Informatics in Education, 2019, Vol. 18, No. 2, 403–426

DOI: 10.15388/infedu.2019.19

The Effect of Unplugged Coding Activities on Computational Thinking Skills of Middle School Students

Betül TONBULOĞLU[1], İsmail TONBULOĞLU[2]

[1]*Ministry of National Education, İstanbul, Turkey*
[2]*Yıldız Technical University, İstanbul, Turkey*
e-mail: betultonbuloglu@gmail.com, ismailtonbuloglu@gmail.com

Received: July 2019

Abstract. The purpose of the study is to examine the effect of unplugged coding activities carried out with middle school students on their computational thinking skills. This study employed nested-mixed design, which is a mixed research method; the data were supported by including the qualitative phase into an experimental study. In this frame, a group of 114 middle school students consisting of 5th graders were given coding training titled "Keşfet Project – I Discover Coding" by using unplugged coding content. The Computational Thinking Scale was applied to the students at the beginning and end of the training; the results obtained from the scale were analyzed by means of a paired t test. Finally, it was found out that unplugged coding activities had a positive effect on the improvement of computational thinking skills of the students. An examination of the subfactors revealed that there is statistically no significant change in the problem solving skill despite the positive impact observed on creativity, algorithmic thinking, collaboration and critical thinking skills. Following the analysis of observation and daily data, the findings obtained revealed that the students usually displayed high levels of motivation and class participation in unplugged coding activities, they had difficulty in concretizing certain concepts as well as subjects requiring mathematical knowledge; various teaching methods and techniques were used in classes; the students liked the activities especially due to their appealing nature and their relation to daily life; however, there were occasional problems with scheduling of activities and teamwork due to overcrowded class size; the students experienced problems in achieving outcomes such as perceiving the relationship between computer science and mathematics and analyzing the given problem, and could have difficulty in associating between computer science and mathematics or between the subjects learned and the computer lesson, and in analyzing a given problem.

Keywords: unplugged coding, computational thinking, coding class in primary education.

1. Introduction

The term 'computational thinking skill' was first used by Papert in 1980. Wing (2006) defines this term as "the cognitive processes involved in formulating problems so that they are represented in a form that can be carried out by a human or machine." The main objective in computational thinking is not to force people to think like computers but to reveal a multidimensional thinking process by using the abstract reasoning skill like a computer scientist (Wing, 2006). ACARA defines computational thinking as "a problem-solving method that involves various techniques and strategies which may include organizing data logically, breaking down problems into parts, designing and using algorithms and models" (ACARA, 2012). Understanding computational thinking requires perceiving basic concepts of computer sciences and the abstraction- creation skill which offers solutions to problems (Bower & Falkner, 2015). It is considered that learners can become able to solve the problems in different fields by using the computational thinking skill (Barr, Harrison and Conery, 2011). Kalelioğlu and Keskinkılıç (2017) also defined computational thinking as an important skill required to design products for computer technologies and use these technologies, and emphasized the importance of assisting children in acquiring computational thinking skills, beginning from early years of life.

The use of technology involves computational thinking skill, and computer science is used for the acquisition of computational thinking skill. The rapid progress of technology has, in addition to shaping the qualifications aimed in the new century, also raised awareness and interest in computer sciences. It is getting more and more widely recognized that teaching children the basic concepts and skills of computer sciences is of considerable value. It seems that, among many technological applications such as software, hardware and the Internet, software applications have gained prominence in computer sciences in the recent years. Kert (2017) indicated that the integration of courses on programming languages in syllabuses beginning from early years of life constitutes the most critical aspect of the new approach that emerged among computer science instructors. Bocconi *et al.* (2016), pointed out that coding and related software are among important steps of computational thinking.

There are numerous recent studies suggesting that coding skills should be considered among basic skills and they are of equivalent importance as reading – writing skills. It is also suggested that coding needs to be learned in order to find solutions to problems in daily life and to interact with the physical world (Kalelioğlu and Keskinkılıç, 2017). Resnick (2012) emphasized that the knowledge of coding does not only mean that the learner will build a career as a computer scientist or software developer but also that the learner can thereby improve creative, systematic and analytic thinking skills; and that it contributes to collaborative work and easier manifestation of ideas and feelings. Barger (2008), who pointed out to the importance of teaching coding at schools, suggested that the purpose of computer education should be to show students how to program a computer rather than to teach them how to become a computer operator using available applications; Barger also criticized the non-inclusion of basic programming language into the process of education, by the analogy of teaching how to read without teaching how to write.

Coding can be taught by means of unplugged applications, block-based programming environments, robot programming, text- based programming or various methods such as interdisciplinary applications. Computer Science Unplugged Activities (B^3)

which may be preferred before any other programming activities (Kalelioğlu, 2017) have a wider scope of subjects than teaching programming, and address many areas such as human- computer interaction, data compression, and artificial intelligence, etc. (Bell, 2014). Bell, Witten and Fellows (1998) argued that learners who learn by means of such activities are able to learn many concepts of computer science such as binary numbers, image processing, data compression, search and ranking algorithms without using a computer. Basic features of B^3 activities are summarized as follows: they do not involve the use of computer; they are usually based on a game or contest which will raise interest, curiosity and motivation of children; kinesthetic context is provided by using physical objects such as cards and weights; the activities usually involve interaction with other students and encourage them to discover the answers by trial & error; most of the activities are feasible to implement by using only inexpensive equipment which are available at schools; they involve exchange of ideas and diversification of activities, and use of storytelling to attract the interest of students (Nishida *et al*., 2009). Taub, Armoni and Ben-Ari (2012) indicated that the main objective of computer science unplugged is to change attitudes and intentions of students towards the computer science, making it more interesting for students and ensuring them to choose research. They noted that CS unplugged has specifically changed the opinions of students on the nature of computer sciences, ensuring them to understand that the key concepts of computer sciences do not focus on the computer, that computer sciences are more than programming and require mathematical thinking.

Bell, Witten and Fellows (1998), who performed the earliest studies on B^3 activities, presented unplugged coding activities for free of charge in the project CS Unplugged; the activities were later collected in a book for free of charge through improvements and updates made, including language support for many countries. Additionally, there are many unplugged coding activities such as code.org Unplugged, Hour of Code, Keşfet Projesi- Kodlamayı Keşfediyorum (Discover Project – I Discover Coding), Tospaa (Unplugged Coding Game) and Bilge Kunduz (Beaver). Cortina (2015) suggested that computer science unplugged activities enable students to work in collaboration, supporting the use of their creativity and problem – solving skills. Paul (2015) mentioned that computer science unplugged activities enable students to ask questions to their classmates, set them in motion and give them directions for purpose of achieving the goal, and to actually learn how to make a similar effort in order to execute a process on the computer. On the other hand, Feaster *et al*. (2011) conducted a research at high school level by using computer science unplugged activities, whereupon they reached the conclusion that there was no positive change in students' attitudes towards the computer science in terms of conceptual understanding and perceived content. Taub, Armoni and Ben-Ari (2012) found out that computer science unplugged activities decreased motivation to study while ensuring partial improvement in students' ideas, and that some of the activities were distantly related to computer sciences.

It is indicated that the studies on computer science unplugged are not much exhaustive although the studies on coding are quite common in the literature (Rodriguez, Rader and Camp, 2016) and that there are very few studies aiming at identifying what kind of effects computer science unplugged activities have on students (Taub, Armoni and

Ben-Ari, 2012). In this frame, the study was aimed to examine the effect of computer science unplugged activities on computational thinking skills of middle school students. The study sought answers to the following research questions:

1. Do the unplugged coding activities have an effect on the computational thinking skills of middle school students?
    a. Is there a significant difference between pre-test and post-test scores regarding computational thinking skills of students?
2. How are the experiences of students and teachers in terms of unplugged coding activities?

2. Methodology

2.1. *Research Model*

Mixed method was used in the research. Mixed method focuses on the collection, analysis and mixture of both qualitative and quantitative data in a single study or a series of studies; the combined use of qualitative and quantitative data ensures better understanding of the research problem than of any other method used alone (Creswell & Plano Clark, 2017). This study employed nested-mixed design, which is a mixed research method. In the nested- mixed design, the researcher may include a qualitative phase into a quantitative phase such as an experimental study, or a quantitative phase into a qualitative phase such as a case study (Creswell & Plano Clark, 2014, p.80). As for this study, the data were supported by including a qualitative phase into an experimental study. "Pre-test – post-test experiment design", which is an experimental model, was taken as basis in the quantitative part of the study. The mentioned design is used in circumstances where there are no comparison groups but only the experimental group in the research (Erden, 1998).

In the qualitative part of the research, the daybooks kept by the teacher on weekly basis and observation forms were used to acquire information about the process. The research design is shown in Table 1.

Table 1
Research Design

Research Design	**Before Practice**	**Practice** Unplugged Coding Activities	**After Practice**
Quantitative Research Design	Computational Thinking Scale		Computational Thinking Scale
Qualitative Research Design		Observation Form Daybooks	

2.2. *Study Group*

The study group of the research consists of 114 students studying at the 5th grade of a public middle school in Ataşehir district of Istanbul province. The school currently has 4 divisions for 5th graders, who have taken computer lesson for an equivalent period but not had any previous training on coding. The group consists of 63 male and 51 female students. The daily average time spent on the computer is 1–3 hours for 78 students, 3–5 hours for 22 students, and 5–8 hours for 14 students. The students were given coding education for 10 weeks, during which the unplugged coding activities in the educational content of "Discover Project – I Discover Coding" were used. Information Technologies lesson is 2 hours a week at the school where coding education is provided.

