Adv Mobile Learn Educ Res, 2023, 3(1): 525-540
DOI: 10.25082/AMLER.2023.01.001

REVIEW

Programming environments for the development of CT in preschool education: A systematic literature review

Konstantina Louka

Department of Preschool Education, University of Crete, Rethymno, Crete 74100, Greece

Correspondence to: Konstantina Louka, Department of Preschool Education, University of Crete, Rethymno, Crete 74100, Greece;
Email: ptpep521@edc.uoc.gr

Received: August 23, 2022;
Accepted: October 20, 2022;
Published: October 24, 2022.

Citation: Louka, K. (2022). Programming environments for the development of CT in preschool education: A systematic literature review. *Advances in Mobile Learning Educational Research*, 3(1), 525-540. https://doi.org/10.25082/AMLER.2023.01.001

Abstract: Computational Thinking (CT) and coding skills are internationally acknowledged as necessary for today's students and 21st century citizens. Nowadays, despite the multifaceted nature of CT, the introduction of CT and associated concepts is regarded as developmentally acceptable for preschool and kindergarten children. Furthermore, there is a considerable influx of software offering various interfaces and styles which facilitate the introduction of children aged four to six to essential CT, coding, and problem-solving skills. Although the creators of these environments claim that they bear educational value, there is no formal or scientifically documented evaluative system certifying this value. For instance, the fast-paced developers produce apps, and the breadth of the available apps has gone beyond what is reasonable for researchers and experts in the domain to evaluate. This article presents a literature review on the available software to encourage preschoolers' introduction to CT, coding and general literacy skills.

Keywords: programming environments, computational thinking, coding skills, educational robotics, preschool education

1 Introduction

Today's digital society needs citizens who can constructively with creativity and a developed capability to solve problems (Lye & Koh, 2014). As Computer Science (CS) makes its presence felt in all aspects of our lives, the need to educate students on the basic coding principles and concepts is imperative (Flórez et al., 2017). It is characteristically reported that CS related professions from 2014 to 2024 will create 500,000 new job openings in the US, according to research by Fayer et al. (2017). It is now considered critical for students to equip themselves with skills related to CT, which can developed through learning programming (Flórez et al., 2017).

Already existing research suggests that preschoolers can use not only programs (Bers et al., 2014) but also develop skills related to the basic dimensions of CT, such as: debugging, concepts of sequencing and sequence (Bers et al., 2019; Papadakis & Orfanakis, 2018). In recent years, programming environments have been designed to teach basic programming concepts to preschool children (Papadakis et al., 2021). Although there are a considerable number of digital environments designed to attract interest in coding, little empirical research has been conducted on the effectiveness of these applications in developing coding skills in early childhood education and, by extension, in cultivating CT (Pila et al., 2019). At the same time, the large influx of applications that have flooded the digital market makes it challenging to identify those programming environments of actual educational value (Kalogiannakis & Papadakis, 2017a; 2017b; 2020). In addition, the fast pace at which developers produce, specifically mobile apps and the range of available applications, have gone beyond what is reasonable for researchers, experts in the field as well as educators and parents to evaluate (Papadakis, 2018, 2021). On top of that, they are designed by creators who characterize them as educational (Hirsh-Pasek et al., 2015).

The literature review will briefly present computational learning environments that cultivate and develop CT in preschool children. The article aims to provide a supportive guide for early childhood education teachers to plan activities to promote CT.

2 Theoretical background

Although Papert (1980) was the first to use "Computational Thinking" (CT), heated debates around this concept began in 2006 as this was the landmark year for CT as Wing (2006) defined it as the total of mental processes aimed at solving a problem with the simultaneous involvement of the human and mechanical factors. Moreover, the need for CT integration into compulsory education was stressed as the value of its cultivation is proportional to teaching reading, writing and arithmetic (Papadakis et al., 2021). Since then, several European governments have formulated different education policies by reviewing their curricula to develop CT by teaching programming (Bocconi et al., 2016).

Prominent figures from the technology field, such as Bill Gates and Mark Zuckerberg, have stressed the importance of CT and taken initiatives for its development (Walsh & Campbell, 2018). Consequently, preschool education could not remain unaffected by the new data. The impact of Wing's (2006) publication, which argues that the cultivation of CT should be introduced into formal education, has led many European governments to pursue education policies that promote CT (Papadakis & Kalogiannakis, 2022). By revising their curricula, they strengthened the teaching of coding in compulsory education at the national level by adopting different approaches (Bocconi et al., 2016).

Although researchers Bers et al. (2019) argue that cultivating CT through learning programming is possible at preschool age, empirical studies are still relatively low (Jung & Wong, 2018). Using developmentally appropriate technologies is crucial for successfully introducing programming in kindergarten (Macrides et al., 2021). However, the vast number of programming environments that have flooded the market impedes their evaluation (Vaiopoulou et al., 2021), making the right programming environment choice a complicated process (Hirsh-Pasek et al., 2015). Consequently, the reasonable question that follows arises:

Which programming environments are suitable for cultivating CT in early childhood education?

This present systematic literature review aims to provide a supportive guide for early childhood education teachers to plan activities to promote CT and coding skills using environments with educational value, as demonstrated by empirical studies. The systematic literature review has been performed by searching the most popular bibliographic databases of all the available English literature with published articles from January 2012 to December 2021. Due to the diverse ways, the data was entered in each database, conditional formatting was implemented to find and highlight duplicate data to obtain those studies that met the criteria for inclusion. This systematic literature review revealed thirty-eight empirical studies that used developmentally appropriate programming environments to cultivate CT in preschool children.

3 Methodology

The systematic literature review was the method to investigate the research question, which is the proper tool to identify all available studies related to the question under study. It is a process that includes specific implementation stages to explore the existing literature comprehensively. Consequently, the summary reports of these earlier studies were used to examine new hypotheses and theories and identify potential gaps that need further study (Xiao & Watson, 2017). This present systematic literature review was implemented to identify the existing empirical studies in whose interventions eligible programmatic environments for the cultivation of CT in preschool-age are included.

The starting point of empirical studies on the promotion of CT in preschool children was the research of Kazakoff & Bers (2012) and Fessakis et al. (2013). This study includes research into articles from January 2012 to December 2021. On the contrary the study of Lye & Koh's (2014) entitled "Review on teaching and learning of CT through programming: What is next for K-12?" found only two empirical studies relative to preschool age.

