analysis

After completing the third phase, the complete text of each potential eligible study was read and the studies were analyzed. During this phase, the studies were scrutinized based on the traits presented in Table 3.

Data extraction

The data from the studies was extracted using a Microsoft Excel-based form created for this purpose. Table 3 shows the characteristics that were considered to extract data.

Results

Studies results

The initial search yielded 1410 results. The total number of papers gathered throughout the first phase was 175, with the remainder discarded as irrelevant or duplicate and after the second phase 150 studies remained as possible included papers. Next, the papers were filtered and after applying the inclusion and exclusion criteria, 61

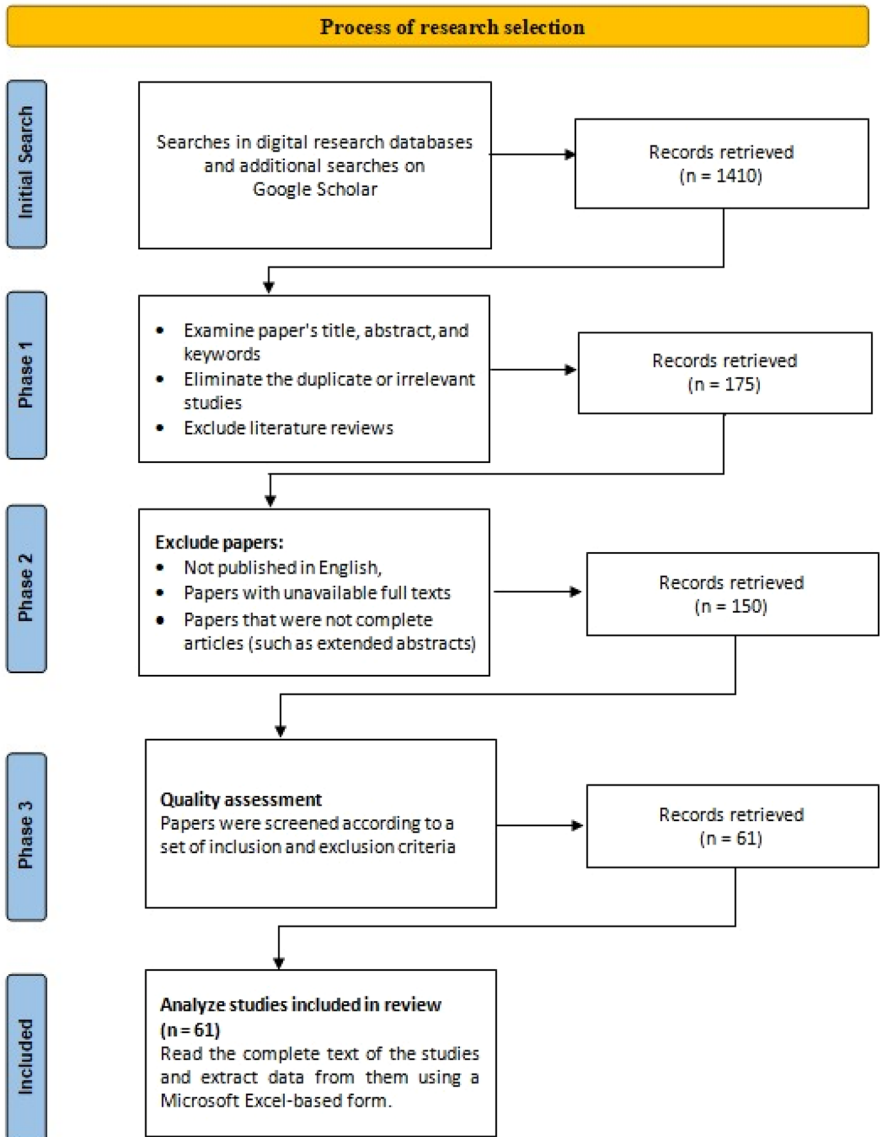


Fig. 1 Process of research selection

main studies were included in the SLR. The distribution of the studies between 2010 and September 2022 is represented in Fig. 2.

According to the yearly distribution of the 61 articles represented in Fig. 2, a rising trend in the publication of empirical studies on the use of SGs to teach young students CT skills and fundamental programming concepts has been seen in recent years, which is consistent with the growing popularity and importance of CT. In

Table 2 Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
Papers published between 2010 and September 2022	Papers published before 2010
Solutions related to games	Board games, Tangible User Interfaces (UI), unplugged activities, studies with game design from students (game maker – scratch)
Research works that comprise details for the evaluation process	Research studies that do not address the evaluation process
Educational games that have been evaluated based on their intended purpose or performance within the game	
Educational games that have undergone evaluation for their level of acceptance or usability	

Table 3 Characteristics used for analyzing papers**General characteristics**

The title of the game or project, if it is available

Researcher names

Accessibility of game or project

Characteristics related to the purpose or objective of the study

CT skills

Programming Concepts

Purpose—Goals of the study

Characteristics associated with the evaluation of the study

Number of individuals involved in the study

Age of individuals involved in the study

Testing method—Instruments

Methodology

Duration

Context of evaluation

Evaluation conclusions

particular, 48 relevant empirical studies were discovered after 2017, representing about 80% of all empirical research conducted after 2010.

The majority of the papers published are from conferences (33), with the remaining 25 coming from journals. Three (3) empirical studies are also included that were part of master's (2) and Ph.D. theses (1).

Figure 3 shows the main subject areas of journal and conference publications. As illustrated in Fig. 3, the primary subject area of journal publication is Educational Computing (24). Computer Science or Information Technology (14) and Serious Games (10) are two more prominent subjects. According to the results of the subject areas of the publications, the majority of empirical investigations involve Information Technology, Computer Science and Education. Taking this

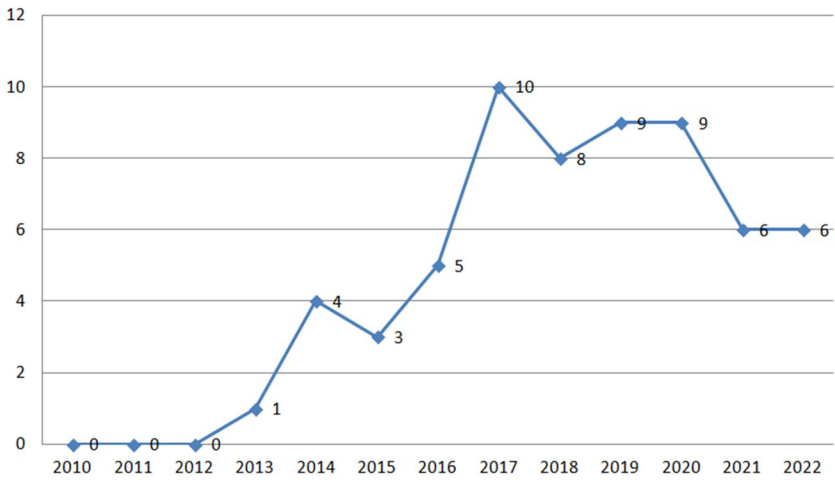


Fig. 2 Flowchart of papers by publication year

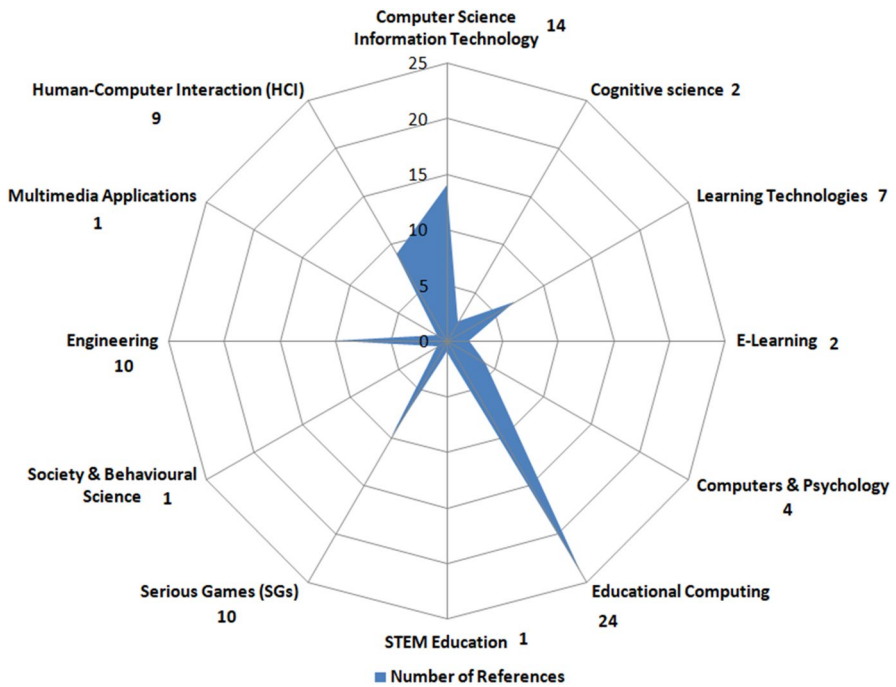


Fig. 3 Journal-Conference publication subjects. Each journal-Conference may cover several topics

into consideration, we can infer that collaboration between researchers and those involved in the educational process (educators, students) is critical for developing games aimed at cultivating CT in young children.

Discussion

This section covers the analysis of the studies' results in relation to the research questions that were established.

RQ1: Which facets of CT and programming are addressed by the available studies?

Table 4 shows how the empirical studies were classified based on the programming topics they addressed, while Table 5 shows how the empirical studies were classified based on CT skills they addressed.

Regarding the programming concepts, it was found that the majority of the studies (50) cover the "Sequence commands execution" while a large proportion also investigate iterative structures (45). Conditional statements are investigated in 33 publications, and it is observed that just a small percentage of studies address concepts such as functions or procedures (18 studies), as well as the idea of variable (13 studies). Furthermore, just three (3) empirical studies (Esper et al., 2014; Lotfi & Mohammed, 2018; Ramírez-Rosales et al., 2016) dealing with principles of object-oriented programming were detected, with only two (2) studies supporting the more complicated concept of recursion (Rose et al., 2019; Lehat et al., 2014). As a result, the most extensively covered notions by researchers and game developers are fundamental programming concepts such as sequencing, conditional statements, and repetition structures whereas OOP concepts and recursion are the least covered. This may be attributed to the highly abstract nature of these concepts, making them difficult to present to primary school children who have not yet developed abstract thinking (Zaharija et al., 2013). Similarly, there is a restricted focus on variables in existing studies, possibly because young students encounter challenges in comprehending abstract representations like variables (Shahid et al., 2019; Papadakis, 2021). As noted by Relkin et al. (2021), it is crucial to consider these difficulties and other developmental factors when designing educational programs aimed at teaching CT to young children (Papadakis, 2021).

Regarding the CT skills covered in the studies, the vast majority of them (39 studies) assist the advancement of *algorithmic thinking* among students by engaging them in programming exercises that require building algorithms to accomplish the given tasks.

In addition, 33 studies endorse *decomposition* by assigning tasks to students that require them to break down an algorithm into smaller, more manageable parts whereas 30 studies advocate for pattern recognition and modularity, often through exercises that involve identifying a pattern to accomplish a task that needs to be programmed and replicated with repetition, reducing redundancy in code. For instance, the game BOTS (Liu et al., 2017) presents players with puzzles that are intentionally crafted with repetitive patterns, allowing students to explore ways to optimize their programs by utilizing loops and functions. Additionally, ILE (Ríos et al., 2020) promotes pattern recognition by offering exercises where students are tasked with adjusting the properties of a 3D object using sliders, aligning the 3D object with the pattern depicted in the background.

