

ANKARA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI
EĞİTİM TEKNOLOJİSİ PROGRAMI

**OKUL ÖNCESİ ÖĞRETİM PROGRAMINA ALGORİTMA VE
KODLAMA EĞİTİMİ ENTEGRASYONUNUN ÖĞRENCİLERİN
PROBLEM ÇÖZME BECERİSİNE ETKİSİ**

YÜKSEK LİSANS TEZİ

CANAN AKYOL ALTUN

Ankara, Ekim, 2018

ANKARA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI
EĞİTİM TEKNOLOJİSİ PROGRAMI

**OKUL ÖNCESİ ÖĞRETİM PROGRAMINA ALGORİTMA VE
KODLAMA EĞİTİMİ ENTEGRASYONUNUN ÖĞRENCİLERİN
PROBLEM ÇÖZME BECERİSİNE ETKİSİ**

YÜKSEK LİSANS TEZİ

CANAN AKYOL ALTUN

DANIŞMAN: DOÇ. DR. AYFER ALPER

Ankara, Ekim, 2018

Eđitim Bilimleri Enstitüsü M¼d¼rl¼đ¼'ne

Canan AKYOL ALTUN'un hazırladıđı "Okul Öncesi Öđretim Programına Algoritma ve Kodlama Eđitimi Entegrasyonunun Öđrencilerin Problem Çözme Becerisine Etkisi" bařlıklı bu çalıřma j¼rimiz tarafından Bilgisayar ve Öđretim Teknolojileri Eđitimi Ana Bilim Dalı/ Eđitim Teknolojisi Programı'nda Yüksek Lisans Tezi olarak kabul edilmiřtir.

İmza

Bařkan: Doç.Dr. Gülg¼n BANGİR ALPAN



¼ye: Doç. Dr. Özl¼m ÇAKIR



¼ye: Doç.Dr. Ayfer ALPER (Danıřman)



ONAY

Bu tez Ankara Üniversitesi Lisans¼st¼ Eđitim-Öđretim ve Sınav Yönetmeliđi'nin ilgili maddeleri uyarınca yukarıdaki j¼ri üyeleri tarafından 03/10/2018 tarihinde uygun gör¼lm¼ř ve Enstit¼ Yönetim Kurulu'nca .../.../20... tarihinde kabul edilmiřtir.

Prof. Dr. Yasemin KEPENEKÇİ
Eđitim Bilimleri Enstitüsü M¼d¼r¼

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.



İmza

Canan AKYOL ALTUN

ÖZET

OKUL ÖNCESİ ÖĞRETİM PROGRAMINA ALGORİTMA VE KODLAMA EĞİTİMİ ENTEGRASYONUNUN ÖĞRENCİLERİN PROBLEM ÇÖZME BECERİSİNE ETKİSİ

AKYOL ALTUN, Canan

Yüksek Lisans Tezi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı
Eğitim Teknolojisi Programı

Danışman: Doç. Dr. Ayfer ALPER

Ekim 2018, xii + 89 sayfa

Bu araştırma, okul öncesi eğitime devam eden beş yaş grubu öğrencilerine verilecek algoritma ve temel kodlama eğitiminin çocukların problem çözme becerilerinin etkisinin saptanması amaçlanmış sıralayıcı karma desen modelinde bir araştırmadır. Araştırmanın çalışma grubunu 2016-2017 eğitim öğretim yılında Ankara ilindeki bir özel anaokuluna devam eden beş yaş grubu 30 öğrenci oluşturmuştur.

Bu araştırmanın nicel verileri, "Problem Çözme Becerisi Ölçeği" ile; nitel verileri ise gözlem formu ve odak grup görüşmeleri esnasında yapılan ses kayıtları yoluyla toplanmıştır. Araştırmada 4 hafta süren algoritma eğitimi ve sonra 4 hafta süren temel kodlama eğitimi adı altında OSMO Kodung eğitimi verilmiştir. Tüm öğrencilere "Problem Çözme Becerisi Ölçeği" uygulama öncesinde ve uygulama sonrasında ön test ve son test olarak uygulanmıştır. Uygulamalar esnasında öğrencilere ait gözlemler gözlem formu yardımıyla toplanmış, uygulama sonrasında ise odak grup görüşmeleri ile veriler toplanmıştır.

Araştırmada elde edilen verilerin çözümlenmesinde betimsel istatistikler ile birlikte Shapiro-Wilk testi ve t-testi kullanılmıştır. Elde edilen bulgulara göre problem çözme ölçeğinden alınan ön test ve son test puanlarına bakıldığında algoritma eğitiminin problem çözme becerisi ölçeğinden alınan puanlarında anlamlı bir farklılık yarattığı görülmüştür. Problem çözme becerisi ölçeğinden alınan son test puanları karşılaştırıldığında ise algoritma eğitimi alanların son test puanlarının algoritma eğitimi almayanların son test puanlarından yüksek olduğu görülmüştür.

Problem Çözme Becerisi Ölçeđi puanları yüksek olan öğrencilerin daha ayrıntılı, daha kapsamlı görüşler bildirdikleri, gerçek hayatlarından örneklerle durumları destekledikleri görölmüştür.

Anahtar Kelimeler: Okul Öncesi Kodlama Eğitimi, Algoritma Eğitimi, Problem Çözme Becerisi, Bilişimsel Düşünme, Kodlama, Programlama



ABSTRACT**ALGORITHM AND CODING EDUCATION IN PRE-SCHOOL TEACHING
PROGRAM INTEGRATION THE EFFECTIVENESS OF PROBLEM SOLVING
SKILLS IN STUDENTS**

AKYOL ALTUN, Canan

Master Thesis, Department of Computer and Instructional Technologies

Educational Technology Program

Supervisor: Doç. Dr. Ayfer ALPER

October 2018, xii + 89 pages

This research, which is a research in the sequential mixed pattern model, aimed at determining the effect of on children 's problem solving skills the algorithm and basic coding training to be given to the students of five age groups who are going to pre - school education. The study group of the study consisted of 30 students aged five years in a private kindergarten in the province of Ankara in the academic year 2016-2017.

The quantitative data of this study were analyzed with "Problem Solving Skills Scale"; qualitative data were collected through voice recordings during observation form and focus group interviews. In the study, 4 weeks of algorithm training and 4 weeks of OSMO Koding training was given under the name of basic coding training. The "Problem Solving Skills Scale" was applied to all students as a pre-test and post-test before and after the application. Observations belonging to the students during the applications were collected with the observation form, and after the application's data were collected with focus group interviews.

Shaphiro-Wilk test and t-test were used together with descriptive statistics in the analysis obtained in the study. According to the findings, when the pre-test and post-test scores taken from the problem-solving scale were examined, it was seen that the training of the algorithm produced a significant difference in the scores obtained from the problem solving skill scale. When the post test scores obtained from the problem solving skill scale are compared, it is seen that the post test scores of the algorithm training areas are higher than the final test scores of those who do not have the algorithm training.

It was seen that the students, who had high scores of Problem Solving Skills Scale, had more detailed, more comprehensive opinions and they supported the situations with examples from their real life.

Keywords: Pre-School Coding Education, Algorithm Education, Problem Solving Skills, Computational Thinking, Coding, Programming.