2.3. *Data Collection Tools*

The use of multiple data collection methods in a research is called "data triangulation". Data triangulation improves validity and reliability of findings by ensuring that the limitedness of a particular data collection method is overcome by another data collection method (Yıldırım and Şimşek, 2008). Below are the data collection tools employed in this research:

2.3.1. *Computational Thinking Skill Levels Scale*

The "Computational Thinking Skill Levels Scale", developed by Korkmaz, Çakır and Özden (2017) and adapted to Turkish language and middle school level by Korkmaz, Çakır and Özden (2015) was used to collect data on the computational thinking skills of the students. The scale consists of 22 items and it was designed as a 5-point Likert type scale. The scale contains five factors which are creativity, algorithmic thinking, collaboration, critical thinking and problem solving. The lowest score of the scale is 22 points and the highest score is 110 points. Cronbach alpha value for the Turkish version of Computational Thinking Skill Levels Scale was found to be 0.809. Cronbach Alpha values for the sub-dimensions of the scale are between 0,640 and 0,867.

2.3.2. *Observation Form and Daybooks*

The observation form was prepared by the researcher in order to enable the teacher to monitor the classroom during the lesson, observe student behavior within the process and reflect opinions about them; the form was filled out by the teacher after each lesson. Furthermore, the teacher regularly took field notes of the experiences in the implementation phase, activities performed, his/her observations related to the flow of lesson within the process, reactions of the students, subject which students have difficulty in learning, and positive & negative items in the lesson plan.

2.4. *Empiric Process*

At the beginning of the research, the Computational Thinking Skill Scale was applied as a pre-test to measure computational thinking skills of students (Korkmaz, Çakır and Özden, 2015); later, the students were given coding education for 10 weeks by using the unplugged coding activities in the content of the Keşfet Project- I Discover Coding (Kodlamayı Keşfediyorum).

The Keşfet Project (www.kesfetprojesi.org) was created and implemented in 2014 in collaboration with Google and the Istanbul Provincial Directorate of the Ministry of National Education. Within the scope of this project, different contents such as education, classroom activities, video and game cards for "conscious and secure internet use" for 5th grade students were made available to teachers, parents and children. Based on this project experience, "Discover Coding Project" was developed in addition to these contents. The contents developed within the "discover coding project" have been used as supportive educational content in many schools and classroom activities. Unplugged coding activities in this project aims to improve students' Algorithmic Thinking, Creating Algorithms, Parsing, Coping with Uncertainty, Evaluating the Solution, Generalization, Debugging, Working in Cooperation, Logical Inquiry, Cause and Effect Relationship, Automation, Parallel Processing, Sequencing, Abstraction, Systematic Thinking, Data Collection, Data Visualization, Data Analysis skills. At the end of these activities, students are expected to develop the following skills (Keşfet Project web site):

- Recognize the basic concepts and approaches in problem solving,
- Bring solutions to the problems they face in daily life,
- Analyze problems, identify and solve appropriate steps,
- Operators will be able to use expressions and equations in place in problem solving and realize the importance of process priority,
- Recognize the concept of algorithms, create algorithms for the solution of problems, recognize and edit errors in algorithms,
- Will be able to recognize flow chart components and functions and will be able to create flow chart of an algorithm.

In the first week of the project activities, the students were firstly informed about the problem, problem solving and problem solving steps, and then divided into groups in order to ensure them to reinforce the learned skills. The groups were introduced to the wolf- sheep – cabbage problem and the students were encouraged to bring solution offers to the problem in line with the problem solving steps. Following the discussion on the solution of this problem, each group was given a Tower of Hanoi puzzle as the second activity; rules of the game for two, three and four items were announced and the students were given time to solve the problem. Then, the students were asked about the steps and strategies they followed for the solution of the problem, with the problem- solving steps repeated. (Fig. 1. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf; http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)

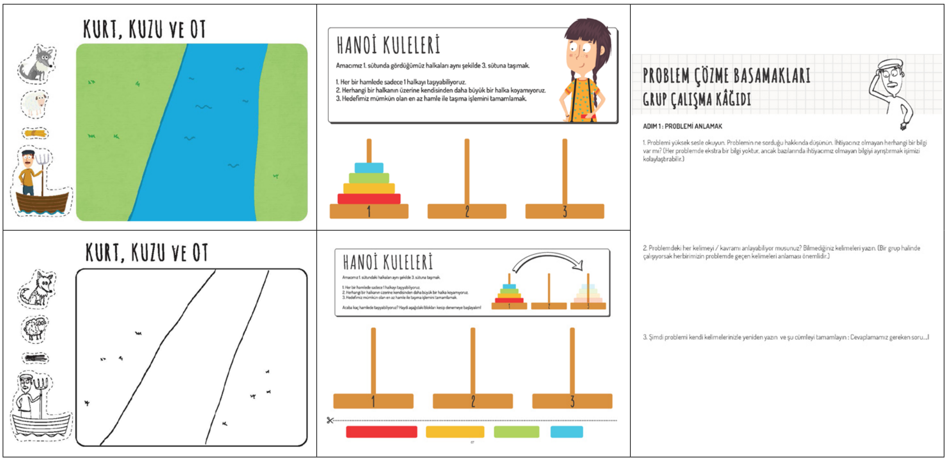
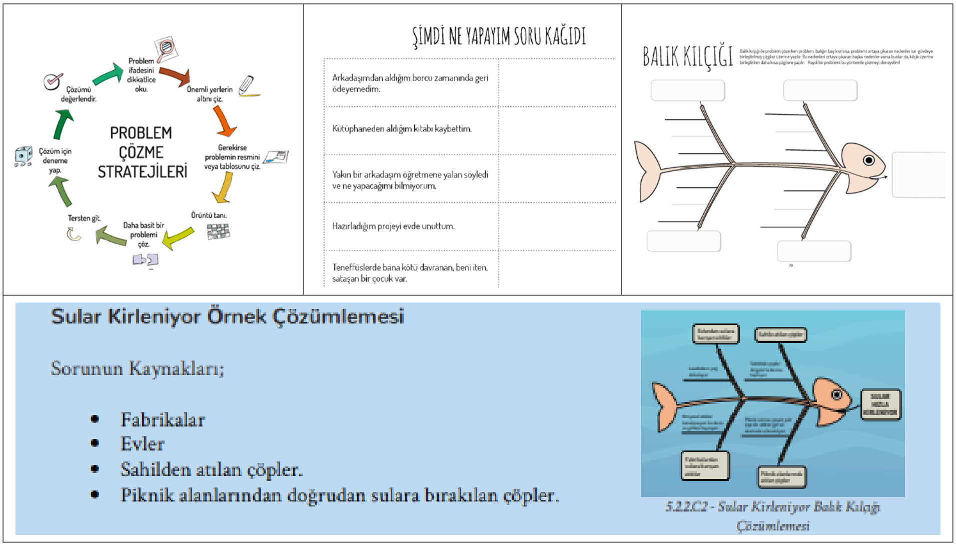

KURT, KUZU VE OT
HANOİ KULELERİ
Amacımız 1. sütunda gördüğümüz halkaları aynı şekilde 3. sütuna taşımak.
1. Her bir hamlede sadece 1 halkayı taşıyabiliyoruz.
2. Herhangi bir halkanın üzerine kendisinden daha büyük bir halka koyamıyoruz.
3. Hedefimiz mümkün olan en az hamle ile taşıma işlemini tamamlamak.
1
2
3
PROBLEM ÇÖZME BASAMAKLARI
GRUP ÇALIŞMA KÂĞIDI
ADIM 1 : PROBLEMİ ANLAMAK
1. Problemi yüksek sesle okuyun. Problemin ne sorduğu hakkında düşünün. İhtiyacınız olmayan herhangi bir bilgi var mı? (Her problemde ekstra bir bilgi yoktur, ancak bazılarında ihtiyacımız olmayan bilgiyi ayrıştırmak işimizi kolaylaştırabilir.)
2. Problemdeki her kelimeyi / kavramı anlayabiliyor musunuz? Bilmediğiniz kelimeleri yazın. (Bir grup halinde çalışıyorsak herbirimizin problemde geçen kelimeleri anlaması önemlidir.)
3. Şimdi problemi kendi kelimelerinizle yeniden yazın ve şu cümleyi tamamlayın : Cevaplamamız gereken soru...!
KURT, KUZU VE OT
HANOİ KULELERİ
Amacımız 1. sütundaki halkaları aynı şekilde 3. sütuna taşımak.
1. Her bir hamlede sadece 1 halkayı taşıyabiliyoruz.
2. Herhangi bir halkanın üzerine kendisinden daha büyük bir halka koyamıyoruz.
3. Hedefimiz mümkün olan en az hamle ile taşıma işlemini tamamlamak.
Acaba kaç hamlede taşıyabiliyoruz? Haydi aşağıdaki blokları kesip denemeye başlayalım!
1
2
3

Fig. 1. Wolf- sheep – cabbage problem and Tower of Hanoi puzzle student worksheets.

