International Journal of Elementary Education
2017; 6(4): 32-37
http://www.sciencepublishinggroup.com/j/ijeedu
doi: 10.11648/j.ijeedu.20170604.11
ISSN: 2328-7632 (Print); ISSN: 2328-7640 (Online)

Teaching Coding to Children: A Methodology for Kids 5+

Ugur Tevfik Kaplancali[1], Zafer Demirkol[2]

[1]Management Information Systems, Yeditepe University, Istanbul, Turkey

[2]Computer Engineering, Bahcesehir University, Istanbul, Turkey

Email address:

ugur.kaplancali@yeditepe.edu.tr (U. T. Kaplancali), zaferdemirkol@gmail.com (Z. Demirkol)

To cite this article:

Ugur Tevfik Kaplancali, Zafer Demirkol. Teaching Coding to Children: A Methodology for Kids 5+. *International Journal of Elementary Education.* Vol. 6, No. 4, 2017, pp. 32-37. doi: 10.11648/j.ijeedu.20170604.11

Received: June 10, 2017; **Accepted**: August 12, 2017; **Published**: September 14, 2017

Abstract: As new education models gain traction, discussions about how to incorporate programming skills and the "maker mindset" throughout the curriculum intensified. A growing realization among educators that teaching coding to children will, not only make it easier for them to understand how the information technology works, but give them a skill for life. In 2013, England wanted to be the first country in the world to make computer programming a compulsory school subject at all levels. In the US, administration was a strong proponent of expansion of programming teaching, President Obama stated "everybody's got to learn how to code early." Many different programming language tools exist for teaching children how to code, but none of them are based on comprehensive methodologies so that, fundamentals of coding can be easily understood by minors. Tools like Scratch Jr. and Tynker provide intuitive suite of visual-programming language for children ages 5 and above and they are more game oriented. This paper argues that there are seven aspects of coding that are fundamental in teaching coding to children, and three out of this seven is a must at any level. A new open source platform called CiK is also introduced.

Keywords: Coding, Programming Tools, Children Learning, Scratch

1. Introduction

Since their launch, tablets have become increasingly popular among preschool children and, in recent years, children's exposure to similar devices is increasing fast. Most children learn to use these electronic devices before they learn how to read and write. They become computer-literate, in part, at an early age but without knowing the workings of the computing technology. Moreover, older kids now realized the opportunities that makerspace presents to them, however maker mindset often requires some programming skills.

Many parents and teachers are now encouraging children to learn how to code so that they can understand how the technology really works. A growing realization among educators is that teaching coding to children will give them a skill for life. As youth unemployment rates are high in most of the developed countries, such computing skills are required to fill the "skills gap" between the number of technology jobs and the people qualified to fill them. As the maker movement become more mainstream, the absence of such skills, especially coding, will prevent young people from connecting with people from all over the world involved in a wide array of digital maker projects, such as robotics, machines and even games.

Visual programming tools based on blocks are the most common way of exposing children to coding at an early age. A research shown that 10th graders who had taken a *Scratch* course in 9th grade learned more quickly, and understood loops better while learning C# or Java [1]. Today, some undergraduate level introductory computer science courses use a blocks-before-text approach such as in Harvard's CS 50, a transition from *Scratch* to C, and in Berkeley's CS 10, students move from *Snap!* to Phyton [2]. The main reason for using block-based tools is to make programming learnable. Difficulty of learning coding during university education causes a high percentage of students in United States to change their computer science major before graduation or drop-out from college education [3]. More countries having similar concerns in computer science education turned their attention to develop new or revised curriculum that can advance "computational thinking" of students, a term introduced by Wing [4].

Some governments, such as those of Britain, Estonia and Finland, have started serious initiatives to make coding part of

the school curriculum. In the US, administration was a strong proponent of expansion of programming teaching, President Obama announced Computer Science For All Initiative in January 2016 by stating "In the coming years, we should built on the progress, by… offering every student the hands-on computer science and math classes that make them job-ready on day one". The United States has no national curriculum, but the biggest public school systems started programs to integrate coding and computer skills to their curriculum, such as New York City and Los Angeles Unified [5]. The worldwide movement towards exposing all students to computer science is elaborated next.

