Review

Not peer-reviewed version

How to Learn and How to Teach the Five Content Areas: Programming Languages, Game Design, Computer Thinking, Algorithmic Thinking and Robotics Programming - A Systematic Review of Journal Publications From 2015 to 2020”

Dragan Cvetkovic , Dragan Rastovac * , Dejan Viduka , Ana Basic

Posted Date: 27 January 2023

doi: 10.20944/preprints202301.0505.v1

Keywords: computer thinking; game design; programming; review; robotics

Review

How to Learn and How to Teach the Five Content Areas: Programming Languages, Game Design, Computer Thinking, Algorithmic Thinking and Robotics Programming—A Systematic Review of Journal Publications from 2015 to 2020"

Dragan Cvetkovic [1], Dragan Rastovac [2,*], Dejan Viduka [3] and Ana Basic [4]

[1] Faculty of Education, University of Novi Sad, Serbia; dcvele@pef.uns.ac.rs
[2] The Higher Education Technical School of Professional Studies in Novi Sad and Preschool Teacher Training and Business Informatics College of Applied Studies, Sremska Mitrovica, Serbia; rastovacd@gmail.com; rastovac@vtsns.edu.rs
[3] Faculty of applied management, economics and finance, University Business Academy in Novi Sad, Serbia; dejan.viduka@mef.edu.rs
[4] Information Tehnology School - ITS, New Belgrade, Serbia; ana.basic@its.edu.rs
* Correspondence: rastovacd@gmail.com; rastovac@vtsns.edu.rs;

Abstract: This study presents review of 7580 papers in 13 academic journals, published from January 2015 to January 2020. After a detailed analysis of all papers, 46 papers were further selected showing research on a student population aged between 6 and 15 years old. In order for the paper to be included in the research, the condition was that the paper deals with teaching at least one of the following content areas: programming languages, game design, computer thinking (CT), algorithmic thinking and robotics programming. This study shows the representation of the listed content area in reviewed papers for the specified time period as well as a detailed analysis of the selected papers. Available data about study, participants and education level, country (first author), learning domain, teaching tools, research questions, hypothesis, pre / post-test results, interviews, control groups, course duration, research design, previous experience, project or grant and research purpose in detected papers were analysed. In addition, impact of studying some of the listed content areas on student learning performance, motivation, attitude and perception were investigated.

Keywords: computer thinking; game design; programming; review; robotics

1. Introduction

With the technology development, the way of education is changing [1]. Therefore, there is a need to include technology [2] and computer science [3] in school curricula. Since 2011, the importance of e-skills has been recognized, so relevant organizations have been involved in defining computer science in schools and it is concluded that computer science topics should be an integral part of the school curricula [4, 5, 6]. According to [7] curriculum changes have been made throughout the world like: the UK [8, 9, 10], Europe [11] Australia [12] and New Zealand [13].

The great commitment to K-12 computer science education indicates the recognition of its importance and helps to solve the lack of computer experts around the world. The creation of a viable model of computer science curriculum and its implementation at the K-12 level is a necessary first step toward reaching these goals [14](p. 3).

In the 1990s, people's interest in programs that use the graphical environment for programming, debugging, etc. began to rise. Increased use of the graphical environment for programming learning can be considered as expected, since learning programming using a traditional approach is more complex for some students. Therefore, it is necessary to realize a graphical environment that will be easy to manage and available to students and teachers interested in "graphic programming" [15] [16]. Numerous studies suggest the use of visual programming languages (when learning programming) like: Logo [17, 18, 19, 20], Scratch [21, 22, 23, 24], Alice [25,26,27]. However, with the growing interest of students in learning graphically oriented programming languages, the gradual introduction of textual programming languages began in schools and, thus, a new field of research was opened for authors [28, 29, 30,31].

In parallel with the introduction of computer science (learning programming, above all) in the curricula of primary school, there was a need to consider how to develop computer thinking (CT) in children. Computational thinking is a type of analytical thinking that implies approach to problem solving. It is about mathematical thinking where we can tackle problems in a variety of ways. It can also be expanded to imply the understanding of human behavior and intelligence [32]. Lu and Fletcher [33] state that "proficiency in computational thinking helps us to systematically and efficiently process information and tasks", while Dede, Mishra, and Voogt [34] give the following definition "Computational thinking is seen as a skill set that every child needs to develop". Voogt, Fisser, Good, Mishra and Yadav [35] state that "programming is an important tool to help develop computational thinking skills", and Román-González, Pérez-González, and Jiménez-Fernández [36]: "computational thinking must be acquired by the new generations of students to thrive in the digital world". Kong, Chiu, and Lai [37] in his study confirms the fact of Brennan and Resnick [38] and Papert [39] that, with the development of CT, students can creatively express their ideas. In addition to computer thinking [33, 34, 40, 41, 42, 43] algorithmic thinking has been studied in the literature [44, 45, 46, 47].

Authors Wilson, Hainey and Connolly [48] present empirical proofs and further guidance on the assessment of the ability of elementary school students to do games-based programming. The after-school club named "Code club" was founded in Great Britain in 2012 to support the elementary school in the field of programming. Students are trained together to create games in the Scratch program [49]. Many other studies have shown increased interest in game design (games-based programming) [50, 51, 52, 53, 54].

Robotics is a growing field that has the potential to significantly impact the nature of engineering and science education at all levels, from K-12 to graduate school [55] (p. 1). Robotics provides "source of energy" that can be used to motivate children to learn [56] and, more specifically, to interest students to learn many aspects of robotics, programming, and computational thinking [57, 58]. Also, a large number of authors gave their views related to robotics programming issues [59, 60, 61, 62, 63].

In the study, presented in this paper, the representation of five content areas (programming languages, game design, computer thinking, algorithmic thinking, robotics programming) in published papers of the cited journals has been investigated. Also, in order to be included in this study, it was necessary that research presented in analysed papers includes participants aged between 6 and 15 years. The research questions are modified version of [64]:

1. What are the journal name and year of publication of the articles, country context, education level and age of the participants, learning domain, teaching tools, research design, previous participant experience, research methodology, course duration, grant or project and research purpose?
2. What is the impact of the content areas on student learning performance, motivation, attitude and perception?

The paper has been organized as follows: section 2 analyses related work; section 3 presents research method (review process, analysis framework and coding); section 3 analyses the research results and gives the discussion of the results; section 4 provides limitations and future research direction; section 5 presents main conclusions.

2. Related Work

In numerous papers, researchers have studied learning achievement of primary and secondary schools' students in the following areas: programming language, computer thinking, algorithmic thinking, game design and robotics programming. These areas are usually an integral part of computer science curricula. Some of the listed areas have been also included in a large number of review papers.

In the programming field, Popat and Starkey [65] reviewed research papers that analyse educational outcomes of learning students to code at school, while Costa and Miranda [66] presents a systematic review of the literature about the effectiveness of the use of Alice software in learning programming comparing with using a conventional programming language. The papers [67, 68, 69, 70, 71, 72, 73] deal with research of learning programming languages as well.

According to Hsu, Chang, and Hung [74], the basic problem of computational thinking (CT) is how to teach it. The authors reviewed the academic journals and searched in them adopted learning strategies and teaching tools in CT education. It has been established that CT is applied in the field of computer science and that there are also studies related to other subjects. The systematic review of computational thinking development through programming in Scratch was presented by Zhang and Nouri [75]. Also, authors [76, 77, 78, 79, 80] show the review in the computer thinking area.

In the article [81] the methodologies, frameworks, and models applied to game designs, and phases of game development software are given. The review of programming curricula in seven countries and programming games were analysed by Lindberg, Laine and Haaranen [82], as well by [83, 84, 85, 86]. Benitti [87] reviews recently published articles in the use of robotics in schools in three main directions: identifying the potential contribution of the applying robotics as educational tool in schools; presenting a synthesis of the available empirical evidence on the educational effectiveness of robotics; defining future research perspectives concerning educational robotics. Xia and Zhong [88] also present review of empirical studies on teaching and learning robotics content knowledge in K-12 and explore future research perspectives of robotics education (RE), based on the reviewed papers. References [89, 90, 91, 92] show research review in robotics area as well.

3. Methods

In this paper, the general guidelines proposed by Kitchenham and Carters [93], have been adapted. It allows to collect empirical evidence about formulated research questions.

3.1. Review process

In this study, an analysis of the articles published in 13 academic journals has been shown. Analysed journals are: Computers in Human Behavior (CHB), Computers & Education (CAE), British Journal of Educational Technology (BJET), Journal of Educational Technology and Society (JETS), Interactive Learning Environments (ILEs), Learning and Instruction (EARLI), Electronic Liberary (EL), Educational Technology Research & Development (ETR&D), Journal of Computer Assisted Learning (JCAL), The Turkish Online Journal of Educational Technology (TOJET), Journal of Educational Computing Research (JECR), IEEE Transactions on Learning Technologies (TLT), International Journal of Computer-Supported Collaborative Learning (IJCSCL) from 1st January, 2015 to 31st December, 2019. The list of journals can be observed as modification of the list considered in Chauhan [94]. The search process was carried out between January 2020 and June 2020. A detailed description of the review procedure is given below.

We reviewed 7580 articles, i.e. all editions of the journal, listed above (Table 1) were counted and it was systematized 46 articles (Table 2) based on selected content areas: programming languages, game design, computer thinking, algorithmic thinking, robotics programming. The condition was that the review paper covers the listed topics in the context of learning, achievements, etc. of school students (aged 6 to 15 years old). Firstly, it was necessary to exclude journals that are not relevant to our research. The research "exclusion strategy" was performed manually through two phases. The first phase involved reading the title and the abstract of each article. In the next phase, the papers that seemed relevant were reviewed in detail by reading the full text [65]. The articles were limited to those published in academic journals. Book reviews, editorial materials, PhD dissertations, meeting abstracts, proceedings papers, were not included in this study. A detailed overview of the literature research and review process is shown in block diagram in Figure 1.

Table 1. Number of articles published in 13 academic journals by year.

Journal Name	**2015**	**2016**	**2017**	**2018**	**2019**	**Total**
Computers in Human Behavior	686	855	693	447	455	3097
Computers & Education	235	161	147	212	198	953
British Journal of Educational Technology	116	89	103	80	201	589
Educational Technology and Society	115	103	92	83	30	423
Interactive Learning Environments	45	118	72	74	79	388
Learning and Instruction	62	58	71	83	73	347
Electronic Library	70	60	72	72	61	335
Educational Technology Research and Development	43	61	71	71	67	313
The Turkish Online Journal of Educational Technology	76	61	65	60	38	300
Journal of Computer Assisted Learning	47	44	46	89	63	289
Journal of Educational Computing Research	48	46	46	54	82	276
IEEE Transactions on Learning Technologies	29	32	42	43	39	185
International Journal of Computer-Supported Collaborative Learning	16	17	17	18	17	85
Total						**7580**

Table 2. Number of selected articles published in academic journals.

Journal Name	Number of Studies
Computers & Education	15
Computers in Human Behavior	10
Journal of Educational Computing Research	5

Journal of Computer Assisted Learning	5
Educational Technology and Society	4
Educational Technology Research and Development	3
Interactive Learning Environments	2
British Journal of Educational Technology	2
Total	*46*

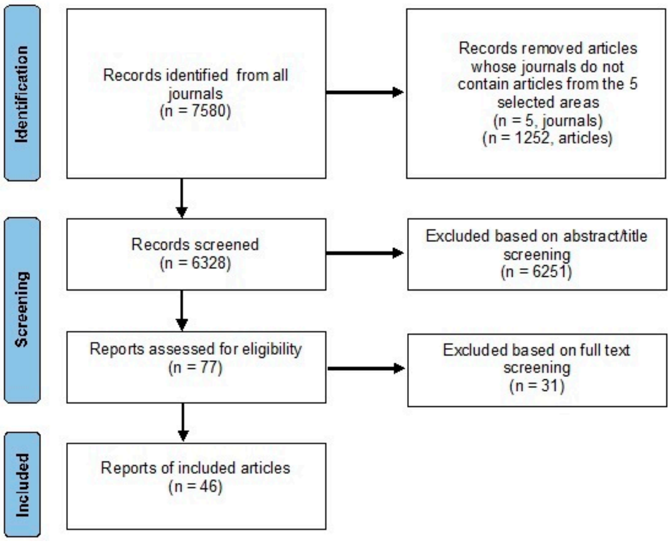

Figure 1. The review process (PRISMA flowchart).