ÖNSÖZ

Okul öncesi öğretim programına algoritma ve kodlama eğitimi entegrasyonunun öğrencilerin problem çözme becerisine etkisinin incelenmesi amacıyla yapılan bu çalışma 5 bölümden oluşmaktadır. Birinci bölümde, araştırmanın problemi, amacı, önemi, sayıtları, sınırlılıkları ve kısaltmalar ile ilgili bilgiler verilmiştir. Araştırmanın yöntemine ilişkin bilgiler ikinci bölümde yer verilmiştir. Üçüncü bölümde araştırmaya konu edilen, bilişimsel düşünme ve problem çözme, algoritma eğitimi, eğitimde programlama, okul öncesinde programlama ile ilgili kuramsal çerçeve, dördüncü bölümde araştırmanın amaçları ve alt amaçları doğrultusunda elde edilen bulgulara ve bu bulgulara ilişkin yorumlara yer verilmiştir. Beşinci bölümde ise araştırmanın sonuç ve önerileri sunulmuştur.

Yüksek lisans eğitimine başladığım ilk günden bugüne bana her zaman desteğini esirgemeyen ve yol gösteren danışmanım Doç.Dr.Ayfer ALPER'e, tezin uygulanmasında her türlü desteği veren Beştepe Koleji kurucusu Gökhan ÜNLÜER'e, Beştepe Koleji Anaokulu Müdürü Begüm KOPAN'a, yardım ve desteğini hiç esirgemeyen Beştepe Koleji kadrosuna, teşekkürlerimi sunuyorum. Araştırma sürecince her an destek olan yüksek lisans eğitimi ile tanıştığım dostlarıma, en çıkmaza girdiğim anlarda destekleyen tüm arkadaşlarıma, hayatım boyunca beni hep destekleyen, hep yanımda olan, beni yetiştiren anneme ve babama teşekkür ediyorum.

Araştırmamda her an destek olan tüm hocalarıma, kaynaklarını kullandığım tüm araştırmacılara ve uygulama süresinde bana yardım eden güzel çocuklarıma ayrıca teşekkür ediyorum.

Canan AKYOL ALTUN

İÇİNDEKİLER

ONAY.....	ii
İMZA SAYFASI	iii
ÖZET	iv
ABSTRACT	vi
ÖNSÖZ.....	viii
İÇİNDEKİLER.....	ix
TABLolar DİZİNİ.....	xi
ŞEKİLLER DİZİNİ	xii
BÖLÜM I	1
GİRİŞ.....	1
Problem.....	1
Amaç.....	11
Önem	12
Sayıtlar	13
Sınırlılıklar.....	13
Kısaltmalar	13
BÖLÜM II	14
YÖNTEM.....	14
Araştırma Modeli.....	14
Çalışma Grubu.....	15
Veriler ve Toplanması	15
Veri Toplama Araçları.....	16
Uygulama	19
Verilerin Analizi.....	31

BÖLÜM III.....	33
KURAMSAL ÇERÇEVE VE İLGİLİ ARAŞTIRMALAR.....	33
Bilişimsel Düşünme ve Problem Çözme.....	33
Algoritma Eğitimi.....	38
Eğitimde Programlama.....	40
Okul Öncesinde Programlama.....	43
Okul Öncesinde Kullanılan Kodlama Siteleri ve Uygulamaları.....	46
BÖLÜM IV	57
BULGULAR VE YORUMLAR	57
Problem Çözme Becerisi Ölçeği Analizinden Elde Edilen Bulgular.....	57
Gözlem Formu Analizinden Elde Edilen Bulgular.....	60
Odak Grup Görüşmesi Analizinden Elde Edilen Bulgular.....	65
BÖLÜM V.....	71
SONUÇLAR VE ÖNERİLER	71
Sonuçlar.....	71
Öneriler.....	72
KAYNAKLAR.....	74
EKLER	84
EK 1: Problem Çözme Becerisi Ölçeği (PÇBÖ) Maddeler.....	84
EK 2: Problem Çözme Becerisi Ölçeği (PÇBÖ) Örnek Sorular ve Cevaplar.....	85
EK 3: Problem Çözme Becerisi Ölçeği (PÇBÖ) Bir Örnek Çizim.....	86
EK 4: Problem Çözme Becerisi Ölçeği Kullanım İzni.....	87
ÖZGEÇMİŞ.....	88

TABLOLAR DİZİNİ

Tablo 1. Öğretim Programına Programlamayı Entegre Etmek İçin Gerekçe.....	7
Tablo 2. Çalışma Grubu Durumu	15
Tablo 3. Çalışma Grubu Dağılımı	15
Tablo 4. Planlanan Odak Grup Görüşmesi Tablosu	17
Tablo 5. Gerçekleştirilen Odak Grup Grup Görüşmesi Tablosu	18
Tablo 6. Program Blokları	22
Tablo 7. Dağılımların Test Edilmesi	31
Tablo 8. Baz(2018), <i>Çocuklar için kodlama yazılımlarının karşılaştırmalı incelemesi</i>	47
Tablo 9. Deney ve kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test puanlarına ilişkin t testi sonuçları	57
Tablo 10. Kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test ve son test puanlarının t-testi sonuçları	57
Tablo 11. Deney grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test ve son test puanlarının t-testi sonuçları	58
Tablo 12. Deney ve kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları son test puanlarına ilişkin t testi sonuçları.....	59
Tablo 13. Gözlem Formu (Kontrol Grubu)	60
Tablo 14. Gözlem Formu (Deney Grubu)	61

ŞEKİLLER DİZİNİ

Şekil 1. Zemin Çalışma Ortamı	21
Şekil 2. OSMO Koding Seti	21
Şekil 3. Nilüfer Çiçeği Örneği	23
Şekil 4. OSMO Başlangıç Ekranı	23
Şekil 5. Örnek Uygulama Adımları	24
Şekil 6. Örnek Uygulama Adımları - 2.....	24
Şekil 7. Örnek Uygulama Adımları - 3.....	25
Şekil 8. Örnek Uygulama Adımları - 4.....	25
Şekil 9. İlk Hafta Örnekler	26
Şekil 10. İkinci Hafta Örnekleri	27
Şekil 11. İkinci Hafta Örnekleri-2	28
Şekil 12. Üçüncü Hafta Örnekleri	29
Şekil 13. Dördüncü Hafta Örnekleri.....	30
Şekil 14. ScratchJr	48
Şekil 15. ScratchJr Ekran Görüntüleri	48
Şekil 16. Kurslar ve etkinlikler listeleri.....	49
Şekil 17. Kurs 1 Dersleri	49
Şekil 18. Kodable Kullanıcı Seçimi	50
Şekil 19. Kodable Öğrenci Ekleme	51
Şekil 20. Kodable Öğretmen Kontrol Ekranı	51
Şekil 21. Kodable Komut Sıralama ve Çalıştırma Ekranı	51
Şekil 22. Kids Koding with The Foos Ekran Görüntüleri	52
Şekil 23. Box Island Ekran Görüntüleri	53
Şekil 24.Cargo Bot Tasarım Aşaması	53
Şekil 25. Daisy Dinosaur Arayüzü	54
Şekil 26. Daisy Dinosaur Sürükle-Bırak	54
Şekil 27. Blockly Games Ekranı	55
Şekil 28. Blockly Problem Çözümü	55
Şekil 29. Move the Turtle. Learn to Code	56

BÖLÜM I

GİRİŞ

Bu bölümde araştırmanın problemi, amacı, önemi, sınırlılıkları ve kullanılan temel kavramların açıklamalarına yer verilmiştir.