2. Coding in Curriculum

2.1. Programming Skills in Europe's Curriculum

European Schoolnet 2015 publication reported that 16 European countries integrated coding in the curriculum at national, regional or local level, including Austria, Hungary, Denmark, France, Spain, Portugal, and Bulgaria [6]. Coding has been part of Estonia's elementary education since 2012 with a pilot program called "Proge Tiiger" [7]. In 2013, England wanted to be the first country in the world to make computer programming a compulsory school subject at all levels. Department for Education announced that by September 2014, computing was going to replace ICT as a national curriculum subject at all key stages. Each key stage was expressed as a series of bullet points in the program of study document.

For children between ages 5 to 14, there are three distinct stages for the England's new computing curriculum and the range of computing skills included in key stage 3 (11-14 year-olds) is as follows [8]:

(1) designing computational abstractions that model the state and behavior of real-world objects;

(2) understanding at least two commonly used algorithms for sorting and searching;

(3) using two or more programming languages, at least one of which should be a written programming language, to solve a variety of computational problems. They should be able to create data structures such as arrays and use functions to write modular programs;

(4) understanding how simple Boolean logic, such as AND, OR and NOT, is used to determine which part of a program is executed;

(5) understanding the hardware and software components that make up networked computer systems and how they interact;

(6) understanding how various data types can be represented and manipulated in the form of binary digits.

Although England's program of study document covers all the strands of a new curriculum with a focus on programming, teachers were left to their own devices in introducing the subjects' content (or the bullet points of each key stage) and to do it with some flexibility. As stated by Kemp "The order of these points does not denote their significance, nor should it influence the sequence of your teaching. The amount of time given to any one aspect is up to you. However, it would be unwise to ignore one strand or give too much emphasis to one aspect to the detriment of the others. How you deliver the course content remains in your hands"[9].

Students in Finland, from ages 7 to 16, will learn the basics of programming as a part of its national core curriculum starting from autumn 2016. To reach new education goals for digital skills, Finland is following a path similar to England, "every school will have a different approach to how programming will be taught as the Finish education system is highly decentralized" Thomas Vikberg explained, counselor of education at the Finish Ministry of Education [10].

2.2. Coding and Curriculum in United States

As mentioned before, the absence of a national curriculum in United States leaves federal administration in each state on their own in case of education policies. However, Obama administration's *No Children Left Behind* programs urged many school districts and education departments to take action. In September 2016 New York City mayor Bill de Blasio announced that *Computer Science For All Initiative* (CS 4 ALL) has raised $ 20 million in private donations. The initiative CS ALL also funded an early project of New York City Department of Education called "Software Engineering Program" (SEP) [11].

Between 2015 and 2016, the SEP served 42 teachers and more than 3000 students in grades 6, 7, 9, 10, and 11th grades across nine high schools and nine middle schools. SEP goals were: (1) to increase the number of high school graduates, particularly from traditionally under-represented groups, who are ready to enter new and emerging high-tech fields, and (2) to develop students' computational thinking and problem solving skills in real-world contexts [12]. Later CS 4 ALL initiative was expanded by adding SEPjr program for reaching elementary students. SEPjr program promotes a more collaborative culture with teacher team structure, where a team including 3-7 teachers with one lead teacher per grade level.

A broad policy shift in state of California was needed for putting computer science instructions to more public schools. One of the statewide actions was a push to add basic computer science curriculum standards to the *Next Generation Science Standards* (NGSS), that represents the blueprint of how science is taught in kindergarten through 12 grades. Also, Los Angeles Unified and San Francisco Unified school districts decided "to introduce K-12 computer science curriculum in advance of the implementation of statewide standards on the subject"[13].

Recent examples of teaching guidance for coding mentioned above indicates individual teachers will have considerable degree of autonomy over how they teach computing topics and especially coding mandated by the national curriculum in their countries. This paper presents a methodology for teaching coding to children as young as five and it is possible to apply this methodology in classes through using open source tools for coding. Next section introduces

some of the tools developed for teaching young children how to code.