3.2. Analysis framework and coding

3.2.1. Research design

Elements were chosen for analysis in accordance with the research questions and represent modification of the elements developed by [95, 96, 97]: 1) the journal of publication, 2) year of publication, 3) the country context (of the first author), 4) the education level and age of the participants (elementary school, secondary school, middle school and mixed subjects- aged 6 to 15 years old), 5) learning domain (programming, robotics programming, game design, computer thinking, algorithmic thinking, and review paper), 6) teaching tools (Scratch, Alice ...etc.), 7) research design, 8) previous participant experience 9) research methodology (research questions, hypotheses, pre/post-test, interviews, control group), 10) the duration of the course, 11) grant or project, 12) research purpose. In this review, each of the research elements were coded individually or were given descriptively. The methods of analysis of each element are described in the continuation:

- journal of publication, year of publication, country context (of the first author). The basic information about listed elements, described in the published articles, is discussed. The aim is to establish which countries have more frequently published articles.
- education level and age of participants – information about: elementary school, primary school, secondary school, middle school (depending on the appropriate level of study of the country in which the research was conducted) and a combination thereof, are coded with the correct name of

the study level and grade (considering appropriate interval from 6 to 15 year old). Also, the age of the participants is given numerically (Appendix). The main purpose of the analysis is to determine which educational level(s) are covered in chosen papers, as well as to compare the number of participants in articles per educational level.

- learning domain- covers topics: programming, robotics programming, game design, computer thinking, algorithmic thinking, and review paper from those topics. Topics are coded by their name (Appendix). The aim is to understand which topic is most represented in published articles.
- teaching tools – represents programming languages and software. This category includes subgroups like Scratch (Scratch Jnr, Blockly), Alice, code.org (The Maze, The Canvas', The Artist 2), programs designed by authors of articles (Koios at al...) (ROBOTCG Graphical - graphical interface, Simulation Creation Toolkit, Unity's 2D development tools, Turtle Graphics Tutorial System, whereby one or more teaching tools may be listed in the analysed papers. Teaching tools are coded by their exact name (Appendix). If the teaching tools are not specified in the article it is coded with "not specified". The main purpose of the analysis of these elements is to determine which teaching tools are used, if any.
- research design – category include: quantitative, qualitative or mixed design. They are coded with their exact name, and if the research design is not specified in the article, it is coded with "not specified".
- previous participant experience – represents the previous experience of participants in working with teaching tools that are done in workshops realized by the article' author(s). They are coded with "Yes" if the authors stated that the participants have experience, with "No" if it is stated that they have no experience and "not specified" if neither of the previous two attitudes is stated.
- Research methodology - category include: research questions, hypotheses, pre/post-test, interviews, control group. They are coded with "Yes" if the authors have defined them in their papers and with "No" if they are not defined. As the next step of the analysis, a counting is performed in order to determine the exact number for each parameter in relation to the number of papers as well as to determine the paper that has the most represented parameters.
- the duration of the course – is defined by the exact time period specified in the article or by "not specified" if not specified. The aim is to understand which papers had the longest time period of course realization.
- grant or project – coding includes the exact name of the grant or project if the paper was realized within the project, or "not specified" if the grant or project is not specified.
- research purpose – this category is defined by a brief description of the purpose of the research stated by the authors in their papers.

4. Research results and discussions

4.1. Research question one

4.1.1. Journal of publication, year of publication, country context

The Appendix contains the following information: the journal where the article is published, the year of publication, the country context (first author) of articles. The papers were published in 13 educational journals for the period from January 2015 to January 2020. The most of the selected articles (15) were published in the CAE journal, followed by the CHB with 10 articles, JECR and JACL with 5 articles each, JETS with 4, ETR&D with 3, BJET and ILEs journals with 2 articles (see Table 2). The review method, adopted in this paper, was based on the concept of systematic review proposed by (Hsu et al., 2018). The Papers were published by authors from 15 countries: USA 9, Turkey 7, Spain 6, China 5 articles, Taiwan 4, The United Kingdom 3, Greece, Hong Kong and New Zealand 2, Belgium, Italy, Norway, Portugal, Slovenia, Sweden with 1 article each. Figure 2. shows distribution status of the authors by countries. In our study, we only listed the nationality information of the first author in country context. The number of published articles by year is illustrated by Figure 3. From the Figure 3 it can be seen that the trend in the number of publications for the period (2015-2020) is constantly increasing.

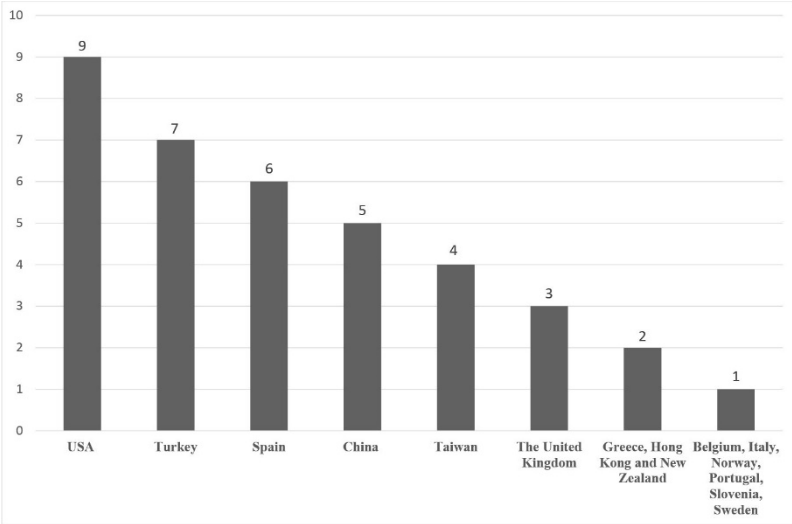

Figure 2. The number of published articles by country.

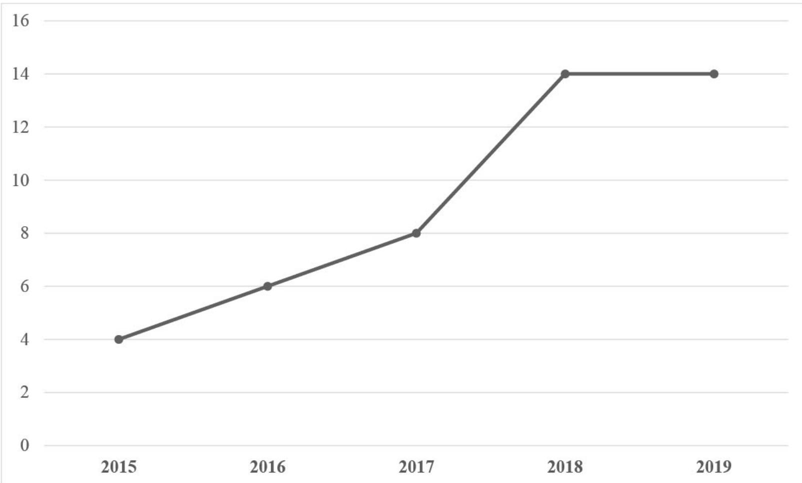

Figure 3. The number of published articles by year.

4.1.2. Education level and age of participants

In order to have a clearer and more consistent analysis, we have adjusted the Elementary Education category as in [94], to include: Elementary Education, Elementary School, Elementary student, Primary Education, Primary School, Primary Student. The research studies involved elementary students (58%), followed by secondary students (28%) and students in middle education (14%). All students are between 6 and 15 years old (Figure 4). Two papers [98, 99] presented a study with the participation of students with special needs (elementary and secondary level, respective).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 January 2023 doi:10.20944/preprints202301.0505.v1

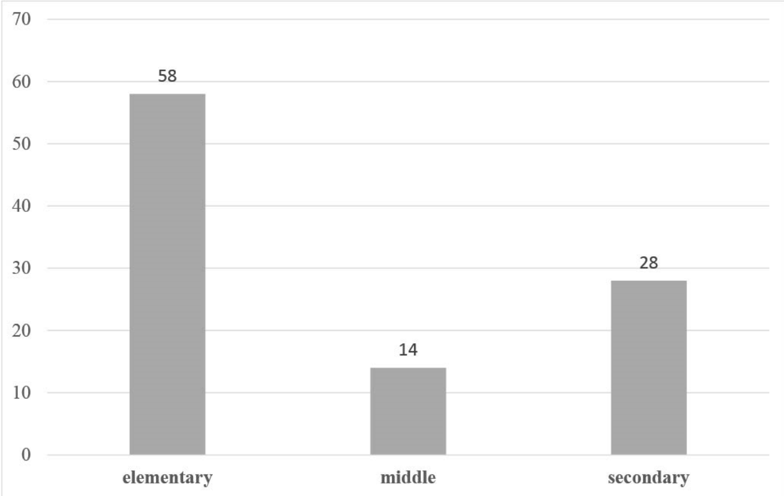

Figure 4. Classification of participants by education level.

4.1.3. Learning domain

In systematized papers, the areas of computer thinking and programming are most represented, (16 and 13 articles, respective), which makes (63%) of the total number of papers, followed by game design (7 articles), review articles (6). The area of algorithmic thinking is the least represented (1 article), (see last column a Table 3). This relationship is not surprising because, according to [100], many of the countries in Europe have been attempting to incorporate CT courses into their K-12 education curricula. European Ministries of Education give an overview of their current initiatives and plans regarding computer programming [74, 101]. However, it opens up the possibility of further research about the extent to which the computer thinking is included in the existing curricula. Also, in last row the Table 3 we see the number of papers in relation to the education level. The similar analysis was performed in [74](see Table 4, page 306).

Table 3. The number of articles in relation to the content area and education level.

	Primary	Middle	Secondary	Primary and secondary	Total
programming	9	1	2	1	13
CT	9 (1 special needs)	3	1	3	16
Game design	-	4	3 (1 special needs)	-	7
Robotics programming	1	1	1	-	3
Algorithmic thinking	1	-	-	-	1
Review articles	-	-	-	-	6
Total	*20*	*9*	*7*	*4*	*40/46*

4.1.4. Teaching tools

Some of teaching tools were used in 40 selected papers. Figure 5 shows the representation of teaching tools in selected papers. The most common is the Scratch programming language, which has been used in 14 papers. Two or more programming languages [36, 102, 103] were used in three papers, while one paper used programming software developed by the author of the paper. Such representation of Scratch and similar block-based software is based on the fact that many authors consider visual programming languages as one of the best for "novice programmers" [76, 104, 105].

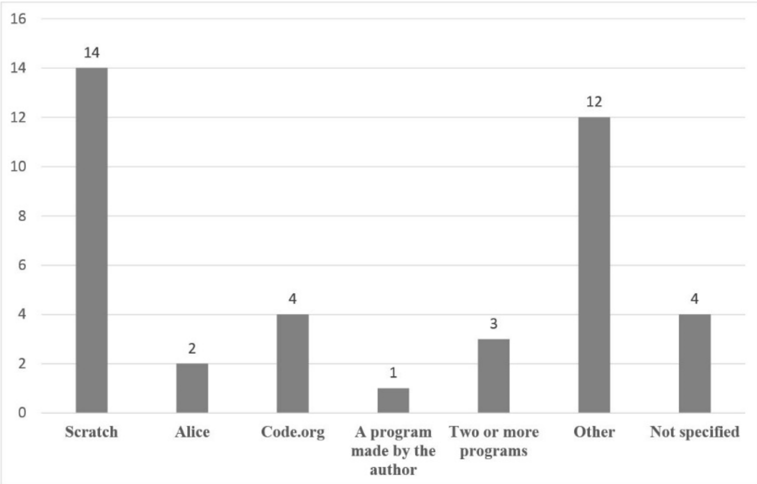

Figure 5. The number of programming languages used in the papers.

4.1.5. Research design

A total of 40 research articles were analysed. In accordance with [106], we defined three types of design (see subsection 3.2.1, item research design) and adjusted our coding based on [64]. All reported methods were included in the analysis. Thirteen articles (32,5%) are based on a quantitative design, followed by eleven articles (27,5%) with a mixed design, and three (7,5%) based on a qualitative research design. The authors in thirteen articles (32,5%) did not explicitly define the research design and, in our coding, we categorized such papers as "not specified". This result of the analysis was obtained because some authors did not use only one research design but mixed. Also, the significant proportion of papers marked as "not specified" is a consequence of the fact that the authors did not explicitly define research design [107]. For example, the authors in [108] predominantly used the qualitative method, and, to a lesser extent, the descriptive analysis. However, since they did not clearly state the research design, for research design "not specified" is put.