In the second week, the following scenario on water pollution was shared with the students after informing them about problem solving strategies and receiving their opinions by the question – answer method. The students were asked to analyze the causes of this problem and bring solution offers to the problem by using the fish bone method in groups; different fish bone design were developed for different problems. (Fig. 2. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)


PROBLEM ÇÖZME STRATEJİLERİ
Problem ifadesini dikkatlice oku.
Önemli yerlerin altını çiz.
Gerekirse problemin resmini veya tablosunu çiz.
Örüntü tanı.
Daha basit bir problemi çöz.
Tersten git.
Çözüm için deneme yap.
Çözümü değerlendir.
ŞİMDİ NE YAPAYIM SORU KAĞIDI
Arkadaşımdan aldığım borcu zamanında geri ödeyemedim.
Kütüphaneden aldığım kitabı kaybettim.
Yakın bir arkadaşım öğretmene yalan söyledi ve ne yapacağımı bilmiyorum.
Hazırladığım projeyi evde unuttum.
Teneffüslerde bana kötü davranan, beni iten, sataşan bir çocuk var.
BALIK KILÇIĞI
Balık kılçığı ile problem çözerken problem, balığın baş kısmına, problemi ortaya çıkaran nedenler ise gövdeye birleştirilmiş çizgiler üzerine yazılır. Bu nedenleri ortaya çıkaran başka nedenler varsa bunlar da, kılçık üzerine birleştirilen daha kısa çizgilere yazılır. Haydi bir problemi bu yöntemle çözmeyi deneyelim!
70
SULAR HIZLA KİRLENİYOR
5.2.2.C2 - Sular Kirleniyor Balık Kılçığı Çözümlemesi

Sular Kirleniyor Örnek Çözümlemesi

Sorunun Kaynakları;

- Fabrikalar
- Evler
- Sahilden atılan çöpler.
- Piknik alanlarından doğrudan sulara bırakılan çöpler.

Fig. 2. Problem solving strategies and fish bone method student worksheets.

In the third week, the students were firstly shown a map image to notice the steps which must be taken in the problem solving process and make use of the suitable steps, and they were later asked to give directions to each other according to given scenarios. As the second activity, voluntary students in the classrooms were asked to give directions to the classmates on how to draw a certain shape, and the classmates were asked to draw that shape on a blank paper by listening to the instructions; the drawings were compared and the results were discussed. (Fig. 3. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)

In the fourth week, the concepts of data and information were explained for the comprehension of variables, constants and operations required for the solution of the problem; later, the concepts of constant and variable were introduced by means of the images of examples in daily life. The students were expected to create a data table showing their breakfast habits and to analyze that table; later, games in daily life were discussed in terms of their constants and variables. (Fig. 4. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)

In the fifth week, the students were asked to do practices on how to use the operators in mathematical operations for the comprehension of the concept of operator. For the comprehension of logical operators, the students were ensured to comprehend the 'and-or – not' operators by using them in bridge crossing activity; number guessing game was played. (Fig. 5. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)

In the sixth week, the students were read an algorithm story in order to introduce them to the concept of algorithm. Later, the students were divided into groups; each group was asked to split the operations in the instructions given to them into steps, respectively; the steps were described in a mix and the audience was expected to estimate

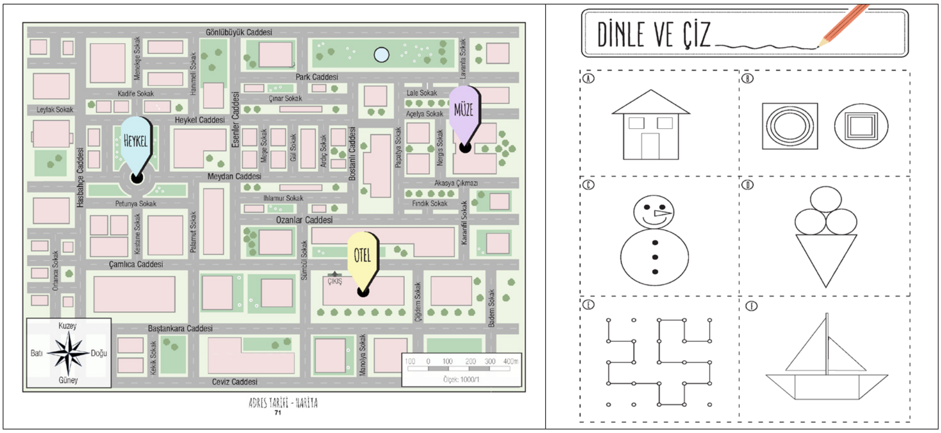
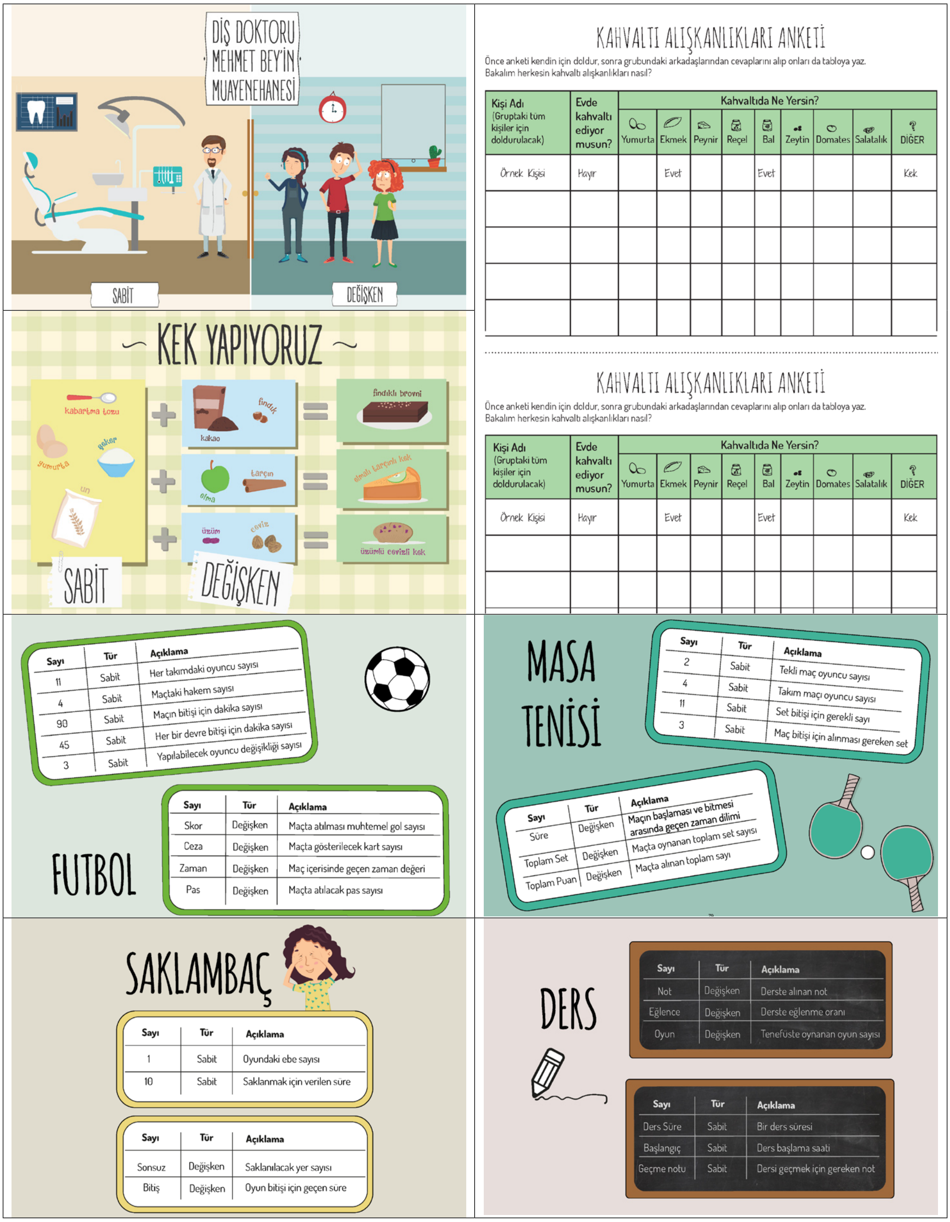


Fig. 3. "Map image" and "listen and draw" activities student worksheets.



KAHVALTI ALIŞKANLIKLARI ANKETİ

Önce anketi kendin için doldur, sonra grubundaki arkadaşlarından cevaplarını alıp onları da tabloya yaz. Bakalım herkesin kahvaltı alışkanlıkları nasıl?

Kişi Adı (Gruptaki tüm kişiler için doldurulacak)	Evde kahvaltı ediyor musun?	Kahvaltıda Ne Yersin?								
		Yumurta	Ekmek	Peynir	Reçel	Bal	Zeytin	Domates	Salatalık	DİĞER
Örnek Kişisi	Hayır		Evet			Evet				Kek

KAHVALTI ALIŞKANLIKLARI ANKETİ

Önce anketi kendin için doldur, sonra grubundaki arkadaşlarından cevaplarını alıp onları da tabloya yaz. Bakalım herkesin kahvaltı alışkanlıkları nasıl?

