

Looking at Algorithm Visualization through the Eyes of Pre-service ICT Teachers

Fatih Saltan

Department of Computer Education & Instructional Technology, Amasya University, Turkey

Copyright © 2016 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract The study investigated pre-service ICT teachers' perceptions of algorithm visualization (AV) with regard to appropriateness of teaching levels and contribution to learning and motivation. In order to achieve this aim, a qualitative case study was carried out. The participants consisted of 218 pre-service ICT teachers from four different universities. Data were obtained through an open-ended questionnaire (n=210) and interviews (n=8). The qualitative data were analyzed using content analysis techniques. The results indicated that about half of the pre-service teachers thought AV to be appropriate for use in elementary and middle schools. A smaller number of participants thought that using AV is appropriate in high schools and colleges. Almost all of the participants thought that AV effectively contributes to learning and teaching computer programming. Participants explained this effect in terms of seven properties of algorithm visualization environments: Visualization, Algorithmic Thinking, Enjoyable Learning, and Progressive Learning, Learning by Doing, Game-based Learning, and Scaffolding. Moreover, results showed that most of the pre-service ICT teachers believed that AV contributes effectively to motivation. They explained this positive contribution to motivation in terms of six properties of AV: Easy to use, Visual, Fun, Quick Produced, Active and Game-based.

Keywords Algorithm Visualization, ICT Teachers, Teaching/Learning Strategies, Computer Programming

1. Introduction

Teachers have a dual role as educators and innovators in information societies. Until recent decades, it was generally accepted that qualified teachers possessed content knowledge and pedagogical knowledge, but today, it is expected that good teachers learn emerging technologies and utilize them in the classroom, along with content and pedagogical knowledge [20]. In programming education, teachers have utilized a technology supported strategy called

algorithm visualization (AV). AV is defined as the graphical illustration of algorithms and data structure via software developed for this purpose [13, 35]. Almost all educational communities have a positive perception of AV [28]. AV is preferred in programming education to facilitate student understanding of the way computer algorithms function [14]. New strategies are needed in this field, because learning computer programming is not easy, even for college students enrolled in computer-related disciplines. College students who have basic programming knowledge perceive programming courses as difficult, because these courses require higher-order thinking skills [37]. Researchers have investigated several problems that may cause failure of some novice programming students and dropout of the programming course [8]. Some of the major problems identified for college students are abstract concepts, lack of understanding of the larger entities, lack of applications, and lack of practical and concrete learning situations [22]. A recently conducted study categorized the problems a novice learner faces as “programming knowledge (56%), programming skills (17%), understanding semantics (13%), and debugging (13%)” [29]. Regarding these problems, computer science educators and instructional technologists have been studying technological and pedagogical ways to make programming instruction easier and more effective. Surely, it is critical to employ instructional time in the most appropriate ways possible. In this regard, much algorithm visualization software has been developed to facilitate learning and teaching, and especially to support novice programmers. One of the first examples of algorithm visualization software is the Brown Algorithm Simulator and Animator (BALSA) that assists students in learning the fundamentals of computer programming through system-generated visualizations [3]. While some AVs can be accessed for free, only a few of them maintain an educational quality [35]. It is also important to highlight that after the concept was debuted in the mid-1970s, more than 500 algorithm visualization soft wares have been developed, but a comprehensive repository of AV software has yet to be established.