4.1.6. Previous experience

In accordance with the coding of this element, out of 40 papers, in 25 (62.5%) papers the authors conducted research with participants who have no previous experience in the field of research. This is followed by 9 (22.5%) papers whose participants had previous experience and 6 (15%) papers where there were no notes on the previous experience of participants. Since some curricula has only recently introduced programming and CT (in primary and/or secondary schools), there is a higher percentage of lack of previous experience in these content areas [12, 109, 110].

4.1.7. Research methodology

In accordance with the presented coding rules (see subsection 3.2.1, item research methodology), research methodology was modified from the research methodology developed by Crompton and Burke [107]. Many articles reported more than one type of research methodology and all reported methods were included in the analysis. The distribution of the number of research methodologies by papers is shown in Figure 6. However, it should be noted that, of 40 papers included in this paper, only one paper [111] contains all five components of research methodology that we defined in our review.

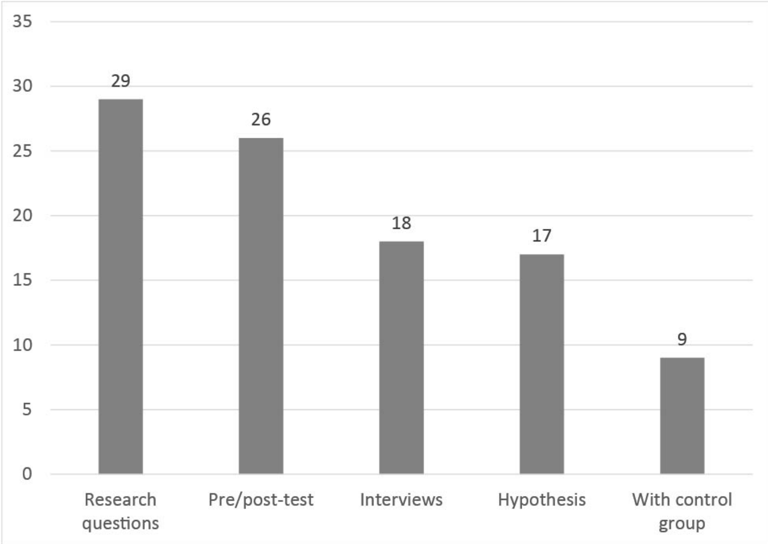

Figure 6. The number of research methodologies' types by papers.

4.1.8. Course duration

Motivation for the analysis of this aspect was found in the work of the authors Lye and Koh ([95]; see Table 2) who presented the time period of the course in their meta-analysis of review papers. In order to determine the duration of the course as objectively as possible, we considered only those papers that explicitly stated at least two of three parameters month / week / hour (number of weeks and hours per week) (24 papers were considered). Based on that, the longest time courses were conducted in the study presented in (Su, Huang, Yang, Ding, and Hsieh [112] ("three hours per week, 4-month, March to June 2013" (in primary school)) and Ruggiero and Green [99] ("Six-months, 30 workshops, 90 min. per workshop " (on special needs school for young people)). On the other hand, Basawapatna [113] ("4 days, 16 pattern implementations") and Çakır, Gass, Foster and Lee [114] ("a full day event during the weekend, two workshops"), (both middle school) conducted the research in the shortest time. These results are based on the fact that the longest courses were conducted by authors in primary and special needs schools, where students are introduced to programming for the first time.

However, it is important to note that this analysis did not include papers that realized their workshops as an integral part of compulsory or elective curriculum subjects with a time period of 1-2 semesters [36, 108, 115, 116, 117, 118], (see Appendix).

4.1.9. Research purpose

In accordance with a significant number of related review papers, in this paper, we have not separately analysed research purpose. For example, the authors of the papers [74, 119] in their reviews analysed only two of three elements in their reviews (learning domain, teaching tools and research purpose). Accordingly, in the previous sections we have made a detailed classification in the following categories: learning domain - review of papers based on the content area and teaching tools - review of papers based on the use of software.

4.2. Research question two

4.2.1. Student learning performance, motivation, attitude and perception

In the table 4 there are 40 articles for which we analyzed the impact of studying the specific content area on the following elements: student learning performance, motivation, attitude and

 doi:10.20944/preprints202301.0505.v1

perception (based on [64], see Table 5 page 115). Review articles were not included in this analysis. The coding in the Table 4 is performed as follows: listing the previous elements in a positive sense, confirming hypotheses and research questions is marked with "positive"; if the authors stated that the hypotheses and research questions were not confirmed and/or if there are negative attitudes related to the listed elements "negative" mark is put; when it is noted that there is no significant statistical difference or the "positive" or "negative" impact cannot be determined with certainty, the field in the table is marked as "neutral"; in case that there is no statement that caused the use of the aforementioned marks or if the elements are not listed, "not specified" is put.

Table 4. Impact of studying any of the content area on student learning performance.

Study	Impact on learning performance	Impact on learning motivation	Impact on learning attitudes	Impact on learning perceptions
Howland and Good (2015)	not specified	positive	not specified	not specified
Sáez-López et al. (2016)	positive	positive	positive	not specified
Snodgrass Israel and Reese (2016)	not specified	not specified	not specified	not specified
Çakır et al. (2017)	not specified	positive	positive	positive
Chen et al. (2017).	neutral	not specified	not specified	not specified
Durak and Saritepeci (2018)	not specified	not specified	negative	not specified
Hsu and Wang (2018)	positive	positive	positive	not specified
Kong et al. (2018)	not specified	not specified	neutral	not specified
Città et al. (2019)	not specified	not specified	not specified	positive
Zhao and Shute (2019)	not specified	neutral	positive	negative
Schlegel et al. (2019)	not specified	positive	not specified	positive
Kalelioğlu, F. (2015)	positive	not specified	positive	positive
Zhong et al. (2016)	positive	not specified	not specified	positive
Román-González et al. (2017)	positive	not specified	not specified	not specified
Ruggiero and Green (2017)	not specified	not specified	not specified	not specified
Pérez-Marín et al. (2018)	positive	not specified	not specified	not specified
Basogain et al. (2018)	not specified	not specified	not specified	not specified

Román-González et al. (2018)	positive	not specified	positive	positive
Cheng, G. (2019)	positive	not specified	positive	positive
Yücel and Rızvanoğlu (2019)	positive	positive	neutral	negative
Papavlasopoulou et al. (2019)	not specified	positive	positive	not specified
Akpinar and Aslan (2015)	not specified	not specified	not specified	not specified
Zhong et al. (2016)	not specified	not specified	not specified	not specified
Jakoš and Verber (2017).	positive	not specified	not specified	not specified
Tran, Y. (2019)	not specified	positive	positive	positive
Vasilopoulos and Van Schaik (2019)	positive	not specified	not specified	not specified
Falloon, G. (2016)	not specified	not specified	not specified	not specified
Witherspoon et al. (2018)	positive	negative	not specified	not specified
Benton et al. (2018)	not specified	not specified	not specified	not specified
Yildiz Durak, H. (2018a)	not specified	positive	positive	not specified
Yildiz Durak, H. (2018b)	positive	positive	positive	not specified
Su et al. (2015).	positive	not specified	not specified	positive
Basawapatna, A. (2016)	not specified	positive	not specified	not specified
Zhong et al. (2017)	not specified	not specified	positive	not specified
Wang et al. (2017)	positive	not specified	positive	not specified
Akcaoglu and Green (2019)	positive	not specified	not specified	not specified
Sáez-López et al. (2019)	positive	positive	not specified	not specified
Strawhacker and Bers (2019)	positive	not specified	not specified	not specified
Garneli and Chorianopoulos (2018)	positive	positive	not specified	not specified
Chiang and Qin (2018)	positive	not specified	positive	positive

4.2.1.1. Learning performance

A significant percentage (50%) of the reviewed articles reported a positive impact of the studying some of the listed content areas on student learning performance. Neutral impact was reported by 2% of the articles. This report can be considered as a consequence of the fact that elementary students have limited programming experience (fifth grade). Results showed that there was no statistical difference found for student performance [58]. However, 48% of the articles did not investigate impact on student learning performance (Figure 7).

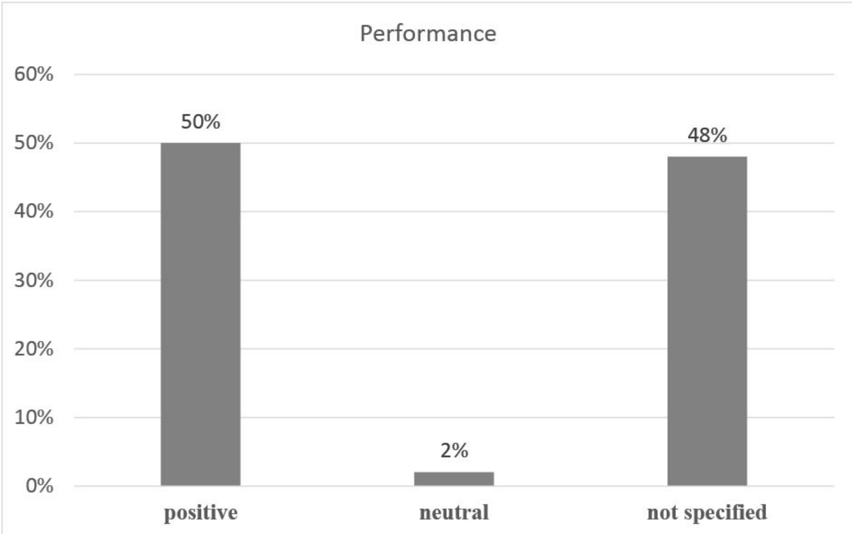

Figure 7. Students' learning performance.

4.2.1.2. Learning motivation

Positive impact of the content area studying on student learning motivation has been found in 32% of reviewed articles. Only two studies of the reviewed articles reported negative and neutral impact (every per 3%). Witherspoon, Schunn, Higashi and Shoop [120], in results of motivational analyse, show a decline in all our motivational measures, therefore we coded their paper with negative mark. We conclude that coding is neutral for the paper Zhao and Shute [121] since the game, used in the paper, did not have an additional reward system, for what is believed to increase the motivation of players. The rest (62%) of the articles in our review did not investigate the motivational impact of the content area studying (Figure 8).

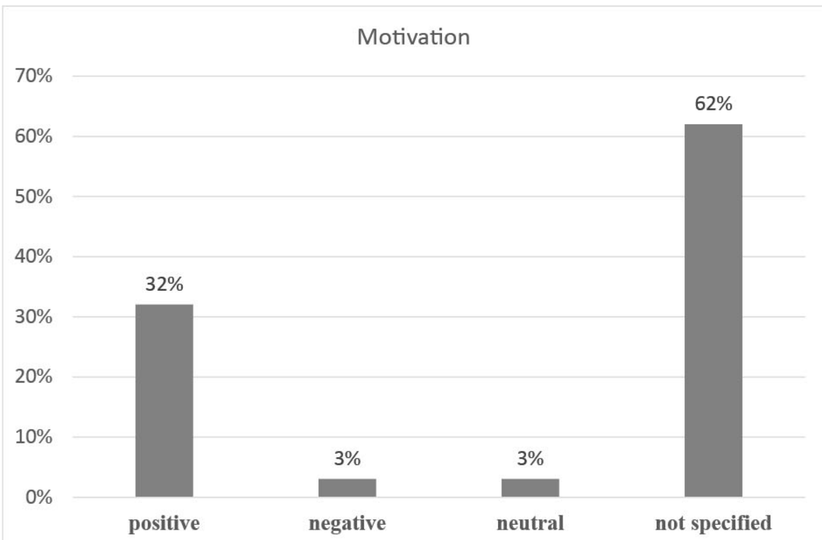

Figure 8. Students' learning motivation.

4.2.1.3. Students' attitude

For 35% of review journals impact according to student attitudes is coded as "positive". Only one article has a negative impact (3%; [122]), because the hypotheses dealing with students' attitude are rejected in the study. However, two articles are coded with "neutral". The authors Kong et al. [37] in their results pointed out the fact that "Students with better attitudes towards collaboration had more creative self-efficacy but not more programming self-efficacy", while the authors Yücel and Rızvanoğlu [123] concluded that girls have "negative attitudes towards serious educational games which were rather the opposite in their male counterparts". Based on quoted claims that girls have a negative attitude and men a positive one, we coded these cases with "neutral". Most of the reviewed articles 57% did not refer to student attitudes (Figure 9).