Kişi Adı (Gruptaki tüm kişiler için doldurulacak)	Evde kahvaltı ediyor musun?	Kahvaltıda Ne Yersin?								
		Yumurta	Ekmek	Peynir	Reçel	Bal	Zeytin	Domates	Salatalık	DİĞER
Örnek Kişisi	Hayır		Evet			Evet				Kek

FUTBOL

Sayı	Tür	Açıklama
11	Sabit	Her takımdaki oyuncu sayısı
4	Sabit	Maçtaki hakem sayısı
90	Sabit	Maçın bitişi için dakika sayısı
45	Sabit	Her bir devre bitişi için dakika sayısı
3	Sabit	Yapılabilecek oyuncu değişikliği sayısı

Sayı	Tür	Açıklama
Skor	Değişken	Maçta atılması muhtemel gol sayısı
Ceza	Değişken	Maçta gösterilecek kart sayısı
Zaman	Değişken	Maç içerisinde geçen zaman değeri
Pas	Değişken	Maçta atılacak pas sayısı

MASA TENİSİ

Sayı	Tür	Açıklama
2	Sabit	Tekli maç oyuncu sayısı
4	Sabit	Takım maçı oyuncu sayısı
11	Sabit	Set bitişi için gerekli sayı
3	Sabit	Maç bitişi için alınması gereken set

Sayı	Tür	Açıklama
Süre	Değişken	Maçın başlaması ve bitmesi arasında geçen zaman dilimi
Toplam Set	Değişken	Maçta oynanan toplam set sayısı
Toplam Puan	Değişken	Maçta alınan toplam sayı

SAKLAMBAÇ

Sayı	Tür	Açıklama
1	Sabit	Oyundaki ebe sayısı
10	Sabit	Saklanmak için verilen süre

Sayı	Tür	Açıklama
Sonsuz	Değişken	Saklanılacak yer sayısı
Bitiş	Değişken	Oyun bitişi için geçen süre

DERS

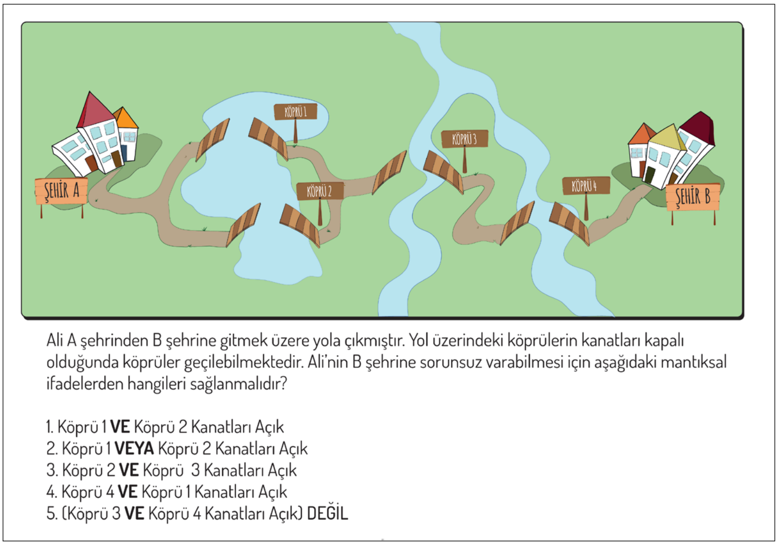
Sayı	Tür	Açıklama
Not	Değişken	Derste alınan not
Eğlence	Değişken	Derste eğlenme oranı
Oyun	Değişken	Tenefüste oynanan oyun sayısı

Sayı	Tür	Açıklama
Ders Süre	Sabit	Bir ders süresi
Başlangıç	Sabit	Ders başlama saati
Geçme notu	Sabit	Dersi geçmek için gereken not

Fig. 4. Variables, constants, breakfast habits table and games in daily life activities student worksheets.

the operation. The audience then tried to put the mixed steps of the operation into order. (Fig. 6. *Source:* http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf; http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf)

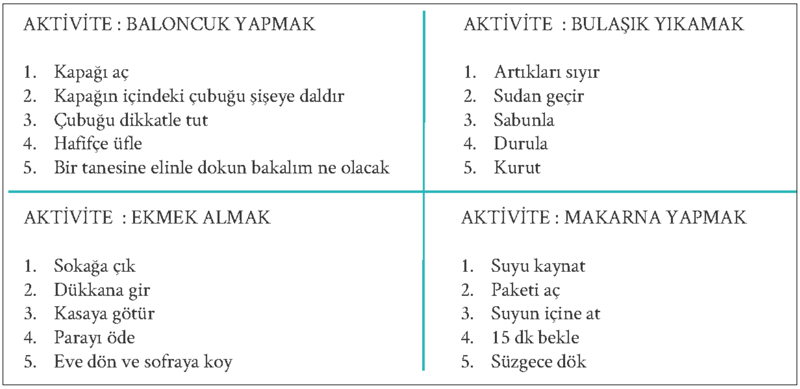
In the seventh week, the students were introduced to the components of the flow scheme and their functions; later, they were read a story and asked to draw the flow



Ali A şehrinden B şehrine gitmek üzere yola çıkmıştır. Yol üzerindeki köprülerin kanatları kapalı olduğunda köprüler geçilebilmektedir. Ali'nin B şehrine sorunsuz varabilmesi için aşağıdaki mantıksal ifadelerden hangileri sağlanmalıdır?

1. Köprü 1 **VE** Köprü 2 Kanatları Açık
2. Köprü 1 **VEYA** Köprü 2 Kanatları Açık
3. Köprü 2 **VE** Köprü 3 Kanatları Açık
4. Köprü 4 **VE** Köprü 1 Kanatları Açık
5. (Köprü 3 **VE** Köprü 4 Kanatları Açık) DEĞİL

Fig. 5. Bridge crossing activity student worksheet.



AKTİVİTE : BALONCUK YAPMAK

1. Kapağı aç
2. Kapağın içindeki çubuğu şişeye daldır
3. Çubuğu dikkatle tut
4. Hafifçe üfle
5. Bir tanesine elinle dokun bakalım ne olacak

AKTİVİTE : BULAŞIK YIKAMAK

1. Artıkları sıyır
2. Sudan geçir
3. Sabunla
4. Durula
5. Kurut

AKTİVİTE : EKMEK ALMAK

1. Sokağa çık
2. Dükkana gir
3. Kasaya götür
4. Parayı öde
5. Eve dön ve sofraya koy

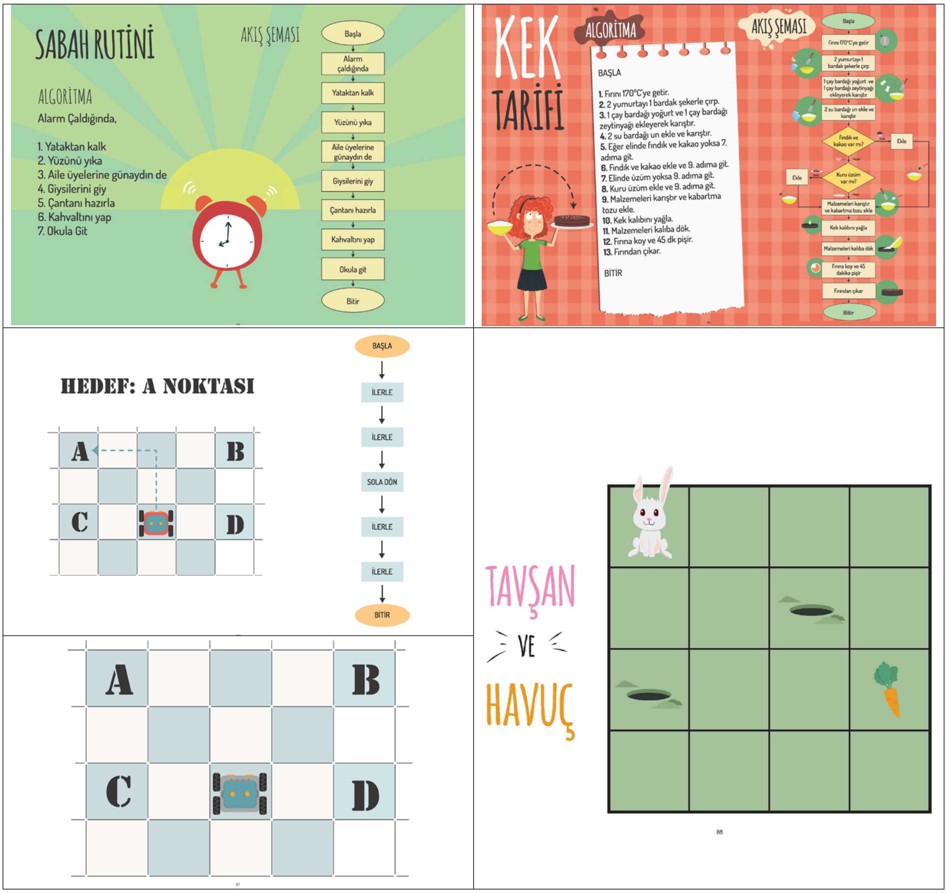
AKTİVİTE : MAKARNA YAPMAK

1. Suyu kaynat
2. Paketi aç
3. Suyun içine at
4. 15 dk bekle
5. Süzgece dök

Fig. 6. Confusion game activity student worksheet.

schemes of algorithms followed by the character in the story in different circumstances. Later, another scenario was presented to the classroom and the students were asked to build the flow scheme of the scenario by themselves. A robot and a map image were shown, and the students were asked to draw the flow scheme required for the robot to arrive at a certain spot. (Fig. 7. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)

In the eighth week, the students in groups were given instructions by their classmates on how to reach the target within a shape drawn, they were ensured to create mul-



SABAH RUTİNİ

ALGORİTMA

Alarm Çaldığında,

1. Yataktan kalk
2. Yüzünü yıka
3. Aile üyelerine günaydın de
4. Giysilerini giy
5. Çantanı hazırla
6. Kahvaltını yap
7. Okula Git