3.1 Search strategy using PRISMA

This literature review presents the search stages using the PRISMA 2009 flow chart (see Figure 1) (Page et al., 2021). The PRISMA 2009 methodology comprises four stages: identification, screening, suitability and inclusion (Moher et al., 2010).

3.2 Databases

The search was performed in the 20 most well-known bibliographic databases, which are: ACM (Association for Computing Machinery), Digital Library, Cambridge Core – Journals and Books Online, CiteSeerX, Digital Library - CSDL | IEEE Computer Society, EBSCO Education Research Complete, Emerald Insight, ERIC – Education Resources Information Center, Google Scholar, Ingenta Connect, JSTOR, Learning & Technology Library (LearnTechLib) (formerly EdITLib), ProQuest, SAGE Journals, ScienceDirect, Scopus, Springer Science+Business Media, Taylor & Francis, Web of Science and Wiley. This search method was selected since no database is considered complete (Xiao & Watson, 2019).

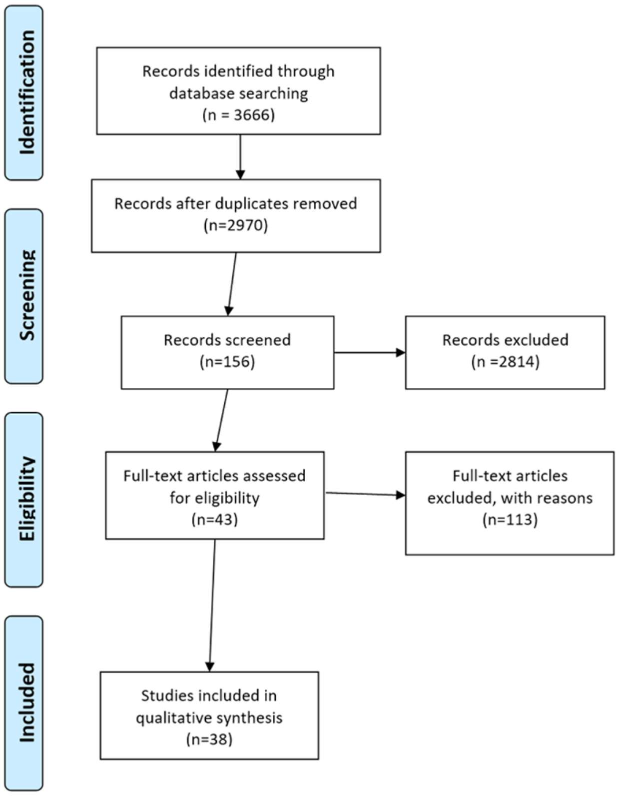

Figure 1 The PRISMA flow diagram of the literature search and review process

3.3 Inclusion and exclusion criteria

After thoroughly reviewing the selected articles' titles and summaries, inclusion and exclusion criteria were applied. The specific inclusion criteria used were the following:

(1) Journal articles with empirical data;

(2) Empirical studies on programming environments for the development of CT in their interventions;

(3) Articles referred to or focused their conclusions on basic CT concepts in preschool children;

(4) Articles published from January 2012 to December 2021.

Concerning the exclusion criteria, the following were used:

(1) Non-English publications;

(2) Participants in empirical studies were not restricted to preschoolers;

(3) Publications were already registered in other bibliographic databases;

(4) Publications were not accessible;

(5) Publications included a simplified description of a programming environment;

(6) Published empirical studies with a focus on preschoolers with special needs.

3.4 Systematic review process

In the first stage, during which the research is identified, the authors sought a series of keywords derived from the research question. Since bibliographic databases follow different archiving standards, the search criteria were adapted to the requirements of each database. At the same time, the Boolean operators were also used for a more specific assessment of all the bases. The keywords used in the search of all the bibliographic bases were: "preschool education", "early childhood education", "kindergarten", "prekindergarten", "young child", "nursery school", "infant school", "pre-primary school", "CT", "robotics", "coding", "computer programming".

Although Boolean operators were used for a comprehensive and more specific search combined with the exclusion and inclusion criteria, the resulting entries were numerous in specific databases, as shown below in Table 1.

Levy & Ellis (2006) cite a range of factors that could lead to the cessation of the bibliographic search. However, Haddaway et al. (2015) suggest focusing on the search's first 200 to 300 results in various bibliographic bases. In the present systematic literature review, following

Table 1 Returned results per data source

Data source	Initial search	1st stage (Identification)	2nd stage (Screening)	3rd stage (Eligibility)	4th stage (Included)
ACM (Association for Computing Machinery) Digital Library	145385	300	9	1	1
Cambridge Core – Journals and Books Online	84446	300	0	0	0
CiteSeerX	114	114	0	0	0
Digital Library - CSDL | IEEE Computer Society	227	7	0	0	0
EBSCO Education Research Complete	11656	300	3	2	2
Emerald Insight	23	23	0	0	0
ERIC – Education Resources Information Center	640	300	10	2	1
Google Scholar	17800	300	62	14	10
Ingenta Connect	3	3	1	0	0
JSTOR	209	209	0	0	0
Learning & Technology Library	18	18	3	1	1
LISTA (Library, Information Science &Technology)	16	16	15	0	0
ProQuest	51272	300	9	1	1
SAGE Journals	167	167	0	0	0
ScienceDirect	2448	300	10	6	6
Scopus	46	46	5	2	2
Springer Science + Business Media	678	300	16	8	8
Taylor & Francis	405	300	5	2	2
Web of Science	63	63	7	3	3
Wiley	607	300	1	1	1
Summary	316223	3666	156	43	38

the recommendation of Haddaway et al. (2015), only the titles and summaries of the first 300 entries from the bases that yielded too many results were examined. The analysis included all the databases articles that returned fewer than 300 entries. Moreover, articles already registered in other databases and those considered irrelevant were removed.

In the second stage of the bibliographic search, the entries were sorted. After applying the criteria for integration and exclusion, only 156 articles resulting from the first stage articles remained. Their selection was made after thoroughly examining these articles' titles and summaries.

In the third stage of the systematic review, 43 studies were included. At this stage, the suitability stage, the 43 remaining articles were meticulously studied to validate that each article met the requirements for inclusion in the present study based on the criteria mentioned above, paying particular attention to the relevance to the research question. This process was carried out by carefully reading the entire text. Four studies were removed from this process, and 38 remained.

In the last stage, the final articles were submitted to the supervising professor to evaluate their suitability for inclusion in the review. After being considered appropriate, they were included in the study and will constitute the final data set for analysis, as shown in Table 2 and 3. Table 2 includes the applied programming environments frequency in the reviewed studies.