Problem

Kritik düşünme, problem çözme, yaratıcılık, üst düzey düşünme, iletişim, işbirliği, bilgi ve teknoloji okuryazarlığı, esneklik ve adapte olabilme, küresel yetkinlikler ve finansal okur-yazarlık temel 21. yüzyıl becerileri olarak tanımlanmaktadır. Bu açıdan bakıldığında 21. yüzyıl öğrencilerinden beklenen temel beceriler; eleştirel düşünebilme, problem çözebilme, yaratıcı olabilme, analiz ve sentez yapabilme, işbirlikçi ve takımla birlikte çalışabilme, doğru iletişim kurabilme, doğru ve güncel bilgiye kolay erişebilme şeklinde sıralanabilmektedir. Öğrencilerin bu becerilere sahip olmasını sağlayacak yöntemlerden birinin bilgisayar programlamanın ve bilgisayar biliminin öğretilmesi olduğunu gösteren araştırmalar ve uygulamalar vardır (Akpınar ve Altun, 2014; Çakıroğlu, Sarı ve Akkan, 2011; The Partnership for 21st Century Skill,2009; Monroy-Hernandez, ve Resnick, 2008; Karabak ve Güneş, 2013; Shin, Park ve Bae, 2013). Kodlamanın, günümüzün okur-yazarlığı olduğu ve problem çözme, takım çalışması ve analitik düşünme gibi 21. yüzyıl becerilerinin kazandırılmasına yardımcı olduğu savunulmaktadır (Coding - the 21st century skill, 2014).

Programlama, bilgisayarın donanıma nasıl davranacağını anlatan, bilgisayara yön veren komutlar ve işlemler bütünüdür (Gülbahar, Kalelioğlu ve Karataş,2017). Programlama, problemleri bir algoritma ile çözmeyi, analiz etmeyi, anlamayı, algoritmanın gereklerini doğrulamayı ve algoritmanın hedef programlama dilinde uygulanması, test edilmesi, hata ayıklanması, kaynak kodun saklanması, yapı sisteminin uygulanması gibi faaliyetleri içerir (Michael ve Omoloye, 2014). Bir programlama sisteminin iki bileşeni vardır (Gülbahar, Kalelioğlu ve Karataş,2017):

1. Bilgisayara kurulmuş olan bileşen - programlama ortamı

2. Programcı tarafından oluşturulan algoritma ve program kodları

Kullandığımız programlama ortamı ile programcı tarafından kullanılacak kelime ve komutlar oluşturulabilir, program akışı ve mevcut durum kontrol edilebilir, adım adım işlemler takip edilebilir, oluşturduğumuz işlemler genelleştirerek soyutlaştırılabilir.

Programlama kavramı ile birlikte sıkça rastladığımız programlama eğitimi ise, bilgisayar teknolojisinin gelişim sürecinde önemli bir yapıtaşı ve yönetim aracı olan yazılım çalışmalarının var olmasını sağlayan önemli bir eğitim alanıdır (Kert ve Uğraş,2009). Burada karşımıza çıkan bir başka kavram olan algoritma eğitimi ise, programlama eğitiminin temelini oluşturmaktadır. Algoritma, bir problemi çözmek için izlenmesi gereken adımlardan oluşan yapıdır. Algoritma birbirini takip eden sıralı adımlardan oluşur. Arabacıoğlu (2006) algoritmayı tüm bilgisayar programlarının tasarımı aşamasında yararlanılan ve işin tamamlanması için gerekli işlemlerin kendi dilimizde tarif edildiği bir belge olarak tanımlamıştır. Gündelik hayatta yapmış olduğumuz işleri aslında basit-karmaşık algoritmalar ile yapmaktayız. Programlama öğretimi öncesinde verilen programlamanın temeli olan algoritma öğretiminde konuşma dili kullanılmaktadır. Algoritma tasarımı sırasında kullanılan komutlar kısa, öz ve anlaşılır olmalıdır (Arabacıoğlu, Bülbül ve Filiz, 2007). Futschek (2006)'e göre çözülmesi gereken problemler çok basit olmamalı, ama problem durumları kolayca anlaşılabilir olmalıdır.

Pedagojik bir bakış açısıyla, algoritmik problem çözme becerilerinin yüksek düzey düşüncenin geliştirilmesinde yararlı olduğu düşünülmektedir (diSessa, 2000; Papert, 1991). Bilişsel gelişim Piaget'e (1973) göre 5-7 yaş arasında daha mantıklı ve daha sezgisel bir düşünce biçimine (işlem öncesi öğrenme aşamasından somut işlemler aşamasına) geçmektedir. Flannery ve Bers (2013), bu geçiş evresinin çocukların program yapma ve kod yazma becerilerinin gelişiminde kritik bir zaman olduğunu keşfetti. Algoritmik düşünme, günümüzdeki mezunlar için önemli bir yetenektir ve algoritmik kavramlar, öğrencilerin okul eğitiminin başlarında kademeli olarak verilmelidir (Mittermeier, 2013). Algoritma eğitiminin bilgisayar bilimleri ile olan yakın ilişkisinin kısıtlayıcı bir faktör olarak görülmemesi, disiplinler arası platformda değerlendirilerek diğer derslerde de kullanılabilmesi, sadece akademik bir amaç olarak düşünülmemesi gerektiği, hayatın temelinde var olan problem çözme becerisini

geliştirdiği açıkça görülmüştür (Çırpılı, 2016). Şabanoviç ve Yannier (2003) robotik eğitimlerin öğrencilerin problemlere yapıcı çözümler bularak ortaya fiziksel bir ürün yaratılmasını sağlarken, teknoloji konusunda doğru bilgilenmeyi, araştırma ve bulduklarını bir araya getirmeyi, takım çalışmasını öğretebileceğini belirtmişlerdir. .

Alan yazında erken yaşlarda verilen algoritma eğitimi konusunda yapılan araştırmalara rastlanmaktadır. İlkokul öğrencileri ile yapmış oldukları algoritmik düşünmenin temel kavramlarını öğrenmesini sağlayan bir öğrenme senaryosu sundukları çalışmalarında öğrencilere bazı materyallerle basit algoritmayı anlatmışlar, SCRATCH ortamına sorunsuz bir geçişi sağlayarak programlamanın ve sanal çevrenin ilk adımlarını öğrenmelerine nasıl yardımcı olabileceğini gösterdiklerini belirtmişlerdir (Futschek ve Moschitz, 2011). Futschek ve Moschitz (2011), bu çalışmalarında öğrencilerin grup halinde çalışmalarını önerdikleri *Tim the Train ile* yeni başlayanlara algoritmik düşünmeyi eğlenceli bir şekilde öğretmeye yardımcı olunacağını belirtmektedirler. Karadeniz, Samur ve Özden (2014), 5 yaş öğrencilerine yönelik yapmış oldukları çalışmada öğrencilere vermiş oldukları çalışma yapraklarının ortalama puanlarına bakıldığında öğrencilerin algoritma ve temel programlama kavramlarını başarılı bir şekilde öğrendikleri sonucuna varmışlardır. Mittermeier (2013), okuma yazmaya başvurmadan basit algoritmaları keşfetmesi için anaokulu gruplarıyla yaptığı çalışmada, 4-6 yaş aralığındaki çocukların algoritma geliştirme kapasitesine sahip olduklarını, geliştirdikleri algoritmayı sözlü olarak ifade edebildiklerini ve farklı çözümler ile algoritmalarını verimli bir şekilde iyileştirme veya revize etmek için ipuçlarını uygulayabildiklerini belirtmiştir. Gibson (2012), 5 - 17 yaşlarındaki okul çocuklarına grafik teorisi ve algoritmalar öğretme konusundaki deneyimlerini bildirdiği çalışmada algoritmaların oyunlara veya bulmacalara olası çözümler sunduğu probleme dayalı öğrenmenin önemine değinmiştir. Bu çalışmada 5-11 yaş arasındaki çocukların algoritma ve hesaplamayı öğrenme konusunda oldukça yüksek potansiyele sahip olduklarını belirtmiştir.