Table 4 Studies Classification according to the Programming Concepts covered

Concepts	Studies: game (reference)
Sequential structures 50 studies	4 Games (Lehat et al., 2014); ARcode (Mina et al., 2022); AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a); Bookworms (Korpi, 2014); BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); ChIP (Sorrentino et al., 2017); Code Venture (Hussain et al., 2015); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018); CodeFruits (Goyal et al., 2017); CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodePlanet (Baek & Oh, 2019); CodeSpells (Esper et al., 2014); CodeTrasures (Nche et al., 2019); Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); Crocro 's Adventure (Forquesato, 2018); Daisy the Dinosaur vs Kodable (Pila et al., 2019); ENGAGE (Min et al., 2020); Kids Block Coding Game (Forquesato & Borin, 2018); Kodable (Karadeniz et al., 2014); Kodable vs Scratch Junior (Kandroudi & Bratitsis, 2016); Kodetu (Eguiluz et al., 2017); Kodetu (Guenaga et al., 2021); KODU vs Kodable (Fokides & Atsikpasi, 2017); Ladybug (Fessakis et al., 2013); Lighthbot Code hour (Yallihep & Kutlu, 2020); Looking for Pets (Pessoa et al., 2019); Minecraft Hour of Code (DEMİRKIRAN & TANSU HOCANIN, 2021); Minerva (Lindberg et al., 2017); Minerva (Lindberg & Laine, 2018); Minicolon (Ayman et al., 2018); Nameless game (Alghamdi et al., 2016); Nameless game (Alghamdi, 2017); Nameless game (Yuliana et al., 2019)*; NanoDoc (Toukiloglou & Xinogalos, 2022); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Jeong, 2022); Pic2Program (Utesch et al., 2020); Pirate Plunder (Rose et al., 2019); RoboTIC (Schez-Sobrinio et al., 2020); Run Marco (Giannakoulas & Xinogalos, 2018); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020); ScratchJr and Lightbot (Rose et al., 2017); StoryCoder (Dietz et al., 2021); VR-OCKS (Segura et al., 2020)
Repetition structures 45 studies	4 Games (Lehat et al., 2014); ARcode (Mina et al., 2022); AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a); BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); ChIP (Sorrentino et al., 2017); cMinds (Tsalapatras, 2015); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018); CodeFruits (Goyal et al., 2017); CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodePlanet (Baek & Oh, 2019); CodeSpells (Esper et al., 2014); CodeTrasures (Nche et al., 2019); Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); Crocro 's Adventure (Forquesato, 2018); ENGAGE (Min et al., 2020); Kids Block Coding Game (Forquesato & Borin, 2018); Kodable (Karadeniz et al., 2014); Kodable vs Scratch Junior (Kandroudi & Bratitsis, 2016); Kodetu (Eguiluz et al., 2017); Kodetu (Guenaga et al., 2021); KODU vs Kodable (Fokides & Atsikpasi, 2017); Lighthbot Code hour (Yallihep & Kutlu, 2020); Minecraft Hour of Code (DEMİRKIRAN & TANSU HOCANIN, 2021); Minerva (Lindberg et al., 2017); Minerva (Lindberg & Laine, 2018); Minicolon (Ayman et al., 2018); Nameless game (Alghamdi et al., 2016); Nameless game (Alghamdi, 2017); NanoDoc (Toukiloglou & Xinogalos, 2022); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Jeong, 2022); Pic2Program (Utesch et al., 2020); Pirate Plunder (Rose et al., 2019); RoboTIC (Schez-Sobrinio et al., 2020); Run Marco (Giannakoulas & Xinogalos, 2018); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020); ScratchJr and Lightbot (Rose et al., 2017); StoryCoder (Dietz et al., 2021); VR-OCKS (Segura et al., 2020)

Table 4 (continued)

Concepts	Studies: game (reference)
Conditional structures 33 studies	4 Games (Lehat et al., 2014); ARcode (Mina et al., 2022); AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a); BOTS (Liu et al., 2017); ChIP (Sorrentino et al., 2017); cMinds (Tsalapatas, 2015); Code.org (Kalelioğlu, 2015); CodePlanet (Baek & Oh, 2019); CodeSpells (Esper et al., 2014); CodeTrasures (Nche et al., 2019); Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); Crocro 's Adventure (Forquesato, 2018); Daisy the Dinosaur vs Kodable (Pila et al., 2019); Kids Block Coding Game (Forquesato & Borin, 2018); Kodable (Karadeniz et al., 2014); Kodetu (Eguiluz et al., 2017); Kodetu (Guenaga et al., 2021); KODU vs Kodable (Fokides & Atsikpasi, 2017); Minecraft Hour of Code (DEMİRKIRAN & TANSU HOCANIN, 2021); Minerva (Lindberg et al., 2017); Minerva (Lindberg & Laine, 2018); Minicolon (Ayman et al., 2018); Nameless game (Alghamdi et al., 2016); Nameless game (Alghamdi, 2017); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Jeong, 2022); Pic2Program (Utesch et al., 2020); RoboTIC (Schez-Sobrinho et al., 2020); Scool (Steinmaurer et al., 2020); ScratchJr and Lightbot (Rose et al., 2017); VR-OCKS (Segura et al., 2020)
Functions/procedures 18 studies	4 Games (Lehat et al., 2014); ARcode (Mina et al., 2022); BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); ChIP (Sorrentino et al., 2017); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018); CodeFruits (Goyal et al., 2017); CodePlanet (Baek & Oh, 2019); CodeSpells (Esper et al., 2014); Coding Galaxy (Zhang et al., 2022a); Daisy the Dinosaur vs Kodable (Pila et al., 2019); KODU vs Kodable (Fokides & Atsikpasi, 2017); Lightbot Code hour (Yallihep & Kutlu, 2020); Penguin Go (Zhao & Shute, 2019); Pirate Plunder (Rose et al., 2019); ScratchJr and Lightbot (Rose et al., 2017); Software Kids (Ramírez-Rosales et al., 2016)
Variables 13 studies	ChIP (Sorrentino et al., 2017); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018); CodeFruits (Goyal et al., 2017); CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodeSpells (Esper et al., 2014); CodeTrasures (Nche et al., 2019); ENGAGE (Min et al., 2020); KODU vs Kodable (Fokides & Atsikpasi, 2017); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020); StoryCoder (Dietz et al., 2021)
OOP Concepts 3 studies	CodeSpells (Esper et al., 2014); OOP Serious Game (Lotfi & Mohammed, 2018); Software Kids (Ramírez-Rosales et al., 2016)
Recursion 2 studies	4 Games (Lehat et al., 2014); Pirate Plunder (Rose et al., 2019)

Empirical studies in Table 4 with an asterisk (*) have been classified based on the programming topics they address according to our estimate

Additionally, various publications advocate for the teaching of abstraction (33) and generalization (13) through the implementation of functions, whereas other studies in this domain involve exercises related to fundamental object-oriented programming (OOP) concepts like classes and objects (refer to Table 5). The game Pirate Plunder (Rose et al., 2019) introduces players to the concept of abstraction by teaching them how to identify duplicated code and eliminate it through the use of repetitions and procedures. Additionally, OOP Serious Game (Lotfi & Mohammed, 2018) instructs players on abstraction by engaging them

Table 5 Studies classification according to the CT skills covered

CT Skills	Studies: game (reference)
Algorithmic thinking 39 studies	ARcode (Mina et al., 2022)*; AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a); Bookworms (Korpi, 2014)*; BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); ChIP (Sorrentino et al., 2017)*; Code Venture (Hussain et al., 2015); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018)*; CodeFruits (Goyal et al., 2017)*; CodePlanet (Baek & Oh, 2019)*; CodeSpells (Esper et al., 2014)*; CodeTracasure (Nche et al., 2019)*; Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); Crocro 's Adventure (Forquesato, 2018); ENGAGE (Min et al., 2020); GrACE (Horn et al., 2016a); GrACE (Horn et al., 2016b); ILE (Ríos et al., 2020); Kodetu (Eguiluz et al., 2017); Kodetu (Guenaga et al., 2021); Ladybug (Fessakis et al., 2013); Ligthbot Code hour (Yallihep & Kutlu, 2020)*; Looking for Pets (Pessoa et al., 2019); Minerva (Lindberg & Laine, 2018)*; Minicolon (Ayman et al., 2018); Nameless game (Yuliana et al., 2019); NanoDoc (Toukiloglou & Xinogalos, 2022)*; Penguin Go (Zhao & Shute, 2019); Pic2Program (Utesch et al., 2020); Pirate Plunder (Rose et al., 2019); Run Marco (Giannakoulas & Xinogalos, 2018)*; ScratchJr and Lightbot (Rose et al., 2017); VR-OCKS (Segura et al., 2020); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b); Zoombinis (Asbell-Clarke et al., 2021)
Decomposition 33 studies	AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a); Bookworms (Korpi, 2014)*; BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); ChIP (Sorrentino et al., 2017)*; Code Venture (Hussain et al., 2015); Code.org (Kalelioğlu, 2015); CodeFruits (Goyal et al., 2017)*; CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodeSpells (Esper et al., 2014)*; CodeTracasure (Nche et al., 2019)*; Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); Crocro 's Adventure (Forquesato, 2018); ENGAGE (Min et al., 2020); ILE (Ríos et al., 2020); Kodetu (Eguiluz et al., 2017); Kodetu (Guenaga et al., 2021); Ligthbot Code hour (Yallihep & Kutlu, 2020)*; Looking for Pets (Pessoa et al., 2019); Minerva (Lindberg & Laine, 2018)*; Minicolon (Ayman et al., 2018); Nameless game (Yuliana et al., 2019); NanoDoc (Toukiloglou & Xinogalos, 2022); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Jeong, 2022); Pirate Plunder (Rose et al., 2019); Run Marco (Giannakoulas & Xinogalos, 2018)*; ScratchJr and Lightbot (Rose et al., 2017); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b); Zoombinis (Asbell-Clarke et al., 2021)
Abstraction 33 studies	ARcode (Mina et al., 2022)*; BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); ChIP (Sorrentino et al., 2017)*; cMinds (Tsalapatas, 2015); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018)*; CodeFruits (Goyal et al., 2017)*; CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodePlanet (Baek & Oh, 2019)*; CodeSpells (Esper et al., 2014)*; Coding Galaxy (Zhang et al., 2022a); Crocro 's Adventure (Forquesato, 2018); ENGAGE (Min et al., 2020)*; GrACE (Horn et al., 2016a); ILE (Ríos et al., 2020); Kids Block Coding Game (Forquesato & Borin, 2018); Kodetu (Eguiluz et al., 2017); Kodetu (Guenaga et al., 2021); Ligthbot Code hour (Yallihep & Kutlu, 2020)*; Looking for Pets (Pessoa et al., 2019); Nameless game (Yuliana et al., 2019); OOP Serious Game (Lotfi & Mohammed, 2018); Penguin Go (Zhao & Shute, 2019); Pic2Program (Utesch et al., 2020); Pirate Plunder (Rose et al., 2019); Run Marco (Giannakoulas & Xinogalos, 2018)*; ScratchJr and Lightbot (Rose et al., 2017); Software Kids (Ramírez-Rosaes et al., 2016); StoryCoder (Dietz et al., 2021); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b); Zoombinis (Asbell-Clarke et al., 2021)