3. Tools for Coding

There exist many tools for teaching children how to program or code. Even some games designed for kids help teach basic problem solving methods. For example, *Lightbot Jr* is specifically designed for ages 4-8 and player can choose Girlbot or Boybot with the goal of solving puzzles that use game mechanics. *Lightbot Jr* is sold as an app but open source tools for coding are available free of charge such as the popular *Scratch*. *Scratch* was developed in 2005, by Mitchel Resnick of the Massachusetts Institute of Technology (MIT) and then, in 2014, released in a simplified version called Scratch Jr for young children between ages 5-7.

3.1. Scratch Jr. and Tynker

As a derivative of the Scratch language, *Scratch Jr* was developed by MIT Media Lab in cooperation with Tufts University and The Playful Invention Company. In *Scratch* environment, dragging blocks into a coding area and then snapping them together creates code. Developers of this programming tool noted that "We have always been intrigued and inspired by the way children play and build with Lego bricks. Given a box full of them, they immediately start tinkering, snapping together a few bricks, and the emerging structure then gives them new ideas" [14].

Tynker is a visual programming language and multimedia-authoring tool launched in April 2013. Similar to Scratch, *Tynker* uses visual code blocks to introduce logic and programming concepts to children. Through their "Hour of Code" page, this platform provides free activities, mainly games and stories, for children to learn code and be creative at the same time. In school, *Tynker*'s curriculum allows students to work on their projects and learn the fundamentals programming found in all object-oriented programming languages.

3.2. Blockly and Code.org

Google's *Blockly* was launched in June 2012, as a graphical language implemented in JavaScript let's you program by dragging and dropping code blocks onto a design surface. At its core, *Blockly* is a client-side JavaScript library for creating visual block programming editors that can also be compiled into Dart and Phyton code too. Two years later, intended to "engage kids in constructive, meaningful learning" according to Google executive assistant Jennifer Vaden Barth, Google Research launched *Blockly Games* [15]. Similar to *Blockly,* this was an open-source project of Google and the new platform incorporated a set of educational games that teach programming concepts -such as loops and conditionals- to children.

A child can engage with seven graduated activities in *Blockly Games* starting with a simple puzzle to introduce the idea snapping together Blockly's interlocking pieces. 'Maze' is the next game that starts simple "but gets progressively more difficult in order to provide an introduction to loops and conditionals and in 'Bird' control flow is explored with increasingly complex conditions" [16]. Fourth game is the 'Turtle' introducing nested loops to kids so that they can paint a picture and then publish their creation to Reddit.

According to Google, the games are designed to be self-paced and self-teaching, they minimize the use of syntax except for the seventh "Pond JS" game, in which a transition from blocks to conventional text-based programming is made [17]. The player writes actual JavaScript code in this game to beat the faster moving objects (see Figure 1).

Founded by Hadi and Ali Partovi in 2013, *code.org* is a non-profit organization that particularly aims school students in teaching computer science. In 2014, alongside their 'Hour of Code' campaign, *code.org* posted a one-hour tutorial to build and customize a *Flappy Bird* video game through site's visual programming language using blocks. Students can easily form a string of commands by clicking and then dragging blocks in *code.org*. Hadi Partovi explained that "these kinds of block commands are how most computer programmers first learn the basics" of coding [18]. "The platform that creates these visual blocks of code is *Blockly*" and its open source library is available for free as mentioned above [19].