Günümüzde, dünyanın her yerinde öğrencilerin programlama mantığını öğrenmesi gerektiği fikri öne çıkmakta bu görüş sıklıkla her mecrada dile getirilmektedir. DiSessa ve Abelson(1986), programlama ortamlarını, problem çözme süreci boyunca öğrencilerin düşüncelerini net ve özlü bir şekilde analiz etmelerini, organize etmelerini, ifade etmelerini ve değerlendirmelerini destekleyen güçlü yeniden yapılandırılabilir bilgi işleme ortamı olarak görmektedir (akt. Fessakis, Gouli ve

Mavroudi,2013). Programlamanın küçük yaştaki öğrencilerin bilişsel alandaki gelişimlerine katkısı yıllardır üzerinde çalışılan bir konudur (Papert, 1980; Billings, 1983; Clements ve Gullo, 1984). Öğrenciler, mantıksal işlemlerle programlar yazarak ve programlama dilleri yoluyla mantıksal düşünmeyi öğrenerek bilişimsel düşünme yeteneklerini geliştirdikleri belirtilmektedir (Chang, 2014). Programlama eğitimine yeni başlayanların bilişim teknolojilerine daha uyumlu olmalarını sağlamak ve programlamanın anlaşılması zor yapısını ve öğrenme güçlüğü en aza indirmek amacıyla çocuk programlama dilleri geliştirilmiştir (Schwartz, Stagner ve Morrison, 2006). Çocukluk döneminden itibaren verilecek programlama eğitimi düşünme becerilerini olumlu yönde etkileyebileceği gibi, süreç ve ürün boyutlarında ayrı ayrı ele alındığında öğrenme sürecinde öz-düzenlemeye, işbirlikçi öğrenmeye ve keşfetmeye dayalı yapılandırmacı yaklaşıma paralel bir öğrenme altyapısı oluşturulmasına katkıda bulunabileceği belirtilmektedir (Kert ve Uğraş, 2009).

Son 5 yıldaki araştırma ve politika değişiklikleri, çocukluk döneminde STEM (bilim, teknoloji, mühendislik ve matematik) eğitimini yeni keşfedilen bir odak haline getirdi (Sesame Workshop 2009; White House 2011). Robotik ve programlama, öğretmenlere erken 'çocukluk dönemi STEM eğitiminde en çok ihmal edilen teknoloji' 'T' 've' 'E' 'mühendislik konularına yeni ve heyecan verici bir yol sunmaktadır. Sullivan ve Bers (2016) tarafından okul öncesinden 2.sınıfa kadar olan öğrencilerle yapılan çalışma, anaokulundan itibaren bir robotun programlanması ile ilgili temel kavramlara hakim olabildiğini ve 7 yaşındaki çocukların, koşullu ifadeler kullanarak bir robot programlama gibi kompleks kavramlar edindiklerini ortaya koymaktadır. Creative Little Scientist projesinin bulgularına göre STEM konularındaki yaratıcılık kavramı, "birey veya toplum olarak fikir ve stratejiler oluşturmak, bunlar arasında eleştirel olarak mantık yürütmek ve mevcut kanıtlarla tutarlı makul açıklamalar ve stratejiler üretmek" olarak tanımlanabilmektedir (Agogi, Rossis and Stylianidou, 2014, p. 8).

Programlamanın eğitimsel anlamda kullanımını 1960'lı yıllarda, Seymour Papert, çocukların oyunları oluşturmak, müzik bestelemek veya resim çizimlerini yapmak için bilgisayar kullanmasına izin vermeyi amaçlayan Logo programlama dilini geliştirilmesiyle başlamıştır (Calao, Moreno-León, Correa, ve Robles, 2015). Öğrencilerin kendi oyunlarını üretme konusunda çok motive olduklarını bu motivasyonlarını kullanarak programlamayı bir amaç için araç olarak tanıtır ve kendi fikirlerini ifade ederek, gelişen programlama becerilerini inşa etmenin bir yolunu ortaya

koyma amacıyla yapılan arařtırmada fikirlerini anlatımsal bir bakıř aısıyla biliřimsel bir bakıř aısına geirebildikleri grlmřtr (Howland ve Good, 2015). Dalton (1986), ğretmen ynetimli problem zme ğretiminde LOGO programının kullanımının ortaokul ğrencilerinin problem zme becerilerine, bařarılarına ve tutumlarına etkisini arařtırmıřtır. Bu alıřmada hem problem zme stratejilerini uygulayan grubun bařarılı olduėu, hem de LOGO programlama dili kullanan ğrencilerin problem zmeye ynelik tutumlarında artıř olduėu gzlenmiřtir (etin, 2014).

LOGO programının eėitimde kullanımına ynelik birok arařtırma bulunmaktadır. Clements ve Sarama'da (2003) belirtildiėi gibi 1984'te 3 ve 4 yařındaki ocuklarla yapılan bir alıřma sırasında Logo ortamında yaptıkları izimler esnasında kaėıt zerine yaptıkları izimlere gre daha fazla etkileřimde buldukları grlmřtr. Ayrıca, anaokulları ve kk ocukları karřılařtıran bir arařtırmada, Logo'nun anaokulları ğretmenleri arasındaki iřbirliėine dayalı davranıřı kolaylařtırdıėı ve sosyal ve dil becerilerini geliřtirdiėi, aynı zamanda daha gen yařtaki ocukların alıřmalarına odaklandıkları da belirtilmiřtir (Strand, 1986). Fessakis, Gouli ve Mavroudi (2013) yaptıėı arařtırmada, anaokulu ocuklarının cretsiz Logo benzeri programlama ortamlarındaki sorunların zmne ynelik bir rnek olay incelemesi tasarlanmıř, uygulanmıř ve sunulmuřtur. Arařtırma verileri, ocukların aktif olarak katıldıėını ve ortaya ıkan sorunlardan memnun olduklarını ortaya koymuřtur. nerilen etkinlikler (bire bir yazıřma, sayma, sayı karřılařtırması, ynlendirme becerilerinin tanıtılması ve glendirilmesi kavramları, aı dnř kavramları) ėrenme potansiyeli, problem zme stratejilerinin geliřtirilmesi, iletiřim ve iřbirliėi becerilerini, gerek bir iletiřimsel erevede matematiksel becerilerini iermektedir.