In the 1990s, TANGO, software which allowed learners to

create visual algorithm animations by utilizing C language, was developed [36]. Pierson and Rodger [31] also created a Java based system, named JAWAA, which was a “simple command language for creating animations of data structures and displaying them with a Web browser” (p.267). In the present decade, more interactive algorithm visualization environments have been introduced. For example, Carnegie Mellon University developed Alice, which enables novice programmers to develop attractive three-dimensional graphic animations [6]. Another popular algorithm visualization software is Scratch that developed by Massachusetts Institute of Technology, where allows learners to develop “interactive, media-rich projects” (p.1) [24, 39, 41]. Scratch allowed learners to begin programming with creating animations instead of writing code that make learners more motivated [39]. This nature of the Scratch provides creative developments and discovery learning [40]. Recent studies indicated that Scratch take learners interest for programming education [42]. It allowed learners utilize computational constructs [43] and improve programming skills [44]. Surely, it was helpful to development of positive perceptions toward programming [45]. In her study Ke also reported that Scratch is useful to make reflections about daily experiences [46]. Like the Scratch, Greenfoot is a recently developed algorithm visualization program that aims to teach computer programming in a visual way to those who have no prior programming experience, and it has gained great popularity [21]. App Inventor (AI) is a new algorithm visualization platform that allowed learners create mobile applications for Android-based smart phones without writing code [47]. Alpha version of AI was launched in 2009[48]. Graphical interface of AI is like Scratch [50]. Morelli and Colleagues asserted that AI is easy to learn, accessible and lead students to problem solving instead of coding [47]. It is also indicated that AI is a highly motivating and powerful instructional tool [49].

Instructors have been using AV software for different educational aims in different contexts. Generally they use AV to maintain individual learning, give homework, facilitate lectures and support laboratory assignments [15, 27]. In K-12 levels, Information and Communication Technology (ICT) teachers utilize AV software to make it easier for students to start learning computer programming. Starting programming education at an early age has been shown to foster mathematical and problem solving skills [5]. Another study, which investigated the contribution of specific AV software to students’ achievement, showed that AV contributes to the development of mathematical and analytical thinking, problem solving and development of logic [4]. Generally, studies on algorithm visualization have focused on the contributions of AV to learning, and students’ motivation and perceptions. Few studies have investigated the perceptions of teachers about the usage of AV. Teachers are decision maker and key actors in instructional environments; their beliefs shape all instructional activities [7, 11]. According to Robertson, Macvean, and Howland

[32], the long term impact of algorithm visualization, like other innovations in education, depends on the degree to which teachers utilize them in practice.

It is important to know what teachers think about AV specifically ICT teachers. This study attempts to address this important need in the literature by investigating the perception of pre-service ICT teachers about the use of AV for teaching computer programming.

1.1. The purpose of the study

The purpose of this study was to investigate pre-service ICT teachers’ perceptions of algorithm visualization with regard to appropriateness of teaching levels and contribution to learning and motivation. Within the scope of this purpose, the following specific research questions were addressed:

1. What do pre-service ICT teachers think about the teaching levels for which AV use is suitable?
2. What are the views of pre-service ICT teachers regarding the contribution of AV to teaching and learning computer programming?
3. What are the views of pre-service ICT teachers regarding the contribution of AV to motivation for learning programming?

2. Method

In order to understand pre-service ICT teachers’ perception of algorithm visualization, this study employed a qualitative study. This approach was chosen in order to gain comprehensive and in-depth information from the selected subjects [1, 30, 38]. In this study, the researcher focused on perceived appropriateness of AV to teaching levels and perceived contribution of AV to learning and motivation. In order to ensure the trustworthiness of the findings, the credibility of the study was addressed [23]. Triangulation, an important technique to establish credibility, was utilized in this study. Data were gathered using two common techniques: interview and questionnaire. Furthermore, coder reliability was calculated for analysis.

2.1. Samples

The participants in this study consisted of 214 pre-service ICT teachers who were in their last year of study in the computer education and instructional technology departments at four different universities. Most of the participants (71%) graduated from a vocational high school. The participants included 104 females and 110 males. Both female ($M=3.93$) and male ($M=4.03$) pre-service teachers had about four years of previous programming experience. The pre-service teachers had all experienced using algorithm visualization to some degree. At a minimum, they had used one AV program like Scratch, Alice or Greenfoot. Demographics of the participants are summarized in Table 1.