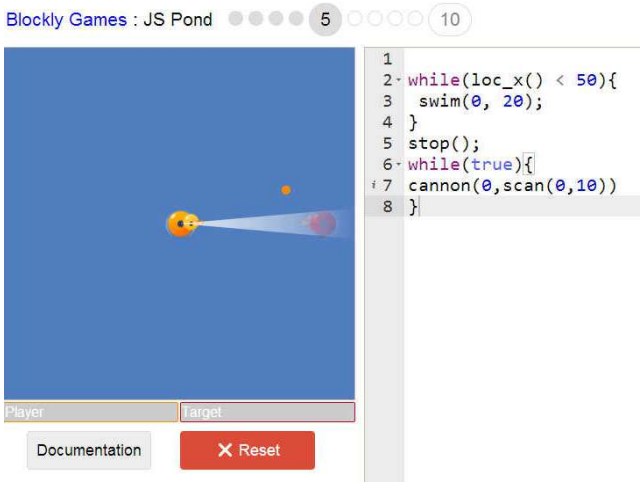

Figure 1. *Pond JS game of Blockly Games.*

The first "Hour of Code Challenge" was launched at the end of 2013 on website of *code.org*. Supported by US President Barack Obama this initiative of *code.org* had the goal of enticing young people to write short snippets of code using *Blockly*. The initiative was successful as more than 20 million students from different countries completed hour of code tutorials and written 668,008,671 lines of code in just weeks following the launch [20]. Code Studio is a website run by *code.org* as the source of coding for children and requires signing up. Code Studio is different than 'Hour of Code' activities as they are not limited to one hour of short coding tutorials.

Almost all of the tools reviewed in this study are open source and work on any web browser. They run on computers

and mobile devices such as tablets. The ease of access and use may provide an advantage in the process of teaching coding skills to children. However, these tools and their websites do not offer any comprehensive or the most appropriate strategy to teach young children how to code. Tynker's curriculum and *code.org's* 'Hour of Code' activities can be effective but from the instructor's point of view, a complete guideline is still missing when they want to design a course that utilizes a methodology for instructing the basic concepts of programming. A new methodology for instructors is included in the next section.

4. Methodology for Teaching Coding to Kids

4.1. Approaches to Teaching Coding

Previous research and studies show positive effects of visual programming environments on students' learning. Gülbahar and Kalelioğlu found "a slight improvement in the students' self-confidence in their problem solving ability" while working with Scratch [21]. Lee concluded that, to unleash learners' imagination, creative, entertaining and interdisciplinary curriculum can be crated by educators through Scratch [22]. "Hour of Code" initiative is also considered for tackling the diversity problem in computer science education [23] and Code Studio reached their benchmark for 1 million girls and 1 million African-Americans and Hispanic students at January 8, 2015 [24]. More studies can be cited as they provided an extensive literature review on studies based on Scratch tool [25]. Scratch is heavily investigated because of its language support, simple interface and appeal to more age groups, yet none of these studies addressed to the problem of methodical introduction of coding to children.

As an example of coding instruction in practice based on direct instruction, CS 4 ALL's SEPjr. program, focused on professional development of teachers "to prepare them to offer a scope and sequence to engage all grades while giving them the flexibility to focus on a grade band implementation" [26]. This program aims to introduce foundational computer science concepts through open-ended creative computing platforms and block-based programming languages, such as Scratch, robotics and maker education. SEPjr's curriculum is comprised of four primary units – Computer Science Fundamentals, Robots, Project Based Learning, and Physical Computing [27].

A more advanced, multi-year, standards-aligned computer science education program SEP provided a curriculum for grades 6 to 12. SEP curriculum intends to give students instruction and experience in the following areas: computer programming, robotics, web design, physical computing, game design and 3D printing.

SEP program also made a call for programmers and educators to design curriculum for NYC public middle and high school teachers without a Computer Science (CS) background. Some of the curriculum requirements for all topics were listed as [28]:

(1) One 45-minute lesson per day

(2) Solidify CS concepts explored in the SEP curriculum via project-based learning teaching (PBL) methodology

(3) Engaging and full independent and group learning activities to teach concepts, foment and enable practice, and create artifacts.

SEP curriculum for 6th grade students has "Computer programming with Scratch" topic covered in fall term made up of 3 units: (1) Basic Computing Concepts, (2) Structure of a Program, (3) Solutions for Efficient Programming. Structure of a Program learning activities include:

(1) Make Different Things Happen at The Same Time

a. Exercise #1: Translating ideas into Scratch script

b. Exercise #2: Making Sprites Do Different Things

(2) Make Multiple Sprites Interact With Each Other

a. Exercise #1: Making 2 Sprites Have a Conversation

b. Exercise #2: Introducing the Broadcast Block

(3) Design a Flow of a Program: Loops, Booleans and Conditional Statements

a. Exercise #1: Making loops and repeats: The Repeat (), Forever and Repeat Until () Blocks

b. Exercise #2: Making conditional statements: The If () Then and If () Then, Else Blocks

c. Exercise #3: Create and read scripts

To facilitate learning authors argue that coding concepts must be instructed following an order. We ranked seven fundamental concepts of coding and these are algorithms, loops, if statements, functions and procedures, graphics, variables, and finally lists and indexes. Instructors must start teaching algorithms, loops and if statements first out these seven fundamental concepts. Especially, starting with these three concepts plays an important role in teaching coding to kids ages 4 and up.