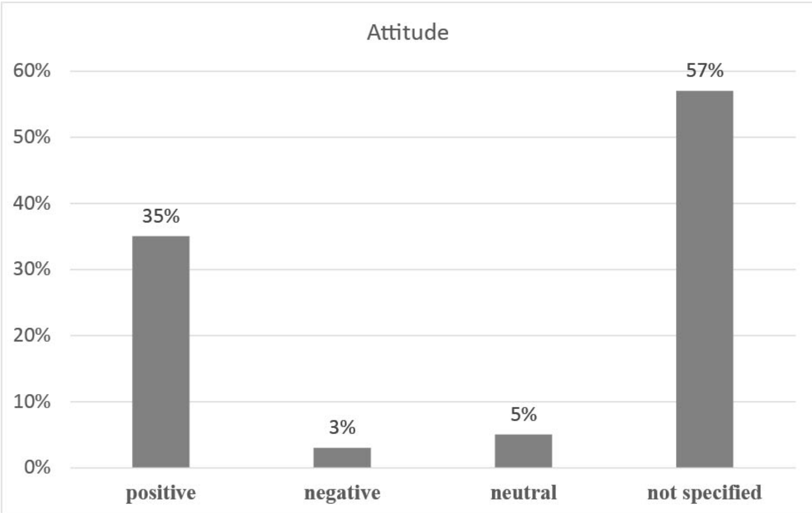

Figure 9. Students' attitudes.

4.2.1.4. Students' perception

Only 25 percentage of the reviewed articles reported a positive impact of the studying some of the listed content areas on student perception. Zhao and Shute [121], in the discussion results of the article, conclude that "the levels more challenging to solve, and thus may have lowered students' perceptions of competency.", while Yücel and Rızvanoğlu [123] in the results show that "failure perception in a code learning game". The majority of the reviewed articles (70%) did not investigate the perception impact (Figure 10).

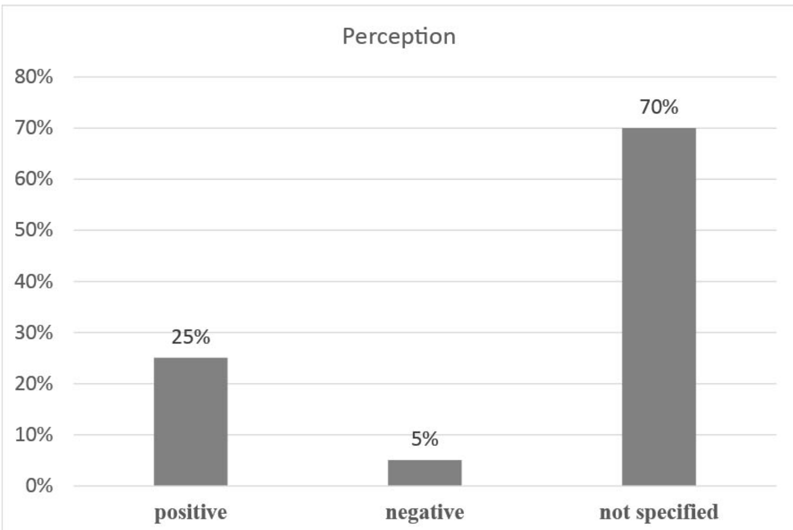

Figure 10. Students' perceptions.

5. Limitation and future research

The literature review, given in this paper, includes articles published in thirteen academic journals in the last five years (January 2015 - January 2020). Therefore, this study can be observed as limited since only chosen journals in predefined time period were analysed. These journals may not include all significant research results related to the following content areas: programming languages, computer thinking, algorithmic thinking, game design, robotics programming. Also, given that the level of education in different countries may differ according to age, the authors took into account the age of children from 6 to 15 years old as well as the names of individual levels (primary, elementary, secondary, middle) that cover the mentioned age range.

There were analysed papers where the authors did not state the age or exact educational level of the participants. Based on the information available on the official website of the Ministry of Education of the countries in which the authors conducted the research, we defined the missing data to perform the analysis.

For the sake of better credibility of the analysis of the results, we coded only those elements that the authors explicitly stated in their works, while in all other cases we coded with "not specified". It is important to emphasize that the "not specified" label does not necessarily mean that individual elements are not explained in detail in the analysed papers, but that they are not specified in accordance with the defined categories.

The two authors have implemented review and analysis, which potentially increases the relevance of comprehensive review, shown in this paper, in comparison with those that the review carried out by a single person. Nevertheless, although the papers in the field were analysed in detail and all relevant data were entered in the tables, the lack of a broader expert team could be seen as a limitation of the research.

Also, journals written in English were analysed, and, thus, papers written in other languages were not included in shown analysis.

Given these limitations, the future research will be aimed in several directions. One research direction may be targeted at respondents over the age of 15 (high school and college level education) and can include a larger scope of papers. Another future research direction involves considering the involvement of a wider expert team in the analysis of a larger number of papers. One possible future research direction could include defining subcategories so that the percentage of "not specified" data is reduced.

6. Conclusions

This study presents a systematic review of empirical studies in academic journals between January 2015 and January 2020. The purpose of this study is to review, analyse, and classify data from selected papers (research content areas: programming languages, computer thinking, algorithmic thinking, game design, robotics programming; participants in the studies were between 6 and 15 years old). It is important to note that the largest number of papers belongs to the field of CT, where two papers had students with special needs as participants (field: CT and game design). It was found that the number of CT works increased significantly in recent years, and that including CT in curricula has received positive comments from scientists in numerous countries ([74]; (p. 308).

In general, analysed paper indicates that students showed a satisfactory degree of progress in learning programming language, game design and robotics programming. On the practical level, teachers are the keystone in the implementation of the selected content areas. It was noticed that the students began to develop the ability of computer thinking through their active participation in the course of their teaching (by implementing programming languages, game design and robotics programming exercises on a computer). The papers mainly focused on programming skills training and CT, while some papers applied Project-Based Learning as part of their research workshops.

In the process of analysing the papers, it was noticed that some papers lack detailed descriptions in their research methodology. That is, they do not provide a complete background description of the research. The most common shortcomings are: lack of data on the age and number of participants

 doi:10.20944/preprints202301.0505.v1

and the length of the workshop, which, according to [75], can impede analysing individual articles and comparison between different articles.

Given research, from the point of view of the analysed works, describes a promising result in selected content areas: programming languages, computer thinking, algorithmic thinking, game design and robotics programming, for the participants between 6 and 15 ages. The future work, mentioned in the previous section, should address potential research limitations.

Supplementary Materials: The following supporting information can be downloaded at the website of this paper posted on Preprints.org

Author Contributions: Rastovac designed the review process; Cvetkovic, Viduka and Basic searched, screened the reviewed literature. Rastovac categorized the reviewed literature; Viduka and Basic analyzed the data; Rastovac and Cvetkovic wrote the manuscript; Viduka and Basic revised the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest: The authors declare that they have no conflict of interest.

7. Appendix Available data about studies (N = 46).

Study	Participants and education level	Country (first author)	Learning domain	Teaching tools	Research questions	Hypothesis	Pre/post-test	Interviews	With control group	Course duration	Research design	Previous experience	Project and grant	Research purpose
Computers & Education														
Howland and Good (2015) [124].	55 young people aged 12-13, one secondary school,	The United Kingdom	Game creation	Flip	Yes	No	Yes	Yes	No	8 weeks (2 lesson per week, each lesson 53 min.)	Not specified	No	Grant EP/G006989/1 from the Engineering and Physical Sciences Research Council.	How young people can use commercial game creation software to develop their own 3D video games.
Sáez-López, Román-González, and Vázquez-Cano (2016) [125].	107 primary school students from 5th to 6th grade, five different schools	Spain	Visual Programming	Scratch	No	No	Yes	No	Yes	The academic years 2013-14 and 2014-15 in 20 one-hour sessions	Mixed	Not specified	Not specified	To evaluate the use of Scratch in school lessons as an introduction to programming for total novices, in a younger age group at primary school.
Snodgrass Israel and Reese	2 students who had different disabilities,	USA	CT activities within visual	Scratch	Yes	Yes	No	Yes	No	One unit at the end of the Spring	Not specified	No	Not specified	Examine the participation of students with disabilities and

(2016) [98].	4th and 5th grade, one elementary school		Programming							2015 school year, 45 min/week				their support needs during computing instruction.
Çakır, Gass, Foster, and Lee (2017) [114].	21 girls in grades five through eight, one middle school	Turkey and USA	Game-design	Unity's 2D development tools and C# (one or two functions and variables)	No	No	Yes	Yes	No	A full day event during the weekend, two workshops	Mixed	Yes	A grant from the Entertainment Software Association Foundation (ESAF).	To help young girls explore a sense of identity as a game designer was through the introductions of female role models in the game industry.
Chen, Shen, Barth-Cohen, Jiang, Huang, and Eltoukhy (2017) [58].	121 students 5th grade, one an elementary school	USA	CT and Robotics Programming	Text-based and visual programming language (similar to Scratch)	Yes	No	Yes	No	No	Six months, between 45 and 60 min per week	Not specified	Yes	Supported by a grant from the Entertainment Software Association Foundation.	Develope an instrument to assess fifth grade students' CT.
Durak and Saritepeci (2018) [122].	152 students, 5th-12th grade (21,7% of them were secondary sch.), different school	Turkey	CT	Not specified	Yes	Yes	No	No	No	2015/16 school year	Quantitative	Not specified	Not specified	Determine how much various variables explain students' (CT) skills.
Hsu and Wang (2018) [126].	242 students 4th-grade, one elementary school	Taiwan	Puzzle-based game learning system, algorithmic thinking skills	TGTS (Turtle Graphics Tutorial System)	Yes	No	Yes	No	Yes	Ten weeks (once a week)	Quantitative	No	Not specified	Examine the effects of using game mechanics and a student-generated questions strategy to promote algoritm. thinking skills in TGTS.
Kong, Chiu, and Lai (2018) [37].	287, 4th to 6th grades, one school	Hong Kong	CT- part of a larger project	Not specified	Yes	Yes	No	No	No	Not specified	Quantitative	No	Project from the Hong Kong Jockey	Promote CT education among primary schools.

			that aims to promote CT education										Club Charities Trust	
Città et al. (2019) [127].	92 students 1st to 5th grade (6 to 10 years), one school	Italy	Mental rotation and CT	LEGO, "My Robotic Friends" and "Graph Paper Programming" lessons from code.org	Yes	Yes	No	No	No	Two 90-min sessions in each class	Not specified	Not specified	Not specified	Contextualize CT and programming concepts in the field of Enactivism.
Zhao and Shute (2019) [121].	69 eighth grade students (one middl school)	USA	Components of CT skills: Algorithmic thinking and Conditional logic	Video game Penguin Go	Yes	Yes	Yes	No	Yes	Three 60-min sessions during three weeks in 2017	Qualitative	No	Not specified	Investigate the cognitive and attitudinal impacts of playing a video game that targeted the development of CT skills among middle school students.
Schlegel, et al. (2019) [117].	190 students, 64 were in the program both years, 3th to 5th-grade (one elementary school)	USA	Basic programming through a block-based interface	Not specified	No	Yes	Yes	No	No	Two academic school years (2015-2016, 2016-2017)	Quantitative	Not specified	NSF grant #DRL-1433770	Whether engaging in Making led to changes in self efficacy, interest, and identification with both Making and science in elementary school.
Hsu, Chang, & Hung (2018) [74].	Review article	Taiwan	A meta-review of the studies published in academic journals from 2006 to 2017 was conducted to analyze application courses, adopted learning strategies, and course categories of CT education.										Database SCOPUS, 1133 articles.	
Xia and Zhong (2018) [88].	Review article	China	This paper aims to review high-qualified empirical studies on teaching and learning robotics content knowledge in K-12 and explore future research										22 SSCI journal papers are included in this review.	