Table 2 Summary of the applied programming environments

Programming Environments	Studies
Bee-Bot	1,7,8, 19, 26, 29, 22
Blue-Bot	13
Colby robotic mouse	16, 28
Code-a-pillar	18, 39
Lady bug Leaf & Lady bug mazes	11
Scratch Jr	10, 23, 25, 30, 32, 36, 38
Daisy the Dinosaur	10, 24
Kodable	24
KIBO robotics kit	4,9, 33, 34, 35, 36
LEGO Education WeDo	37, 31, 15, 5, 3
Evo	16
The Dash & Dot robot	14
Turtle Robot	21
Cubetto	20
Sphero Mini Robot Ball	22
Social Robot (SoRo)	12
Code.org mBot, Arduino robot car	2, 6 17

Table 3 Studies included in the review

No.	Authors	Year	No. of Participants	Age of Participants	Tool(s)
1	Angeli & Valanides	2020	50	5-6 years	Bee-Bot
2	Arfé, Vardanega, Montuori & Lavanga	2019	80	5-6 years	Code.org
3	Bers, Flannery, Kazakoff & Sullivan	2014	53	5-6 years	CHERP and Lego robotic kit
4	Bers, Gonzalez & Torres	2019	172	3-5 years	KIBO
5	Çakır, Korkmaz, İdil & Erdoğmuş	2021	40	5 years	LEGO® WeDo 2.0®
6	Çiftci & Bildiren	2020	28	4-5 years	Code.org
7	Critten, Hagon & Messer	2021	15	2-4 years	Bee-Bot
8	Di Lieto, Inguaggiato, Casto, Cecchi, Cioni & Dario	2017	12	3-5 years	Bee-Bot
9	Elkin, Sullivan & Bers	2016	64	3-5 years	KIBO
10	Falloon	2016	32	5-6 years	Daisy the Dinosaur & ScratchJr
11	Fessakis, Gouli & Mavroudi	2013	10	5-6 years	Ladybug Leaf & Ladybug mazes
12	Gordon, Rivera, Ackermann & Breazeal	2015	22	4-6 years	SoRo Toolkit
13	Heikkilä & Mannila	2018	-	4-5 years	Bluebot
14	Heljakka & Ihamaki	2019	20	5-6 years	Dash robot
15	Kazakoff, Sullivan & Bers	2013	29	4-5 years	CHERP & LEGO WeDo
16	Khoo	2020	3	5 years	Colby, the mouse robot & OZO-bot
17	Lin, Chien, Hsiao, Hsia & Chao	2020	7	5-6 years	mBot, Arduino robot car
18	Clarke-Midura, Kozlowski, Shumway & Lee	2021	16	5-6 years	Code-a-pillar
19	Misirli & Komis	2014	674	4-6 years	Bee-Bot
20	Murcia & Tang	2019	8	3-4 years	Cubetto
21	Nam, Kim & Lee	2019	53	5-6 years	TurtleBot
22	Newhouse, Cooper & Cordery	2017	50	5-6 years	Bee-Bot &Sphero
23	Papadakis, Kalogiannakis & Zaranis	2016	43	5 years	ScratchJr
24	Pila, Alade, Sheehan, Lauricella & Wartella	2019	28	4-5 years	Daisy the Dinosaur &Kodable
25	Portelance, Strawhacker & Bers	2016	62	5-8 years	ScratchJr
26	Repiso & González	2019	131	3-6 years	Bee-Bot
27	Roussou & Rangoussi	2019	18	4-6 years	Colby, the mouse robot
28	Saxena, Lo, Hew & Wong	2020	11	3-6 years	Bee-Bot
29	Strawhacker & Bers	2019	57	5-7 years	ScratchJr
30	Strawhacker & Bers	2015	35	5 years	CHERP & LEGO WeDo
31	Strawhacker, Lee & Bers	2018	222	5-7 years	ScratchJr
32	Sullivan & Bers	2016	60	4-7 years	CHERP & Kiwi robotic kit
33	Sullivan & Bers	2018	98	3-6 years	KIBO robotics kit
34	Sullivan & Bers	2019	105	5-7 years	KIBO robotics kit
35	Sullivan, Bers & Pugnali	2017	28	4-7 years	KIBO robotics kit & ScratchJr
36	Sullivan, Kazakoff & Bers	2013	37	Four years	CHERP & Lego WeDo
37	Sung, Ahn & Black	2017	66	5-7 years	ScratchJr
38	Wang, Choi, Benson, Eggleston & Weber	2020	3	Four years	Code-a-pillar

3.5 Data analyses

The data that emerged from the systematic literature review were classified based on the categorization proposed by Fessakis et al. (2019). A programming environment should be based on pedagogical and didactic criteria in preschool and primary education. Therefore, a programming environment should be developmentally appropriate, support the solution of a problem creatively and offer opportunities for digital expression. In addition, a good selection of a programming environment depends on how well its programming model facilitates learning, designing learning activities and making the problem area accessible (Fessakis et al., 2019). Given the preceding, the classification proposed by the researchers was based on five axes: the axis related to the abstraction levels of the computational system, the axis of age suitability, the axis of the supported programming model, the axis of the abstract approach to the programming process and the axis of the supported methods used in programming languages.

The classification includes the following categories:

(1) Logo family programming environments;
(2) Visual programming environments;
(3) Commercial programming learning environments for entertainment purposes;
(4) Physical Computing environments;
(5) Miscellaneous unplugged applications and environments.

This classification aims to assist preschool teachers in choosing the appropriate programming environment to design and organize programming teaching activities to cultivate CT in preschool and early childhood children.

4 Educational programming environments for preschoolers

4.1 Logo family programming environments

These programming environments are based on the first programming language created for children by Papert (1980), Logo. Children were first introduced to programming using the Logo programming language, which involved a turtle that a child could move around the screen with just a few commands. Notably, they continue to exist and inspire the construction of new systems, as listed below in subcategories (Fessakis et al., 2019).

4.1.1 Roamers

(1) Bee-Bot

Bee-Bot (Figure 2) is a programmable floor robot with a honeybee face. Bee-Bot can store over 40 commands given using the buttons located on the top of its back. It comprises four directional keys (front, back, 90° left turn, 90° right turn), a start button (GO), a clean memory button (CLEAR) and a pause button. The start button GO executes the commands while the pause button interrupts their execution. The CLEAR button erases everything that has been stored in the memory of Bee-Bot. New instructions are inserted into Bee-Bot's memory if this button is not used and added to the already saved ones. Bee-Bot's eyes flash every time a command is completed, while a sound is heard as soon as all the commands are completed. No computer is involved while engaging with the Bee-Bot robot (Angeli & Valanides, 2020).