Table 1. Demographics of the Pre-service ICT teachers

Gender	Vocational High School	Non-Vocational High School	Total
Female	72	32	104
Male	81	29	110
Total	153	61	214

2.2. Data Collection

Data was gathered through a questionnaire and interviews. The questionnaire consisted of 4 demographic questions and 5 open-ended questions. All pre-service teachers answered the questionnaire voluntarily. Interviews consisting of semi-structured questions were conducted with eight pre-service teachers, and the interviews were recorded via a recording device.

2.3. Data Analysis

The data were analyzed using content analysis techniques [30, 26]. In this study, the analysis was organized according to the four stages defined by Yildirim and Simsek [38]. In the first two stages, data were coded and themes were developed from these codes. In the last two stages, themes were organized and described.

First, the interviews were transcribed. The answers to the open-ended questions were coded using Nvivo8, a qualitative data analysis software. Then, themes were developed from the codes. In this process, the researcher made sure that themes were internally consistent and distinct from each other [25]. The themes were verified and confirmed by two independent researchers. Inter-coder reliability was calculated as 78 percent.

3. Results

3.1. What Do Pre-service ICT Teachers Think about the Teaching Levels for Which Use of AV Is Suitable?

Pre-service ICT teachers were asked for which teaching levels AV use is suitable and why. The data analysis indicated that about half of the participants (55%) thought that AV should be used in elementary or middle school. While some of the pre-service ICT teachers (13%) thought that AV was appropriate only for high school, some participants (10%) regarded AV as suitable for both middle school and high school. Moreover, some of the participants (9%) indicated that AV can be used in all grades, K-12. Also, some participants (13%) regarded the use of AV to be suitable in grades K-12 and college. The participants' opinions are presented under the seven categories in Table 2.

When it was investigated why participants chose lower grades instead of high grades, content analysis showed that participants mainly prefer using AV in lower grades because they believe that it is simple, visual and helpful for children.

In this regard, one pre-service ICT teacher said, "AV is simple and suitable for young ages. It can improve learning and memory at those ages."

Likewise, pre-service ICT teachers who highlighted that AV is not useful for students in upper grades believe that AV is too simplistic and does not cover the important topics in computer programming. For example, one pre-service ICT teacher said,

"It is unnecessary for us to use AV in college and high school. The logic behind algorithms is an issue that can be understood in one lesson. It will be better for us to learn more helpful computer programs. At least, this software should not be use any more."

Table 2. The appropriate teaching grades for using AV

Themes/Categories	Frequency (n=196)
Elementary and middle school	51(26%)
Only middle school	32(16%)
Only elementary school	25(13%)
Only high school	25(13%)
K12+ University	25(13%)
Middle and high school	20(10%)
K12	18(9%)

3.2. What Are the Views of Pre-service ICT Teachers Regarding the Contribution of AV to Teaching and Learning Computer Programming?

Analysis of the interviews and the responses of the pre-service ICT teachers to open-ended questions showed that all but fourteen participants (7%) thought that AV contributes effectively to learning and teaching computer programming. For example one participant stated,

"AV software is an easy and enjoyable program that every teacher should learn and should teach to students. It uses a drag and drop approach, like Legos, and does not require writing code. It is very good because it ensures that students learn computer programming while using their imaginations to design a game-like application."

Table 3. Reasons why pre-service ICT teachers perceive AV as useful

Themes/Categories	Frequency(n=128)
Visualization	45 (35%)
Algorithmic Thinking	34 (27%)
Enjoyable Learning	20 (16%)
Progressive Learning	10(8%)
Learning by Doing	9(7%)
Game-based Learning	6(5%)
Clues as Scaffold	4(3%)

Results showed that pre-service ICT teachers regard AV as a useful instructional strategy for several reasons. Resulting categories are listed in Table 3. On the other hand, participants who did not find AV useful for teaching and

learning highlight that AV is too simple for learning coding. One of them said,

“It (AV software) does not make it easier to learn computer programming. On the contrary, it takes users to a utopian world. AV software is not appropriate for the seriousness of learning computer programming.”