KEK TARİFİ

ALGORİTMA

BAŞLA

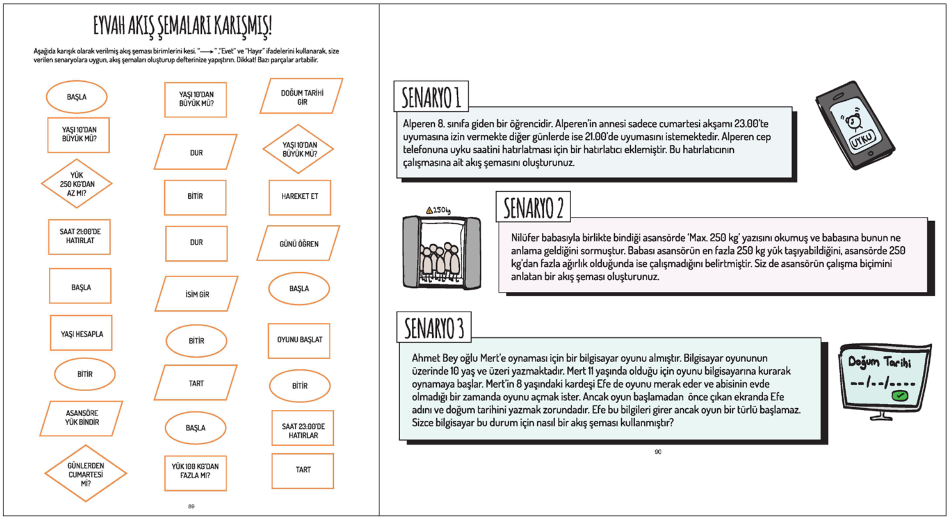
1. Fırını 170°C'ye getir.
2. 2 yumurtayı 1 bardak şekerle çırp.
3. 1 çay bardağı yoğurt ve 1 çay bardağı zeytinyağı ekleyerek karıştır.
4. 2 su bardağı un ekle ve karıştır.
5. Eğer elinde fındık ve kakao yoksa 7. adıma git.
6. Fındık ve kakao ekle ve 9. adıma git.
7. Elinde üzüm yoksa 9. adıma git.
8. Kuru üzüm ekle ve 9. adıma git.
9. Malzemeleri karıştır ve kabartma tozu ekle.
10. Kek kalıbını yağla.
11. Malzemeleri kalıba dök.
12. Fırına koy ve 45 dk pişir.
13. Fırından çıkar.

BİTİR

Fig. 7. Flow scheme activities student worksheets.

tiple and correct flow schemes. As the second activity, sheets with a mixed flow scheme were later handed out to the students in groups; the students were asked to create correct flow schemes by selecting suitable parts for three different scenarios on the sheets. (Fig. 8. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)

In the ninth week, the class was divided into groups and tangram sets were handed out to each group; the students were asked to look at the tangram shape created by the teacher and to give verbal instructions to their classmates on how to do the shape. During the second activity, the concept of variable was introduced to the classroom; a name in a poem was constantly changed by means of an envelope so that the concept of variable was identified with the envelope. (Fig. 9. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-dersi-ogrenci-materyalleri.pdf`; `http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf`)



EYVAH AKIŞ ŞEMALARI KARIŞMIŞ!

Aşağıda karışık olarak verilmiş akış şeması birimlerini kesi. "⟶", "Evet" ve "Hayır" ifadelerini kullanarak, size verilen senaryolara uygun, akış şemaları oluşturup defterinize yapıştırın. Dikkat! Bazı parçalar artabilir.

BAŞLA
YAŞI 10'DAN BÜYÜK MÜ?
YÜK 250 KG'DAN AZ MI?
SAAT 21:00'DE HATIRLAT
BAŞLA
YAŞI HESAPLA
BİTİR
ASANSÖRE YÜK BİNDİR
GÜNLERDEN CUMARTESİ Mİ?
YAŞI 10'DAN BÜYÜK MÜ?
DUR
BİTİR
DUR
İSİM GİR
BİTİR
TART
BAŞLA
YÜK 100 KG'DAN FAZLA MI?
DOĞUM TARİHİ GİR
YAŞI 10'DAN BÜYÜK MÜ?
HAREKET ET
GÜNÜ ÖĞREN
BAŞLA
OYUNU BAŞLAT
BİTİR
SAAT 23:00'DE HATIRLAR
TART
89

SENARYO 1

Alperen 8. sınıfa giden bir öğrencidir. Alperen'in annesi sadece cumartesi akşamı 23.00'te uyumasına izin vermekte diğer günlerde ise 21.00'de uyumasını istemektedir. Alperen cep telefonuna uyku saatini hatırlatması için bir hatırlatıcı eklemiştir. Bu hatırlatıcının çalışmasına ait akış şemasını oluşturunuz.

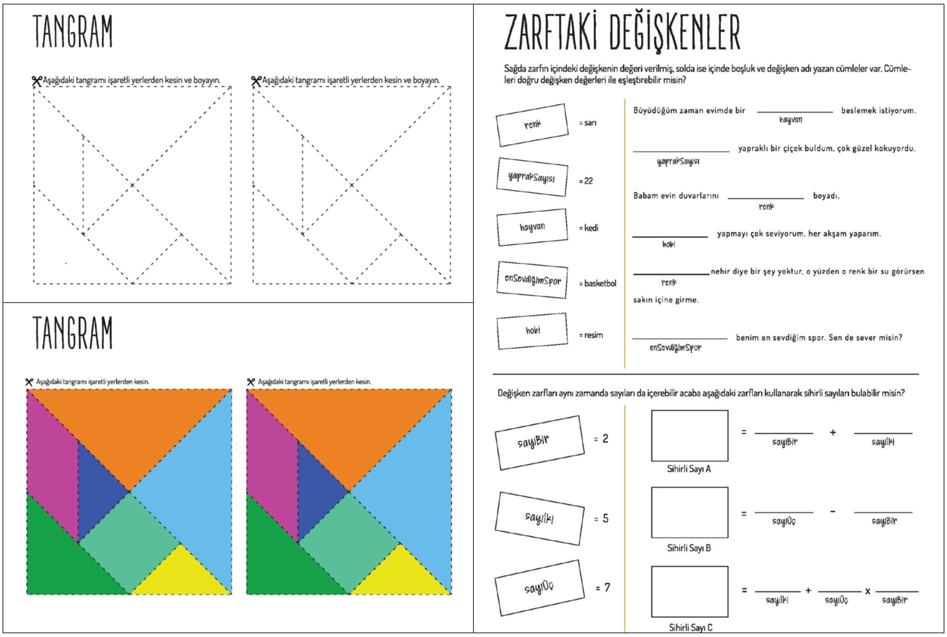
SENARYO 2

Nilüfer babasıyla birlikte bindiği asansörde 'Max. 250 kg' yazısını okumuş ve babasına bunun ne anlama geldiğini sormuştur. Babası asansörün en fazla 250 kg yük taşıyabildiğini, asansörde 250 kg'dan fazla ağırlık olduğunda ise çalışmadığını belirtmiştir. Siz de asansörün çalışma biçimini anlatan bir akış şeması oluşturunuz.

SENARYO 3

Ahmet Bey oğlu Mert'e oynaması için bir bilgisayar oyunu almıştır. Bilgisayar oyununun üzerinde 10 yaş ve üzeri yazmaktadır. Mert 11 yaşında olduğu için oyunu bilgisayarına kurarak oynamaya başlar. Mert'in 8 yaşındaki kardeşi Efe de oyunu merak eder ve abisinin evde olmadığı bir zamanda oyunu açmak ister. Ancak oyun başlamadan önce çıkan ekranda Efe adını ve doğum tarihini yazmak zorundadır. Efe bu bilgileri girer ancak oyun bir türlü başlamaz. Sizce bilgisayar bu durum için nasıl bir akış şeması kullanmıştır?

Fig. 8. Flow scheme activities student worksheets.



TANGRAM

Aşağıdaki tangramı işaretli yerlerden kesin ve boyayın.

Aşağıdaki tangramı işaretli yerlerden kesin ve boyayın.

TANGRAM

Aşağıdaki tangramı işaretli yerlerden kesin.

Aşağıdaki tangramı işaretli yerlerden kesin.

ZARFTAKİ DEĞİŞKENLER

Sağda zarfın içindeki değişkenin değeri verilmiş, solda ise içinde boşluk ve değişken adı yazan cümleler var. Cümleleri doğru değişken değerleri ile eşleştirebilir misin?

renk = sarı

yaprakSayısı = 22

hayvan = kedi

enSevdiğimSpor = basketbol

hobi = resim

Büyüdüğüm zaman evimde bir ________ (hayvan) beslemek istiyorum.

________ (yaprakSayısı) yapraklı bir çiçek buldum, çok güzel kokuyordu.

Babam evin duvarlarını ________ (renk) boyadı.

________ (hobi) yapmayı çok seviyorum, her akşam yaparım.

________ (renk) nehir diye bir şey yoktur, o yüzden o renk bir su görürsen sakın içine girme.

________ (enSevdiğimSpor) benim en sevdiğim spor. Sen de sever misin?

Değişken zarfları aynı zamanda sayıları da içerebilir acaba aşağıdaki zarfları kullanarak sihirli sayıları bulabilir misin?

sayıBir = 2

sayıİki = 5

sayıÜç = 7

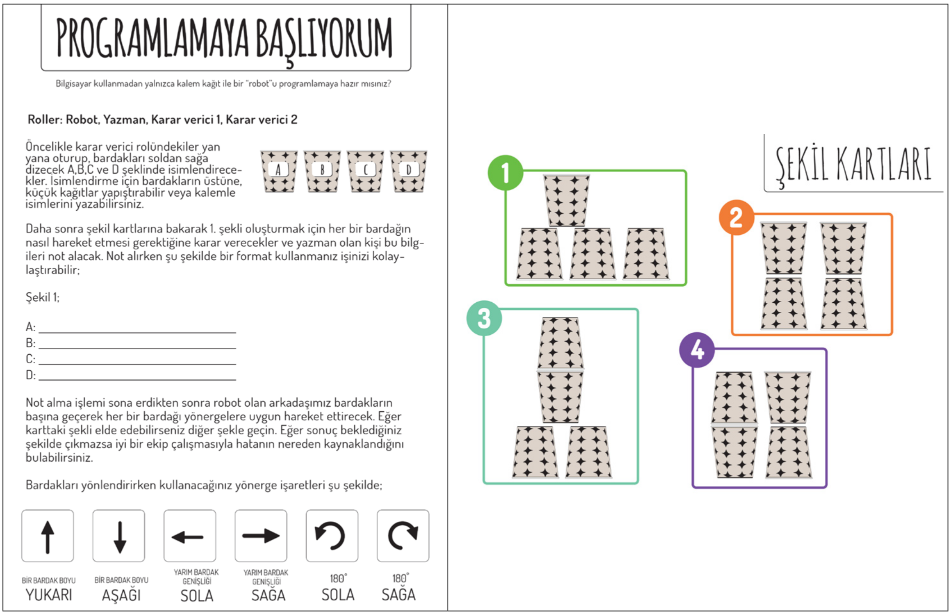
□ Sihirli Sayı A = ________ (sayıBir) + ________ (sayıİki)

□ Sihirli Sayı B = ________ (sayıÜç) - ________ (sayıBir)

□ Sihirli Sayı C = ________ (sayıİki) + ________ (sayıÜç) x ________ (sayıBir)

Fig. 9. Tangram and variables in envelope activities student worksheets.