Figure 2 The Bee-Bot Programmable floor robot

(2) Blue-Bot

Blue-Bot (Figure 3) is a small mobile robot similar to Bee-Bot. It is also used by programming it with the buttons on the top of its back and can store 200 commands. Blue-Bot executes the entire sequence of commands saved with a slight pause between each command when the GO key is pressed. Each saved command moves the floor robot in the desired direction by 15cm. Blue-Bot has a transparent shell so children can have visual contact with its interior. When the command is completed, Blue-Bot stops and a sound effect is produced, which indicates the end. Turning the sound on and off is possible using the switch on the underside. Blue-Bot can connect wirelessly to mobile devices as well as a computer. This way, children can use the free application to engage in remote programming experiences.

Figure 3 The Blue-Bot Programmable floor robot

(3) Colby robotic mouse

Colby is a small floor robot in the form of a mouse, and its operation is relevant to that of Bee-bot and Blue-Bot. It features coloured arrow-shaped navigation buttons, a delete key for saved commands, and a start button to execute commands (Figure 4). It features two speeds and can make sounds. Engaging with this robot provides fun opportunities for teaching programming to preschool children. Additional accessories, such as square shapes joined together to form a grid, are accompanied by additional accessories. Adding walls and tunnels in different places on the grid can also create a maze. The 2-speed levels enable its use on the floor or any other surface besides the grid.

Figure 4 Robot mouse Colby

(4) Code-a-pillar

Code-a-pillar is a programmable game with a shape caterpillar. It features a motorized head and eight reconnecting segments for multiple combinations: three for forwarding movement, two for turning 90°right, two for turning 90° left and a segment for audible action indicators (Figure 5). Each segment can be detached and reconnected. Every time the segments are connected, the sequence of instructions shows the route the Code-a-pillar will follow (Clarke-Midura et al., 2021). The motorized head segment features lights, amusing sounds and flashing eyes to bring Code-a-pillar to life. In case of an obstacle, this programmable game stops and continues when the motorized head is pressed.

Figure 5 Code-a-pillar

4.1.2 Software roamers

Ladybug leaf & Ladybug mazes are two programming environments available at the National Library of Virtual Manipulatives of the University of UTAH, USA, with free access. The specific environments are based on the Logo programming language, and the commands are in the form of virtual tiles and include forward movement, rearward movement, 45° or 90° left turn and 45°or 90°right turn. The program appears as a sequence of tiles-commands at the bottom left of the screen. Adding or removing tiles and executing a specific command or the entire algorithm is possible. The ladybug moves with the program's execution while leaving traces indicating the completion of each command (Fessakis et al., 2013).

According to Brennan and Resnick's Computational Thinking Framework (2012), developmentally appropriate coding skills expected in young children are sequence, debugging, looping, and conditionals. Overall, studies using the Logo family programming environments have positively cultivated sequence and problem-solving skills. Also, abstraction and decomposition skills were developed.

4.2 Visual programming environments

Visual programming environments are software that allows a user to create programs by connecting appropriate blocks of code (Cheng, 2019). Programming environments in this category use digital optical tiles to represent the building blocks used by programming languages that enable the creation of complex structures to create programs (Fessakis et al., 2019). Users with the drag-and-drop function can select suitable blocks, and each block has a command (Cheng, 2019). Some visual programming environments provide visual cues to help users understand the functioning of the tiles (Lye & Koh, 2014).

4.2.1 ScratchJr

This is a programming environment where children aged 5-7 can learn fundamental programming concepts by creating games and interactive stories. ScratchJr was developed in collaboration with the DevTech Research Group of Tufts University and MIT Lifelong Kindergarten Group. The main screen features a theatrical stage in which users create objects and six command categories: yellow Trigger blocks, blue Motion blocks, purple Looks blocks, green Sound blocks, orange Control flow blocks, and red End blocks (Figure 6).

Figure 6 Screenshot of ScratchJr interface

Interaction between objects is feasible by using the programming tiles. Programmatic tiles are selected and transferred to the programming area in which they are activated. The connection between the tiles creates programs that run from left to right (Bers,2018).

4.2.2 Daisy the Dinosaur

Daisy the Dinosaur is a free digital application designed to teach basic programming concepts. Interaction with the application's environment is performed through drag and drop. It aims for children to understand the concepts of sequence, loops and conditionals and is compatible with the iOS operating system. By selecting the "challenge mode", Daisy must complete tasks of graded difficulty. For example, in the first task, Daisy is asked to move to a star a few centimetres away from her (Figure 7). The user should drag the tile from the left side of the screen with the "move" command and drop it in a frame that forms a programming area (Pila et al., 2019).

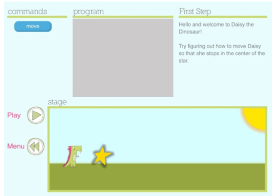

Figure 7 Screenshot of daisy the dinosaur in challenge mode

4.2.3 Code.org

Code.org is a non-profit institution with the primary aim of expanding access to computer science in as many schools as possible and increasing the participation of sensitive populations. It is an open-source programming platform with educational scenarios designed for all student age groups while providing personalized feedback. It includes activities that promote CT through learning programming using a computer and without one (Arfé et al., 2019).

Generally, the findings of the studies using visual programming environments showed the development of coding skills in preschool children, such as sequence comprehension, debugging, symbol interpretation and repetition.

4.3 Commercial programming learning environments for entertainment purposes

This category includes programming environments that do not present any innovative characteristics, their access to the learning process is not on equal terms and are commercial products (Fessakis et al.,2019)

Kodable is a digital application for children aged 4 to 10 that aims to teach programming. The user interacts with the application's environment using directional arrows for commands in a drag-and-drop interface. Children must choose the appropriate arrows to help the hero of the game cross a maze and move on to the next level. The app comes with over 70 lesson plans for teachers. At the same time, it also provides a comprehensive curriculum that focuses on teaching basic programming principles to young children while creating and writing their code is provided for older children (Figure 8).

Figure 8 Screenshot of Kodable

Results from the study used commercial programming learning environments demonstrate that kindergarteners can successfully learn fundamental programming skills, including sequence, fixing errors and correspondence skills (Pila et al., 2019).