Many pre-service ICT teachers stated that AV enhances learning and teaching because it provides a visual learning environment. They stated that AV makes programming codes and structures tangible. Therefore, they perceived AV to be most suitable for when students first start learning programming. They also indicated that AV programs support learning by providing clear and apprehensible interfaces. In this regard, one of them stated, “Yeah, easy. Programming code is abstract, but students can see the structures and outputs in AV software. It is easier to learn.”

Another theme that explains why pre-service ICT teachers regard AV as effective is algorithmic thinking (27%). Many pre-service ICT teachers stated that AV facilitates algorithmic thinking by letting users understand the logic behind computer programs. Therefore, it contributes to teaching and learning computer programming.

It was also found that pre-service ICT teachers believe that AV contributes to learning programming because AV ensures the learning process is fun. It was highlighted that using AV programs were enjoyable on the country of the traditional instructional approaches. A participant indicated

“AV facilitates learning computer programming. It is because compare to other programming language, AV is easier to understand and more enjoyable. AV software is more suitable and beneficial because it contains visual expressions and visual content which the students like.”

Three other important themes emerged regarding the instructional methods that AV employed. Some pre-service ICT teachers indicated that AV programs utilize progressive learning (8%). Learners can progress in stages with this approach. Participants also asserted that AV employs the learning-by-doing method (7%). Learners are actively involved when using AV.

The third instructional method highlighted by pre-service ICT teachers was game-based learning (5%). They indicated that, like the other two methods, usage of game-method AV facilitates learning and teaching.

Lastly, a few participants noted that AV programs provide clues as scaffolding (3%), which contributes to instruction.

3.3. What Are the Views of Pre-service ICT Teachers Regarding the Contribution of AV to Motivation for Learning Programming?

Results showed that while most of the pre-service ICT teachers (87%) thought that AV contributes effectively to motivation, some of them (13%) did not. Further analysis indicated that participants who believe using AV making a positive contribution to motivation gave reasons that fell under six main categories. The resulting categories are listed in Table 4.

Table 4. Reasons why using AV motivates learners

Themes/Categories	Frequency(n=135)
Simple & Easy	41 (30%)
Visual	29 (21%)
Funny	26 (19%)
Quick Development	21 (16%)
Interaction & Active Learning	12(9%)
Game	6(4%)

About one in three participants noted that the reason using AV motives learners is because it is simple and easy to use (30%). Pre-service teachers also indicated that using AV makes learning visual (21%) and fun (19%), so it generates necessary motivation for learners. For example, an interview participant said,

“It is being visual and working without writing code makes students more motivated. Therefore, AV encourages learners to work on programming.” Another important category that emerged as a part of the motivation cultivated by AV was quick development (16%). Pre-service ICT teachers thought that learners feel satisfied when they immediately see the results of what they do in AV programs. Thus, using AV drives them to continue learning. In this regard, one participant said,

“I think AV is motivating. It is because we can immediately see results of code block we created using AV. The resulting products (programs) let individuals to be more motivated and willing to progress create more complex programs”.

Moreover, some of the ICT teachers emphasized interaction with AV programs (9%) as a source of motivation for learners. The analysis showed that these teachers believed that active involvement in AV environments successfully keeps learners’ attention. Lastly, a few participants noted that the game-like nature (4%) of AV programs improves learners’ motivation. They highlighted that students, especially young students, want to play computer games, so AV appeals to them.