			perspectives of robotics education (RE) based on the reviewed papers.										
Popat and Starkey (2019) [65].	Review article	New Zealand	This study reviewed research to analyse educational outcomes for children learning to code at school.									Identified 172 potentially relevant research articles, ten articles were used in the review and included quantitative data.	
Zhang and Nouri (2019) [75].	Review article	Sweden	This systematic review presents a synthesis of 55 empirical studies, providing evidence of the development of computational thinking through programming in Scratch.									Systematic overview of CT education for K-9.	
						Computers in Human Behavior							
Kalelioğlu, F. (2015) [128].	32 primary school students, 10 years, one school	Turkey	Teaching programming skills (block-code)	Code.org site (The Maze, The Artist 2)	Yes	No	Yes	Yes	Yes	Five-week (one hour per week)	Mixed	No	Not specified	Explore the effects of code.org programming on 4th grade primary school students' reflective thinking skills towards problem solving skills.
Zhong, Wang, and Chen (2016) [129].	154, 6th grade pupils, one a primary School	China	Programming	Alice	Yes	No	Yes	Yes	No	13 weeks, in the 2015 spring semester	Quantitative	Yes	The project "Collaborative Innovation Center for Talent Cultivating Mode in Basic Education.	Explore the impacts of two social factors on pair programming effectiveness.
Román-González et al. (2017) [36].	1251 Spanish students from 5th to 10th grade, 1110 students 10-15 age (24 different schools)	Spain	CT	Code.org, site (The Maze, The Canvas)	No	Yes	No	No	No	Elective subject of Computer Science, which is held twice a week (1 h each)	Quantitative	No	Not specified	Provide a new instrument for measuring CT and additionally giving evidence of the correlations between CT and other well-established psychological constructs in the study of cognitive abilities.
Ruggiero and Green	11 students, 14 - 17 age (average 14	United	Design game, the	Not specified	No	No	No	No	No	Six-months (30	Quantitative	No	Not specified	Draw from the game iterations a list of

(2017) [99].	age). students have of special needs young people, secondary school - special needs	Kingdom	Project Tech							workshops, 90 min. per workshop)				empirically grounded problem solving attributes that are associated with digital game design in a special needs classroom.
Pérez-Marín, Hijón-Neira, Bacelo, and Pizarro (2018) [130].	132, primary education students 4th – 5th grade (9–12 years in age), more than 32 schools	Spain	CT, a methodology based on metaphors	Scratch	Yes	Yes	Yes	No	No	6 weeks,1 h per week	Quantitative	No	Research funded by the projects TIN 2015-66731-C2-1-R and S2013/ICE-2715.	Analyse whether MECOPROG has an impact on the students' programming knowledge and whether it can improve computational thinking in students.
Basogain, Olabe, Olabe and Rico (2018) [103].	No Number, students of primary and secondary education (10-15 years old), 21 schools	Spain	CT	Scratch and Alice	No	No	No	Yes	No	Study-1, April-June 2016; and Study-2, December - 2016/March-2017, 10 sessions, each lasting 2 hours	Not specified	No	The Research Development Grants of the University Basque System (2016-18)	Processes of CT aided by the visual programming environments.
Román-González et al. (2018) [116].	1251, 5th to 10th grade, 24 different schools	Spain	CT	The Maze, The Canvas	Yes	Yes	Yes	No	No	Optional Computer Science contents (in Primary School), with a frequency of twice a week.	Quantitative	No	Not specified	To extend the nomological network of CT with non-cognitive factors, through the study of the correlations between CT, self-efficacy and the several dimensions from the 'Big Five' model of human personality.
Cheng, G. (2019) [131].	431 students in 38	Hong Kong	Visual programming	App Inventor	No	Yes	No	Yes	No	Between December 2015 and	Mixed	No	Not specified	Designing an extension of the technology

	primary schools.		environment (VPE)							March 2016, 1 to 5 hours on programming activities in the last 4 weeks.				acceptance model to identify determinants influencing boys' and girls' behavioural intention to use VPE in the primary school context.
Yücel and Rızvanoğlu (2019) [123].	16 children (age between 11 and 14), one middle school	Turkey	A code learning game	Code Combat game	Yes	Yes	Yes	Yes	No	Play the first 10 levels maximum 1 h.	Mixed	No	Not specified	Provide insights about the first-time user experience in a home environment of 16 middle school children with a code learning game named "Code Combat".
Papavlasopoulou, Giannakos, and Jaccheri (2019) [132].	44 children (8–17) -cycle 1, 105 children (13–16) years-cycle 2, 8 girls (10-14) years- cycle 3, one school	Norway	A block-based programming environment and collaboratively created a socially meaningful artifact (i.e., a game).	Scratch	Yes	No	Yes	Yes	No	Over the two years (cycle 1 two and cycle 2 six weeks, cycle 3 two day)	Mixed	No	The European Commission's Horizon 2020 SwafS-11-2017 Program (Project Number: 787476)	In this study, investigate children's learning experience as they constructed their own knowledge by using a digital programming tool (Scratch) and collaboratively creating socially meaningful artifacts: games.
							Journal of Educational Computing Research							
Akpinar and Aslan (2015) [133].	18 fifth grade and 12 sixth graders (12–14) age, one middle school	Turkey	Programming-video game	Scratch	Yes	No	Yes	No	No	Nine workshops (two 40-min, four 30 min of hands-on Scratch programming instruction and 50 min of	Quantitative	Not specified	Not specified	Explore the effects of middle school students' development of video games with Scratch on their achievement of independent events in probability.

										developing games)				
Zhong, Wang, Chen, and Li (2016) [108].	144 pupils sixth grade, one primary school	China	CT	3D programming language Alice 2.4	Yes	No	Yes	No	No	(18 weeks, 40 minutes per week)	Not specified	Yes	(13YJC880121) granted by Chinese Ministry of Education.	Propose what types of tasks could be made accessible and meaningful for assessing students' CT.
Jakoš and Verber (2017) [134].	107 sixth grade pupils, three primary schools	Slovenia	Learning programing	Game "Aladdin and his flying carpet"	Yes	Yes	Yes	Yes	No	2 months – 2 weeks for Phases 1 and 3, 45 min, and 1 month for Phase 2, 135 min	Not specified	No	Not specified	Investigate the effectiveness of using educational games for learning basic programing skills.
Tran, Y. (2019) [135].	Over 200 students, five elementary schools	USA	CT	Blockly programming language-code.org	Yes	No	Yes	Yes	No	10-weeks, an hour each week	Mixed	Yes	Not specified	Pre- and posttest changes in CT using adapted lessons from code.org's.
Vasilopoulos and Van Schaik (2019) [136].	66 third-grade students (mean age 14), one secondary school	Greece	Visual programming	Koios programming language	No	Yes	No	No	No	Nine lessons (one per week)	Quantitative	No	Not specified	Produce a programming environment that could serve as an efficient tool for improving the teaching and learning of introductory programming in Greece.

Journal of Computer Assisted Learning

Falloon, G. (2016) [137].	32, 5- and 6-year-old students, one primary school	New Zealand	CT	Scratch Jnr	Yes	No	No	No	No	4 sessions, 25-40 min, February 2015 to April 2015	Not specified	No	Not specified	Students thinking skills when they have completed the basics of programming
Witherspoon, Schunn, Higashi, and Shoop (2018) [120].	136 (6th–8th grade), two middle school	USA	Virtual robotics curriculum, visual programming	ROBOTCGGraphical, VEX IQ robots	Yes	No	Yes	No	No	6- to 9-week course, treba proveriti	Not specified	Not specified	Grant/Award Number: 1418199; National Science Foundation.	Effects of units with different programming content within a virtual robotics context on both learning gains and motivational

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 January 2023 doi:10.20944/preprints202301.0505.v1

			language											changes in middle school (6th–8th grade) robotics classrooms.
Benton, Kalas, Saunders, Hoyles, and Noss (2018) [138].	181 pupils (aged 10–11), from 6 primary schools	UK	Computational and mathematical thinking, visual blocks-based language	Scratch Maths	No	No	No	Yes	No	2-year, computing and mathematics curriculums, six modules (three per year)	Not specified	Yes	Education Endowment Foundation, the SM project schools.	Develop computational and mathematical thinking skills through learning to program.
Yildiz Durak, H. (2018a) [111].	62 fifth-grade students, one secondary school.	Turkey	Digital story use in programming teaching	Scratch	Yes	Yes	Yes	Yes	Yes	10-week application process, course name Information Technologies and Software	Mixed	No	Not specified	Determine the effects and experiences of the use of digital story design activities in teaching applications of programming on academic achievement.
Yildiz Durak, H. (2018b) [115].	371 students, 5th to 8th grade, two middle schools	Turkey	flipped learning readiness (FLR), programming	Scratch	No	Yes	No	No	No	15-week programming teaching during the spring semester of 2017	Quantitative	Yes	Not specified	Investigate the effect of students' (FLR) on engagement, programming self-efficacy, attitude towards programming.
				Educational Technology and Society										
Su, Huang, Yang, Ding, and Hsieh (2015) [112].	37 students sixth-grade (average age 12), one elementary school	Taiwan	Programming course	Scratch	Yes	No	Yes	Yes	Yes	Three hours per week, 4-month, March to June 2013	Quantitative	No	Not specified	Explore the effects of annotations and homework on learning achievement.
Basawapatna, A. (2016) [113].	45 7th grade students, one middle school	USA	visual programming, game design, IPAK JE CT, Pattern	Simulation Creation Toolkit	No	No	No	No	No	4 days, 16 pattern implementations	Not specified	Yes	The National Science Foundation under Grant Numbers 0833612,	Design game in the integration of Computational Thinking activities through simulation construction in

			Programming										0848962, 1138526.	the classroom environment.
Zhong, Wang, Chen, and Li (2017) [102].	150 pupils 6th grade, one primary school	China	Programming course	Alice and Scratch	Yes	No	Yes	Yes	No	13 weeks	Not specified	Yes	The project "Collaborative Innovation Center for Talent Cultivating Mode in Basic Education.	Compare the learning achievement and attitude in different periods of switching roles.
Wang, Hwang, Liang, and Wang (2017) [139].	166 ninth graders, one junior high school	Taiwan	Visual programming	Scratch	Yes	No	Yes	No	Yes	10 weeks of two hours per week	Quantitative	Not specified	Supported in part by the Ministry of Science and Technology, China, numbers NSC 102 2511 S 011 007 MY3 and MOST 104.	Evaluate the studen ts' competence of using the programming statements and operations to develop Scratch programs based on the topics specified by the teacher.

Educational Technology Research and Development

Akcaoglu and Green (2019) [140].	19 6th grade students (average = 11 age), one middle school	USA	Game design course	Microsoft Kodu software	Yes	No	Yes	Yes	Yes	A once-a-week, hour-long session, the school year	Mixed	No	Not specified	If middle school students who attended a game design course showed improvements in their system analysis and design skills.
Sáez-López, Sevillano-García, and Vazquez-Cano (2019) [118].	93 sixth-grade students, four primary schools	Spain	Robotics and programming	mBot	Yes	Yes	Yes	No	Yes	Academic year 2016–2017	Quantitative	Not specified	Not specified	Analyze the potential of visual block programming and robotics for use in primary education.
Strawhacker and Bers (2019) [141].	57 K-2nd grade participant children (One Kindergarten, one 1th and 2th	USA	Programming	ScratchJr	Yes	No	No	No	No	Twice-weekly 1-h lessons over 6 weeks	Mixed	No	Grant No. DRL1118664.	Cognitive domain that young children leverage when learning programming for the first time.

	grade classroom)												
					Interactive Learning Environments								
Garneli and Chorianopoulos (2017) [142].	34-students, 15 age, third grade, one middle school	Greece	CT over video-game	Scratch	Yes	Yes	Yes	Yes	No	Five weeks, two-hour sessions per week	Qualitative	No	Not specified	Potential effects of constructing video games and simulations on student learning.
Chiang and Qin (2018) [143].	89 seventh grade students, one secondary school	China	Game-based construction learning	Scratch	Yes	No	Yes	Yes	No	A ten-week period during a weekly 45-minute session	Quantitative	No	By Beijing Wangjing Experiment School, grant number KJHX2015322.	Examine the impacts of Scratch-based games made by seventh grade students to solve equations, on their equation-solving performance and attitudes towards learning mathematics with the assistance of technology.
					British Journal of Educational Technology								
Costa and Miranda (2017) [66].	Review article	Portugal	A systematic review of the literature include 232 studies published between the years 2000 and 2014 in the main databases.										The effectiveness of the use of Alice software in programming learning when compared to the use of a conventional programming language.
Lindberg, Laine, and Haaranen (2019) [82].	Review article	Belgium	An investigation on the guidelines on programming education in K-12 in seven countries.										Review of existing acquirable games that utilize programming topics in their gameplay was conducted by searching popular game stores.