4.4 Physical Computing environments

Physical Computing environments refer to electromechanical systems regulated by software and incorporate mechanical components such as sensors, motors and switches (Fessakis et al., 2019).

4.4.1 KIBO robotics kit

The KIBO robotics kit (Figure 9) is designed for children aged 4-7 years and enables them to engage in fun activities derived from engineering and programming. Children can create and program their robots while simultaneously devising activities incorporating various art forms such as music and dance (Sullivan & Bers, 2018; Bers et al., 2019). The KIBO robotics kit includes both software and hardware. The software comprises tangible wooden blocks with barcodes and a colour image representing symbols and text, while the hardware features sensors, wheels, motors, and a scanner. The child creates a sequence of instructions (a program) using the wooden blocks, and KIBO reads the barcodes with the embedded scanner. No connection to a computer device is required. The original version of KIBO was called KIWI, and using a computer, and a camera were prerequisites for its operation. The camera took pictures of the wooden programming blocks, and the algorithm was sent to the robot with the help of a USB cable (Bers, 2018).

Figure 9 KIBO robotics kit

4.4.2 LEGO Education WeDo

It combines popular LEGO blocks with a motor, motion, two tilt sensors and the LEGO USB Hub used for connecting the sensors and motors to the computer (Figure 10). Models created with LEGO blocks can be programmed using a computer (PC or laptop) or a tablet on which the software LEGO WeDo 2.0 is installed. WeDo 2.0 software uses visual programming techniques and enables drag and drop operation (Çakır et al., 2021).

Figure 10 LEGO Education WeDo 2.0

4.4.3 Evo Ozobot

It is a small robot that teaches programming to children aged five years and up. Evo features a built-in speaker, proximity sensors and Bluetooth connectivity (Figure 11). The proximity sensors connect to the collision avoidance detection system and help it avoid obstacles as it moves. Evo can recognize lines, colours, and codes on digital surfaces like an iPad and natural surfaces like paper. It offers two ways of coding and uses sensors to follow lines and read the colour codes created by the user with Colour Code markers or stickers. However, it can also be programmed with its software, Ozo Blockly, which enables students to create programs using different blocks – starting from simple and moving on to more complex coding structures.

Figure 11 Evo Ozobot

4.4.4 The Dash & Dot robot

Wonder Workshop's Dash is a programmable robot, and Dot is its sidekick (Figure 12). Dash's head is fully mobile, and the two powered wheels of its body enable it to run and rotate. It can be operated in two ways; either wirelessly, via Bluetooth connection, or manually. It features position, proximity, microphones, a gyroscope, and led lights. Dash can be programmed to do several tasks and interact with its surroundings. It can also respond to voice messages and commands based on the applications of a mobile device. At the same time, it can record the user's voice and comes with applications that allow Dash's operation and programming and design of new behaviours.

Figure 12 The Dash & Dot robot

4.4.5 mBot, Arduino robot car

mBot, Arduino robot car is a programmable Make block robot that is not delivered assembled. Indeed, step-by-step instructions are provided to build a robot from scratch using a screwdriver (Figure 13). The construction process presents an opportunity for children to learn the basics of robotic mechanisms and electronic components. It offers three modes of operation: obstacle avoidance mode, infrared remote-control operation and line sequence function. Bluetooth facilitates the wireless connection between a computer or a smart portable device, and mBot and programming are executed using an application with visual programming features.

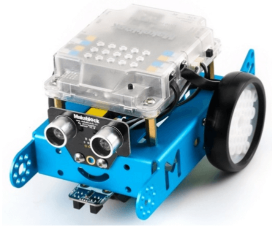

Figure 13 mBot, Arduino robot car

4.4.6 Turtle robot

Turtle robot is a handsome and cute programmable robot in the form of a turtle (Figure 14). The robot finds its way to the target, as the coding is implemented through a colour sensor using 24 cards consisting of the robot's software. Each of these cards gives instructions for different directions. Once the target path has been determined, the cards needed for the action are selected and placed in the planned order. The cards are inserted into the robot by placing them in order under the head of the turtle robot where the colour sensor is located. After all the cards have been inserted, the robot is launched to execute the programmed path. In addition, the turtle robot can also be programmed by following the path along the black line through five colour codes. While no computer is required, the connection is also feasible with Bluetooth.

Figure 14 Turtle robot

4.4.7 Cubetto

Cubetto is a robotic tool that is not connected to a computer or any other computational portable device. It includes the Cubetto robot, a control board on which the start button and function bar are located, 16 physical tiles of different colours and shapes that include different commands: 4 forward-facing tiles, 4 90° left-turn tiles, 4 90° right-turn tiles, four-mode tiles and a squared World Map Mat (Figure 15). To create a program, the user first needs to place the tiles on the board in a row and then by pressing the enter button, Cubetto executes the sequence of commands. If a set of instructions needs to be executed more than once, the sequence is placed in the function row, and the algorithm is revoked each time.

Figure 15 Cubetto

4.4.8 Sphero mini robot ball

Sphero mini robot ball is a small robot equipped with a gyroscope, acceleration sensors, led lights and motor encoders. It is accompanied by six tiny bowling pins and three traffic cones

(Figure 16) and is equipped with built-in Bluetooth to connect to smart mobile devices. Through Sphero applications, children engage in code learning activities by designing a path for their robot to follow. By selecting the drive mode in Sphero Edu app, the user can generate the code in the programming area of the application by performing the drag-and-drop operation.

Figure 16 Sphero mini robot ball

4.4.9 Social Robot (SoRo) tool kit

Social Robot (SoRo) tool kit (Figure 17) allows preschoolers to experiment with computational concepts while teaching a social robot new rules. The tool kit also provides a platform for developing interpersonal skills through storytelling that integrates interpersonal and computational concepts. Soro harnesses preschoolers' natural interest in social interaction to familiarize them with new concepts. The programming interface is composed of colourful reusable vinyl stickers (Figure 17). Children are taught how to create rules that make the robot do new things. On top of that, they show new rules to the robot while the experimenter teleoperates the rules on a tablet, allowing children to later interact with the robot and try the new rule.

Figure 17 Social Robot (SoRo) tool kit

Reviewed research using physical computing environments has shown that it is possible to teach basic CT skills to preschoolers, such as sequence, debugging, looping and conditionals (Brennan & Resnick, 2012). The research results on the cultivation of sequence, error correction and repetition skills were mainly positive, with the older children scoring higher on the more complex concepts of CT, such as conditionals. It is worth mentioning that the conclusions concerning children aged three years were also positive (Bers et al., 2019).