In essence, our methodology proposes that an instructor must start teaching algorithms first, then loops and if conditionals as the third concept when dealing with young children. Instructors can follow a two-tiered methodology to teach coding concepts to young kids (see Figure 2). Traditionally programming is taught at high school or undergraduate level starting with variables. Early research also revealed that students in grade 7 and younger responded best to the graphics-rich programming environment whereas high school students preferred more conventional textual environment [29].

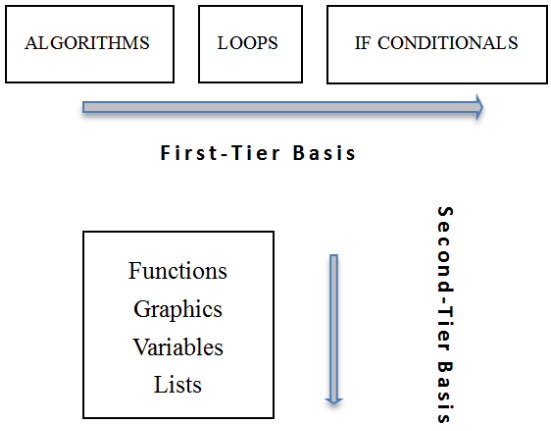

Figure 2. *Methodology for teaching coding to kids 4+.*

Our experience shows that starting with the concept of variables can cause problems with younger children. Visual programming tools such as Scratch, Blockly Games and *code.org* provide more interactive and fun sessions to kids. For example, second game "Maze" in Blockly Games is an introduction to loops and conditionals. Third game "Bird" presents more complex conditionals and in *code.org*, lesson 1, scene 1 and scene 2 are simple examples for algorithms as shown in Table 1 [30].

Table 1. *Examples of code.org practices.*

Code.org	Lesson	Scene	Ages
Algorithms	1	1-2-3-4-5-6	4 and up
Loops	2	5-6-7-8-10	6 and up
If conditionals	3	7-8-14	8 and up

4.2. CiK Platform for Coding Instruction

Author of this study developed a coding library as a programing platform for kids called "Çocuklar için Kodlama Kütüphanesi" or CiK [31]. This library was designed to serve many purposes, such as having a platform in Turkish that is expandable. CiK library is accessible through web and coded in JavaScript therefore it works on any web browser (main library is called cik.js). Offline use of the library is also possible by downloading cik.zip files. Also any CiK application can be developed easily through its APIs.

CiK Library is made up of methods and properties similar to any high-level programming language and uses natural language elements. Chain methods are also usable with CiK. An editor is also created for CiK where user can write codes and then see the results on bi-colored screens (see Figure 3). Editor of CiK is based on CSS and HTML technologies.