Table 5 (continued)

CT Skills	Studies: game (reference)
Pattern recognition and Modularity 30 studies	ARcode (Mina et al., 2022); AutoThinking (Hooshyar et al., 2021a); Bookworms (Korpi, 2014)*; BOTS (Liu et al., 2017); ChIP (Sorrentino et al., 2017)*; cMinds (Tsalapatras, 2015); Code.org (Kalelioğlu, 2015); Codecombat (Ádámkó, 2018)*; CodeFruits (Goyal et al., 2017)*; CodePlanet (Baek & Oh, 2019)*; CodeSpells (Esper et al., 2014)*; CodeTracesure (Nche et al., 2019)*; Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); ILE (Ríos et al., 2020); Lighthbot Code hour (Yallihep & Kutlu, 2020)*; Looking for Pets (Pessoa et al., 2019); Minerva (Lindberg & Laine, 2018)*; Minicolon (Ayman et al., 2018)*; Nameless game (Yuliana et al., 2019); NanoDoc (Toukiloglou & Xinogalos, 2022); Patrony (Barrón-Estrada et al., 2022); Penguin Go (Zhao & Shute, 2019); Pic2Program (Utesch et al., 2020); Pirate Plunder (Rose et al., 2019); Run Marco (Giannakoulas & Xinogalos, 2018)*; StoryCoder (Dietz et al., 2021); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b); Zoombinis (Asbell-Clarke et al., 2021)
Problem solving 17 studies	BOTS (Liu et al., 2017); cMinds (Tsalapatras, 2015); Code.org (Kalelioğlu, 2015); CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodePlanet (Baek & Oh, 2019); Coding Galaxy (Zhang et al., 2022b); ENGAGE (Min et al., 2020); GrACE (Horn et al., 2016b); Kids Block Coding Game (Forquesato & Borin, 2018); Kodetu (Eguiluz et al., 2017); Ladybug (Fessakis et al., 2013); Nameless game (Alghamdi et al., 2016); Nameless game (Alghamdi, 2017); Penguin Go (Liu & Moon, 2021); Penguin Go (Liu & Jeong, 2022); Pic2Program (Utesch et al., 2020); RoboTIC (Schez-Sobrinio et al., 2020)
Generalization 13 studies	ARcode (Mina et al., 2022)*; ChIP (Sorrentino et al., 2017)*; Codecombat (Ádámkó, 2018)*; CodePlanet (Baek & Oh, 2019)*; CodeSpells (Esper et al., 2014)*; ENGAGE (Min et al., 2020)*; Kodetu (Guenaga et al., 2021); Lighthbot Code hour (Yallihep & Kutlu, 2020)*; OOP Serious Game (Lotfi & Mohammed, 2018); Penguin Go (Zhao & Shute, 2019)*; Run Marco (Giannakoulas & Xinogalos, 2018)*; ScratchJr and Lighthbot (Rose et al., 2017); Software Kids (Ramírez-Rosales et al., 2016)*
Debugging 13 studies	AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a); BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); Crocro's Adventure (Forquesato, 2018); ENGAGE (Min et al., 2020); Kids Block Coding Game (Forquesato & Borin, 2018); Ladybug (Fessakis et al., 2013); Penguin Go (Zhao & Shute, 2019); Pic2Program (Utesch et al., 2020); ScratchJr and Lighthbot (Rose et al., 2017)
Simulation 2 studies	AutoThinking (Malva et al., 2020); AutoThinking (Hooshyar et al., 2021a)

Empirical studies in Table 5 with an asterisk (*) have been classified based on CT skills they address according to our estimate

in activities that involve creating objects, categorizing objects based on their classes, and appropriately assigning them to their respective parent classes.

Moreover, problem-solving activities were observed in 17 empirical studies whereas testing and debugging activities were found in 13 studies. As an illustration, Liu et al. (2017) offered students debugging exercises within a game called BOTS, in order to introduce them to basic programming principles and examine their problem-solving behaviors. Furthermore, the game Crocro's Adventure (Forquesato,

2018) embraces a trial and error methodology by providing various means to demonstrate to the player the unfolding events and pinpointing their mistakes, resembling a debugging phase in programming. While debugging holds significant importance in CT (Grover & Pea, 2018; Shute et al., 2017), it is worth mentioning that there is a limited body of empirical research investigating this particular facet of CT.

Due to the fact that multiple researchers have described CT using various programming or computational concepts (Brennan & Resnick, 2012; Shute et al., 2017; Tang et al., 2020), it becomes clear that in numerous educational interventions utilizing programming games, the researchers evaluated students' programming abilities as an indication of their CT proficiency.

RQ2: What evaluation methods are utilized to gauge the effectiveness of educational games produced to teach young students CT and programming?

Evaluating the effectiveness of games is a crucial aspect of implementing the game-based approach. Researchers have employed a variety of instruments, often in combination, to gather data and evaluate how effective games created for inexperienced programmers are. A list of assessment methods currently utilized for the previously mentioned games is displayed in Fig. 4.

The most popular testing method to examine non-cognitive learning outcomes related to emotions and attitudes towards game-based learning methods is properly designed perceptions/attitudes questionnaires (35), as shown in Fig. 4. In-Game Evaluations (22) using learning analytics, or game logs, is another approach of evaluating the effectiveness of games in learners' acquisition of new knowledge that has been widely used by academics in recent years, as indicated by the data obtained from the included empirical studies. Furthermore, researchers employed knowledge pretests and posttests in also 22 experiments, where students' knowledge was examined before and after using the suggested game-based technique. In addition, 13 studies were identified that used the observation method, in which the teacher used informal assessment techniques during the teaching intervention, such as asking questions or simply monitoring the students' current performance on a task, while 11 studies used interviews to collect data.

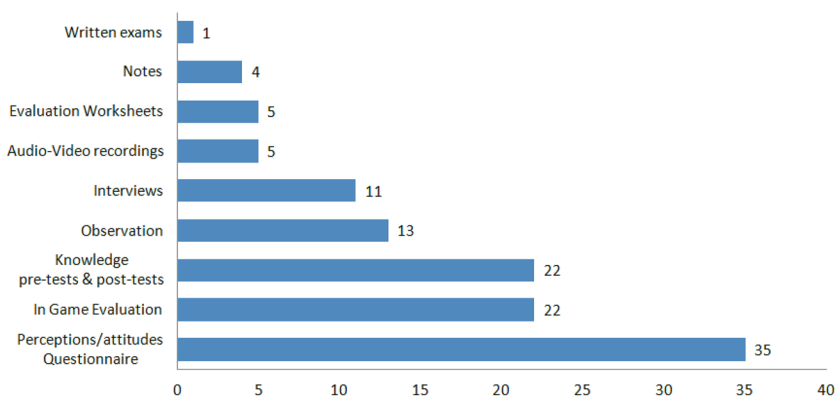


Fig. 4 Measurement tools used in the studies

Researchers prepared specially designed worksheets in five studies to examine if students can transfer the learned concepts to another field or to evaluate the knowledge they have acquired, while, respectively, in five studies, the children's actions were video-recorded and transcribed or players' audio recordings were carried out during the game experience. Finally, in four empirical studies, researchers took notes during the teaching intervention about the child's performance or current understanding, as well as any areas in which the student may require assistance (Sorrentino et al., 2017), whereas only 1 study used the written exam method, in which students were asked to write and trace some code in CodeSpells (Esper et al., 2014). Table 6 categorizes the studies based on the measurement instruments employed.

If we move away from the measurement tools to analyzing the evaluation methods utilized in the reviewed studies, we can see that both formative and summative assessments are used. Formative and summative assessments are two commonly utilized tools by teachers to evaluate student learning of new material and track their progress in acquiring new knowledge (Dixson & Worrell, 2016). Formative assessment refers to an ongoing process on collecting data during the learning process to support student growth, while summative assessment utilizes data to evaluate the extent of a student's knowledge upon completing a learning process (Dixson & Worrell, 2016).

The primary use of the knowledge pre-test/post-test method was for summative purposes, aiming to present proof of the extent to which students have achieved proficiency in the knowledge and CT skills covered in the learning unit after the intervention. Additionally, it sought to determine whether the intervention implemented through a game had a notable influence. Utilizing learning analytics has the capacity to offer valuable understandings into the process and results of students' CT learning (Guenaga et al., 2021). Learning analytics were used in the studies both for summative and formative purposes depending on their utilization and the objectives they served in the educational context. For instance, Lindberg et al. (2017) conducted a formative assessment involving 32 elementary students to evaluate Minerva's adaptation model. They gathered diverse game data and suggested enhancements for the adaptation model based on the findings of the evaluation. Conversely, in Forquesato's (2018) study, learning analytics were employed for summative purposes to investigate whether they could be a reliable alternative for determining if a user gained knowledge while engaging with a CT mobile game.

Furthermore, observation predominantly served as a formative assessment in the majority of the cases, aiming to offer assistance to students encountering challenges (Karadeniz et al., 2014), evaluate children's engagement and interest during gameplay (Forquesato, 2018), and assess their current performance on a task (Sorrentino et al., 2017). Additionally, interviews were primarily employed for formative assessment, with the main objectives of investigating challenges encountered during gameplay (Kalelioğlu, 2015; Dietz et al., 2021), delving deeper into students' motivation and intention to use the game (Dietz et al., 2021; Zhang et al., 2022a), exploring their perceptions of the game and the overall gaming experience (Malva et al., 2020; Zhang et al., 2022a; Pila et al., 2019), gathering their opinions on learning programming through a game (Demirkiran & Tansu Hocanin, 2021; Dietz et al., 2021; Lindberg et al., 2017), and exploring students'

Table 6 Studies Classification according to the measurement tools used

Instrument	Number of studies	Studies: game (reference)
Perceptions/attitudes Questionnaire (prior and/or after the intervention)	35	4 Games (Lehat et al., 2014); ARcode (Mina et al., 2022); AutoThinking (Hooshiyar et al., 2021a); BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); Code Venture (Hussain et al., 2015); CodePlanet (Baek & Oh, 2019); Coding Galaxy (Zhang et al., 2022a); Coding Galaxy (Zhang et al., 2022b); ENGAGE (Min et al., 2020); GrACE (Horn et al., 2016a); GrACE (Horn et al., 2016b); ILE (Ríos et al., 2020); Kodetu (Eguíluz et al., 2017); Kodetu (Guenaga et al., 2021); KODU vs Kodable (Fokides & Asikpasi, 2017); Lightbot Code hour (Yällihev & Kutlu, 2020); Looking for Pets (Pessoa et al., 2019); Minecraft Hour of Code (DEMIRKIRAN & TANSU HOCANIN, 2021); Minerva (Lindberg & Laine, 2018); Mimicron (Ayman et al., 2018); Nameless game (Alghamdi et al., 2016); Nameless game (Alghamdi, 2017); Nameless game (Yuliana et al., 2019); NanoDoc (Toukiloglou & Xinogalos, 2022); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Jeong, 2022); Pic2Program (Utesch et al., 2020); RoboTIC (Schez-Sobrinho et al., 2020); Run Marco (Giannakoulas & Xinogalos, 2018); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020); ScratchJr and Lightbot (Rose et al., 2017); Software Kids (Ramírez-Rosales et al., 2016); VR-OCKS (Segura et al., 2020)
In Game Evaluation	22	BOTS (Liu et al., 2017); BOTS (Zhi et al., 2018); Code.org (Kalelioğlu, 2015); CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); CodeTraceure (Nche et al., 2019); Coding Galaxy (Zhang et al., 2022a); Crocro 's Adventure (Forquesato, 2018); GrACE (Horn et al., 2016b); Kodable (Karadeniz et al., 2014); Kodetu (Eguíluz et al., 2017); Kodetu (Guenaga et al., 2021); Minerva (Lindberg et al., 2017); OOP Serious Game (Lofii & Mohammed, 2018); Penguin Go (Liu & Moon, 2021); Penguin Go (Liu & Jeong, 2021); Run Marco (Giannakoulas & Xinogalos, 2018); ScratchJr and Lightbot (Rose et al., 2017); VR-OCKS (Segura et al., 2020); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b); Zoombinis (Asbell-Clarke et al., 2021)