4.5 Miscellaneous unplugged applications and environments

In this category, Computational Thinking is cultivated through learning scenarios without using a computer (Fessakis et al., 2019). The systematic literature review revealed no empirical studies corresponding to this category.

5 Discussion

The present systematic literature review documented the undiminished interest in cultivating CT in preschoolers through learning programming has become apparent over the last decade. Using developmentally appropriate programming environments is the cornerstone for successfully integrating kindergarten programming (Macrides et al., 2021; Papadakis et al., 2022). The present research shows that programming environments can be used in early childhood education and positively promote CT (Kalogiannakis & Papadakis, 2017c). ScratchJr is a well-known programming environment for developing basic CT and coding concepts and was used in seven of the research reviewed with positive results. Floor robots (roamers) have also proved popular; Bee-Bot holds a prominent place in researchers' preferences since it was used in seven studies aimed at developing CT in early childhood pupils.

Regarding educational robotic environments, the review highlighted six studies in which KIBO is used as a programming environment for teaching introductory concepts of programming. However, it is worth mentioning that all the reviewed research that used this robotic kit was conducted by Bers, the creator, and her collaborators. Five studies with LEGO WeDo used as a programming environment for teaching introductory programming concepts (see Figure 18).

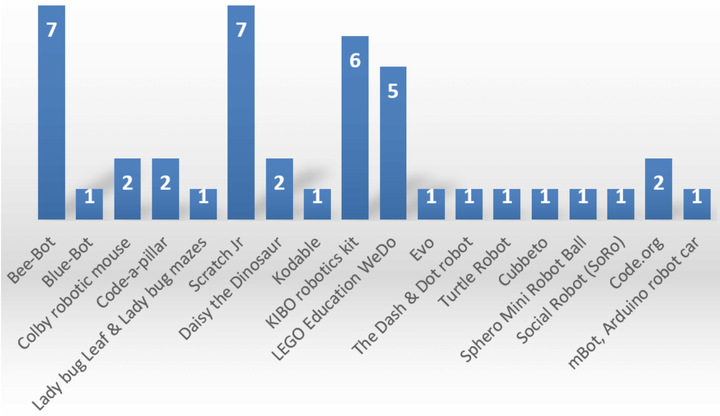

Figure 18 Popular programming environments based on research

Fessakis et al. (2013) argued that the practical and methodical integration of programming in kindergartens is determined by the programming environments and by designing and implementing appropriate teaching interventions that can be thoroughly integrated into the classroom. Pila et al. (2019) also featured the attractiveness of a programming environment as an essential factor for the effective learning of basic programming concepts. Research into developing educational curricula that comprehensively introduce programming in early childhood education is still at an early stage (Macrides et al., 2021). However, educators who wish to integrate CT into their classrooms could implement interventions using educational technologies accompanied by integrated curricula (Kikilias et al., 2009), such as ScratchJr and Kodable (Ching et al., 2018). However, an issue arises again related to the prohibitive costs required to purchase specific programming environments, making them difficult for both the teacher and the school unit (Ching et al., 2018). This is linked to the growing trend facing learning programming as a commercial product (Fessakis et al., 2019). These challenges need to be addressed immediately so that the cultivation of CT can be successfully integrated into preschool education. Therefore, the need to develop curricula (Macrides et al., 2021) and teacher vocational training programs are imperative to successfully integrate educational technology into the classroom (Lavidas et al., 2022). The scientific community can significantly help this endeavour by supporting the development of these programs based on scientifically substantiated data and up-to-date technological resources (Bakala et al., 2021; Papadakis, 2022).

Conflicts of interest

The author declares that they have no conflict of interest.

References

Angeli, C., & Valanides, N. (2019). Developing young children's CT with educational robotics: An interaction effect between gender and scaffolding strategy. Computers in Human Behavior,105, 1-13. https://doi.org/10.1016/j.chb.2019.03.018

Arfé, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in primary grades boosts children's executive functions. Frontiers in Psychology, 10, Article 2713. https://doi.org/10.3389/fpsyg.2019.02713

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and CT: a systematic review. International Journal of Child-Computer Interaction, 29. https://doi.org/10.1016/j.ijcci.2021.100337

Bers, M. U. (2018). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. In 2018 IEEE global engineering education conference (EDUCON) (pp. 2094-2102). IEEE. https://doi.org/10.1109/EDUCON.2018.8363498

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). CT and tinkering: exploration of an early childhood robotics curriculum. Computers & Education, 72, 145-157. https://doi.org/10.1016/j.compedu.2013.10.020

Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: promoting positive learning experiences in childhood classrooms. Computers & Education, 138, 130-145. https://doi.org/10.1016/j.compedu.2019.04.013

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. European Commission, JRC Science

for Policy Report, 68.
https://doi.org/10.2791/792158

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Canada, (pp. 1-25).

Çakır, R., Korkmaz, Ö., İdil, Ö., & Erdoğmuş, F. U. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. Thinking Skills and Creativity, 40, Article 100812.
https://doi.org/10.1016/j.tsc.2021.100812

Cheng, G. (2019). Exploring factors influencing the acceptance of visual programming environment among boys and girls in primary schools. Computers in human behavior, 92, 361-372.
https://doi.org/10.1016/j.chb.2018.11.043

Ching, Y. H., Hsu, Y. C., & Baldwin, S. (2018). Developing CT with educational technologies for young learners. TechTrends, 62, 563-573.
https://doi.org/10.1007/s11528-018-0292-7

Çiftci, S., & Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. Computer science education, 30(1), 3-21.
https://doi.org/10.1080/08993408.2019.1696169

Clarke-Midura, J., Kozlowski, J. S., Shumway, J. F., & Lee, V. R. (2021). How young children engage in and shift between reference frames when playing with coding toys. International Journal of Child-Computer Interaction, 28, Article 100250.
https://doi.org/10.1016/j.ijcci.2021.100250

Critten, V., Hagon, H., & Messer, D. (2021). Can preschool children learn programming and coding through guided play activities? A case study in CT. Early Childhood Education Journal, 1-13.
https://doi.org/10.1007/s10643-021-01236-8

Di Lieto, M. C., Inguaggiato, E., Castro, E., Cecchi, F., Cioni, G., Dell'Omo, M., Laschi, C., Pecini, C., Santerini G., Sgandurra, G., & Dario, P. (2017). Educational Robotics intervention on Executive Functions in preschool children: a pilot study. Computers in human behavior, 71, 16-23.
https://doi.org/10.1016/j.chb.2017.01.018