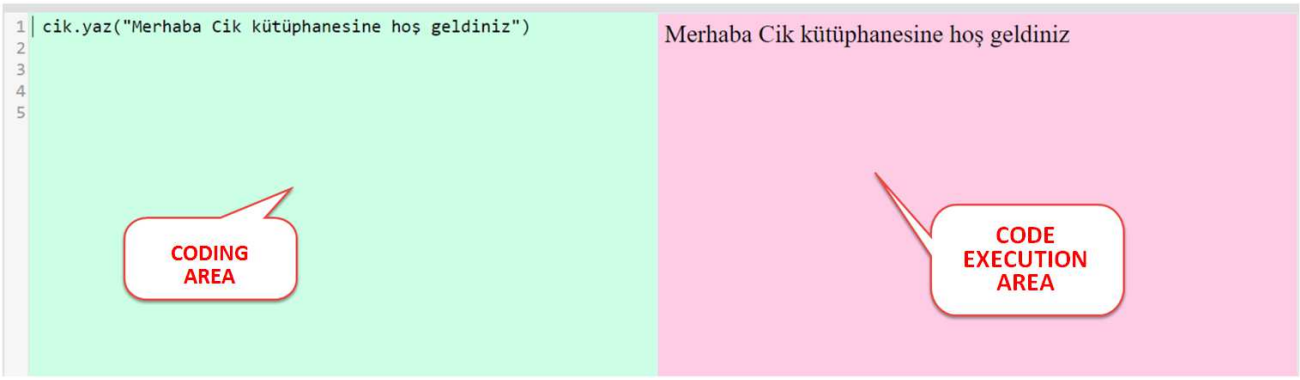

Figure 3. *CiK editor screen.*

CiK library also supports drawing methods along with some objects like circle, rectangle, and star where user can assign values. Finally, CiK library comes with pre-loaded pictures that are added by using method "cik.resim.", for example "cik.resim.balık" method presents a fish picture in the editor (see Figure 4). CiK platform is also accessible through GitHub and still under development.

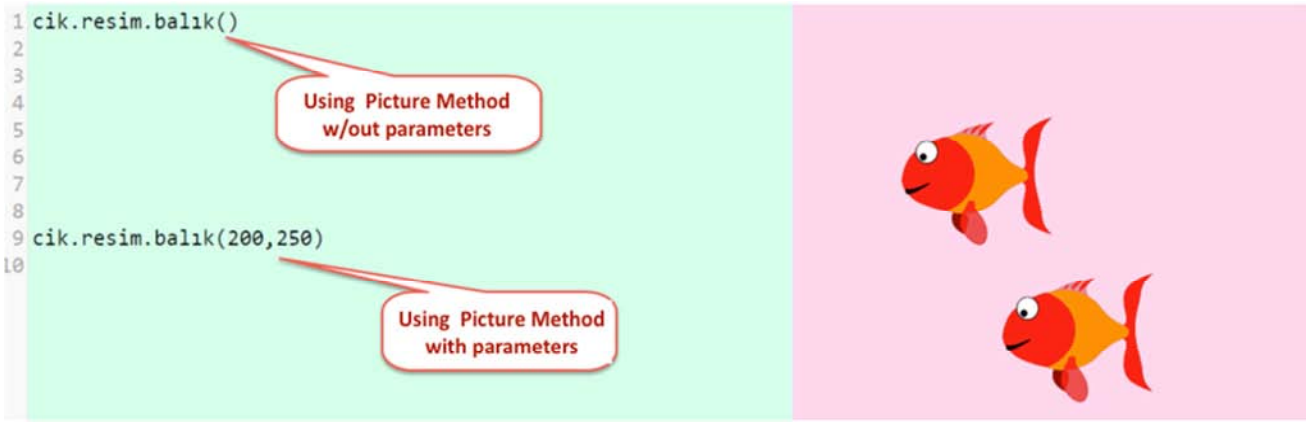

Figure 4. *CiK library graphics coding.*

5. Conclusion

Teaching coding to children will give them a skill for life. Many different programming language tools exist for teaching children how to code. Some of these tolls are developed as open source and free ones reached to millions of children around the world. However, none of these tools guide instructors by providing comprehensive methodologies so that, fundamentals of coding can be easily understood by minors.

This study introduced a methodology for instructors based on authors' ranking of fundamental concepts in coding. Seven

of these concepts were listed and the methodology is defined through a two-tier approach. When dealing with young children, two-tier approach proposes that an instructor must start teaching algorithms first, then loops and if conditionals as the third concept within the first-tier. Rest of the programming basics such as functions, graphics, variables and lists fall into second-tier in this methodology. A new coding platform for kids called "CiK" is also presented. Compatible to all browsers, CiK is a platform independent coding library that supports Turkish with its editor. Based on JavaScript, CiK is an open-source coding environment still under development.

References

[1] M. Armoni, O. Meerbaum-Salant, and M. Ben-Ari, "From Scratch to 'Real' Programming", ACM Transactions on Computing Education, 14 (4), 2015, pp. 25:1-15.

[2] D. Bau, J. Gray, C. Kelleher, J. Sheldon, and F. Turbak, "Learnable Programming: Blocks and Beyond", Communications of the ACM, 60 (6), 2017, pp. 72-80.