Table 6 (continued)

Instrument	Number of studies	Studies: game (reference)
Knowledge pre-tests & post-tests	22	ARcode (Mina et al., 2022); AutoThinking (Hooshyar et al., 2021a); Bookworms (Korpi, 2014); BOTS (Zhi et al., 2018); Code.org (Kalelioglu, 2015); CodeSpells (Esper et al., 2014); Coding Galaxy (Zhang et al., 2022b); Daisy the Dinosaur vs Kodable (Pila et al., 2019); GrACE (Horn et al., 2016a); GrACE (Horn et al., 2016b); ILE (Ríos et al., 2020); KODU vs Kodable (Fokides & Atsikpasi, 2017); Lighthot Code hour (Yallihp & Kutlu, 2020); Minerva (Lindberg et al., 2017); Minerva (Lindberg & Laine, 2018); Mimicolon (Ayman et al., 2018); Nameless game (Yuliana et al., 2019); Patrony (Barrón-Estrada et al., 2022); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Jeong, 2022); Pirate Plunder (Rose et al., 2019); Zoombinis (Asbell-Clarke et al., 2021)
Observation	13	ChIP (Sorrentino et al., 2017); cMinds (Tsalapatas, 2015); Codecombat (Ádámkó, 2018); CodeFruits (Goyal et al., 2017); Crocro 's Adventure (Forquesato, 2018); Kids Block Coding Game (Forquesato & Borin, 2018); Kodable (Karadeniz et al., 2014); Kodable vs Scratch Junior (Kandroudi & Bratitsis, 2016); Ladybug (Fessakis et al., 2013); Minecraft Hour of Code (Demirkiran & Tansu Hocanin, 2021); Nameless game (Alghamdi, 2017); Nameless game (Yuliana et al., 2019); Software Kids (Ramírez-Rosales et al., 2016)
Interviews	11	AutoThinking (Malva et al., 2020); Code.org (Kalelioglu, 2015); Coding Galaxy (Zhang et al., 2022a); Daisy the Dinosaur vs Kodable (Pila et al., 2019); Ladybug (Fessakis et al., 2013); Minecraft Hour of Code (DEMIRKIRAN & TANSU HOCANIN, 2021); Minerva (Lindberg et al., 2017); Minerva (Lindberg & Laine, 2018); Nameless game (Alghamdi et al., 2016); Pirate Plunder (Rose et al., 2019); StoryCoder (Dietz et al., 2021)
Audio-Video recordings	5	GrACE (Horn et al., 2016b); Ladybug (Fessakis et al., 2013); StoryCoder(Dietz et al., 2021); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b)
Evaluation Worksheets	5	Kodable (Karadeniz et al., 2014); KODU vs Kodable (Fokides & Atsikpasi, 2017); Run Marco (Giannakoulas & Xinogalos, 2018); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020)
Notes	4	ChIP (Sorrentino et al., 2017); Codecombat (Ádámkó, 2018); Crocro 's Adventure (Forquesato, 2018); Ladybug (Fessakis et al., 2013)
Written exams	1	CodeSpells (Esper et al., 2014)

perspectives on programming (Demirkiran & Tansu Hocanin, 2021; Lindberg & Laine, 2018; Lindberg et al., 2017).

Taking notes (of the child's current understanding) and audio–video recordings were utilized for formative assessment as well, capturing players' activities, facial expressions, gestures, and discussions during gameplay aimed to investigate how participants developed and applied CT throughout the gaming activity. Finally, the written exams method, was used only in one study and is typically a summative assessment while in five separate studies, researchers employed evaluation worksheets as a summative assessment method to ascertain whether students were able to apply the learned concepts to different problems (Steinmaurer et al., 2019, 2020) and to evaluate the extent of knowledge acquired by the children (Fokides & Atsikpasi, 2017; Giannakoulas & Xinogalos, 2018; Karadeniz et al., 2014).

In summary, the empirical studies analyzed employ a combination of summative and formative assessments, which may be utilized independently or concurrently. The choice between these approaches depends on the primary objectives of the teaching intervention and the intended goals specific to each study.

Finally, concerning the methodology employed, the majority of studies are focused on outcomes (outcome oriented), while a few studies concentrate on the process (process oriented). Specifically, among all the examined studies, eight of them compared game-based learning with the traditional learning method. These studies involved the division of students into an experimental group which uses the game in the learning process and a control group using a traditional learning approach, with a subsequent comparison of the learning outcomes between the two groups at the end of the intervention (Hooshyar et al., 2021a; Yallihep & Kutlu, 2020; Lindberg & Laine, 2018; Alghamdi et al., 2016; Segura et al., 2020; Lehat et al., 2014; Ríos et al., 2020; Barrón-Estrada et al., 2022).

The majority of the previous investigations were conducted in a formal learning environment. In particular, 34 teaching interventions (56%) were mostly employed in a formal environment, often in school labs, whereas 12 experiments (20%) were conducted in a non-formal setting. Finally, five studies (8%) were done in an informal learning environment, while ten studies (16%) did not indicate the experimental design. Figure 5 illustrates the learning context in which the empirical experiments included in this research were carried out. In our perspective, the prevalence of studies carried out within a formal setting highlights the significant interest within the research community in integrating these educational programming games as a tool for fostering CT in students in the learning process.

Another significant aspect of the evaluation process in the studies included is the number of participants, which is depicted in Fig. 6. With the exception of five (5) experiments in which more than 700 students participated, the mean sample size of participants in the remaining studies is 46. In particular, 38 empirical studies with up to 50 students were discovered, 13 studies with more than 50 and up to 100 students, and 5 studies with more than 100 and up to 200 students. In some studies, the limited number of participants is stated as an issue that prevents results from being generalized (Giannakoulas & Xinogalos, 2018; Forquesato 2018; Yallihep & Kutlu, 2020). The research of Eguiluz et al. (2017) had the biggest data set with

Fig. 5 Studies' learning context

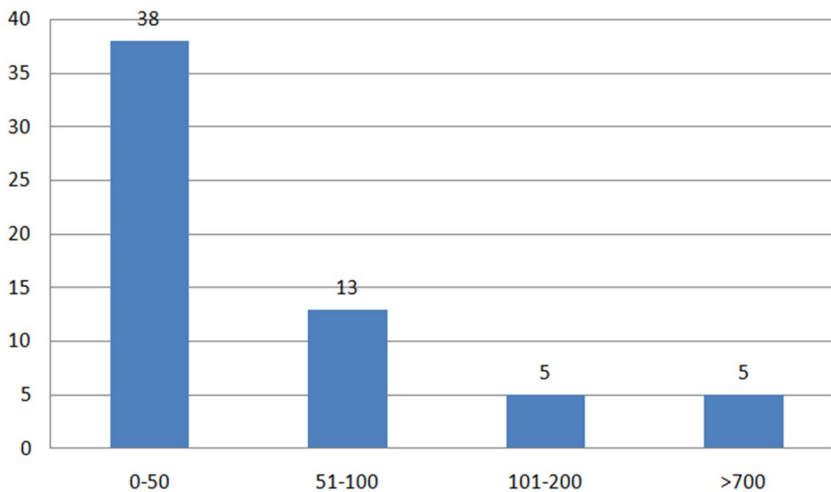
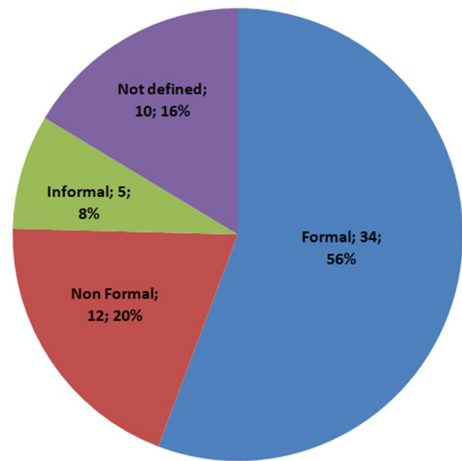


Fig. 6 Studies' participants sample size

3555 participants, meanwhile another study with data from 2040 participants using Codemonkey (Israel-Fishelson & Hershkovitz, 2020) was also recorded.

RQ3: Do educational games for CT and programming targeted to young students have a positive effect on learning the underlying concepts?

In order to answer the research question, the educational games reported in the 61 selected studies were categorized and analyzed. Regarding the studies' cognitive results, the categorization is determined by the skills, the programming concepts or the proposed topics they examine, as well as the evaluation results. Table 7 classifies the studies based on the cognitive outcomes they present. In addition to cognitive results, studies have been conducted to investigate additional factors such as student attitudes, affective outcomes, and the quality of the user interface design.

Table 7 Studies classification according to the cognitive results

Cognitive results	Game (reference) for studies with a negative effect	Game (reference) for studies with a neutral effect	Game (reference) for studies with a positive effect
CT Skills (25 studies)	Code.org (Kalelioğlu, 2015); Coding Galaxy (Zhang et al., 2022b) Kodetu (Guenaga et al., 2021); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020); ScratchJr & Lightbot (Rose et al., 2017);		AutoThinking (Hooshyar et al., 2021a); cMinds (Isalapatas, 2015); CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); Crocro's Adventure (Forquesato, 2018); GrACE (Horn et al., 2016a); GrACE (Horn et al., 2016b); ILE (Ríos et al., 2020); Ladybug (Fes-sakis et al., 2013); Looking for Pets (Pessoa et al., 2019); Micolon (Ayman et al., 2018); Nameless game (Yuliana et al., 2019); Patrony (Barrón-Estrada et al., 2022); Penguin Go (Zhao & Shute, 2019); Penguin Go (Liu & Moon, 2021); Pirate Plunder (Rose et al., 2019); StoryCoder (Dietz et al., 2021); Zoombinis (Rowe et al., 2017a); Zoombinis (Rowe et al., 2017b); Zoombinis (Asbell-Clarke et al., 2021)
Programming Concepts (17 studies)	Bookworms (Korpi, 2014); Minerva (Lindberg & Laïne, 2018);		ARcode (Mina et al., 2022); ChiP (Sorrentino et al., 2017); CodePlanet (Baek & Oh, 2019); CodeSpells (Esper et al., 2014); CodeTracsure (Nche et al., 2019); Daisi the Dinosaur vs Kodable (Pila et al., 2019); Kodable (Karadeniz et al., 2014); Kodable vs Scratch Junior (Kandroudi & Brattis, 2016); KODU vs Kodable (Fokides & Atsikpasi, 2017); Lightbot Code Hour (Yallihp & Kutlu, 2020); Minecraft Hour of Code (DEMIRKIRAN & TANSU HOCANIN, 2021); Nameless game (Alghamdi et al., 2016); Pic2Program (Utesch et al., 2020); Run Marco (Gian-nakoulas & Xinogalos, 2018); VR-OCKS (Segura et al., 2020)
Object Oriented Concepts (1 study)			Software Kids (Ramírez-Rosales et al., 2016)

Table 7 (continued)

Cognitive results	Game (reference) for studies with a negative effect	Game (reference) for studies with a neutral effect	Game (reference) for studies with a positive effect
Supports (3 studies)			BOTS (Zhi et al., 2018); NanoDoc (Toukiloglou & Xinogalos, 2022); Penguin Go (Liu & Jeong, 2022)
Debugging (1 study)			BOTS (Liu et al., 2017)
Number of studies	0	8	39

Table 8 classifies the studies according to these additional factors. Because many studies examine several parameters at the same time, such as participants' cognitive improvements in CT skills or understanding of certain programming concepts after the instructional intervention, or their attitudes toward learning programming both before and after using an educational game, the commentary and analysis of their results are reported in both Tables 7 and 8.