References

1. Chen, X.; Zou, D.; Cheng, G.; Xie, H. Detecting Latent Topics and Trends in Educational Technologies over Four Decades Using Structural Topic Modeling: A Retrospective of All Volumes of Computers & Education. *Comput Educ*, **2020**, *151*, 103855. https://doi.org/10.1016/J.COMPEDU.2020.103855.
2. Berrett, B.; Murphy, J.; Sullivan, J. Administrator Insights and Reflections: Technology Integration in Schools. *Qualitative Report - ERIC*, **2012**, 200–221.
3. Hubwieser, P.; Armoni, M.; Giannakos, M. N.; Mittermeir, R. T. Perspectives and Visions of Computer Science Education in Primary and Secondary (K-12) Schools. *ACM Transactions on Computing Education (TOCE)*, **2014**, *14* (2). https://doi.org/10.1145/2602482.
4. e-skills UK Board members. *Technology Insights 2012*; London, 2012.
5. Livingstone, I.; Hope, A. Next Gen: Transforming the UK into the World's Leading Talent Hub for the Video Games and Visual Effects Industries; NESTA, 2011.
6. Crick, T.; Sentance, S. Computing at School: Stimulating Computing Education in the UK. *Proceedings - 11th Koli Calling International Conference on Computing Education Research, Koli Calling'11*, **2011**, 122–123. https://doi.org/10.1145/2094131.2094158.

7. Webb, M.; Davis, N.; Bell, T.; Katz, Y.; Reynolds, N.; Chambers, D. P.; Sysło, M. M. Computer Science in K-12 School Curricula of the 2lst Century: Why, What and When? *Educ Inf Technol (Dordr)*, **2017**, *22* (2), 445–468. https://doi.org/10.1007/S10639-016-9493-X/TABLES/4.
8. Royal Society. Shut down or restart? The way forward for computing in UK schools https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf (accessed Jan 2, 2023).
9. Wilson, C.; Sudol, L. A.; Stephenson, C.; Stehlik, M. Running on Empty. **2010**. https://doi.org/10.1145/3414583.
10. Seehorn, D.; Carey, S.; Fuschetto, B.; Lee, I.; Moix, D.; O'Grady-Cunniff, D.; et al. CSTA K–12 computer science standards https://scholar.google.com/scholar?hl=sr&as_sdt=0%2C5&q=CSTA+K%E2%80%9312+computer+science+standards&btnG= (accessed Jan 1, 2023).
11. Europe, J. I. ACM Europe Working Group on Informatics Education.(2013). Informatics education: Europe cannot afford to miss the boat: Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education, **2014**.
12. Falkner, K.; Vivian, R.; Falkner, N. The Australian Digital Technologies Curriculum: Challenge and Opportunity; 2014.
13. Bell, T.; Andreae, P.; Robins, A. Computer Science in NZ High Schools: The First Year of the New Standards. *SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, **2012**, 343–348. https://doi.org/10.1145/2157136.2157240.
14. Tucker, A.; Deek, F.; Jones, J.; Hayden, C.; School, H.; Mccowan, D.; Stephenson, C.; Verno, A. A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee Chair ACM K-12 Task Force Curriculum Committee Committee Members; 2003. https://doi.org/10.1145/2593247.
15. Lewis, C.; Olson, G. Can principles of cognition lower the barriers to programming? | Empirical studies of programmers: second workshop https://dl.acm.org/doi/10.5555/54968.54984 (accessed Jan 1, 2023).
16. Myers, B. A. Taxonomies of Visual Programming and Program Visualization. *J Vis Lang Comput*, **1990**, *1* (1), 97–123. https://doi.org/10.1016/S1045-926X(05)80036-9.
17. Kurland, D. M.; Pea, R. D. Children's Mental Models of Recursive Logo Programs. *http://dx.doi.org/10.2190/JV9Y-5PD0-MX22-9J4Y*, **1995**, *1* (2), 235–243. https://doi.org/10.2190/JV9Y-5PD0-MX22-9J4Y.
18. Clements, D.; Meredith, J. S. Research on Logo: Effects and Efficacy. *Journal of Computing in Childhood Education*, **1993**, *4* (4), 263–290.
19. An, J.; Park, N. Computer Application in Elementary Education Bases on Fractal Geometry Theory Using LOGO Programming. In *Lecture Notes in Electrical Engineering*; Springer, Dordrecht, 2012; Vol. 107 LNEE, pp 241–249. https://doi.org/10.1007/978-94-007-2598-0_26/COVER.
20. Pardamean, B. Enhancement of Creativity through Logo Programming. *Am J Appl Sci*, **2014**, *11* (4), 528–533. https://doi.org/doi: 10.3844/ajassp.2014.528.533.
21. Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. Scratch. *Commun ACM*, **2009**, *52* (11), 60–67. https://doi.org/10.1145/1592761.1592779.
22. Kalelioğlu, F.; Gülbahar, Y. The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education - An International Journal*, **2014**, *13* (1), 33–50.
23. Armoni, M.; Meerbaum-Salant, O.; Ben-Ari, M. From Scratch to Real Programming. *ACM Transactions on Computing Education (TOCE)*, **2015**, *14* (4). https://doi.org/10.1145/2677087.
24. Mladenović, M.; Krpan, D.; Mladenović, S. Learning Programming from Scratch. *The turkish online journal of educational technology*, **2017**, 419–427.
25. Sykes, E. R. Determining the Effectiveness of the 3D Alice Programming Environment at the Computer Science I Level. *http://dx.doi.org/10.2190/J175-Q735-1345-270M*, **2016**, *36* (2), 223–244. https://doi.org/10.2190/J175-Q735-1345-270M.
26. Kelleher, C.; Pausch, R.; Kiesler, S. Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. In *Conference on Human Factors in Computing Systems - Proceedings*; Association for Computing Machinery, 2007; pp 1455–1464. https://doi.org/10.1145/1240624.1240844.
27. Rodger, S. H.; Bashford, M.; Dyck, L.; Hayes, J.; Liang, L.; Nelson, D.; Qin, H. Enhancing K-12 Education with Alice Programming Adventures. In *ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*; 2010; pp 234–238. https://doi.org/10.1145/1822090.1822156.
28. Hromkovič, J.; Kohn, T.; Komm, D.; Serafini, G. Combining the Power of Python with the Simplicity of Logo for a Sustainable Computer Science Education. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **2016**, *9973 LNCS*, 155–166. https://doi.org/10.1007/978-3-319-46747-4_13/COVER.

29. Tsukamoto, H.; Takemura, Y.; Nagumo, H.; Ikeda, I.; Monden, A.; Matsumoto, K. I. Programming Education for Primary School Children Using a Textual Programming Language. In *Proceedings - Frontiers in Education Conference, FIE*; Institute of Electrical and Electronics Engineers Inc., 2015; Vol. 2015, pp 1–8. https://doi.org/10.1109/FIE.2015.7344187.
30. Price, T. W.; Barnes, T. Comparing Textual and Block Interfaces in a Novice Programming Environment. In *ICER 2015 - Proceedings of the 2015 ACM Conference on International Computing Education Research*; Association for Computing Machinery, Inc, 2015; pp 91–100. https://doi.org/10.1145/2787622.2787712.
31. Noone, M.; Mooney, A. First Programming Language: Visual or Textual? **2017**. https://doi.org/10.48550/arxiv.1710.11557.
32. Wing, J. M. Computational Thinking and Thinking about Computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **2008**, *366* (1881), 3717–3725. https://doi.org/10.1098/RSTA.2008.0118.
33. Lu, J. J.; Fletcher, G. H. L. Thinking about Computational Thinking. In *SIGCSE'09 - Proceedings of the 40th ACM Technical Symposium on Computer Science Education*; 2009; pp 260–264. https://doi.org/10.1145/1508865.1508959.
34. Dede, C.; Mishra, P.; Voogt, J. Working Group 6: Advancing Computational Thinking in 21st Century Learning. In *International summit on ICT in education*; 2013; pp 1–6.
35. Voogt, J.; Fisser, P.; Good, J.; Mishra, P.; Yadav, A. Computational Thinking in Compulsory Education: Towards an Agenda for Research and Practice. *Educ Inf Technol (Dordr)*, **2015**, *20* (4), 715–728. https://doi.org/10.1007/S10639-015-9412-6/TABLES/2.
36. Román-González, M.; Pérez-González, J. C.; Jiménez-Fernández, C. Which Cognitive Abilities Underlie Computational Thinking? Criterion Validity of the Computational Thinking Test. *Comput Human Behav*, **2017**, *72*, 678–691. https://doi.org/10.1016/J.CHB.2016.08.047.
37. Kong, S. C.; Chiu, M. M.; Lai, M. A Study of Primary School Students' Interest, Collaboration Attitude, and Programming Empowerment in Computational Thinking Education. *Comput Educ*, **2018**, *127*, 178–189. https://doi.org/10.1016/J.COMPEDU.2018.08.026.
38. Brennan, K.; Resnick. New Frameworks for Studying and Assessing the Development of Computational Thinking. In *In Proceedings of the 2012 annual meeting of the American educational research association*; 2012; Vol. 1, pp 1–25.
39. Papert, S. Teaching Children Thinking. *Programmed Learning and Educational Technology*, **1972**, *9* (5), 245–255. https://doi.org/10.1080/1355800720090503.
40. Denning, P. J. The Profession of ITBeyond Computational Thinking. *Commun ACM*, **2009**, *52* (6), 28–30. https://doi.org/10.1145/1516046.1516054.
41. Seiter, L.; Foreman, B. Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *ICER 2013 - Proceedings of the 2013 ACM Conference on International Computing Education Research*; 2013; pp 59–66. https://doi.org/10.1145/2493394.2493403.
42. Heintz, F.; Mannila, L.; Farnqvist, T. A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education. In *Proceedings - Frontiers in Education Conference, FIE*; Institute of Electrical and Electronics Engineers Inc., 2016; pp 1–9. https://doi.org/10.1109/FIE.2016.7757410.
43. Dolgopolovas, V.; Dagiene V. On the Future of Computational Thinking Education: Moving beyond the Digital Agenda, a Discourse Analysis Perspective. *Sustainability*, **2021**, *13* (24). https://doi.org/10.3390/su132413848.
44. Futschek, G. Algorithmic Thinking: The Key for Understanding Computer Science. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), **2006**, 4226 LNCS, 159–168. https://doi.org/10.1007/11915355_15/COVER.
45. Futschek, G.; Moschitz, J. Learning Algorithmic Thinking with Tangible Objects Eases Transition to Computer Programming. *Springer*, **2011**, *7013 LNCS*, 155–164. https://doi.org/10.1007/978-3-642-24722-4_14.
46. Katai, Z. The Challenge of Promoting Algorithmic Thinking of Both Sciences- and Humanities-Oriented Learners. *J Comput Assist Learn*, **2015**, *31* (4), 287–299. https://doi.org/10.1111/JCAL.12070.
47. Kanaki, K.; Kalogiannakis, M.; Poulakis, E.; Politis, P. Investigating the Association between Algorithmic Thinking and Performance in Environmental Study. *Sustainability 2022, Vol. 14, Page 10672*, **2022**, *14* (17), 10672. https://doi.org/10.3390/SU141710672.
48. Wilson, A.; Hainey, T.; Connolly T. Evaluation of Computer Games Developed by Primary School Children to Gauge Understanding of Programming Concepts. In *In European Conference on Games Based Learning*; 2012; pp 549–558.
49. Smith, N.; Sutcliffe, C.; Sandvik, L. Code Club: Bringing Programming to UK Primary Schools through Scratch. In *SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education*; Association for Computing Machinery, 2014; pp 517–522. https://doi.org/10.1145/2538862.2538919.
50. Basawapatna, A.; Koh, K.; Repenning, A. Using Scalable Game Design to Teach Computer Science from Middle School to Graduate School. In *In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*; 2010; pp 224–228. https://doi.org/10.1145/1822090.1822154.