[3] B. Moskal, D. Lurie, and S. Cooper, "Evaluating the Effectiveness of a New Instructional Approach", Proceedings of the 35th SIGCSE technical symposium on Computer science education, Virginia, USA, 2004, pp. 75-79.

[4] J. M. Wing, "Computational thinking", Communications of the ACM, 49 (3), 2006, pp. 33-35.

[5] A. Kamenetz, (2016), The President Wants Every Student To Learn Computer Science. How Would That Work? http://www.npr.org/sections/ed/2016/01/12/462698966/the-president-wants-every-student-to-learn-computer-science-how-would-that-work.

[6] 2015 European Schoolnet Report http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0

[7] P. Olson, (2012), Why Estonia has started teaching its first-graders to code? *Forbes,* http://www.forbes.com

[8] N. Heath, (2013). Programming set to be core of new computing classes for English kids. *ZDNet,* http://www.zdnet.com

[9] P. Kemp, (2014) Computing in the national curriculum: A guide for secondary teachers. *Computing at School,* http://www.computingatschool.org.uk/data/uploads/cas_secondary.pdf

[10] E. Haaramo, (2015) Building a digital future should coding be mandatory for every schoolchild. *ZDNet,* http://www.zdnet.com

[11] NYC Department of Education website http://cs4all.nyc/2016/09/23/cs4all-anniversary/

[12] SEP website http://sepnyc.org/about/

[13] P. Matio, (2016), California moves to catch up on K-12 computer science curriculum. EdSource https://edsource.org/2016/california-moves-to-catch-up-on-k-12-computer-science-curriculum/565265

[14] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, A. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: programming for all," Communications of the ACM, *52* (11), 2009, pp. 60-67.

[15] J. V. Barth, (2014). Summer games: Learn to program http://googleresearch.blogspot.com.tr/2014/08/summer-games-learn-to-program.html

[16] S. Gee, (2014) Blockly Games introduce kids to code. http://www.i-programmer.info/news/150-training-a-education/7673-blockly-games-introduce-kids-to-code.html

[17] T. R. Weiss, (2014) Google introduces kids to coding through Blockly Games Project. http://www.eweek.com

[18] J. Demmitt, (2015). Microsoft and *Code.org* will use Minecraft to teach kids basics of computer programming. http://www.geekwire.com

[19] L. Ibanez, (2015). Blockly makes it easier to learn code. https://opensource.com

[20] J. F. Lefferts, (2013). Preview of writing code for future. *The Boston Globe.* http://www.bostonglobe.com

[21] Y. Gülbahar, and F. Kalelioğlu, "The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective," Informatics in Education-An International Journal*,* vol. 13 (1), 2014, pp. 33-50.

[22] Y. Lee, Y. "Scratch: multimedia programming environment for young gifted learners," Gifted Child Today, vol. 34 (2), 2011, pp. 26–31.

[23] C. Wilson, "Hour of code: we can solve the diversity problem in computer science," ACM Inroads*,* vol. 5 (4), 2014, pp. 22-24.

[24] N. Ungerleider, (2015). *Code.org*: 1 million girls, 1 million African-Americans and Hispanic students learn to program. http://www.fastcompany.com

[25] Ş. Çatlak, M Tekdal F.Ç, Baz, "Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir DokümanİnceleMe Çalışması," Journal of Instructional Technologies & Teacher Education, vol. 4, 2015, pp. 13-25.

[26] SEP Jr. website http://sepnyc.org/sepjr/program-overview/

[27] Computer Science For All website http://cs4all.nyc/academic-programs/software-engineering-program-jr/sepjr-curriculum/

[28] SEP Curriculum Development Opportunities website http://cs4all.nyc/sep-curriculum-development-opportunities/

[29] C. Y. Cheung, G. Ngai, C. F. Chan, and W. Y. Lau, "Filling the gap in programming instruction: a text-enhanced graphical programming environment for junior high students," ACM SIGCSE Bulletin, vol. 41 (1), 2009, pp. 276-280.

[30] Z. Demirkol, Çocuklar için kodlama. Istanbul, 2016, Pusula.