In the tenth week, the students in the classroom were divided into groups, with four cups given to each group; the groups were asked to create necessary algorithm in order to create the indicated shape, and share them with the class. (Fig. 10. *Source:* `http://kesfetprojesi.org/files/uploads/5-sinif-bilisim-teknolojileri-der-`



PROGRAMLAMAYA BAŞLIYORUM

Bilgisayar kullanmadan yalnızca kalem kağıt ile bir "robot"u programlamaya hazır mısınız?

Roller: Robot, Yazman, Karar verici 1, Karar verici 2

Öncelikle karar verici rolündekiler yan yana oturup, bardakları soldan sağa dizecek A,B,C ve D şeklinde isimlendirecekler. İsimlendirme için bardakların üstüne, küçük kağıtlar yapıştırabilir veya kalemle isimlerini yazabilirsiniz.

Daha sonra şekil kartlarına bakarak 1. şekli oluşturmak için her bir bardağın nasıl hareket etmesi gerektiğine karar verecekler ve yazman olan kişi bu bilgileri not alacak. Not alırken şu şekilde bir format kullanmanız işinizi kolaylaştırabilir;

Şekil 1;

A: ____________________
B: ____________________
C: ____________________
D: ____________________

Not alma işlemi sona erdikten sonra robot olan arkadaşımız bardakların başına geçerek her bir bardağı yönergelere uygun hareket ettirecek. Eğer karttaki şekli elde edebilirseniz diğer şekle geçin. Eğer sonuç beklediğiniz şekilde çıkmazsa iyi bir ekip çalışmasıyla hatanın nereden kaynaklandığını bulabilirsiniz.

Bardakları yönlendirirken kullanacağınız yönerge işaretleri şu şekilde;

↑	↓	←	→	↺	↻
BİR BARDAK BOYU YUKARI	BİR BARDAK BOYU AŞAĞI	YARIM BARDAK GENİŞLİĞİ SOLA	YARIM BARDAK GENİŞLİĞİ SAĞA	180° SOLA	180° SAĞA

ŞEKİL KARTLARI

Fig. 10. Shape cards activity student worksheet.

si-ogrenci-materyalleri.pdf; http://kesfetprojesi.org/files/uploads/kodlamayi-kesfet-ogretmen-rehberi.pdf)

During the classes, the teacher giving the coding lesson kept daily notes of the experiences, students' feelings and ideas, and their approach towards the coding education. The Computational Thinking Skill Scale was re-applied as the post-test with the aim to measure the change in computational thinking skills of students at the end of the process.

2.5. *Analysis of Data*

The paired t test was applied between pre-test and post-test measurements for the analysis of quantitative data. The data collected were analyzed via SPSS 22 software. The independent variable of the research was formed by the process of unplugged coding education, and the dependent variable by the computational thinking skill. Ks value was examined to test whether the scores received from the scales displayed a normal distribution; the data were seen to have a normal distribution (value p > .05).

The qualitative findings obtained from the observation form and daybooks were analyzed by content analysis method, and the results were reported. The observation is a method used to define a behaviour in a particular setting or institution in detail; the most important quality of observation is that it allows the researcher to access the data at first hand (Yıldırım and Şimşek, 2008). The unstructured field study, which is the preferred type of observation, is performed in natural environments where the behavior takes place, and in most cases, through the method named ‘participant observation’, involving participation of the researcher (Yıldırım and Şimşek, 2008). In the participant observation, the observer accompanies the observed audience and acts like a member of them (Karasar, 2007). Glesne (2012) noted that participant observation makes it possible to learn at first hand to what extent the participants’ words match their actions, to see their behavioral patterns and to experience the unexpected in addition to the expected. During observations, data can be digitized more easily by developing and using observation guides / checklists / rating scales (Best, 1970). This research employed participant observation, with observation forms used to ensure easier recording of the data obtained.

2.6. *Validity and Reliability*

a) Triangulation method was used to increase internal validity of the study. The findings were presented on basis of the data obtained from the comparison of pre-test – post- test results as well as from observations and daybook notes. The teacher’s confirmation was again received for the observations and daybook notes recorded; the parts that are misunderstood were corrected. The students’ confirmation was received for the qualitative findings; the students verified their own behavior and opinions and eliminated misunderstandings. As the study had a ten-week period, a long-term interaction was assured with the students; the data were examined in depth and collected.
b) In order to increase reliability (consistency) of the study, direct quotations were used in the presentation of qualitative findings. Observation findings of the researcher and the teacher were compared at the reporting stage in order to achieve a shared outcome.
c) Cronbach alpha value for the Turkish version of Computational Thinking Skill Levels Scale, which is used as the data collection tool, was found to be 0.809. Cronbach Alpha values for the sub-dimensions of the scale are between 0,640 and 0,867 (Korkmaz, Çakır and Özden, 2015).

3. Findings

3.1. *Findings on the Effect of Unplugged Coding Activities on the Computational Thinking Skills of Middle School Students*

This question of the research investigated the effect of unplugged coding activities on computational thinking skills of middle school students; in this frame, pre-test and post-test scores of the students were analyzed by means of paired group t test. The findings are presented in Table 2.

As it can be seen from Table 2, it was concluded that there was a significant difference between pre-test and post-test scores of the Computational Thinking Skills scale

Table 2

Results of t Test on Computational Thinking Skills of Students

Factors	Test	n	m	sd	df	t	p
Computational thinking skills	Pre-test	114	3.55	0.58	113	-3.67	.00
	Post-test	114	3.86	0.68			

Table 3

Results of t Test on Sub-dimensions of Computational Thinking Skills of Students

Factors	Test	n	m	sd	df	t	p
Creativity	Pre-test	114	3.83	0.78	113	-2.35	.02
	Post-test	114	4.07	0.85			
Algorithmic Thinking	Pre-test	114	3.30	0.84	113	-5.30	.00
	Post-test	114	3.91	0.81			
Collaboration	Pre-test	114	3.37	0.48	113	-6.96	.00
	Post-test	114	4.05	0.97			
Critical Thinking	Pre-test	114	3.25	0.90	113	-4.87	.00
	Post-test	114	3.85	0.94			
Problem Solving	Pre-test	114	3.44	0.92	113	-1.05	.29
	Post-test	114	3.57	1.05			

which aims to measure computational thinking skills of students, so the unplugged coding activities had a positive effect on the improvement of computational thinking skills.

In order to examine whether there is a significant difference between pre-test and post-test scores for the sub-dimensions of computational thinking skills, t test for each dimension was repeated, and the findings obtained are presented in Table 3.

As it is concluded from Table 3, an examination of the results of t test for the sub-dimensions of the computational thinking skills of students revealed that there was a significant difference ($\alpha < 0,05$) between pre-test and post-test scores of the students in terms of sub-dimensions such as creativity, algorithmic thinking, collaboration and critical thinking, and that only the α value regarding problem- solving skills is higher than .05. Thus, it was found out that unplugged coding activities had a positive effect on creativity, algorithmic thinking, collaboration and critical thinking skills, which are among computational thinking skills; and no statistically significant change was observed in the problem solving skill.

3.2. *Findings on the Experience of the Students and the Teacher within the Process*

Observation findings and daybook records entered at the end of each class were used to learn the experience of the students and of the teacher within the process regarding unplugged coding activities.

When the observation findings were analyzed, it was seen that the students usually made adequate use of the sources and text provided for lecturing. It was reported that the duration assigned for the activities remained inadequate for all other classes except for the sixth week in which the algorithm story was told. It was observed that the class was mainly performed as planned in other weeks than the fourth week in which data - variable – constant and operations were explained, and the flow of lesson was satisfactory. Moreover, it was reported that most of the class materials attracted the attention of students and they were also liked in terms of visual design. The students were observed to have achieved the designated learning outcomes in all weeks other than the second week in which the problem solving strategies were explained and the ninth week in which the concept of variable was explained.

Upon an analysis of the daybooks kept by the teacher at the end of each lesson in relation to the flow of lesson, the experience gained in the process and student reactions, it was seen that the findings can be classified in five main categories which are:

a) Student reactions,

b) Teaching methods and techniques used,

c) Popular features of coding activities,

d) Flaws in the flow of lesson,

e) Learning outcomes which students had difficulty in achieving.

The mentioned categories and their codes are summarized in Table 4.