The majority of the studies (39) that investigated the effectiveness or the impact of games on the acquisition of the proposed topics reported positive results. Also, 8 studies found no significant improvement after the teaching intervention, whereas no study demonstrated a negative effect.

Table 8 shows that the affective outcomes and students' attitudes regarding learning programming, particularly utilizing a game-based method, are generally positive. The majority of studies that examined students' views on learning programming through games found beneficial results. The research discovered that incorporating programming games in the teaching process is an effective way to encourage and engage young students in CT activities. In our estimate, this approach also helps boost their self-assurance (Esper et al., 2014; Zhang et al., 2022b) and initiates their integration into the field of computer science.

Eguiluz et al. (2017) and Guenaga et al. (2021) employed questionnaires to collect data on students' personal characteristics such as age, gender, educational level, prior programming expertise, and attitudes towards technology, in two empirical studies employing Kodetu. The extent to which students' personal characteristics impact their performance in grasping CT skills and their learning progress constituted one of the study issues addressed in these investigations. Additionally, Guenaga et al. (2021) created a specialized tool with the objective of interpreting Kodetu logs. This tool facilitates the examination of the log data enabling the generation of metrics that aid in the comprehension of the CT acquisition process and outcomes in novice programmers. Authors concluded that age and gender are variables that affect CT performance and furthermore, combining Learning Analytics and other evaluation techniques can help alleviate the burden on teachers when managing a substantial number of students, particularly in obtaining data that would be strenuous to gather manually.

The objective of Forquesato's (2018) study was to assess the efficacy of Learning Analytics as a method for determining whether user acquired knowledge while playing a mobile game centered on CT. The teaching intervention involved allowing 29 children, aged between 7 and 11, to play the game on a smart phone for around 50 min on two separate days. The results indicated that the game is a valuable tool for promoting the development of CT skills in children as young as seven. Additionally, the study affirmed the effectiveness of analytics as a technique for recognizing the acquisition of CT knowledge.

Zhao and Shute (2019) assessed the cognitive and emotional impacts of participating in the video game Penguin Go as well as the influence of a specific game component, known as "constraints," which limits the number of blocks used in a solution. Pre- and post-tests were administered to assess students' CT abilities, while a survey was utilized to assess students' views regarding CS. The findings indicated that playing Penguin Go significantly improved students' CT skills, whereas the

Table 8 Studies classification according to the affective results/attitudes/aesthetics

Dimensions	Game (reference) for studies with a negative effect	Game (reference) for studies with a neutral effect	Game (reference) for studies with a positive effect
Attitudes towards technology/CS / IT (5 studies)	Penguin Go (Zhao & Shute, 2019) (additional constraints condition)	BOTS (Zhi et al., 2018); Lighthot Code Go (Zhao & Shute, 2019)	Software Kids (Ramírez-Rosales et al., 2016)
Attitude towards learning CT/ Programming/ Debugging activities (9 studies)			AutoThinking (Hooshyar et al., 2021a); Bookworms (Korpi, 2014); BOTS (Liu et al., 2017); Code.org (Kalelioğlu, 2015); Coding Galaxy (Zhang et al., 2022b); Minecraft Hour of Code (Demirkiran & Tansu Hocanin, 2021); NanoDoc (Toukiloglou & Xinogalos, 2022); RoboTIC (Schez-Sobrino et al., 2020); Scool (Steinmaurer et al., 2019)
Usability of the Game / Acceptance (8 studies)		Coding Galaxy (Zhang et al., 2022a)	4 Games (Lehat et al., 2014); Ladybug (Fessakis et al., 2013); Looking for Pets (Pessoa et al., 2019); RoboTIC (Schez-Sobrino et al., 2020); Run Marco (Giannakoulas & Xinogalos, 2018); StoryCoder (Dietz et al., 2021); VR-OCKS (Segura et al., 2020)
User Interface / Game Mechanics (8 studies)			CodeFruits (Goyal et al., 2017); Coding Galaxy (Zhang et al., 2022a); ENGAGE (Min et al., 2020); Looking for Pets (Pessoa et al., 2019); Nameless game (Yuliana et al., 2019); NanoDoc (Toukiloglou & Xinogalos, 2022); Pic2Program (Uttesch et al., 2020); VR-OCKS (Segura et al., 2020)

Table 8 (continued)

Dimensions	Game (reference) for studies with a negative effect	Game (reference) for studies with a neutral effect	Game (reference) for studies with a positive effect
Fun – Entertainment (5 studies)			Codecombat (Ádámkó, 2018); Kids Block Coding Game (Forquessato & Borin, 2018); Kodable (Karadeniz et al., 2014); Looking for Pets (Pessoa et al., 2019); Software Kids (Ramírez-Rosales et al., 2016)
Enjoyment –Satisfaction – Interest (13 studies)		GrACE (Horn et al., 2016b); Penguin Go (Liu & Jeong, 2022)	AutoThinking (Malva et al., 2020); Code Venture (Hussain et al., 2015); Codecombat (Ádámkó, 2018); CodePlanet (Baek & Oh, 2019); Coding Galaxy (Zhang et al., 2022a); GrACE (Horn et al., 2016a); Ladybug (Fessakis et al., 2013); Looking for Pets (Pessoa et al., 2019); OOP Serious Game (Lofli & Mohamed, 2018); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020)

Table 8 (continued)

Dimensions	Game (reference) for studies with a negative effect	Game (reference) for studies with a neutral effect	Game (reference) for studies with a positive effect
Engagement – Motivation (19 studies)			ARcode (Mina et al., 2022); AutoThinking (Malva et al., 2020); cMinds (Tsalapatas, 2015); Codecombat (Adámikó, 2018); CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); CodeSpells (Esper et al., 2014); CodeTraceure (Nche et al., 2019); Coding Galaxy (Zhang et al., 2022a); ENGAGE (Min et al., 2020); ILE (Ríos et al., 2020); Kids Block Coding Game (Forquesato & Borin, 2018); Kodable vs Scratch Junior (Kandroudi & Bratitsis, 2016); Ladybug (Fessakis et al., 2013); Looking for Pets (Pessoa et al., 2019); Minerva (Lindberg & Laine, 2018); Scool (Steinmaurer et al., 2019); Scool (Steinmaurer et al., 2020); Software Kids (Ramírez-Rosales et al., 2016); StoryCoder (Dietz et al., 2021)
Number of studies	1	6	60

added restrictions had negligible impact on their learning. In addition it was found that although the game did not alter the students' perception of computer science, the supplementary limitations in the game had an adverse effect on their attitude towards the subject.

Based on these results, Liu and Jeong (2022) conducted a study using the game Penguin Go without the feature of limitations on the number of blocks, examining the impact of supports in CT skills cultivation. The study aimed to investigate how in-game support affects students' CT skills and their ability to transfer these skills to various problem-solving contexts beyond the game. The findings indicated that the students' performance in the game was not enhanced by the use of supports. However, the students' CT skills were found to have significantly improved at the near transfer level, but not at the far transfer level.

Toukiloglou and Xinogalos (2022) presented a serious game called NanoDoc that aims to teach sequencing and loops to primary school students that incorporates adaptive supports with working examples, providing some preliminary results on the game's usage by students and an evaluation of the user's experience with the game. After participating in a 45-min play session in the school's computer lab under a teacher's supervision, 85 elementary school children from the same school, aged 10 to 12, anonymously responded to an online questionnaire. The students expressed a distinct liking towards learning programming through games and displayed an overall positive attitude towards the game's mechanics, features, and support system.

An experimental research on primary school pupils was done by Hooshyar et al. (2021a) to evaluate the efficacy of Autothinking, an adaptive game, in comparison to a conventional learning approach. The authors utilized a pre-post testing method to assess the CT knowledge of students and pre-post questionnaires to measure students' attitude towards learning CT. According to the findings, the game was more successful than the traditional approach in improving students' CT abilities, and also had a greater positive impact on their attitudes toward CT.

A similar strategy was used in a research by Yallihep and Kutlu (2020), which used a pretest–posttest methodology to determine how students' knowledge of programming concepts and attitudes about technology is affected by mobile serious games. The researchers formed two sets of students, where the experimental group utilized the Lightbot game while the control group followed the traditional teaching approach. Both before and after the research, both groups underwent an achievement test and a survey. Results showed that experiment group pupils significantly improved their knowledge on programming fundamentals in contrast to the other group and that there was no discernible improvement in the students' attitudes toward the "Information Technology".

Barrón-Estrada et al. (2022) presented Patrony, an educational game app for mobile devices, designed to encourage the growth of CT among 8–10 year old children by instructing and exercising their ability to identify patterns. In order to compare the efficiency of learning pattern recognition using the Patrony tool with a conventional learning technique, the authors conducted an experiment with two groups of students using pre-post test evaluation. The findings validated that teaching pattern recognition through the app was more efficient than the conventional learning method.

During their experiment investigation, Kandroudi and Bratitsis (2016) observed two groups of two children. The students first played the game Kodable to introduce themselves to a new programming concept, and then they were requested to create their personal story with ScratchJr using similar game instructions. The sequence of interaction with the two environments was reversed on some days. According to the observation of students, the authors concluded that utilizing the game Kodable initially followed by ScratchJr, facilitated students' comprehension of basic programming concepts.

Karadeniz et al. (2014) conducted another case study that relied on observations. The researchers studied how experiential games and the game Kodable taught fundamental programming concepts to 5-year-old students. According to the results, students had a positive experience playing Kodable and demonstrated proficiency in both the game's activities and the assessment sheets that were designed for the purpose.

In a formative mixed-method review of Minerva, Lindberg and Laine (2018) compared the learning outcomes of 32 Korean sixth graders who played the game against 32 sixth graders who studied the same topics using handouts. Researchers gathered information over a two-week period using a questionnaire, interviews, and a Code.org-inspired memory test that assessed how well players remembered information delivered in the game as opposed to on paper. According to the findings, there was no significant difference in memory retention between the two groups whereas the game was found to increase student engagement.