4. Discussion

The goal of this study was to investigate pre-service ICT teachers’ perceptions of algorithm visualization with regard to appropriateness of teaching levels and contribution to learning and motivation. Data analysis revealed that about half of the pre-service teachers thought that AV is appropriate for use in elementary and middle school. A smaller number of participants thought that using AV is appropriate in high school and college. To date, there has not been an extensive study showing in which teaching levels AV is most preferred and in which teaching levels using AV is most effective. However, the findings of this study correlate with the emphasis of “novice programmer” in previous studies [e.g. 33]. In the literature, there is a significant emphasis on learners who have just begun

learning computer programming. It has been supposed that novice programmers are the potential audience of AV-based instruction [24]. Although the main audience of the most of the popular AV software developers is K-12 students, some developers also address college students and life-long learners.

The study also showed that almost all of the pre-service ICT teachers thought that AV contributes effectively to learning and teaching computer programming. Participants explained this effect with seven properties of algorithm visualization environments: Visualization, Algorithmic Thinking, Enjoyable Learning, and Progressive Learning, Learning by Doing, Game-based Learning, and Scaffolding.

Similarly, a recent study showed that AV allows users, especially novice programmers, to focus mainly on design and development rather than programming syntax [9]. In this respect, another study found that using AV decreases the cognitive load through “avoidance of the handling of the syntax errors” [17]. Moreover, Brennan and Resnick [2] indicate that designing interactive media in visual algorithm environments help learners develop computational thinking skills. On the other hand, several studies found little or no evidence that AV is effective or efficient [e.g. 10,16]. In their comprehensive meta-analysis study, Hundhausen, Douglas and Stasko [12], indicate that educators do not want to use AV, mainly because they do not believe using AV is as effective and efficient as traditional approaches. However, the present study showed that teachers have good reasons to utilize AV and most believe in its effectiveness.

In addition, most of the pre-service ICT teachers thought that AV contributes effectively to motivation. They explained this positive contribution to motivation with six properties of AV which are Ease of Use, Visual, Fun, Quick Production, Active and Game-based Learning. In previous studies, the lively and popular aspect of AV software has been clearly highlighted [12, 17, 36]. In the present study, several ICT teachers indicated that AV cannot contribute to learners’ motivation because it is too simple. However, most ICT teachers saw its simplicity as a motivating factor. Most novice learners want to see an immediate outcome. The simple development environment of AV provides this satisfaction and builds confidence. Both satisfaction and confidence are major aspects of the Kellers’ ARCS motivation theory [17]. Moreover, the attention of the learners, another aspect of the ARCS theory, is gained with the visualization and interaction in AV programs.

Overall, this study showed that pre-service ICT teachers thoroughly evaluate the usage of AV in teaching computer programming. They indicated the contributions and appropriateness to teaching level of AV.

REFERENCES

[1] Bogdan, R. & Biklen, S. K. *Qualitative Research for Education*, Allyn & Bacon, Boston, 1992.