Table 4

Daybook Findings on the Experience of the Students and the Teacher within the Process

Category	Code
Student Reactions	• Entertainment • High level of motivation and class attendance • Difficulty with concretizing certain concepts as well as subjects requiring mathematical knowledge
Teaching methods and techniques used	• Fish bone • Contest • Question - Answer • Case study • Presentation • Problem Solving • Survey
Popular features of coding activities	• Attractive quality • Different solution offers to daily life problems • Good use of examples and visual materials
Flaws in the flow of lesson	• Problems experienced with timing and teamwork due to overcrowded classrooms • Class materials with excessive and distracting details • Demotivating factors
Learning outcomes which students had difficulty in achieving	• Comprehending the relationship between computer science and mathematics • Problem-solving and analyzing a given problem

When student reactions were examined during unplugged coding activities, it was observed that the students were entertained, they had high level of motivation and class attendance, but they also had difficulty in concretizing certain concepts as well as in subjects requiring mathematical knowledge. Below are some of the daily statements reported on the entertainment and high level of motivation and class attendance of students:

> *The students had too much fun in wolf – sheep – cabbage and tower of hanoi activities; their levels of motivation and class attendance were quite high.* (First week)
>
> *The students had lots of fun and enthusiasm at the competition event we organized in the first lesson, they wanted the event to continue in the second lesson.* (Second week)
>
> *Student reactions were positive; they really enjoyed while trying to give directions using the map. On the other hand, especially the first group of students I chose for the listen – draw activity had considerable difficulty in making correct descriptions, but this boosted their motivation and enthusiasm.* (Third week)
>
> *The students especially liked the examples of cake and clinic, which attracted their attention. As for other examples of games, they liked football but had poor interest in other examples, most of them said that they did not know the rules of ping pong.* (Fourth week)

Below are some of the statements reported in the daybook regarding the students' having difficulty in concretizing certain concepts as well as in subjects requiring mathematical knowledge:

> *The students especially liked the example of logical operator. Many students did not have strong basic mathematical knowledge, they did not know about operator precedence; therefore I had to focus on operators for a while.* (Fifth week)
>
> *The students were especially quite interested in the presentation of the flow scheme. They found it a little bit difficult to comprehend the meanings of flow scheme symbols; I had to teach names of certain geometric figures (rhombus and parallelogram) as some of the students did not know about them. But they liked the class materials.* (Seventh week)
>
> *They had difficulty in comprehending how the concept of variable is used in programming. I tried to explain by means of examples, but they had difficulty in fully picturing it in their minds.* (Ninth week)

Upon an analysis of the teacher's statements in the daybook, it is seen that the teaching methods and techniques used include fish bone, contest, question – answer technique, case studies, presentations, problem solving method and surveys. Below are some of the daybook records regarding the implementation of this method and techniques and student reactions to them:

> *Some of the students reported that they already worked on the wolf – sheep – cabbage problem in the selective course titled Mind Games. Although I warned*

> *them not to share the answer until other members of the group found the answers, some of the students disclosed the answers.* (First week)
>
> *The students had lots of fun and enthusiasm at the competition event we organized in the first lesson, they wanted the event to continue in the second lesson. During the second lesson, we made a fish bone on which they wanted to directly write the solution offers, not the causes of the problem. We formed the lesson when I stated that they needed to write the causes on the fish bone first, and we would share the solution offers later.* (Second week)
>
> *After reading the story, the students were able to tell the meaning of the concept of Algorithm and exemplify it.* (Sixth week)
>
> *The students were especially quite interested in the presentation of the flow scheme.* (Seventh week)
>
> *In order to make the students better understand the concept of variable during the class, I told them that I was a variable and would take the value of the hand that I would touch. I touched students' hands, respectively, and said I was the color that they indicated. So I tried to explain that I continuously change and hold a value.* (Ninth week)

The most popular features of unplugged coding activities were reported as "attractive quality, different solution offers to daily life problems, good use of examples and visual materials". Below are some of the notes in the daybook regarding the said qualities:

> *What I liked most is that the activities attracted the attention of the students.* (First week)
>
> *It is beautiful that the activities are able to ensure the students to bring different solutions to the problems which they may encounter in daily life, consider all events from an analytical perspective and think about the causes of problems.* (Second week)
>
> *The materials and activities were good; I think that the class is capable of making students achieve the outcomes targeted.* (Third week)
>
> *The concepts of constant and variable were explained by using examples, which was interesting and nice.* (Fourth week)
>
> *Logical operator was a good example.* (Fifth week)
>
> *I liked that the concept of algorithm was nicely based on a tale. (Sixth week)*
>
> *The visual materials and examples were good.* (Seventh week)
>
> *Case studies and scenarios in the work sheet were well designed.* (Eighth week)
>
> *The poem writing activity for kids is a good idea. It was already attractive for the students, it helped me to explain the subject to them.* (Ninth week)

When the statements about the flaws in the flow of lesson were analyzed, they were found out to be mainly caused by three reasons. These include problems experienced with timing and teamwork due to overcrowded classrooms, class materials with excessive and distracting details, and demotivating factors. Below are some of the flaws caused by overcrowded classrooms:

> *During the lesson in which we analyzed the problem by using fish bone, we run out of time to discuss the solutions and to collect different solution offers from each group.* (Second week)
>
> *I couldn't complete all of the activities, I felt that the time was not enough. I couldn't finish the paper folding activity in any class. The time periods given were not enough for me.* (Third week)
>
> *It took a long while before the students understood the scenario and create the flow schemes for those scenarios, so we couldn't do the rabbit & carrot activity.* (Eighth week)
>
> *Tangram activity is suitable only for small classes. I find it hard to do this activity in crowded classes with 32 to 38 students.* (Ninth week)

The problems experienced with the failure in effective use of some class materials due to their excessive and distracting details; related statements are as follows:

> *The sheet containing the problem solving steps was consisting of eight pages; when I showed and read the sheet to the students, they had difficulty in understanding it, some of them even thought that the sheet was an exam paper. Later, I briefly noted the steps written in the sheet on the board, I told about the activities on basis of the mentioned steps. I think that the students can better comprehend the activities when I tell them that the problem solving methods in frame of the activities are realized by following those problem solving steps. Similarly, I summarized and finished the lesson on basis of the steps that I wrote on the board. I think that the steps can be presented in a shorter form.* (First week)
>
> *The story titled Tortop was too long with distracting details. It might be better if we present the topic to the students with a shorter story and focus on the parts where the flow scheme would be drawn.* (Seventh week)
>
> *The flow schemes were mixed up. This will cause the students to experience confusion as to the meanings of the shapes if they have not fully understood the subject. Therefore, it would be better to avoid presenting incorrect shapes so that the students can better comprehend the subject.* (Eighth week)

Another flaw in the flow of lesson was demotivating factors. Those factors caused the students to have difficulty in building the link between the subjects taught and the computer lesson, thinking that they could not learn anything at the end of the lesson. Some of the statements indicating this situation are as follows:

> *Some of the students had the wolf– sheep – cabbage activity before and they knew the answer. Another problem is that the students may be demotivated as they tend to think that they have not learned anything at the end of the lesson no matter how much fun they had or how much the problem solving steps were emphasized. Therefore, it would be suitable to prepare a work sheet summarizing the lesson they had on the same day.* (First week)

> *These activities are very nice and entertaining, but th students cannot fully understand where and how to use the subjects in computer coding and programming. Although I remind the students that our activities are aimed at improving their thinking skills, sometimes they might be distracted from the lesson and ask me how the activities relate to our lesson. I try to associate activities with the computer more often during the lesson. For example, during the listen & draw activity, I told each of the students to act like a computer and just to implement the command given instead of finishing their drawing according to their estimations. I gave that instruction because some of the students tried to complete their drawings or told their classmates about their drawings on basis of an estimated object.* (Third week)

The teacher reported the learning outcomes which students had difficulty in achieving under two main titles. One of them is about comprehending the relationship between computer science and mathematics, and the other is about problem-solving and analyzing a given problem. Below are some of the statements reported in the daybook in relation to the mentioned situation:

> *The activities performed this week (Tower of Hanoi activity) are not enough to ensure the students to achieve the learning outcome 'comprehending the relationship between computer science and mathematics'.* (First week)
>
> *It is so hard to expect that a single activity will be enough to enable the students to achieve the learning outcome 'Analyzing a given problem'*. (Ninth week)

Upon an analysis of the findings from the observation and daybooks, it was revealed that the students usually had high level of motivation in the unplugged coding activities despite the difficulties they experienced in some of the topics; that different teaching methods and techniques were used, and the classes were found interesting and functional although a number of problems were experienced such as overcrowded classrooms and the failure in achieving certain learning outcomes.

4. Conclusion and Discussion

This study examined the effect of unplugged coding activities on computational thinking skills of middle school students. For this purpose, the study employed nested-mixed design, which is a mixed research method; the data were supported by including the qualitative phase into an experimental study. For the quantitative part of the study, unplugged coding activities were carried out for 10 weeks on basis of the pre-test post-test experiment design, as a result of which it was revealed that the unplugged coding activities had a positive effect on the development of computational thinking skills of the students. An examination of the sub-factors revealed that there is statistically no significant change in the problem solving skill despite the positive impact observed on creativity, algorithmic thinking, collaboration and critical thinking skills. Following the analysis of observation and daily data, the findings obtained revealed that the students usually displayed high levels of motivation and class participation in unplugged coding activities, they had difficulty in concretizing certain concepts as well as subjects requiring mathematical knowledge; various teaching methods and techniques were used in classes; the students liked the activities especially due to their appealing nature and their relation to daily life; however, there were occasional problems with scheduling of activities and teamwork due to over-crowded class size; the students experienced problems in achieving outcomes such as perceiving the relationship between computer science and mathematics and analyzing the given problem, and could have difficulty in associating between computer science and mathematics, and in analyzing a given problem.