Demrkiran and Tansu Hocanin (2021) conducted a study to assess the perceptions of fifth-grade elementary school students regarding the idea of programming, using the Minecraft Hour of Code. Sixty-three pupils were requested to complete the 14 Minecraft Hour of Code challenges, while the researcher observed and encouraged them throughout the process. Following the completion of the specified task in a computer lab, researchers collected information by utilizing a questionnaire and conducting interviews. The findings revealed that the game helped students grasp programming concepts and develop a favorable outlook on programming, despite any preconceived notions they may have had.

In another research, Pessoa et al. (2019) tested an early version of the game "Looking for Pets" to assess usability and game flow from the user's perspective. The tests were conducted in two stages. The usability was confirmed in the first phase with three experienced game players using the Nielson heuristic assessment, while the MEEGA + KIDS test, which assesses twelve categories, was used in the second step, with ten kids aged 10–13 years. In terms of usability, the game obtained an average score over 60% in the majority of metrics, and in terms of game flow, the majority of the assessed criteria received excellent scores, indicating that the game's elements are suitable as an appealing tool for the younger audience. Furthermore, according to preliminary tests, the game helps with the development of CT abilities like breaking a problem into smaller pieces, recognizing patterns in the data, and coming up with a set of steps to reach the solution, demonstrating its potential to work in the educational sphere and provide significant positive effects for learning.

In Table 9 a classification of the studies' results based on some different dimensions they examined is presented.

Table 9 Studies classification, according the results regarding other dimensions

Other dimensions	No studies	Studies: game (reference)
Collaboration vs independent learning	1	GrACE (Horn et al., 2016a)
Difficulty of Game	3	CodeMonkey (Israel-Fishelson & Hershkovitz, 2019); CodeMonkey (Israel-Fishelson & Hershkovitz, 2020); GrACE (Horn et al., 2016b)
Evaluate Adaptation Model of Game	2	Kodetu (Eguíluz et al., 2020); Minerva (Lindberg et al., 2017)

GrACE, a game that promotes CT through Procedural Content Generation (PCG) and collaboration, was the subject of an experimental pilot research by Horn et al. (2016a). Long-term objectives of the authors included investigating the educational value of PCG integration in educational games, both in collaboration and in individual conditions during gameplay, and developing a suitable puzzle design to promote CT in young pupils. Forty-three students participated in the experimental teaching, and they were separated into two groups with one of the groups working individually and the other cooperatively. The students were given pre-tests, pre-questionnaires, and assigned specific tasks in the laboratory. The study's primary results revealed that collaboration had a lesser impact on the students' improvement, suggesting that more investigation is necessary to improve collaboration in games and that PCG improved learning gain, in the group of students who worked individually.

Israel-Fishelson & Hershkovitz (2019), investigated students' micro-persistence in learning CT using a Learning Analytics technique in CodeMonkey. Being persistent in finding the optimal way to finish a task is the behavior of micro-persistence. The authors examined only the correct solution attempts made by the 119 elementary school students who used the learning platform. The findings revealed that the difficulty of the task has an impact on students' micro-persistence, and factors such as the complexity of the task and a support provided by a teacher, may be more effective in explaining persistence than individual traits.

In another study, Lindberg et al. (2017) tested Minerva with 33 South Korean sixth-graders with the objective of analyzing Minerva's adaptation model. The researchers used a post-test survey, interviews, and log data, to gather appropriate information about players and how the game adjusts its level complexity to the player's profile. The evaluation results produced suggestions for Minerva's adaptation model and highlighted a room for improvement, especially in the way educational content is presented to players.

Due to the significant increase in the amount of empirical research incorporating games into the learning process, along with promising initial findings on their effectiveness in promoting CT skills, it is suggested that future studies of a similar nature should strive to include a larger pool of participants. This would enable a more reliable generalization of the learning outcomes. Furthermore, we hold the belief that to enhance our comprehension of the educational impact of games, it is crucial to employ a diverse range of tools. Consequently, we consider the use

of different assessment measures to gauge the effectiveness of games in the analyzed empirical studies as a step in the right direction.

Conclusions

Conclusions

The main objective of this study was to identify the most recent empirical research on the application of educational games for teaching young students' CT skills and basic programming concepts, as well as to explore the effects of these games on the cultivation of the aforementioned skills. To that end, we examined 61 empirical studies concentrating on the primary school age group that were chosen after searching the literature over the previous twelve years (2010 until September 2022).

In terms of the programming concepts covered by the identified studies, it was discovered that the most extensively covered notions by academics and game developers are fundamental programming concepts such as sequence, selection, and iteration, whilst OOP concepts and recursion are the least covered. Maybe this is because these notions need a high degree of abstraction, which makes it challenging to teach them to elementary school students who have not yet mastered abstract thought (Zaharija et al., 2013). As highlighted by Relkin et al. (2021), taking into account these challenges and other developmental aspects is of utmost importance when developing educational programs specifically targeted at teaching CT to young children (Papadakis, 2021). In addition, regarding the CT skills that games seek to cultivate in learners, it appears that the majority of them aim to develop algorithmic thinking, decomposition, abstraction, as well as pattern recognition and modularity, which are skills recognized by numerous researchers (Brennan & Resnick, 2012; Atmazidou & Demetriadis, 2016; Shute et al., 2017; Grover & Pea, 2018), as fundamental elements of CT. Additionally, in terms of debugging, it has been observed that there is a scarcity of empirical studies focusing specifically on this aspect of CT.

Moreover, academics have used a range of tools, frequently in tandem, to collect data and assess the effectiveness of games designed for inexperienced programmers. Most often, perceptions/attitudes questionnaires are used to investigate typical emotional learning results, especially for the purpose of studying the incentives and perspectives related to learning techniques that utilize games. It was also found that in-game evaluations utilizing learning analytics, or game logs, was another popular technique to measure the efficacy of games in learners' acquisition of new knowledge, while researchers in several studies preferred employing pre-test and post-test procedures. Observations during game play, interviews with players, and even relevant worksheets constitute some of the extra data gathering methods employed by the researchers. Additionally, Learning Analytics approaches which may have the potential to provide students with personalized feedback on their performance and compare it with that of the entire group can help alleviate the burden on teachers when managing a substantial number of students, particularly in obtaining data that would be strenuous to gather manually (Guenaga et al., 2021). Although it appears that learning analytics techniques is a useful way to recognize the acquisition of CT,

some details are unlikely to be captured using this approach to gather data alone (Forquesato, 2018). Therefore, it is necessary to use additional techniques like observing and conducting interviews, which all together could form “integrated assessment systems” providing an overall view of student understanding (Guenaga et al., 2021). Consequently, we believe that in order to improve our understanding of the educational influence of games that aim to cultivate CT skills, it is essential to utilize a wide variety of tools to measure their effectiveness.

Shifting our focus from the measurement tools to the evaluation methods employed in the analyzed studies, we observe the utilization of both formative and summative assessments. The selection of these approaches is contingent upon the main objectives of the teaching intervention and the particular goals set for each study. Moreover, in relation to the methodology employed, it was observed that the majority of studies have a primary focus on outcomes, adopting an outcome-oriented approach. On the other hand, a smaller number of studies center their attention on the process itself, adopting a process-oriented perspective.

The majority of the studies that looked into the efficiency or impact of games on the acquisition of the suggested topics (39 out of 47) reported positive results, demonstrating that educational programming games can assist elementary school students in developing CT abilities or understanding fundamental programming concepts. Moreover, no study showed a negative impact, whereas 8 investigations reported no appreciable change following the instructional intervention. The results of this research unequivocally demonstrated a strong positive impact of using an educational game as a teaching tool for CT skills and basic programming concepts to elementary school kids. Students established a good attitude toward programming through game-based activities, and the utilization of programming games proved to be a powerful motivator for young children in engaging them in CT activities. Furthermore, it is advised that in future studies of a similar nature, which examine the educational impact of games specifically designed to foster CT skills, researchers should make an effort to involve a broader range of participants. This would facilitate a more dependable generalization of the learning outcomes.

Nevertheless, one aspect that appears to be absent from the relevant research included in this study is player collaboration during the game. There was just one publication that examined the aspect of collaboration, indicating that future research may look into how participants communicate during the game and how this impacts their efficacy in obtaining new knowledge. Additionally, the research findings show that another area that has not been thoroughly examined thus far is the adaptation of the game’s level to the player’s profile, as well as the manner in which the educational information is delivered to the player. Taking these factors into account, we can infer that there is still opportunity for progress in these domains and that further research is necessary to tap into these possibilities.

Although this study has some limitations that are presented in the next subsection, it also has a distinct contribution in the field. Specifically, this study could be used as a basis for further experimental research on the impact of games on programming and CT skills among students. This could also include exploring the acceptance of using such games for teaching programming and acquiring CT skills, among both teachers and students. Also, this study may be helpful to educators who

often express a lack of knowledge on effectively integrating a game-based learning strategy into their lessons or to policymakers looking for methods to incorporate CT education into the curriculum. Finally, the SLR recorded various important approaches, such as collaboration and adaptation, which are missing from existing games and has highlighted areas of future work that can have a unique impact on the design and more effective application of educational games for cultivating CT skills.

Limitations

There are some limitations in the provided literature review that need to be pointed out. To start with, it is possible that there are research papers that were not incorporated into the articles we have selected because some digital research databases could not be accessed. As a result, certain publications could not be collected or analyzed. Furthermore, a considerable number of empirical studies involved a limited number of individuals in the intervention process, thus making it challenging to extend the efficacy of these investigations to a broader population.

Future work

In line with the research questions of this paper, emphasis is placed on investigating CT skills and programming concepts, as well as the methodology and the results of the relevant studies. In the analysis and discussion of results, the games on which the studies were based are listed, as we believe that the type and the elements of a game may have an effect on learning outcomes. However, the study of the specific characteristics of the games was not included in the objectives of this SLR. Although the majority of the related studies demonstrated positive cognitive and affective results, it is crucial for future research to conduct an SLR that explores what types of educational games for programming and CT exist, what their distinct characteristics are, as well as their suitability as tools for cultivating CT skills considering children's age.

Additionally, it would be intriguing for researchers to look into unresolved concerns and missing characteristics from the games being examined, such as collaborative play, adaptation to learner profiles and assistance for developing personalized lesson plans. Afterwards, they can integrate these characteristics into an improved educational game platform, and conduct additional experiments to investigate how these features affect the learning process.

Funding Open access funding provided by HEAL-Link Greece. This research received no external funding.