- [2] Brennan, K., & Resnick, M. New frameworks for studying and assessing the development of computational thinking, 2012 annual meeting of the American Educational Research Association, 1-25, 2012.
- [3] Brown, M. H., & Sedgewick, R. A system for algorithm animation, *ACM SIGGRAPH Computer Graphics*, Vol.18, Issue.3, 177-186, 1984.
- [4] Calder, N. Using Scratch: An Integrated Problem-Solving Approach to Mathematical Thinking, *Australian Primary Mathematics Classroom*, Vol.15, Issue.4, 9-14, 2010.
- [5] Çetin, E. The Effect of Computer Programming Education on Children’s Problem-Solving Skills, (Unpublished master’s thesis), Gazi University, Ankara, 2012.
- [6] Cooper, S., Dann, W., Pausch, R. Alice: a 3-D Tool for Introductory Programming Concepts, *Journal of Computing Sciences in Colleges*, Vol.15, Issue.5, 107-116, 2000.
- [7] Ertmer P. A. Teacher pedagogical beliefs: the final frontier in our quest for technology integration?, *Educational Technology Research and Development*, Vol.53, Issue.4, 25-39, 2005.
- [8] Federici, S. A minimal, extensible, drag-and-drop implementation of the C programming language, 2011 conference on Information technology education, 191-196, 2011.
- [9] Grover, S., & Pea, R. Computational Thinking in K–12 A Review of the State of the Field, *Educational Researcher*, Vol.42, No.1, 38-43, 2013.
- [10] Gurka, J., Citrin, W. Testing effectiveness of algorithm animation, *IEEE Symposium on Visual Languages (VL’96)*. 182–189, 1996.
- [11] Hammond L. D. Constructing 21st-century teacher education, *Journal of Teacher Education*, Vol.57, No.3, 300-314, 2006.
- [12] Hundhausen, C. D. Toward effective algorithm visualization artifacts: designing for participation and communication in an undergraduate algorithms course, (Unpublished Ph.D. dissertation), Department of Computer and Information Science, University of Oregon, 1999.
- [13] Hundhausen, C. D. Integrating algorithm visualization technology into an undergraduate algorithms course: ethnographic studies of a social constructivist approach, *Computers & Education*, Vol.39, Issue.3, 237-260, 2002.
- [14] Hundhausen, C. D., & Brown, J. L. Designing, visualizing, and discussing algorithms within a CS 1 studio experience: An empirical study, *Computers & Education*, Vol.50, Issue.1, 301-326, 2008.
- [15] Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. A meta-study of algorithm visualization effectiveness, *Journal of Visual Languages and Computing*, Vol.13, Issue.3, 259-290, 2002.
- [16] Jarc, D. J., Feldman, M. B., and Heller, R. S. Assessing the benefits of interactive prediction using Web-based algorithm animation courseware, 31st SIGCSE Technical Symposium on Computer Science Education (SIGCSE’00), 377–381, 2000.
- [17] Kelleher, C., & Pausch, R. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers, *ACM Computing Surveys (CSUR)*, Vol.37, Issue.2, 83-137, 2005.