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. Computers in the Schools, 33(3), 169-186.
https://doi.org/10.1080/07380569.2016.1216251

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. Journal of Computer Assisted Learning, 32(6), 576–593.
https://doi.org/10.1111/jcal.12155

Fayer, S., Lacey, A., & Watson, A. (2017). STEM occupations: past, present, and future. Spotlight on Statistics, pp. 1, 1–35.
https://www.bls.gov

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. Computers & Education, 63, 87-97.
https://doi.org/10.1016/j.compedu.2012.11.016

Fessakis, G., Komis, V., Dimitracopoulou, A., & Prantsoudi, S. (2019). Overview of the Computer Programming Learning Environments for primary education. Review of Science, Mathematics and ICT Education, 13(1), 7-33. . doi:
https://doi.org/10.26220/rev.3140

Flórez, B. F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: teaching CT through programming. Review of Educational Research, 87(4), 834-860.
https://doi.org/10.3102%2F0034654317710096

García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y. A. (2019). Robotics to develop CT in early Childhood Education. Comunicar. Media Education Research Journal, 27(59), 63-72.
https://doi.org/10.3916/C59-2019-06

Gordon, M., Rivera, E., Ackermann, E., & Breazeal, C. (2015). Designing a relational social robot toolkit for preschool children to explore computational concepts. In Proceedings of the 14th International Conference on Interaction Design and Children (pp. 355-358). ACM.
https://doi.org/10.1145/2771839.2771915

Haddaway, N. R., Collins, A. M., Coughlin, D., & Kirk, S. (2015). The role of Google Scholar in evidence reviews and its applicability to grey literature searching. PLoS ONE, 10(9), Article e0138237.
https://doi.org/10.1371/journal.pone.0138237

Heikkilä, M., & Mannila, L. (2018). Debugging in programming as a multimodal practice in early childhood education settings. Multimodal Technologies and Interaction, 2(3), 42.
https://doi.org/10.3390/mti2030042

Heljakka, K., & Ihamäki, P. (2019). Ready, steady, move! Coding toys, preschoolers, and mobile playful learning. In P. Zaphiris, & A. Ioannou (Eds), Learning and Collaboration Technologies. Ubiquitous and Virtual Environments for Learning and Collaboration (pp. 68-79). Springer.
https://doi.org/10.1007/978-3-030-21817-1_6

Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting education in "educational" apps: lessons from the Science of learning. Psychological Science in the Public Interest, 16(1), 3-34.
https://doi.org/10.1177%2F1529100615569721

Jung, S. E., & Won, E. S. (2018). Systematic review of research trends in robotics education for young children. Sustainability, 10(4), 905.
https://doi.org/10.3390/su10040905

Kalogiannakis, M., & Papadakis, S. (2017b). A proposal for teaching ScratchJr programming environment in preservice kindergarten teachers. In Proceedings of the 12th Conference of the European Science Education Research Association (ESERA) (pp. 21-25).

Kalogiannakis, M., & Papadakis, S. (2017a). An evaluation of Greek educational Android apps for preschoolers. In proceedings of the 12th Conference of the European Science Education Research Association (ESERA), Research, Practice and Collaboration in Science Education, Dublin City University and the University of Limerick, Dublin, Ireland (pp. 21-25).

Kalogiannakis, M., & Papadakis, S. (2017c). An evaluation of Greek educational Android apps for preschoolers. In proceedings of the 12th Conference of the European Science Education Research Association (ESERA), Research, Practice and Collaboration in Science Education, Dublin City University and the University of Limerick, Dublin, Ireland (pp. 21-25).

Kalogiannakis, M., & Papadakis, S. (2020). The use of developmentally mobile applications for preparing pre-service teachers to promote STEM activities in preschool classrooms. In Mobile Learning Applications in Early Childhood Education (pp. 82-100). IGI Global.
https://doi.org/10.4018/978-1-7998-1486-3.ch005

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Childhood Education Journal, pp. 41, 245-255.
https://doi.org/10.1007/s10643-012-0554-5

Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. Journal of Educational Multimedia and Hypermedia, 21(4), 371-391.

Khoo, K. Y. (2020). A Case Study on How Children Develop CT Collaboratively with Robotics Toys. International Journal of Educational Technology and Learning, 9(1), 39-51.
https://doi.org/10.20448/2003.91.39.51

Kikilias, P., Papachristos, D., Alafodimos, N., Kalogiannakis, M. & Papadakis, St. (2009). An Educational Model for Asynchronous E-Learning. A case study in a Higher Technology Education, In D. Guralnick (ed.) Proceedings of the International Conference on E-Learning in the Workplace (ICELW-09), 10-12 June 2009, New York: Kaleidoscope Learning (CD-Rom).

Lavidas, K., Apostolou, Z., & Papadakis, S. (2022). Challenges and opportunities of mathematics in digital times: Preschool teachers' views. Education Sciences, 12(7), 459.
https://doi.org/10.3390/educsci12070459

Levy, Y., & Ellis, T. J. (2006). A systems approach to conduct an effective literature review in support of information systems research. Informing Science Journal, 9, 181-212.
https://doi.org/10.28945/479

Lin, S. Y., Chien, S. Y., Hsiao, C. L., Hsia, C. H., & Chao, K. M. (2020). Enhancing CT Capability of Preschool Children by Game-based Smart Toys. Electronic Commerce Research and Applications, 44, Article 101011.
https://doi.org/10.1016/j.elerap.2020.101011

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of CT through programming: what is next for K-12? Computers in Human Behavior, 41, 51-61.
https://doi.org/10.1016/j.chb.2014.09.012

Macrides, E., Miliou, O., & Angeli, C. (2021). Programming in early childhood education: a systematic review. International Journal of Child-Computer Interaction, Article 100396.
https://doi.org/10.1016/j.ijcci.2021.100396

Misirli, A., & Komis, V. (2014). Robotics and programming concepts in Early Childhood Education: a conceptual framework for designing educational scenarios. In C. Karagiannidis, P. Politis, & I. Karasavvidis (Eds) Research on e-Learning and ICT in Education (pp. 99-118). Springer.
https://doi.org/10.1007/978-1-4614-6501-0_8

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., Altman, D., & The PRISMA Group (2010). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. International Journal of Surgery, 8, 336-341.
https://doi.org/10.1016/j.ijsu.2010.02.007

Murcia, K. J., & Tang, K. S. (2019). Exploring the multimodality of young children's coding. Australian Educational Computing, 34(1).