Data availability All data are included in the article.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical statements We hereby declare that this manuscript is the result of our independent creation under the reviewers' comments. Except for the quoted contents, this manuscript does not contain any research achievements that have been published or written by other individuals or groups. We are the only authors of this manuscript. The legal responsibility of this statement shall be borne by us.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Ádámkó, É. (2018). Gamification in programming—A short introductory session in programming with online games. *International Journal of Engineering and Management Sciences*. <https://doi.org/10.21791/ijems.2018.5.2>
- Alghamdi, M., Al-Jumeily, D., Hussain, A., & Dermott, Á. mac. (2016). A proposed tutoring system to support the learning of computer programming: Case Study. *Proceedings of 2016 SAI Computing Conference, SAI 2016*. <https://doi.org/10.1109/SAI.2016.7556082>
- Alghamdi, M. Y. (2017). *Supporting the learning of computer programming in an early years education*. Doctoral thesis, Liverpool John Moores University
- Asbell-Clarke, J., Rowe, E., Almeda, V., Edwards, T., Bardar, E., Gasca, S., Baker, R. S., & Scruggs, R. (2021). The development of students' computational thinking practices in elementary- and middle-school classes using the learning game, Zoombinis. *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2020.106587>
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75B, 661–670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Atmatzidou, S., & Demetriadis, S. (2017). A didactical model for educational robotics activities: A study on improving skills through strong or minimal guidance. *Advances in Intelligent Systems and Computing*, 560, 58–72. https://doi.org/10.1007/978-3-319-55553-9_5
- 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. https://doi.org/10.1007/978-3-030-02762-9_9
- Baek, J., & Oh, G. (2019). Development of a Puzzle Game to Learn Coding for Elementary Students. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11786 LNCS. https://doi.org/10.1007/978-3-030-30033-3_21
- Barrón-Estrada, M. L., Zatarain-Cabada, R., Romero-Polo, J. A., & Monroy, J. N. (2022). Patrony: A mobile application for pattern recognition learning. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-021-10636-7>
- Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-021-10700-2>
- Bebras Homepage, <https://www.bebas.org/>, last accessed 2023/01/15.
- Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, 24(5), 628–647. <https://doi.org/10.1007/s10956-015-9552-x>

- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Paper presented at the annual American educational research association meeting*, Vancouver, BC, Canada (p. 25).
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages: A way to learn programming principles. *Education and Information Technologies*, 2(1), 65–83. <https://doi.org/10.1023/A:1018636507883>
- Budiyanto, C. W., Shahbodin, F., Umam, M. U. K., Isnaini, R., Rahmawati, A., & Widiastuti, I. (2021). Developing computational thinking ability in early childhood education: The influence of programming toy on parent-children engagement. *International Journal of Pedagogy and Teacher Education*, 5(1), 19–25. <https://doi.org/10.20961/ijpte.v5i1.44397>
- Constantinou, V., & Ioannou, A. (2018). Development of Computational Thinking Skills through Educational Robotics. *European Conference on Technology Enhanced Learning*.
- Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. In *Journal of Science Education and Technology* (Vol. 28, Issue 6, pp. 651–676). Springer. <https://doi.org/10.1007/s10956-019-09799-3>
- Demirkiran, M. C., & Tansu Hocanin, F. (2021). An investigation on primary school students' dispositions towards programming with game-based learning. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-021-10430-5>
- Dietz, G., Le, J. K., Tamer, N., Han, J., Gweon, H., Murnane, E. L., & Landay, J. A. (2021). Storycoder: Teaching computational thinking concepts through storytelling in a voice-guided app for children. *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3411764.3445039>
- Dixson, D. D., & Worrell, F. C. (2016). Formative and summative assessment in the classroom. *Theory into Practice*, 55(2), 153–159. <https://doi.org/10.1080/00405841.2016.1148989>
- Djelil, F., Albouy-Kissi, A., Albouy-Kissi, B., Sanchez, E., & Lavest, J. (2016). Microworlds for learning object-oriented programming: Considerations from research to practice. *The Journal of Interactive Learning Research*, 27, 247–266.
- Eguiluz, A., Garaizar, P., & Guenaga, M. (2018). An Evaluation of Open Digital Gaming Platforms for Developing Computational Thinking Skills. In *Simulation and Gaming*. InTech. <https://doi.org/10.5772/intechopen.71339>
- Eguiluz, A., Guenaga, M., Garaizar, P., & Olivares-Rodriguez, C. (2017). Exploring the progression of early programmers in a set of computational thinking challenges via clickstream analysis. *IEEE Transactions on Emerging Topics in Computing*, 6750(c), 1–6. <https://doi.org/10.1109/TETC.2017.2768550>
- Esper, S., Foster, S. R., Griswold, W. G., Herrera, C., & Snyder, W. (2014). CodeSpells: Bridging educational language features with industry-standard languages. *ACM International Conference Proceeding Series, 2014-November*(November). <https://doi.org/10.1145/2674683.2674684>
- Ríos, F. J. M., Cabada, R. Z., Estrada, M. L. B., & Vara, J. F. (2020). An intelligent learning environment for computational thinking. *Computacion y Sistemas*. <https://doi.org/10.13053/CYS-24-3-3480>
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers and Education*. <https://doi.org/10.1016/j.compedu.2012.11.016>
- Fessakis, G., Komis, V., Dimitracopoulou, A., & Prantsoudi, S. (2019). Overview of the computer programming learning environments for primary education. *Review of Science, Mathematics and ICT Education*, 13(1), 7–33. <https://doi.org/10.26220/rev.3140>
- Fokides, E., & Atsikpasi, P. (2017). Redefining the framework for teaching programming to primary school students: Results from three pilot projects. *British Journal of Education, Society & Behavioural Science*, 20(3), 1–11. <https://doi.org/10.9734/bjesbs/2017/33520>
- Forquesato, L. E. T., & Borin, J. F. (2018). *Kids Block Coding Game: A game to introduce programming to kids*. In *Anais do XXVI Workshop sobre Educação em Computação. Porto Alegre: SBC*. <https://doi.org/10.5753/wei.2018.3502>
- García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. In *Computers in Human Behavior* (Vol. 80, pp. 407–411). Elsevier Ltd. <https://doi.org/10.1016/j.chb.2017.12.005>
- Giannakoulas, A., & Xinogalos, S. (2020). A review of educational games for teaching programming to primary school students. In: *Handbook of research on tools for teaching computational thinking in P-12 education* (pp. 1–30). IGI Global.

- Giannakoulas, A., Terzopoulos, G., Xinogalos, S., & Satratzemi, M. (2020). A Proposal for an Educational Game Platform for Teaching Programming to Primary School Students. *TECH-EDU*.
- Giannakoulas, A., & Xinogalos, S. (2018). A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-018-9702-x>
- Goyal, S., Chopra, S., & Mohanan, D. (2017). CodeFruits: Teaching Computational Thinking Skills Through Hand Gestures. *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play - CHI PLAY '17 Extended Abstracts*, 291–298. <https://doi.org/10.1145/3130859.3131335>
- Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. In *Educational Researcher* (Vol. 42, Issue 1, pp. 38–43). <https://doi.org/10.3102/0013189X12463051>
- Grover, S., & Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. In S. Sentance, E. Barendsen & C. Schulte (Eds.). *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 19–38). London: Bloomsbury Academic. <https://doi.org/10.5040/9781350057142.ch-003>
- Guenaga, M., Eguíluz, A., Garaizar, P., & Gibaja, J. (2021). How do students develop computational thinking? Assessing early programmers in a maze-based online game. *Computer Science Education*, 31(2), 259–289. <https://doi.org/10.1080/08993408.2021.1903248>
- Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M., & Lim, H. (2021a). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2020.106575>
- Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G. J., Wang, M., Lim, H., & Delev, D. (2021b). 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>
- Horn, B., Folajimi, Y., Hoover, A. K., Smith, G., Barnes, J., & Hartevelde, C. (2016b). Opening the black box of play: Strategy analysis of an educational game. *CHI PLAY 2016 - Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. <https://doi.org/10.1145/2967934.2968109>
- Horn, B., Clark, C., Strom, O., Chao, H., Stahl, A. J., Hartevelde, C., & Smith, G. (2016a). Design insights into the creation and evaluation of a computer science educational game. *SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. <https://doi.org/10.1145/2839509.2844656>
- Hour of Code Homepage, <https://hourofcode.com/gr>, last accessed 2023/01/15.
- 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 and Education*. <https://doi.org/10.1016/j.compedu.2018.07.004>
- Hussain, A. J., Fergus, P., Al-jumeily, D., Pich, A., & Hind, J. (2015). Teaching primary school children the concept of computer programming. *International Conference on Developments of E-Systems Engineering (DeSE), 2015*, 180–184. <https://doi.org/10.1109/DeSE.2015.57>
- Israel-Fishelson, R., & Hershkovitz, A. (2020). Persistence in a game-based learning environment: The case of elementary school students learning computational thinking. *Journal of Educational Computing Research*. <https://doi.org/10.1177/0735633119887187>
- Kaleliolu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2015.05.047>
- Kandroudi, M., & Bratitsis, T. (2016). Teaching programming at young ages with mobile devices through the Kodable game and ScratchJr: A case study. In T. A. Mikropoulos, A. Tsiara, P. Chalki (ed.), *Proceedings of the 8th Panhellenic Conference "Informatics Teaching"* (pp. 133–140). Ioannina: ETPE. September 23–25, 2016.
- Karadeniz, S., Samur, Y., & Özden, M.Y. (2014). Playing with Algorithms to Learn Programming: A Case Study on 5 Years Old Children, *9th International Conference on Information Technology and Applications (ICITA2014)*, 1 - 4 July, 2014, Sydney, Australia , <http://www.icita.org>.
- Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, 9, 522–531.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele University, UK and National ICT Australia*, 33, 1–26.
- Korpi, J. (2014). *Teaching programming to children through games*. Master of Science Thesis. Tampere University of Technology. <https://urn.fi/URN:NBN:fi:tyy-201412031565>

- Laporte, L., & Zaman, B. (2016). Informing Content-driven Design of Computer Programming Games: a problems analysis and a game review. *NordiCHI'16*, 1–10. <https://doi.org/10.1145/2971485.2971499>
- Lehat, M. L., Mokhtar, R., Sokman, Y., Ismail, M. I., & Basir, N. M. (2014). Games: An approach to introduce computer programming for upper primary school students. *Proceedings - 2014 3rd International Conference on User Science and Engineering: Experience. Engineer. Engage, i-USER 2014*. <https://doi.org/10.1109/IUSER.2014.7002682>
- Lindberg, R. S. N., Hasanov, A., & Laine, T. H. (2017). Improving play and learning style adaptation in a programming education game. *CSEdu 2017 - Proceedings of the 9th International Conference on Computer Supported Education, 1*. <https://doi.org/10.5220/0006350304500457>
- Lindberg, R. S. N., & Laine, T. H. (2018). Formative evaluation of an adaptive game for engaging learners of programming concepts in K-12. *International Journal of Serious Games*. <https://doi.org/10.17083/ijsg.v5i2.220>
- Liu, Z., & Moon, J. (2021). Investigating Children's Problem-Solving Patterns in Digital Game-Based Learning for Computational Thinking Development. In de Vries, E., Hod, Y., & Ahn, J. (Eds.), *Proceedings of the 15th International Conference of the Learning Sciences - ICLS 2021*. (pp. 949–950). Bochum, Germany: International Society of the Learning Sciences.
- Liu, Z., & Jeong, A. C. (2022). Connecting learning and playing: the effects of in-game cognitive supports on the development and transfer of computational thinking skills. *Educational Technology Research and Development*. <https://doi.org/10.1007/s11423-022-10145-5>
- Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, 27(1), 1–29. <https://doi.org/10.1080/08993408.2017.1308651>
- Lotfi, E., & Mohammed, B. (2018). *Teaching Object Oriented Programming Concepts Through a Mobile Serious Game*. 1–6. <https://doi.org/10.1145/3286606.3286851>
- Luís Eduardo Thibes Forquesato (2018). Using a game to teach computational thinking and assess learning. 2018. 1 online resource (56 p.). Dissertation (master's degree) - State University of Campinas, Institute of Computing, Campinas, SP.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014b). Integrating learning analytics in an educational MMORPG for computer programming. In Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies. Athens, Greece: IEEE Computer Society Press.
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014a). Designing educational games for computer programming: A holistic framework. *Electronic Journal of E-Learning*, 12(3).
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2017). CMX: The effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, 10, 219–235.
- Malva, L., Hooshyar, D., Yang, Y., & Pedaste, M. (2020). Engaging estonian primary school children in computational thinking through adaptive educational games: A qualitative study. *Proceedings - IEEE 20th International Conference on Advanced Learning Technologies, ICALT 2020*. <https://doi.org/10.1109/ICALT49669.2020.00061>
- Min, W., Mott, B., Park, K., Taylor, S., Akram, B., Wiebe, E., Boyer, K. E., & Lester, J. (2020). Promoting Computer Science Learning with Block-Based Programming and Narrative-Centered Gameplay. *IEEE Conference on Computational Intelligence and Games, CIG, 2020-August*. <https://doi.org/10.1109/CoG47356.2020.9231881>
- Mina, D., Salah, J., & Abdennadher, S. (2022). ARcode: Programming for Youngsters Through AR. In: , et al. *Methodologies and Intelligent Systems for Technology Enhanced Learning, 11th International Conference. MIS4TEL 2021. Lecture Notes in Networks and Systems*, 326. https://doi.org/10.1007/978-3-030-86618-1_7
- Moreno-Leon, J., Roman-Gonzalez, M., & Robles, G. (2018). On computational thinking as a universal skill: A review of the latest research on this ability. *IEEE Global Engineering Education Conference, EDUCON, 2018-April*. <https://doi.org/10.1109/EDUCON.2018.8363437>
- Nche, O. M., Welter, J., Che, M., Kraemer, E. T., Sitaraman, M., & Zordan, V. B. (2019). CodeTrasures-Combining Gaming, CS Concepts, and Pedagogy. *Proceedings of the 2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology, RESPECT 2019*. <https://doi.org/10.1109/RESPECT46404.2019.8985865>