It is noted that computer science unplugged activities ensure successful acquisition of especially computational thinking skills (Curzon, McOwan, Plant and Meagher, 2014), and those activities are among the best ways of teaching basic concepts of computer science (Bower and Falkner, 2015). Similarly, it was observed in this study that the students were able to improve their computational thinking skills by using the unplugged coding activities. However, unlike the studies indicating that computer science unplugged activities assist students in using their problem solving skills (Cortina, 2015; Gülbahar and Kalelioğlu, 2018; Bell, Witten and Fellows, 2015), this study revealed that there was no statistically significant change in their problem solving skills. It is considered that the reason for this situation is the ten-week time period which covers the implementation process is inadequate to enable the students to acquire problem solving skills.

Lambert and Guiffre (2009) performed a study on the effects of computer science unplugged activities on the interest in computer science and mathematics, whereby they evaluated the effect of three activities on the interest of fourth grade students in computer science and mathematics, their trust in mathematics and cognitive skills they perceived. Lambert and Guiffre (2009) indicated that the students were observed to have raised self-confidence and higher cognitive skills perceived for mathematics and higher interest in computer science after the activities. However, only one of the activities implemented was the computer science unplugged activity. Furthermore, Taub, Armoni and Ben-Ari (2012) criticized the study in the aspect that the number of experimental subjects was not provided and the results were not given for each question asked. Whereas, in scope of the research, it was seen that students with poor mathematical knowledge had difficulty in unplugged coding activities related to mathematics, they were reluctant to do the activities and they did not fully understand the subject. The reason for this situation can be interpreted as the inadequacy of unplugged coding activities in improving skills of students who have difficulty with basic mathematical concepts and operations or challenges experienced in the improvement of these skills due to the failure in fully focusing on mathematical skills.

On the contrary to the studies suggesting that computer science unplugged enable students to learn the subjects in an entertaining manner and with high motivation, improving motivation, thinking skills and imagination (Nishida *et al.*, 2009) and raise their interest in the computer science (Groover, 2009; Mano, Allan and Cooley, 2010), there are also studies which evidence that the said activities do not lead to any change in the attitudes of students towards computer sciences (Feaster *et al.*, 2011). This study revealed that the students had high level of motivation and class attendance throughout the activities. It was also indicated that the activities were liked as they are interesting and associated with daily life. The demotivating factors, according to which the students had difficulty in establishing the relation between the lessons learned and the computer lesson and thought that they did not learn anything at the end of the class, are consistent with the findings obtained from the study performed by Taub, Armoni and Ben- Ari(2012). Taub, Armoni and Ben- Ari (2012) suggested that the activities should be improved, indicating that some of the computer science unplugged activities are not fundamentally based on computer science, which decrease the students' interest in studying computer science, and that the students mainly comprehend the function of computer science but think that computer is the core of the science.

The findings obtained from the research are consistent with those of the researches suggesting that unplugged coding activities have positive effects on computational thinking skills of students; and it is observed that the integration of such type of activities in classes have positive effects on student motivation. In order to enable teachers to implement such activities more easily, it is important that availability and accessibility of unplugged coding activities should be ensured by way of translation of activities published in different languages, and that such activities should also be integrated to other classes and subjects. It is suggested that unplugged coding activities should be diversified and the results of implementation should be analyzed from different perspectives through scientific studies with longer implementation periods for different grades. Furthermore, it is considered that the increase in the number of studies examining the effect of block- based programming tools, robotics kits and similar educational coding materials on computational thinking skills and the review of factors that influence computational thinking skills under different studies will be helpful.

References

ACARA (2012). The shape of the Australian curriculum: technologies. Retrieved 29 March, 2019, from `http://docs.acara.edu.au/resources/Shape_of_the_Australian_Curriculum_-_Bibliography_-_August_2012.pdf`

Barger, R. N. (2008). Computer Ethics. Cambridge University Press.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology,* 38(6), 20–23.

Bell, T. (2014). The Science in Computer Science: Unplugging computer science to find the science. *Ubiquity Symposium,* 2014 (March), 3.

Bell, T. C., Witten, I. H., & Fellows, M. (1998). *Computer Science Unplugged: Off-line activities and games for all ages*. Computer Science Unplugged.

Bell, T., Witten, I. H., Fellows, M., Adams, R., McKenzie, J., Powell, M., & Jarman, S. (2015). Cs unplugged. *Computer Science Without a Computer. Creative Commons*.

Best, J. W. (1970). *Research In Education*. (Second Ed). Prentice Hall.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A. ve Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education-Implications for policy and practice*. Joint Research Centre (Seville site). doi:10.2791/792158

Bower, M., & Falkner, K. (2015, January). Computational thinking, the notional machine, pre-service teachers, and research opportunities. In *Proceedings of the 17th Australasian Computing Education Conference (ACE 2015)* (Vol. 27, p. 30).

Cortina, T. J. (2015). Reaching a broader population of students through unplugged activities. Communications of the ACM, 58(3), 25–27.

Creswell, J. W. & Plano Clark, V. L. (2014). *Karma Yöntem Araştırmaları, Tasarımı ve Yürütülmesi.* (Y. Dede & S. B. Demir, Translated.) Ankara: Anı Publishing.

Creswell, J. W. & Plano Clark, V. L. (2017). *Designing and conducting mixed methods research*. Thousand Oaks, CA:Sage.

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014).Introducing teachers to computational thinking using unplugged storytelling. *In proceedings of the 9th workshop in primary and secondary computing education* (pp.89–92).

Erden, M. (1998). *Eğitimde Program Değerlendirme [Program Evaluation in Education]***.** Ankara: Anı Publishing.

Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. (2011). Teaching CS unplugged in the high school (with limited success). In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 248–252).

Glesne, C. (2012). *Nitel araştırmaya giriş*. (A. Ersoy & P. Yalçınoğlu, Translated). Ankara: Anı Publishing.

Groover, T. R. (2009). Using games to introduce middle school girls to computer science. Journal of Computing Sciences in Colleges, 24(6), 132–138.

Gülbahar, Y., & Kalelioğlu, F. (2018). Bilişim Teknolojileri Ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci [Information Technology and Computer Science: Curriculum Update Process.]. *Milli Eğitim Journal*, *47*(217), 5–23.

Kalelioğlu, F. (2017). Bilgisayarsız Bilgisayar Bilimi (B^3) Öğretimi [Computer Science without Computer Teaching.]. Gülbahar, Y. (Ed.), *Bilgi işlemsel düşünmeden programlamaya*. (183–206). Ankara: Pegem Akademi Yayıncılık

Kalelioğlu, F., Keskinkılıç, F. (2017). Bilgisayar bilimi eğitimi için öğretim yöntemleri [Teaching methods for computer science education]. Gülbahar, Y. (Ed.), *Bilgi işlemsel düşünmeden programlamaya.* (155–182). Ankara: Pegem Akademi Yayıncılık

Karasar, N. (2007). *Bilimsel Araştırma Yöntemi [Scientific Research Method]*. Ankara: Nobel Publishing.

Kert, S. B. (2017). Bilgisayar Bilimi Eğitimine Giriş [Introduction to Computer Science Education]. Gülbahar, Y. (Ed.), *Bilgi işlemsel düşünmeden programlamaya*. (1–22). Ankara: Pegem Akademi Yayıncılık

Keşfet Projesi [Discover Project- I Discover Coding]. URL:
`http://kesfetprojesi.org/kodlamayi-kesfediyorum, Available Date: 10.09.2019`

Korkmaz, Ö., Çakır, R., Özden, Y. (2015). Bilgisayarca düşünme beceri düzeyleri ölçeğinin (BDBD) ortaokul düzeyine uyarlanması [Adaptation of the 'Scale of computational thinking skill levels' to the level of secondary school]. *Gazi Eğitim Bilimleri Dergisi[Gazi Journal of Educational Sciences],* 1(2), 143–162.

Korkmaz, Ö., Çakır, R., Özden, Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior* 72, 558–569.

Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in colleges,* 24(3), 118–124.

Mano, C., Allan, V., & Cooley, D. (2010). Effective in-class activities for middle school outreach programs. In *Frontiers in Education Conference (FIE), 2010 IEEE*.

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS unplugged design pattern. *ACM SIGCSE Bulletin*, 41(1), 231–235.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. *Basic Books,* Inc.

Paul, A. M. (2015). Teaching computer science-without touching a computer. *The Education Digest*, 80(5), 23–26.

Resnick, M. (2012). Let's Teach Kids to Code. TED Talks.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using student performance to assess cs unplugged activities in a classroom environment. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education.*

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12(2), 8.

Wing, J.M. 2006, 'Computational thinking', Communications of the ACM, March, vol. 49, no. 3, p. 33.

Yıldırım, A. ve Şimşek, H. (2008). *Sosyal bilimlerde nitel araştırma yöntemleri [Qualitative research methods in the social sciences].* Ankara: Seçkin Publishing.

B. Tonbuloğlu graduated from Marmara University, Department of Computer Education and Instructional Technology. She completed her master's degree with a thesis on the usability of educational software at Marmara University. She completed her PhD thesis on Evaluation of Distance Education Programs at Yıldız Technical University in 2017. She has been working as an IT teacher since 2008 at the Ministry of National Education in Turkey. Her research interests include distance education, teaching software, usability, computer-aided education and coding activities for children.

İ. Tonbuloğlu graduated from Marmara University, Department of Computer Education and Instructional Technology. He completed his master's degree with a thesis on the usability of instructional software with eye tracking and video recording methods at Marmara University. He completed PhD thesis on a study about preservice teachers' acceptance of use of mobile technologies for teaching purposes at Sakarya University in 2017. He has been working as a research assistant since 2010 at the Yıldız Technical University in Turkey. His research interests include technology acceptance, mobile technology, social media, mobile learning, technology integration in education teaching software, learning styles, usability and coding activities for children.