51. Tsalapatas, H.; Heidmann, O.; … R. A.-S. B. of the; 2012, undefined. Game-Based Programming towards Developing Algorithmic Thinking Skills in Primary Education. *Scientific Bulletin of the Petru Maior University of Targu Mures*, **2012**, *9* (1), 56–63.
52. Werner, L.; Campe, S.; Denner, J. Children Learning Computer Science Concepts via Alice Game-Programming. In *SIGCSE'12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*; 2012; pp 427–432. https://doi.org/10.1145/2157136.2157263.
53. Denner, J.; Werner, L.; Ortiz, E. Computer Games Created by Middle School Girls: Can They Be Used to Measure Understanding of Computer Science Concepts? *Comput Educ*, **2012**, *58* (1), 240–249. https://doi.org/10.1016/J.COMPEDU.2011.08.006.
54. Gruenbaum, P. Undergraduates Teach Game Programming Using Scratch. *Computer (Long Beach Calif)*, **2014**, *47* (02), 82–84. https://doi.org/10.1109/MC.2014.49.
55. Mataric, M. J. Robotics Education for All Ages. In Proc. AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education; 2004; pp 22–24.
56. Johnson, J. Children, Robotics, and Education. *Artificial Life and Robotics 2003 7:1*, **2003**, *7* (1), 16–21. https://doi.org/10.1007/BF02480880.
57. Leonard, J.; Buss, A.; Gamboa, R.; Mitchell, M.; Fashola, O. S.; Hubert, T.; Almughyirah, S. Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. *J Sci Educ Technol*, **2016**, *25* (6), 860–876. https://doi.org/10.1007/S10956-016-9628-2/TABLES/7.
58. Chen, G.; Shen, J.; Barth-Cohen, L.; Jiang, S.; Huang, X.; Eltoukhy, M. Assessing Elementary Students' Computational Thinking in Everyday Reasoning and Robotics Programming. *Comput Educ*, **2017**, *109*, 162–175. https://doi.org/10.1016/J.COMPEDU.2017.03.001.
59. Hirst, A. J.; Johnson, J.; Petre, M.; Price, B. A.; Richards, M. What Is the Best Programming Environment/Language for Teaching Robotics Using Lego Mindstorms? *Artificial Life and Robotics 2003 7:3*, **2003**, *7* (3), 124–131. https://doi.org/10.1007/BF02481160.
60. Shih, B.; Chen, T.; … S. W.-J. of B. S.; 2013, undefined. The Exploration of Applying Lego Nxt in the Situated Science and Technology Learning. *Journal of Baltic Science Education*, **2013**, *12* (1), 73–91.
61. Okita, S. Y. The Relative Merits of Transparency: Investigating Situations That Support the Use of Robotics in Developing Student Learning Adaptability across Virtual and Physical Computing Platforms. *British Journal of Educational Technology*, **2014**, *45* (5), 844–862. https://doi.org/10.1111/BJET.12101.
62. Sapounidis, T.; 2016, S. D.-I. C. E.; 2017, undefined. Educational Robots Driven by Tangible Programming Languages: A Review on the Field. *Springer*, **2017**, *560*, 205–214. https://doi.org/10.1007/978-3-319-55553-9_16.
63. Nemiro, J.; Larriva, C.; Jawaharlal, M. Developing Creative Behavior in Elementary School Students with Robotics. *J Creat Behav*, **2017**, *51* (1), 70–90. https://doi.org/10.1002/JOCB.87.
64. Nikou, S. A.; Economides, A. A. Mobile-Based Assessment: A Literature Review of Publications in Major Referred Journals from 2009 to 2018. *Comput Educ*, **2018**, *125*, 101–119. https://doi.org/10.1016/J.COMPEDU.2018.06.006.
65. Popat, S.; Starkey, L. Learning to Code or Coding to Learn? A Systematic Review. *Comput Educ*, **2019**, *128*, 365–376. https://doi.org/10.1016/J.COMPEDU.2018.10.005.
66. Costa, J. M.; Miranda, G. L. Relation between Alice Software and Programming Learning: A Systematic Review of the Literature and Meta-Analysis. *British Journal of Educational Technology*, **2017**, *48* (6), 1464–1474. https://doi.org/10.1111/BJET.12496.
67. Robins, A.; Rountree, J.; Rountree, N. Learning and Teaching Programming: A Review and Discussion. *Computer science education*, **2010**, *13* (2), 137–172. https://doi.org/10.1076/CSED.13.2.137.14200.
68. Wang, Y.; Li, H.; Feng, Y.; Jiang, Y.; Liu, Y. Assessment of Programming Language Learning Based on Peer Code Review Model: Implementation and Experience Report. *Comput Educ*, **2012**, *59* (2), 412–422. https://doi.org/10.1016/J.COMPEDU.2012.01.007.
69. Sorva, J.; Karavirta, V.; Malmi, L. A Review of Generic Program Visualization Systems for Introductory Programming Education. *ACM Transactions on Computing Education (TOCE)*, **2013**, *13* (4). https://doi.org/10.1145/2490822.
70. Noone, M.; Mooney, A. Visual and Textual Programming Languages: A Systematic Review of the Literature. *Journal of Computers in Education*, **2018**, *5* (2), 149–174. https://doi.org/10.1007/S40692-018-0101-5/FIGURES/3.
71. Luxton-Reilly, A.; Simon; Albluwi, I.; Becker, B. A.; Giannakos, M.; Kumar, A. N.; Ott, L.; Paterson, J.; Scott, M. J.; Sheard, J.; et al. Introductory Programming: A Systematic Literature Review. In *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*; Association for Computing Machinery, 2018; pp 55–106. https://doi.org/10.1145/3293881.3295779.
72. Sapounidis, T.; Demetriadis, S.; Papadopoulos, P. M.; Stamovlasis, D. Tangible and Graphical Programming with Experienced Children: A Mixed Methods Analysis. *Int J Child Comput Interact*, **2019**, *19*, 67–78. https://doi.org/10.1016/J.IJCCI.2018.12.001.

73. Scherer, R.; Siddiq, F.; Sánchez Viveros, B. A Meta-Analysis of Teaching and Learning Computer Programming: Effective Instructional Approaches and Conditions. *Comput Human Behav*, **2020**, *109*, 106349. https://doi.org/10.1016/J.CHB.2020.106349.
74. Hsu, T. C.; Chang, S. C.; Hung, Y. T. How to Learn and How to Teach Computational Thinking: Suggestions Based on a Review of the Literature. *Comput Educ*, **2018**, *126*, 296–310. https://doi.org/10.1016/J.COMPEDU.2018.07.004.
75. Zhang, L. C.; Nouri, J. A Systematic Review of Learning Computational Thinking through Scratch in K-9. *Comput Educ*, **2019**, *141*, 103607. https://doi.org/10.1016/J.COMPEDU.2019.103607.
76. Grover, S.; Pea, R. Computational Thinking in K–12: A Review of the State of the Field. *Educational researcher*, **2013**, *42* (1), 38–43. https://doi.org/10.3102/0013189X12463051.
77. Kalelioglu, F.; Gulbahar, Y.; Kukul, V. A Framework for Computational Thinking Based on a Systematic Research Review. *BALTIC JOURNAL OF MODERN COMPUTING*, **2016**, *4* (3), 583–596.
78. Lockwood, J.; Mooney, A. Computational Thinking in Education: Where Does It Fit? A Systematic Literary Review. *International Journal of Computer Science Education in Schools*, **2017**, *2* (1), 1–58. https://doi.org/10.48550/arxiv.1703.07659.
79. Moreno-León, J.; Román-González, M.; Robles, G. On Computational Thinking as a Universal Skill: A Review of the Latest Research on This Ability. In *In 2018 IEEE Global Engineering Education Conference (EDUCON)*; 2018; pp 1684–1689. https://doi.org/10.1109/EDUCON.2018.8363437.
80. Berikan, B.; Özdemir, S. Investigating "Problem-Solving With Datasets" as an Implementation of Computational Thinking: A Literature Review. *Journal of Educational Computing Research*, **2020**, *58* (2), 502–534. https://doi.org/10.1177/0735633119845694.
81. Ávila-Pesántez, D.; Rivera, L. A.; Alban, M. S. Approaches for Serious Game Design: A Systematic Literature Review. *The ASEE Computers in Education (CoED) Journal*, **2017**, *8* (3).
82. Lindberg, R. S. N.; Laine, T. H.; Haaranen, L. Gamifying Programming Education in K-12: A Review of Programming Curricula in Seven Countries and Programming Games. *British Journal of Educational Technology*, **2019**, *50* (4), 1979–1995. https://doi.org/10.1111/BJET.12685.
83. Wu, B.; Wang, A. I. A Guideline for Game Development-Based Learning. *International Journal of Computer Games Technology*, **2012**, *2012*, 20. https://doi.org/10.1155/2012/103710.
84. Almeida, M. S. O.; Corrêa Da Silva, F. S. A Systematic Review of Game Design Methods and Tools. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), **2013**, 8215 LNCS, 17–29. https://doi.org/10.1007/978-3-642-41106-9_3.
85. Vahldick, A.; Mendes, A. J.; Marcelino, M. J. A Review of Games Designed to Improve Introductory Computer Programming Competencies. In *Proceedings - Frontiers in Education Conference, FIE*; Institute of Electrical and Electronics Engineers Inc., 2014; Vol. 2015-February, pp 1–7. https://doi.org/10.1109/FIE.2014.7044114.
86. Miljanovic, M. A.; Bradbury, J. S. A Review of Serious Games for Programming. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), **2018**, 11243 LNCS, 204–216. https://doi.org/10.1007/978-3-030-02762-9_21/COVER.
87. Benitti, F. B. V. Exploring the Educational Potential of Robotics in Schools: A Systematic Review. *Comput Educ*, **2012**, *58* (3), 978–988. https://doi.org/10.1016/J.COMPEDU.2011.10.006.
88. Xia, L.; Zhong, B. A Systematic Review on Teaching and Learning Robotics Content Knowledge in K-12. *Comput Educ*, **2018**, *127*, 267–282. https://doi.org/10.1016/J.COMPEDU.2018.09.007.
89. Felicia, A.; Sharif, S. A Review on Educational Robotics as Assistive Tools for Learning Mathematics and Science. *International Journal of Computer Science Trends*, **2014**, *2* (2), 62–84.
90. Scaradozzi, D.; Sorbi, L.; Pedale, A.; Valzano, M.; Vergine, C. Teaching Robotics at the Primary School: An Innovative Approach. *Procedia Soc Behav Sci*, **2015**, *174*, 3838–3846. https://doi.org/10.1016/J.SBSPRO.2015.01.1122.
91. Hong, N. W. W.; Chew, E.; Sze-Meng, J. W. The Review of Educational Robotics Research and the Need for Real-World Interaction Analysis. In *2016 14th International Conference on Control, Automation, Robotics and Vision, (ICARCV) IEEE*; Institute of Electrical and Electronics Engineers Inc., 2016; pp 1–6. https://doi.org/10.1109/ICARCV.2016.7838707.
92. Kubilinskienė, S.; Žilinskienė, I.; Dagienė, V.; Sinkevičius, V. Applying Robotics in School Education: A Systematic Review. *Baltic journal of modern computing*, **2017**, *5* (1), 50–69. https://doi.org/10.22364/BJMC.2017.5.1.04.
93. Kitchenham, B.; Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering; 2007.
94. Chauhan, S. A Meta-Analysis of the Impact of Technology on Learning Effectiveness of Elementary Students. *Comput Educ*, **2017**, *105*, 14–30. https://doi.org/10.1016/J.COMPEDU.2016.11.005.
95. Lye, S. Y.; Koh, J. H. L. Review on Teaching and Learning of Computational Thinking through Programming: What Is next for K-12? *Comput Human Behav*, **2014**, *41*, 51–61. https://doi.org/10.1016/J.CHB.2014.09.012.