- Papadakis, S. (2021). The Impact of Coding Apps to Support Young Children in Computational Thinking and Computational Fluency. A Literature Review. In *Frontiers in Education* (Vol. 6). <https://doi.org/10.3389/feduc.2021.657895>
- Papadakis, S. (2020). Robots and robotics kits for early childhood and first school age. *International Journal of Interactive Mobile Technologies (iJIM)*, 14(18), 34–56. <https://doi.org/10.3991/ijim.v14i18.16631>
- Papert, S. (1980). Computers for children. In *Mindstorms: Children, computers and powerful ideas*.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*. <https://doi.org/10.1007/BF000191473>
- Pessoa, M., Alencar, L. F., Araujo, L., Melo, R., & Pires, F. (2019). Looking for Pets: A game for the logical reasoning development. *Proceedings - Frontiers in Education Conference, FIE, 2019-October*. <https://doi.org/10.1109/FIE43999.2019.9028510>
- Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. *Computers and Education*. <https://doi.org/10.1016/j.compedu.2018.09.006>
- Prensky, M. (2001). Digital natives, digital immigrants part 1. *On the Horizon*, 9(5), 1–6.
- Ramirez-Rosales, S., Vazquez-Reyes, S., Villa-Cisneros, J. L., & de Leon-Sigg, M. (2016). A serious game to promote object oriented programming and software engineering basic concepts learning. *Proceedings - 2016 4th International Conference in Software Engineering Research and Innovation, CONISOFT 2016*, 97–103. <https://doi.org/10.1109/CONISOFT.2016.23>
- Relkin, E., de Rooter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers and Education*. <https://doi.org/10.1016/j.compedu.2021.104222>
- Resnick, M., & Siegel, D. (2015). A different Approach to Coding. *International Journal of People-Oriented Programming*, 4(1).
- Rijo-Garcia, S., Segredo, E., & Leon, C. (2022). Computational thinking and user interfaces: A systematic review. *IEEE Transactions on Education*. <https://doi.org/10.1109/TE.2022.3159765>
- Rose, S. P., Jacob Habgood, M. P., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of E-Learning*, 15(4), 297–309. <https://doi.org/10.2304/elea.2005.2.1.5>
- Rose, S. P., Jay, T., & Jacob Habgood, M. P. (2019). Using pirate plunder to develop children's abstraction skills in scratch. *Conference on Human Factors in Computing Systems - Proceedings*. <https://doi.org/10.1145/3290607.3312871>
- Rowe, E., Asbell-Clarke, J., Gasca, S., & Cunningham, K. (2017a). Assessing implicit computational thinking in Zoombinis Gameplay. *ACM International Conference Proceeding Series, Part F130151*. <https://doi.org/10.1145/3102071.3106352>
- Rowe, E., Asbell-Clarke, J., Cunningham, K., & Gasca, S. (2017b). Assessing implicit computational thinking in Zoombinis gameplay: Pizza pass, fleens & bubblewonder abyss. *CHI PLAY 2017 Extended Abstracts - Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*. <https://doi.org/10.1145/3130859.3131294>
- Schez-Sobrino, S., Vallejo, D., Glez-Morcillo, C., Redondo, M., & Castro-Schez, J. J. (2020). RoboTIC: A serious game based on augmented reality for learning programming. *Multimedia Tools and Applications*, 79(45–46). <https://doi.org/10.1007/s11042-020-09202-z>
- Segura, R. J., del Pino, F. J., Ogáyar, C. J., & Rueda, A. J. (2020). VR-OCKS: A virtual reality game for learning the basic concepts of programming. *Computer Applications in Engineering Education*. <https://doi.org/10.1002/cae.22172>
- Shahid, M., Wajid, A., Haq, K. U., Saleem, I., & Shujja, A. H. (2019). A Review of Gamification for Learning Programming Fundamental. *3rd International Conference on Innovative Computing, ICIC 2019*, pp. 1–8, <https://doi.org/10.1109/ICIC48496.2019.8966685>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. In *Educational Research Review* (Vol. 22, pp. 142–158). Elsevier Ltd. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Israel-Fishelson, R., & Hershkovitz, A. (2019). Micro-Persistence in the Acquisition of Computational Thinking. *Proceedings of International Conference on Computational Thinking Education, CTE 2019* (pp. 18–23). <https://cris.tau.ac.il/en/publications/micro-persistence-in-the-acquisition-of-computational-thinking>

- Sorrentino, F., Spano, L. D., Casti, S., Carcangiu, A., Corda, F., Cherchi, G., Murru, A., Muntoni, A., Nuvoli, S., & Scateni, R. (2017). ChIP: Teaching coding in primary schools. *CEUR Workshop Proceedings, 1910*.
- Steinmaurer, A., Pirker, J., & Gütl, C. (2019). Scool - Game-based learning in computer science class A case study in secondary education. *International Journal of Engineering Pedagogy*. <https://doi.org/10.3991/ijep.v9i2.9942>
- Steinmaurer, A., Pirker, J., & Gütl, C. (2020). sCool - Game based learning in STEM education: A case study in secondary education. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-030-11932-4_58
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*. <https://doi.org/10.1016/j.compedu.2019.103798>
- Taslibeyaz, E., Kursun, E., & Karaman, S. (2020). How to develop computational thinking: A systematic review of empirical studies. *Informatics in Education*. <https://doi.org/10.15388/INFEDU.2020.30>
- Taylor, K., & Baek, Y. (2018). Collaborative robotics, more than just working in groups. *Journal of Educational Computing Research, 56*(7), 979–1004. <https://doi.org/10.1177/0735633117731382>
- Theodoropoulos, A., & Lepouras, G. (2020). *Digital Game-Based Learning and Computational Thinking in P-12 Education: A Systematic Literature Review on Playing Games for Learning Programming* (pp. 159–183). <https://doi.org/10.4018/978-1-7998-4576-8.ch007>
- Toukiloglou, P., & Xinogalos, S. (2022). NanoDoc: Designing an adaptive serious game for programming with working examples support. *European Conference on Games Based Learning*, 16, 628–636. <https://doi.org/10.34190/ecgbl.16.1.463>
- Tsalapatas, H. (2015). Evaluating the use of programming games for building early analytical thinking skills. *EAI Endorsed Transactions on Game-Based Learning*. <https://doi.org/10.4108/eai.5-11-2015.150610>
- Utesch, M. C., Faizan, N. D., Krcmar, H., & Heining, R. (2020). Pic2Program - An educational android application teaching computational thinking. *IEEE Global Engineering Education Conference, EDUCON, 2020-April*. <https://doi.org/10.1109/EDUCON45650.2020.9125087>
- Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). A review of games designed to improve introductory computer programming competencies. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, Madrid, Spain, 2014, pp. 1-7. <https://doi.org/10.1109/FIE.2014.7044114>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society a: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717–3725.
- Xinogalos, S., & Satratzemi, M. (2004). Introducing Novices to Programming : a review of Teaching Approaches and Educational Tools. 2nd International Conference on Education and Information Systems, Technologies and Applications (Vol. 2, pp. 60–65). Academic Press.
- Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers and Education*. <https://doi.org/10.1016/j.compedu.2004.09.005>
- Yallihep, M., & Kutlu, B. (2020). Mobile serious games: Effects on students' understanding of programming concepts and attitudes towards information technology. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-019-10008-2>
- Yuliana, I., Octavia, L. P., Sudarmilah, E., & Matahari, M. (2019). Introducing Computational Thinking Concept Learning in Building Cognitive Capacity and Character for Elementary Student. *Proceedings - 2019 19th International Symposium on Communications and Information Technologies, ISCIT 2019*. <https://doi.org/10.1109/ISCIT.2019.8905149>
- Zaharija, G., Mladenović, S., & Boljat, I. (2013). Introducing basic Programming Concepts to Elementary School Children. *Procedia - Social and Behavioral Sciences, 106*, 1576–1584. <https://doi.org/10.1016/j.sbspro.2013.12.178>
- Zhang, S., Wong, G. K. W., & Chan, P. C. F. (2022b). *Integrating Game-based Learning into Computational Thinking Class for Lower Primary Students: Lesson Design and Course Effect*. <https://doi.org/10.34641/ctestem.2022.458>

- Zhang, S., Wong, G. K. W., & Chan, P. C. F. (2022a). Playing coding games to learn computational thinking: What motivates students to use this tool at home? *Education and Information Technologies*. <https://doi.org/10.1007/s10639-022-11181-7>
- Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers and Education*. <https://doi.org/10.1016/j.compedu.2019.103633>
- Zhi, R., Lytle, N., & Price, T. W. (2018). *Exploring Instructional Support Design in an Educational Game for K-12 Computing Education*. April 2019, 747–752. <https://doi.org/10.1145/3159450.3159519>

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

Andreas Giannakoulas is a PhD Candidate, since 2018, in the Department of Applied Informatics at the University of Macedonia (Thessaloniki, Greece), supervised by Stelios Xinogalos, professor at the Department of Applied Informatics, University of Macedonia, Greece. He received his BSc in Informatics from the Aristotle University of Thessaloniki in 2002 and his MSc in Applied Informatics from the University of Macedonia (Thessaloniki, Greece) in 2017. He also works in education as Informatics teacher since 2004. His current research is focused on programming environments and games for young children.

Stelios Xinogalos is a Professor in the Department of Applied Informatics, University of Macedonia, Greece. He received his BSc and PhD from the Department of Applied Informatics, University of Macedonia, Greece in 1998 and 2002 respectively. His research interests include Programming Environments and Techniques, Object-oriented Design and Programming, Didactics of Programming, Computer Science Education, Educational Technology, and Serious/Educational Games. He has published more than 120 research papers in International journals, conferences and books.