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting plans to action: the effects of a card-coded robotics curriculum and activities on Korean kindergartners. The Asia-Pacific Education Researcher, 28(5), 387-397.
https://doi.org/10.1007/s40299-019-00438-4

Newhouse, C. P., Cooper, M., & Cordery, Z. (2017). Programmable toys and free play in early childhood classrooms. Australian Educational Computing, 32(1).

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Systematic reviews, 10(1), 1-11.
https://doi.org/10.1186/s13643-021-01626-4

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers?: A case study. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 13(1), 1-16.
https://doi.org/10.4018/IJWLTT.2018010101

Papadakis, S. (2021). The impact of coding apps to support young children in computational thinking and computational fluency. A literature review. In Frontiers in Education (p. 183). Frontiers. https://doi.org/10.3389/feduc.2021.657895

Papadakis, S. (2022). Apps to Promote Computational Thinking and Coding Skills to Young Age Children: A Pedagogical Challenge for the 21st Century Learners. Educational Process. International Journal, 11(1), 7-13. https://doi.org/10.22521/edupij.2022.111.1

Papadakis, S., & Kalogiannakis, M. (2022). Learning computational thinking development in young children with Bee-Bot educational robotics. In Research Anthology on Computational Thinking, Programming, and Robotics in the Classroom (pp. 926-947). IGI Global. https://doi.org/10.4018/978-1-6684-2411-7.ch040

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice. International Journal of Technology Enhanced Learning, 10(1-2), 44-72. https://doi.org/10.1504/IJTEL.2018.088333

Papadakis, S., Alexandraki, F., & Zaranis, N. (2022). Mobile device use among preschool-aged children in Greece. Education and Information Technologies, 27(2), 2717-2750. https://doi.org/10.1007/s10639-021-10718-6

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and CT with ScratchJr in preschool education: a case study. International Journal of Mobile Learning and Organisation, 10(3), 187-202. https://doi.org/10.1504/IJMLO.2016.077867

Papadakis, S., Vaiopoulou, J., Sifaki, E., Stamovlasis, D., & Kalogiannakis, M. (2021). Attitudes towards the use of educational robotics: Exploring pre-service and in-service early childhood teacher profiles. Education Sciences, 11(5), 204. https://doi.org/10.3390/educsci11050204

Papadakis, S., Vaiopoulou, J., Sifaki, E., Stamovlasis, D., Kalogiannakis, M., & Vassilakis, K. (2021). Factors That Hinder in-Service Teachers from Incorporating Educational Robotics into Their Daily or Future Teaching Practice. In CSEDU (2) (pp. 55-63). https://doi.org/10.5220/0010413900550063

Papert, S. (1980). Mindstorms: Children, Computers, and powerful ideas. Basic Books.

Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Computers & Education, 128, 52-62. https://doi.org/10.1016/j.compedu.2018.09.006

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr programming language in the early childhood classroom. International Journal of Technology and Design Education, pp. 26, 489-504. https://doi.org/10.1007/s10798-015-9325-0

Roussou, E., & Rangoussi, M. (2019). On the use of robotics for the development of CT in kindergarten: Educational intervention and evaluation. In M. Merdan, W. Lepuschitz, G. Koppensteiner, R. Balogh, & D. ObdrŽálek , Robotics in Education (pp. 31-44). Springer. https://doi.org/10.1007/978-3-030-26945-6_3

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate CT: An exploratory study in early childhood education. The Asia-Pacific Education Researcher, 29(1), 55-66. https://doi.org/10.1007/s40299-019-00478-w

Strawhacker, A., & Bers, M. U. (2015). "I want my robot to look for food": comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. International Journal of Technology and Design Education, pp. 25, 293-319. https://doi.org/10.1007/s10798-014-9287-7

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. Educational Technology Research and Development, pp. 67, 541-575. https://doi.org/10.1007/s11423-018-9622-x

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. International Journal of Technology and Design Education, pp. 28, 347-376. https://doi.org/10.1007/s10798-017-9400-9

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in prekindergarten through second grade. International Journal of Technology and Design Education, pp. 26, 3-20. https://doi.org/10.1007/s10798-015-9304-5

Sullivan, A., & Bers, M. U. (2018). Dancing robots: integrating art, music, and robotics in Singapore's early childhood centers. International Journal of Technology and Design Education, pp. 28, 325-346. https://doi.org/10.1007/s10798-017-9397-0

Sullivan, A., & Bers, M. U. (2019). Investigating the use of robotics to increase girls' interest in engineering during early elementary school. International Journal of Technology and Design Education, pp. 29, 1033-1051. https://doi.org/10.1007/s10798-018-9483-y

Sullivan, A., Bers, M., & Pugnali, A. (2017). The impact of user interface on young children's CT. Journal of Information Technology Education: Innovations in Practice, 16, 171-193. https://doi.org/10.28945/3768

Sullivan, A., Kazakoff, E. R., & Bers, M. U. (2013). The wheels on the bot go round and round: robotics curriculum in prekindergarten. Journal of Information Technology Education, pp. 12, 203-219. https://doi.org/10.28945/1887

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing CT to young learners: practicing computational perspectives through embodiment in mathematics education. Technology, Knowledge and Learning, 22, 443-463. https://doi.org/10.1007/s10758-017-9328-x

Vaiopoulou, J., Papadakis, S., Sifaki, E., Stamovlasis, D., & Kalogiannakis, M. (2021). Parents' perceptions of educational apps use for kindergarten children: development and validation of a new instrument (PEAU-p) and exploration of parents' profiles. Behavioral Sciences, 11(6), 82. https://doi.org/10.3390/bs11060082

Walsh, C., & Campbell, C. (2018). Introducing coding as a Literacy on mobile Devices in the Early Years. In G. Oakley (Ed), Mobile Technologies in Children's Language and Literacy: Innovative Pedagogy in Preschool and Primary Education (51-66). Emerald Publishing Limited. https://doi.org/10.1108/978-1-78714-879-620181004

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2021). Teacher's role in fostering preschoolers' CT: an exploratory case study. Early Education and Development, 32(1), 26-48. https://doi.org/10.1080/10409289.2020.1759012

Wing J. M. (2006). CT. Communications of the ACM 49(3), 33-35. https://doi.org/10.1145/1118178.1118215

Xiao, Y., & Watson, M. (2017). Guidance on conducting a systematic literature review. Journal of Planning Education and Research, 39(1), 93-112. https://doi.org/10.1177/0739456X17723971