96. Jung, S. E.; Won, E. S. Systematic Review of Research Trends in Robotics Education for Young Children. *Sustainability 2018, Vol. 10, Page 905*, **2018**, *10* (4), 905. https://doi.org/10.3390/SU10040905.
97. Farrokhnia, M.; Meulenbroeks, R. F. G.; van Joolingen, W. R. Student-Generated Stop-Motion Animation in Science Classes: A Systematic Literature Review. *J Sci Educ Technol*, **2020**, *29* (6), 797–812. https://doi.org/10.1007/S10956-020-09857-1/TABLES/2.
98. Snodgrass, M. R.; Israel, M.; Reese, G. C. Instructional Supports for Students with Disabilities in K-5 Computing: Findings from a Cross-Case Analysis. *Comput Educ*, **2016**, *100*, 1–17. https://doi.org/10.1016/J.COMPEDU.2016.04.011.
99. Ruggiero, D.; Green, L. Problem Solving through Digital Game Design: A Quantitative Content Analysis. *Comput Human Behav*, **2017**, *73*, 28–37. https://doi.org/10.1016/J.CHB.2017.03.024.
100. Balanskat, A.; Engelhardt, K. Computing Our Future Computer Programming and Coding-Priorities, School Curricula and Initiatives across Europe Computing Our Future-Priorities, School Curricula and Initiatives across Europe Publisher. *European Schoolnet*, **2014**.
101. Moreno-León, J.; Robles, G.; Román-González, M. Code to Learn: Where Does It Belong in the K-12 Curriculum? *Journal of Information Technology Education*, **2016**, *15*, 283–303.
102. Zhong, B.; Wang, Q.; Chen, J.; Li, Y. Investigating the Period of Switching Roles in Pair Programming in a Primary School. *J Educ Techno Soc*, **2017**, *20* (3), 220–233.
103. Basogain, X.; Olabe, M.; Olabe, J.; Behavior, M. R.-C. in H.; 2018, undefined. Computational Thinking in Pre-University Blended Learning Classrooms. *Elsevier*, **2017**. https://doi.org/10.1016/j.chb.2017.04.058.
104. Chao, P. Y. Exploring Students' Computational Practice, Design and Performance of Problem-Solving through a Visual Programming Environment. *Comput Educ*, **2016**, *95*, 202–215. https://doi.org/10.1016/J.COMPEDU.2016.01.010.
105. Erol, O.; Kurt, A. A. The Effects of Teaching Programming with Scratch on Pre-Service Information Technology Teachers' Motivation and Achievement. *Comput Human Behav*, **2017**, *77*, 11–18. https://doi.org/10.1016/J.CHB.2017.08.017.
106. Creswell, J. W. Research Design Qualitative, Quantitative and Mixed Methods Approaches (4th Ed.). Thousand Oaks, CA Sage. - References - Scientific Research Publishing; 2014.
107. Crompton, H.; Burke, D. The Use of Mobile Learning in Higher Education: A Systematic Review. *Comput Educ*, **2018**, *123*, 53–64. https://doi.org/10.1016/J.COMPEDU.2018.04.007.
108. Zhong, B.; Wang, Q.; Chen, J.; Li, Y. An Exploration of Three-Dimensional Integrated Assessment for Computational Thinking. *Journal of Educational Computing Research*, **2016**, *53* (4), 562–590. https://doi.org/10.1177/0735633115608444/ASSET/IMAGES/LARGE/10.1177_0735633115608444-FIG2.JPEG.
109. Garneli, V.; Giannakos, M. N.; Chorianopoulos, K. Computing Education in K-12 Schools: A Review of the Literature. In *IEEE Global Engineering Education Conference, EDUCON*; IEEE Computer Society, 2015; Vol. 2015-April, pp 543–551. https://doi.org/10.1109/EDUCON.2015.7096023.
110. Hubwieser, P.; Giannakos, M. N.; Berges, M.; Brinda, T.; Diethelm, I.; Magenheim, J.; Pal, Y.; Jackova, J.; Jasute, E. A Global Snapshot of Computer Science Education in K-12 Schools. In *In Proceedings of the 2015 ITiCSE on working group reports*; Association for Computing Machinery, Inc, 2015; pp 65–83. https://doi.org/10.1145/2858796.2858799.
111. Yildiz Durak, H. Digital Story Design Activities Used for Teaching Programming Effect on Learning of Programming Concepts, Programming Self-Efficacy, and Participation and Analysis of Student Experiences. *J Comput Assist Learn*, **2018**, *34* (6), 740–752. https://doi.org/10.1111/JCAL.12281.
112. Su, A.; Huang, C.; Yang, S.; Ding, T. Effects of Annotations and Homework on Learning Achievement: An Empirical Study of Scratch Programming Pedagogy. *J Educ Techno Soc*, **2015**, *18* (4), 331–341.
113. Basawapatna, A. Alexander Meets Michotte: A Simulation Tool Based on Pattern Programming and Phenomenology. *J Educ Techno Soc*, **2016**, *19* (1), 277–291.
114. Akkuş Çakır, N.; Gass, A.; Foster, A.; Lee, F. J. Development of a Game-Design Workshop to Promote Young Girls' Interest towards Computing through Identity Exploration. *Comput Educ*, **2017**, *108*, 115–130. https://doi.org/10.1016/J.COMPEDU.2017.02.002.
115. Yildiz Durak, H. Flipped Learning Readiness in Teaching Programming in Middle Schools: Modelling Its Relation to Various Variables. *J Comput Assist Learn*, **2018**, *34* (6), 939–959. https://doi.org/10.1111/JCAL.12302.
116. Román-González, M.; Pérez-González, J. C.; Moreno-León, J.; Robles, G. Extending the Nomological Network of Computational Thinking with Non-Cognitive Factors. *Comput Human Behav*, **2018**, *80*, 441–459. https://doi.org/10.1016/J.CHB.2017.09.030.
117. Schlegel, R. J.; Chu, S. L.; Chen, K.; Deuermeyer, E.; Christy, A. G.; Quek, F. Making in the Classroom: Longitudinal Evidence of Increases in Self-Efficacy and STEM Possible Selves over Time. *Comput Educ*, **2019**, *142*, 103637. https://doi.org/10.1016/J.COMPEDU.2019.103637.

118. Sáez-López, J. M.; Sevillano-García, M. L.; Vazquez-Cano, E. The Effect of Programming on Primary School Students' Mathematical and Scientific Understanding: Educational Use of MBot. *Educational Technology Research and Development*, **2019**, *67* (6), 1405–1425. https://doi.org/10.1007/S11423-019-09648-5/FIGURES/7.
119. Yadegaridehkordi, E.; Noor, N. F. B. M.; Ayub, M. N. bin; Affal, H. B.; Hussin, N. B. Affective Computing in Education: A Systematic Review and Future Research. *Comput Educ*, **2019**, *142*, 103649. https://doi.org/10.1016/J.COMPEDU.2019.103649.
120. Witherspoon, E. B.; Schunn, C. D.; Higashi, R. M.; Shoop, R. Attending to Structural Programming Features Predicts Differences in Learning and Motivation. *J Comput Assist Learn*, **2018**, *34* (2), 115–128. https://doi.org/10.1111/JCAL.12219.
121. Zhao, W.; Shute, V. J. Can Playing a Video Game Foster Computational Thinking Skills? *Comput Educ*, **2019**, *141*, 103633. https://doi.org/10.1016/J.COMPEDU.2019.103633.
122. Durak, H. Y.; Saritepeci, M. Analysis of the Relation between Computational Thinking Skills and Various Variables with the Structural Equation Model. *Comput Educ*, **2018**, *116*, 191–202. https://doi.org/10.1016/J.COMPEDU.2017.09.004.
123. Yücel, Y.; Rızvanoğlu, K. Battling Gender Stereotypes: A User Study of a Code-Learning Game, "Code Combat," with Middle School Children. *Comput Human Behav*, **2019**, *99*, 352–365. https://doi.org/10.1016/J.CHB.2019.05.029.
124. Howland, K.; Good, J. Learning to Communicate Computationally with Flip: A Bi-Modal Programming Language for Game Creation. *Comput Educ*, **2015**, *80*, 224–240. https://doi.org/10.1016/J.COMPEDU.2014.08.014.
125. Sáez-López, J. M.; Román-González, M.; Vázquez-Cano, E. Visual Programming Languages Integrated across the Curriculum in Elementary School: A Two Year Case Study Using "Scratch" in Five Schools. *Comput Educ*, **2016**, *97*, 129–141. https://doi.org/10.1016/J.COMPEDU.2016.03.003.
126. Hsu, C. C.; Wang, T. I. Applying Game Mechanics and Student-Generated Questions to an Online Puzzle-Based Game Learning System to Promote Algorithmic Thinking Skills. *Comput Educ*, **2018**, *121*, 73–88. https://doi.org/10.1016/J.COMPEDU.2018.02.002.
127. Città, G.; Gentile, M.; Allegra, M.; Arrigo, M.; Conti, D.; Ottaviano, S.; Reale, F.; Sciortino, M. The Effects of Mental Rotation on Computational Thinking. *Comput Educ*, **2019**, *141*, 103613. https://doi.org/10.1016/J.COMPEDU.2019.103613.
128. Kalelioılu, F. A New Way of Teaching Programming Skills to K-12 Students: Code.Org. *Comput Human Behav*, **2015**, *52*, 200–210. https://doi.org/10.1016/J.CHB.2015.05.047.
129. Zhong, B.; Wang, Q.; Chen, J. The Impact of Social Factors on Pair Programming in a Primary School. *Comput Human Behav*, **2016**, *64*, 423–431. https://doi.org/10.1016/J.CHB.2016.07.017.
130. Pérez-Marín, D.; Hijón-Neira, R.; Bacelo, A.; Pizarro, C. Can Computational Thinking Be Improved by Using a Methodology Based on Metaphors and Scratch to Teach Computer Programming to Children? *Comput Human Behav*, **2020**, *105*, 105849. https://doi.org/10.1016/J.CHB.2018.12.027.
131. Cheng, G. Exploring Factors Influencing the Acceptance of Visual Programming Environment among Boys and Girls in Primary Schools. *Comput Human Behav*, **2019**, *92*, 361–372. https://doi.org/10.1016/J.CHB.2018.11.043.
132. Papavlasopoulou, S.; Giannakos, M. N.; Jaccheri, L. Exploring Children's Learning Experience in Constructionism-Based Coding Activities through Design-Based Research. *Comput Human Behav*, **2019**, *99*, 415–427. https://doi.org/10.1016/J.CHB.2019.01.008.
133. Akpinar, Y.; Aslan, Ü. Supporting Children's Learning of Probability through Video Game Programming. *Journal of Educational Computing Research*, **2015**, *53* (2), 228–259. https://doi.org/10.1177/0735633115598492.
134. Jakoš, F.; Verber, D. Learning Basic Programing Skills with Educational Games. *Journal of Educational Computing Research*, **2017**, *55* (5), 673–698. https://doi.org/10.1177/0735633116680219/ASSET/IMAGES/LARGE/10.1177_0735633116680219-FIG2.JPEG.
135. Tran, Y. Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research*, **2019**, *57* (1), 3–31. https://doi.org/10.1177/0735633117743918/ASSET/IMAGES/LARGE/10.1177_0735633117743918-FIG2.JPEG.
136. Vasilopoulos, I. v.; van Schaik, P. Koios: Design, Development, and Evaluation of an Educational Visual Tool for Greek Novice Programmers. *Journal of Educational Computing Research*, **2019**, *57* (5), 1227–1259. https://doi.org/10.1177/0735633118781776/ASSET/IMAGES/LARGE/10.1177_0735633118781776-FIG2.JPEG.
137. Falloon, G. An Analysis of Young Students' Thinking When Completing Basic Coding Tasks Using Scratch Jnr. On the IPad. *J Comput Assist Learn*, **2016**, *32* (6), 576–593. https://doi.org/10.1111/JCAL.12155.
138. Benton, L.; Kalas, I.; Saunders, P.; Hoyles, C.; Noss, R. Beyond Jam Sandwiches and Cups of Tea: An Exploration of Primary Pupils' Algorithm-Evaluation Strategies. *J Comput Assist Learn*, **2018**, *34* (5), 590–601. https://doi.org/10.1111/JCAL.12266.

 doi:10.20944/preprints202301.0505.v1

139. Wang, X.; Hwang, G.; Liang, Z.; Wang, H. Enhancing Students' Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online Peer-Assessment Attempt. *J Educ Techno Soc*, **2017**, *20* (4), 58–68.
140. Akcaoglu, M.; Lucy, •; Green, S.; Santos Green, L. Teaching Systems Thinking through Game Design. *Springer*, **2019**, *67* (1), 1–19. https://doi.org/10.1007/s11423-018-9596-8.
141. Strawhacker, A.; Bers, M. U. What They Learn When They Learn Coding: Investigating Cognitive Domains and Computer Programming Knowledge in Young Children. *Educational Technology Research and Development*, **2019**, *67* (3), 541–575. https://doi.org/10.1007/S11423-018-9622-X/FIGURES/11.
142. Garneli, V.; Chorianopoulos, K. Programming Video Games and Simulations in Science Education: Exploring Computational Thinking through Code Analysis. *Interactive Learning Environments*, **2017**, *26* (3), 386–401. https://doi.org/10.1080/10494820.2017.1337036.
143. Chiang, F. kuang; Qin, L. A Pilot Study to Assess the Impacts of Game-Based Construction Learning, Using Scratch, on Students' Multi-Step Equation-Solving Performance. *Interactive Learning Environments*, **2018**, *26* (6), 803–814. https://doi.org/10.1080/10494820.2017.1412990.