Son zamanlarda, kk ocuklar iin tasarlanmıř "eėitsel diller" adı verilebilecek geleneksel komut istemi ve metin tabanlı kodlamaya ihtiya duymayan 5 yař altındaki ocuklar iin eriřilebilir ve kullanılabilir olan grafik temelli kodlama ve programlama deneyimleri sunan bir dizi programlama dili ve sistemi kullanıma sunulmuřtur (Lewis ve diėerleri, 2014; Code.org, 2013; Kafai ve Burke, 2013). Bu programlardan bazıları Scratch, ScratchJr, Code Spells, Tynker ve Lego Mindstorms gibi dillerdir. Bu grsel programlama yapıları, kk yařtaki ğrencilerin geleneksel programlama dillerinin karmařık kod yapılarını ėrenmelerine gerek kalmadan, uygulamalar yazabilmeleri saėlamaktadır (Resnick ve diėerleri, 2009).

Çocuklar için geliştirilmiş, çocukları gelişimsel açıdan daha uygun hale getiren özel özelliklere sahip yazılım programlama ortamlarına ek olarak çocukların fiziksel nesnelere hareket ettirmelerine izin veren sistemler de önerilmektedir. Bu sistemlerin bazıları bir bilgisayarın kullanılmasını gerektiren sistemlerken bazıları da TORTIS sistemi (Perlman, 1976) ve Valiant Roamer Robot (Amethyst Consultancy Ltd, 2003) gibi bağımsız elektromekanik cihazlardır. Bilgisayar programcılığının küçük çocuklarla kullanımı üzerine daha ilginç erken çabalardan birisi de Perlman'a (1974, 1976) aittir. Perlman, Logo kaplumbağasından esinlenen bir robotik cihazın programlanması için kullanılan somut nesnelere oluşan TORTIS adlı bir programlama sistemi geliştirmiştir. Perlman'ın temel tasarım hedefi, metin yazarak program oluşturma konusundaki engeli aşmak ve programlama öğrenmenin faydalarını 3-4 yaşlarındaki çocuklar için erişilebilir hale getirmektir. TORTIS, Perlman'ın hedeflerini başarıyla karşıladı ve buna ek olarak, TEACH (Solomon & Papert, 1975) gibi programlama dillerinin ve Valiant Roamer Kaplumbağa Robotu gibi robotik sistemlerinin geliştirilmesinde diğer araştırmacıları etkilemiştir (Fessakis, Gouli ve Mavroudi, 2013).

Ayrıca, tasarımcılar iPad gibi tablet cihazları aracılığıyla sunulan dokunmatik ekran etkileşiminden yararlandıkça yeni kaynaklar da (örn. Hopscotch, Kodable, Lightbot ve Tynker) artmaktadır. Bir ekran yerine fiziksel arayüzleri kullanan somut tasarımlar küçük yaş gruplarında daha yaygın hale gelmektedir (Manches ve O'Malley, 2012). Örneğin Cubetto, kesilmiş şekiller içeren tahtalara yerleştirilen blokları kullanan, Dash and Dot, 4 yaşından büyük çocukların basit talimatları girme etkilerini keşfetmesini sağlayan oyuncak robotlardan oluşan, Robot Turtles, 3 yaşından büyük çocuklara programlamanın temellerini öğreten, kartlarla oynanan sayısal olmayan bir tahta oyunudur. ScratchJr, Scratch'e dayanan ve tabletler tarafından sağlanan etkileşimlerden faydalanarak 5 ila 7 yaş arasındaki çocuklar için yeniden tasarlanan bir grafiksel programlama dilidir (Manches ve Plowman, 2017).

Ülkemizde ve Dünya'da programlama eğitimi üzerinde dikkatle durulmakta ve toplumsal yaşam içerisine programlama becerisine sahip bireylerin kazandırılmasına, nitelikli programlama uzmanı sayısının her geçen gün artırılmasına önem verilmektedir (Kert ve Uğraş, 2009). Yapılan incelemelerde birçok ülkenin öğretim programlarında programlama derslerine yer verdiği ve bazı ülkelerde okul öncesinden itibaren bu eğitimlerin verilmeye başlandığı görülmüştür. Son yıllarda birçok kar amacı gütmeyen kuruluş tarafından yenilikçi ve ilgi çekici eğitim yaklaşımları ile kodlama eğitimi

verilmeye başlanmıştır (Code.org,2015). 2015 yılında European Schoolnet (Avrupa Okul Ağı) tarafından yapılan bir araştırmada bilgi ve iletişim teknolojilerini entegre etmek için temel mantığı araştırılmıştır. 16 ülke ulusal, bölgesel veya yerel düzeyde öğretim programlarına kodlamayı entegre etmektedir: Avusturya, Bulgaristan, Çek Cumhuriyeti, Danimarka, Estonya, Fransa, Macaristan, İrlanda, İsrail, Litvanya, Malta, İspanya, Polonya, Portekiz, Slovakya ve İngiltere. Finlandiya ve Belçika Flanders ise müfredatta entegre etmeyi planlıyor. Aşağıda yer alan Tablo 1'de, hem müfredatta kodlamayı tamamlamış hem de bunu yapmayı planlayan ülkeler tarafından benimsenen gerekçeler gösterilmektedir:

Tablo 1. Öğretim Programına Programlamayı Entegre Etmek İçin Gerekçe

	Mantıksal Düşünmeyi Destekleme	Problem Çözme Destekleme	BIT'i Öğrenci için Cazip Hale Getirme	Kodlama Becerilerini Destekleme	Bilgi ve İletişim Teknolojileri İstihdamını Destekleme	Diğer Kilit Yetenekleri Destekleme
Avusturya	•	•	•	•	•	•
<i>Belçika Flanders</i>			•		•	•
Bulgaristan	•	•	•	•		
Çek Cumhuriyeti	•	•	•	•	•	•
Danimarka	•	•				•
Estonya	•	•	•			•
<i>Finlandiya</i>	•	•		•		
Fransa			•		•	•
İrlanda	•	•	•	•		•
İsrail	•	•	•	•	•	•
Macaristan	•	•				
Litvanya	•			•		
Malta			•	•		
Polonya	•	•	•	•	•	•
Portekiz	•	•			•	•
İspanya	•	•		•		•
Slovakya	•	•				
İngiltere	•	•	•	•	•	

Tablo incelendiğinde ülkelerin öğretim programına kodlamayı entegre etmek için birden fazla nedenlerinin olduğu görülmektedir. 15 ülkenin mantıksal düşünme becerilerini ve 14 ülkenin problem çözme becerilerini desteklemeyi amaçladıkları, yine

11 ülkenin kilit yeterliliklerin ve kodlama becerilerinin geliştirilmesine odaklandıkları sonucu çıkmaktadır. Ülkelerin çoğunluğunun 21.yüzyıl becerilerini ele almayı amaçladıkları görülmektedir.

İngiltere’de okullarda belirlenen yaş gruplarına özel (5-6, 7-11 ve 11-14) programlama eğitimleri planlanmıştır. Birinci basamak olarak belirlenen 5-6 yaş grubunda algoritmanın ne olduğunun öğretilmesi, 7-11 yaş grubu olarak sınıflandırılan ikinci basamakta daha karmaşık programlar oluşturmaları ve hataları ayıklayabilecek (bulup düzeltebilecek) düzeye gelmeleri, üçüncü basamakta yani 11-14 yaş grubu aralığında ise öğrencilerin iki ya da daha fazla programlama diline hâkim olmaları hedeflenmektedir (Öndeş, 2016). Manches ve Plowman(2017), yapmış oldukları araştırmada erken yıllarda bilgisayar eğitimine yönelik İngiltere örneğini vermiştir. Royal Society (2012) tarafından yayınlanan raporda okul öğretim programlarında bilgi iletişim teknolojisine yer verilmesi gerektiğini belirtmiş, İngiltere bununla ilgili çalışmaları başlatarak Eylül 2014 yılından itibaren 1.kademedden itibaren 5-7 yaşları arasında 2 yılı kapsayan 6 temel alanı (Department for Education, 2013) içeren bir bilgisayar eğitimi öğretim programı geliştirilmiştir.