- [18] Kelleher, C., Pausch, R., & Kiesler, S. Storytelling Alice motivates middle school girls to learn computer programming, SIGCHI conference on Human factors in computing systems, 1455-1464, 2007.
- [19] Keller, J. M. Development and use of the ARCS model of motivational design, *Journal of Instructional Development*, Vol. 10, Issue. 3, 2-10, 1987.
- [20] Koehler, M., & Mishra, P. What is technological pedagogical content knowledge (TPACK)?, *Contemporary issues in technology and teacher education*, Vol.9, No.1, 60-70, 2009.
- [21] Kölling, M. The greenfoot programming environment, *ACM Transactions on Computing Education (TOCE)*, Vol.10, Issue.4, 14, 2010.
- [22] Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. A study of the difficulties of novice programmers, *ACM SIGCSE Bulletin*, Vol. 37, Issue. 3, 14-18, 2005.
- [23] Lincoln, Y. S., & Guba, E. G. *Naturalistic inquiry*, CA: Sage, Beverly Hills, 1985.
- [24] Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. The scratch programming language and environment, *ACM Transactions on Computing Education (TOCE)*, Vol. 10, Issue. 4, 16, 2010.
- [25] Marshall, C., & Rossman, G. B. *Designing Qualitative Research* (4th ed.), Sage, Thousand Oaks, 2006.
- [26] Miles, M. B., & Huberman, A. M. *Qualitative Data Analysis: An Expanded Sourcebook* (2nd ed.), Sage, California, 1994.
- [27] Naps, T. L. Algorithm visualization in computer science laboratories, *ACM SIGCSE Bulletin*, Vol.22, Issue.1, 105-110, 1990.
- [28] Naps, T., Röbling, G., Almström, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., and Ángel Velázquez-Iturbide J. Exploring the role of visualization and engagement in computer science education, *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education (ITiCSE-WGR'02)*, 131-152, 2002.
- [29] Özmen, B., Altun, A. Undergraduate Students' Experiences in Programming: Difficulties and Obstacles. *Turkish Online Journal of Qualitative Inquiry*, Vol. 5, Issue. 3, 9-27, 2014.
- [30] Patton, M. Q. *Qualitative Research and Evaluation Methods* (3th ed), Thousand Oaks, CA: Sage, 2002.
- [31] Pierson, W. C. & Rodger, S. H. Web-based animation of data structures using JAWAA, twenty-ninth SIGCSE technical symposiums on Computer science education, 267-271, 1998.
- [32] Robertson, J., Macvean, A., & Howland, K. Robust evaluation for a maturing field: the train the teacher method, *International Journal of Child-Computer Interaction*, Vol. 1, Issue. 2, 50-60, 2013.
- [33] Robins, A., Rountree, J., & Rountree, N. Learning and teaching programming: A review and discussion, *Computer Science Education*, Vol. 13, Issue. 2, 137-172, 2003.
- [34] Shaffer, C. A., Cooper, M. L., Alon, A. J. D., Akbar, M., Stewart, M., Ponce, S., & Edwards, S. H. Algorithm visualization: The state of the field, *ACM Transactions on Computing Education (TOCE)*, Vol. 10, Issue. 3, 9, 2010.
- [35] Shaffer, C. A., Heath, L. S., & Yang, J. Using the Swan data structure visualization system for computer science education, *ACM SIGCSE Bulletin*, Vol. 28, No. 1, 140-144, 1996.
- [36] Stasko, J. T. Using student-built animations as learning aids, *ACM Technical Symposium on Computer Science Education*, 25-29, 1997.
- [37] Tan, P. H., Ting, C. Y., & Ling, S. W. Learning difficulties in programming courses: undergraduates' perspective and perception, *International Conference on Computer Technology and Development 2009 (ICCTD'09)*, 42-46, 2009.
- [38] Yıldırım, A., & Şimşek, H. *Sosyal Bilimlerde Nitel Araştırma Yöntemleri*, Seçkin Yayıncılık, Ankara, 2006.
- [39] Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. Learning Basic Programming Concepts by Creating Games with Scratch Programming Environment. *Procedia-Social and Behavioral Sciences*, 191, 1479-1482, 2015.
- [40] Kobsiripat, W. Effects of the Media to Promote the Scratch Programming Capabilities Creativity of Elementary School Students. *Procedia-Social and Behavioral Sciences*, 174, 227-232, 2015.
- [41] Brennan, K. Best of Both Worlds: Issues of Structure and Agency in Computational Creation, In and Out of School. Ph.D. Dissertation. MIT Media Lab, 2012.
- [42] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, J., S., Silverman, B., & Kafai, Y. Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67, 2009.
- [43] Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239-264, 2013.
- [44] Begosso, L. C., & da Silva, P. R. Teaching computer programming: A practical review. In 2013 IEEE Frontiers in Education Conference (FIE) (pp. 508-510). IEEE, 2013.
- [45] Genç, Z., & Karakuş, S. Tasarımla Öğrenme: Eğitsel Bilgisayar Oyunları Tasarımında Scratch Kullanımı. 5th International Computer & Instructional Technologies Symposium, 2012.
- [46] Ke, F. An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26-39, 2014.
- [47] Morelli, R., de Lanerolle, T., Lake, P., Limardo, N., Tamotsu, E., & Uche, C. Can android app inventor bring computational thinking to k-12. In Proc. 42nd ACM technical symposium on Computer science education (SIGCSE'11), 2011.
- [48] H. Abelson. App Inventor for Android. Google Research Blog, <http://googleresearch.blogspot.com/2009/07/app-inventor-for-android.html>, July 31, 2009.
- [49] Wagner, A., Gray, J., Corley, J., and Wolber, D., Using App Inventor in a K - 12 Summer Camp, SIGCSE '13 , 621 – 626, 2013.
- [50] Resnick, M., Maloney, J., Monroy - Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y., S cratch: Programming for All, *CACM* , 52, (11), 60 - 67, 2009.