İncelenen ülkelerin birçoğunda programlama eğitimine ilkökul düzeyinde başlandığı görülmüştür. Ülkeler özellikle sivil toplum kuruluşları, yazılım ve teknoloji şirketleri ve devlet desteğiyle kodlama alanında birçok çalışma yapmaktadır. Örneğin, bu çalışmalar arasında Amerika Birleşik Devletleri'nde yaklaşık 6 milyon öğrencinin kullanmakta olduğu kodlamayla ilgili yüzlerce kavramı kullanarak kendi programlarını yazabildiği 2013 yılında kurulan “code.org” platformu öne çıkmaktadır (Öndeş, 2016). Avustralya'da 2015 yılında 5 yaşından itibaren öğrencilere 2 yıl temel kodlama eğitimi, 7 yaşından itibaren de programlama eğitimi verilmesini planlanmıştır. Dersler sonraki senelerde ileri düzey programlama derslerine dönüşerek ve programda 7 yaşında bir öğrencinin temel programlama mantığını çözmüş olacak şekilde bir düzenleme yapılmıştır (Öçalan, 2015 ve Kahraman, 2015). Özellikle Çin ve Hindistan gibi internet pazarının sahibi olan ülkelerde erken çocukluk döneminde programlama eğitiminin zorunlu verildiği gözlenmiştir (Özkaya, 2016). Bu anlamda ülkemizde de programlama eğitiminin daha erken bir yaşta vermeye yönelik çalışmaların yapılması gerektiği belirtilmektedir (Saygıner ve Tüzün, 2017).

Ülkemizde de algoritma ve kodlama eğitimi adına gerek Milli Eğitim Bakanlığı gerekse dernek ve özel kuruluşlar çalışmalar yürütmektedir. BTE Derneği tarafından

2013 yılında yayınlanan “Bilişim Teknolojileri Derslerinin Tarihi” başlıklı yazıda belirtildiği gibi, 1997 yılında 143 sayılı kararla yayınlanan haftalık ders çizelgesinde ilk kez “Bilgisayar” Dersi ilköğretim okulu seçmeli ders listesinde yer almıştır. 4.5.6.7.ve 8. sınıflarda bir veya iki saat seçilerek okutulmuştur. 2005 yılında 192 sayılı kararla derslerin saati 1’e indirilmiş ayrıca seçmeli derslerde notla değerlendirme kaldırılmıştır. 2007 yılında 111 sayılı kararla 4. ve 5.sınıflarda ders saati ikiye çıkarılmıştır. 2010 yılında yayınlanan 75 sayılı kararla 6-7-8.sınıflarda bir saat seçilebilmesine karar verilmiş. 2012 yılında 69 sayılı karar ile Bilişim Teknolojileri ve Yazılım Dersi çizelgede yer almış, ders saati iki saate çıkarılmıştır. Ders yepyeni bir çerçeve öğretim programıyla birlikte 5. sınıflarda uygulanmaya başlamıştır. 2013 yılında 22 sayılı karar ile BT Dersleri haftalık ders çizelgesinde ilk kez zorunlu olarak yer almış ve ders saati eğitim-öğretim bütünlüğünü sağlayabilecek şekilde iki saat olarak belirlenmiştir (BTE Derneği, 2013). Talim ve Terbiye Kurulu Başkanlığı'nın 16/10/2017 tarih ve 144 sayılı kararı ile ilköğretim kurumlarında "Bilişim Teknolojileri ve Yazılım" dersi 5 ile 6.sınıflarda iki saat zorunlu, 7 ile 8.sınıflarda ise 2 saat seçmeli ders olarak belirlenmiştir. Ortaöğretim bilgisayar bilimi dersi (Kur 1, Kur 2) öğretim programı, Talim ve Terbiye Kurulu Başkanlığı'nın 31/08/2016 tarih ve 65 sayılı kararı ile güzel sanatlar ve spor liselerinde 2016-2017, diğer ortaöğretim kurumlarında ise 2017-2018 Eğitim ve Öğretim Yılından itibaren Kur-1 den başlamak üzere kademeli olarak uygulanmaya başlanmıştır.

Ülkemizde Milli Eğitim Bakanlığı'nın BT Dersi çalışmalarının dışında kar amacı gütmeyen kurum ve kuruluşlar kodlama eğitimi konusunda ücretsiz eğitimler, atölye çalışmaları düzenlemektedir. Bu eğitimler ile kişiler kodlama eğitimi ile tanışmakta ve hayata yeni bir bakış açısı oluşturmaktadır. 2014 yılında Türkiye Bilişim Derneği ve çeşitli üniversitelerin desteğiyle ilk, orta ve lise öğrencilerinin bilgisayar ve internet teknolojileriyle kendi programlarını yazabileceklerini ve bunun basit bir şey olduğunu fark etmelerini sağlamak amacı ile “Bilgisayar Programlama Çocuk Oyuncağı” adlı bir etkinlik düzenlenmiştir (TBD, 2014). Bilişim Garaj Akademisi tarafından düzenlenen 7-8, 9-12 ve 13-16 yaş gruplarına yönelik programlama, web tasarımı, 3D tasarım ve robot üretimi gibi eğitimlerin verildiği projede ülkemizde gerçekleştirilen bir diğer önemli proje olarak nitelendirilebilir. Bu eğitimler sayesinde öğrenciler, programlamanın bir problem çözme süreci olduğunun farkına varmaktadırlar (Demirer ve Sak, 2016). Zorlu Holding, Avrupa Kodlama Haftası kapsamında 3 gün

süren “Hayallerini Kodla” etkinliği ile bilişim teknolojileri, internet güvenliği ve temel kodlama eğitimleri düzenleyeceğini duyurmuş (Zorlu Holding,2016) ve 9-13 yaş grubu 420 öğrencinin katıldığı bu etkinlik Zorlu Holding ve Code.org Türkiye sorumlusu Robincode işbirliği ile Ekim 2016’da gerçekleştirilmiştir. Türkiye Vodafone Vakfı çatısı altında, sürdürülebilir kalkınma alanındaki başarılı çalışmalarıyla tanınan Habitat Derneği'nin işbirliğiyle hayata geçirilen "Yarını Kodlayanlar" projesi kapsamında, İstanbul, Samsun, İzmir, Kayseri ve Mardin olmak üzere 5 ilde yaşları 7-14 arasında değişen yaklaşık bin çocuğa 2 ayda kodlama eğitimi verilmiştir (Türkiye Vodafone Vakfı, 2016). General Electric (GE), Robincode.org ile iş birliğinde “Kızlar Kodluyor” programı altında GE Türkiye İnovasyon Merkezi’nde 9-11 yaşları arasındaki avantajlı ve dezavantajlı toplam 100 kız öğrenciye temel programlama ve analitik beceri eğitimi vermiştir. Başarılı olan 30 öğrenci ise toplamda 16 saatten oluşan temel kodlama eğitimi olarak Robincode, Code.org ve Cisco Networking Akademi “get connected” sertifikasına sahip olmuştur (General Electric, 2016). Robincode.org, ülkemizde 5-16 yaş grubu çocuklara bilgisayar bilimleri (kod-3d-robot) ve stem eğitimleri veren, Cisco Networking Akademi ve Akademi Destek Merkezi olan, Oracle Akademi'nin desteklediği bir Türk sosyal girişimi olarak büyük bir başarıya imza atmıştır (Robincode, 2016). Dünyada, çocuklara bilgisayar bilimlerini öğretmek üzere kurulmuş, ortakları arasında Apple, Google, Facebook, Microsoft gibi şirketlerin bulunduğu Code.org, Robincode'u Türkiye sorumlusu ve global ortağı yapmıştır (Code.org, 2016). Borusan tarafından başlatılan “Haydi Kızlar Kodlamaya” adlı proje kapsamında, Borusan Holding, Bursa Gemlik Fabrikası çalışanlarının 6-13 ve 14-17 yaş gruplarındaki kız çocuklarına İngiltere’de bu alanda başarılı sonuçlar alan 'KızCode Platformu' ile kodlama ve algoritma eğitimi vermeye başlanarak 8 hafta süren eğitimlerde kursa katılan öğrencilere Scratch ile oyun, animasyon veya hikaye programlama, elektriğin temel prensip ve parçalarını öğrenme, basit elektronik modüller ve 3D printing entegrasyonu ile basit robot tasarlama gibi dersler verilmiştir. Bu programda sosyal sorumluluk alanları arasında kadının güçlendirilmesi yer aldığı için kız çocuklarıyla ilerlemeyi planladıklarını açıklamışlardır. (Borusan, 2016).

Günümüzde kodlama eğitimine verilen önem ve bu konuda yapılan çalışmalar incelendiğinde, daha önce de belirtildiği gibi kodlama eğitimi öncesinde algoritma eğitimi alınması ve bu eğitimin de erken yaşlarda örneğin okul öncesi eğitimde alınması gerektiği belirtilmektedir (Futschek ve Moschitz,2011; Karadeniz, Samur ve Özden,2014; Mittermeimer,2013; Gibson,2012). Algoritma eğitimi verilirken basit ve

anlaşılır komutlar kullanılacağından okul öncesi eğitimde yer alan öğrencilere verilecek eğitim onların günlük hayatta karşılaşacakları problemlerden oluşabilmektedir. Dolayısıyla erken yaşta verilecek bir algoritma eğitiminin çocukların problem çözme, kritik düşünme, karar verme becerilerini geliştirmeye katkıda bulunacağı düşünülmektedir. Bununla ilgili yapılan araştırmalarda (diSessa ve Avelson, 1986; Çırpılı, 2016; The Partnership for 21st Century Skill,2009) bu ve benzeri becerileri olumlu etkilediği görüşü savunulmuştur. Patan(2016), okul öncesi öğrencileri ile çalışmış okul öncesinde uygulanabilecek bir öğretim programı tasarlamıştır. Okul öncesinde kodlama öğretim programı uygulanan öğrencilerin kodlamaya karşı olumlu tutum geliştirdikleri görülmüştür.

Bilişim teknolojisinin bu kadar hayatımızın içinde olması ile birlikte Çocukların teknoloji ile tanışma yaşları gün geçtikçe küçük yaşlara inmektedir. Aileler bu konuda neler yapabileceklerini araştırmaya girmekte, okullarda da bilişim dersleri önem kazanmakta ve boyut değiştirmektedir. Özellikle okullarda hali hazırdaki öğretim programlarının içerisine entegre edecek şekilde bilişim teknolojilerinden en üst düzeyde fayda sağlamaya, yeni uygulamalar konusuna eğilmeye başlanmıştır. Bu anlamda özellikle kodlama eğitimi üzerinde durulmaya başlanmıştır. Günümüzde öğrencilerin okulla tanışma yaşlarına bakıldığında OECD ülkelerinin çoğunluğunda çocuklar 5 yaşından önce eğitim almaya başladığı ve 4 yaşındaki çocukların üçte ikisi (%84) okul öncesi veya ilköğretime devam ettiği görülmektedir (OECD, 2014). Okul ve bilişim teknolojileri ile tanışma yaşı bu durumdayken algoritma ve kodlama eğitimleri üzerinde durulmaya başlanmış olması doğal bir süreçtir. Bu çalışmada, okul öncesi eğitimine devam eden öğrencilerin kodlama eğitimi öncesinde verilecek olan algoritma eğitimi ve temel kodlama eğitimi ile problem çözme becerileri incelenecektir.

Amaç

Çalışmanın genel amacı, okul öncesi eğitime devam eden beş yaş grubu öğrencilerine verilecek algoritma ve kodlama eğitiminin çocukların problem çözme becerilerine etkisinin saptanmasıdır.

Bu genel amaç çerçevesinde aşağıdaki alt amaçlara cevaplar aranacaktır.

1. Algoritma ve temel kodlama eğitimi alan öğrencilerin problem çözme becerilerine yönelik ön test puanları ile son test puanları arasında anlamlı bir fark var mıdır?

2. Algoritma eğitimi almayan fakat temel kodlama eğitimi alan öğrencilerin problem çözme becerilerine yönelik ön test puanları ile son test puanları arasında anlamlı bir fark var mıdır?
3. Algoritma eğitimi alan ve almayan öğrencilerin problem çözme becerilerine yönelik son test puanları arasında anlamlı bir farklılık var mıdır?
4. Algoritma eğitimi alan Problem Çözme Becerisi Ölçeği puanları düşük ve yüksek olan öğrencilerin kodlamayla problemleri çözmelerine yönelik görüşleri nelerdir?
5. Algoritma eğitimi almayan Problem Çözme Becerisi Ölçeği puanları düşük ve yüksek olan öğrencilerin kodlamayla problemleri çözmelerine yönelik görüşleri nelerdir?

Önem

Günümüzde çocukların teknolojiyle tanışmaları çok erken yaşlara düşmüştür. Okul öncesi dönemde kazandırılan alışkanlıkların hayatlarını etkileyebileceği düşüncesi ile okul öncesi dönemde öğrencilerin teknoloji ile tanışma durumlarını olumlu hale dönüştürebilecek, problem çözme, eleştirel düşünme becerilerini geliştirecek kodlama eğitiminin verilmesinin önerildiği görülmüştür (Strand, 1986; Lewis ve diğerleri, 2014; Resnick, 2013; Perlman, 1976; Futschek ve Moschitz,2011; Mittermeier,2013; Kert ve Uğraş, 2009; Saygıner ve Tüzün,2017). Bu alanda Türkiye’de okul öncesi dönemde kodlama eğitimi ile ilgili yapılan çalışmaların oldukça sınırlı olduğu (Karadeniz, Samur ve Özden,2014; Patan,2016), genellikle ilköğretim ve üniversite öğrencileri ile çalışıldığı görülmüştür (Çetin,2012; Coşar,2013; Gülmez,2009). Benzer ve Erümit (2017), Türkiye’de programlama öğretimine yönelik, deneysel çalışma içeren lisansüstü tezlerini inceledikleri araştırmalarında, yapılan çalışmaların çoğunluğunun lisans ve ortaokul düzeyinde olduğunu belirtmişlerdir. Okul öncesi öğrencilerine yönelik eleştirel düşünme becerisi, problem çözme becerisi, yansıtıcı düşünme becerisi ve algoritmik düşünme becerisi gibi 21. yüzyıl becerilerini geliştirme amacıyla okul öncesi dönemde algoritma ve kodlama eğitiminin problem çözme becerileri ile ilişkilerinin inceleneceği bu araştırmanın ülkemizde okul öncesi dönemde yapılan çalışmaların azlığı nedeniyle alan yazına katkı sağlayacağı ve okul öncesi dönemde verilecek algoritma ve kodlama eğitimi açısından yol gösterici olacağı düşünülmektedir.

Sayıtlar

Öğrencilerin daha önceden tablet kullanma deneyimlerinin olduğu varsayılmaktadır.

Sınırlılıklar

Bu çalışma Ankara ili Yenimahalle ilçesinde yer alan özel bir anaokuluna devam eden beş yaş grubu öğrencileri ile sınırlıdır.

Kısaltmalar

PÇBÖ: Problem Çözme Becerisi Ölçeği

OSMO Kodıng: iPad için geliştirilmiş, çocukların uygulamalı oyun oynama fırsatı verirken kodlamayı öğreten bir ortamdır.

BÖLÜM II

YÖNTEM

Bu bölümde araştırmanın modeli, araştırma grubu, veri toplama araçları ve verilerin toplanması, verilerin analizi ve yorumlanması hakkında bilgi verilmiştir.

Araştırma Modeli

Araştırmada, okul öncesi eğitime devam eden beş yaş grubu öğrencilerine verilecek algoritma ve kodlama eğitiminin çocukların problem çözme becerilerine etkisinin belirlenmesi amacı ile nitel ve nicel verilerin elde edildiği sıralı açıklayıcı karma desen kullanılacaktır.

Karma yöntem araştırmalarının temel varsayımı, nitel ve nicel araştırma yöntemlerinin birlikte veya harmanlanarak kullanılmasının araştırma problem ve sorularının bu yöntemlerin ayrı kullanılmasından daha iyi anlaşılmasını sağladığıdır (Creswell, 2008; akt: Fırat, Kabakçı Yurdakul ve Ersoy, 2014). Nicel ve nitel araştırma tekniklerin aynı anda kullanıldığı karma araştırma yönteminin temel ilkesini Johnson ve Turner (2003) ise, “araştırmacı farklı strateji, yöntem ve yaklaşımları kullanarak çoklu veriler toplamalı” diye ifade etmektedir (Baki ve Gökçek, 2012).

Karma yöntemle uygulama yapan araştırmacı nitel ve nicel verileri sırayla (sequentially) veya aynı anda (concurrently) toplayabilir. Bu seçim araştırmacının amacına bağlıdır (Baki ve Gökçek, 2012). Alanyazın incelendiğinde karma araştırma yöntemiyle ilgili birçok tasarım bulunmaktadır. Creswell(2003), karma yöntem araştırmacılarının seçebilecekleri üçü sıralı, üçü ise eşzamanlı altı temel tasarım ortaya koymuştur. Bu araştırmada bu tasarımlardan baskın olarak nicel verilerin toplandığı, nitel verilerin nicel verileri pekiştirmek ve arttırmak için kullanıldığı sıralı açıklayıcı tasarım kullanılmıştır.

Nitel veriler araştırmacı tarafından geliştirilen gözlem formu ve uygulama sonrasında odak grup görüşmeleri esnasındaki yapılan ses kayıtları yoluyla, nicel veriler ise Problem Çözme Becerisi Ölçeği kullanılarak toplanmıştır.

Çalışma Grubu

Araştırma, 2016-2017 Eğitim Öğretim yılında Ankara ili Yenimahalle ilçesinde yer alan özel bir anaokuluna devam eden beş yaş grubu öğrencileriyle yürütülmüştür. Bu öğrencilere ait dağılım Tablo 2’de verilmiştir.

Tablo 2. Çalışma Grubu Durumu

Grup	Kız	Erkek	Toplam
Kontrol Grubu	8	7	15
Deney Grubu	5	10	15

Deney grubunda yer alan bir erkek öğrenci Problem Çözme Becerisi Ölçeği ön testini tamamlamak istemediği için PÇBÖ sonuçları istatistiksel hesaplamalarda dikkate alınmamış ancak öğrenci uygulama sürecinin dışına çıkarılmamış süreç boyunca sınıfta etkinliklere katılması sağlanmıştır. Çalışma grubu Tablo 3’te belirtildiği gibi güncellenmiştir:

Tablo 3. Çalışma Grubu Dağılımı

Grup	Kız	Erkek	Toplam
Kontrol Grubu	8	7	15
Deney Grubu	5	9	14

Veriler ve Toplanması

Araştırma kapsamında nitel ve nicel veri toplama araçları bir arada kullanılmıştır. Nitel veri toplama aracı olarak öğrenme sürecini izlemek amaçlı araştırmacı tarafından hazırlanan gözlem formu ve odak grup görüşmesi tekniği kullanılmıştır. Nicel veri toplama aracı olarak ise “Problem Çözme Becerisi Ölçeği” kullanılmıştır.

Araştırma kapsamında anaokuluna devam etmekte olan öğrencilerden oluşan toplamda iki grup ile çalışılmıştır. İlk gruba 4 haftalık algoritma eğitimi ve ardından 4 haftalık OSMO Koding eğitimi, ikinci gruba algoritma eğitimi almadan 4 haftalık

OSMO Kodıng eđitimi verilmiřtir. Eđitim ncesinde đrencilerin problem zme becerileri ile ilgili verileri toplamak iin her iki gruba da Problem zme Becerisi leđi n test olarak uygulanmıř, kodlama eđitimi tamamlandıktan sonra deney ve kontrol gruplarına Problem zme Becerisi leđi son test olarak uygulanmıřtır.

Osmo Kodıng eđitimi ařamasında her hafta arařtırmacı tarafından hazırlanan gzlem formuyla yapılan eđitimler izlenmiřtir. Eđitimin bitiminde PB puanları dřk ve yksek olan đrencilerle odak grup grřmesi yapılmıřtır.

Veri Toplama Araları

Gzlem Formu

Gzlem, herhangi bir ortamda veya kurumda oluřan davranıřı ayrıntılı olarak tanımak amacıyla kullanılan bir yntemdir. Eđer arařtırmacı herhangi bir ortamda oluřan bir davranıřa iliřkin ayrıntılı, kapsamlı ve zamana yayılmıř bir fotođraf elde etmek istiyorsa gzlem yntemini kullanabilir (Yıldırım ve řimřek, 2008: 169). Kodlama eđitiminin ařamalarını ieren iřlem basamaklarından oluřan bir gzlem formu arařtırmacı tarafından geliřtirilmiřtir.

Gzlem formlarının hazırlanmasında řu ařamalar gerekleřtirilmiřtir:

- a. Alanyazın taraması yapılarak gzlem formlarının kapsamı belirlenmiřtir.
- b. Hazırlanan gzlem formları kapsam geerliđi iin uzman grřne sunulmuřtur. Gzlem formları tez danıřmanı, okul ncesi blmnden bir đretim yesi, eđitim teknolojisi alanından iki uzman, 2 okul ncesi đretmeni, bir uzman psikolojik danıřman tarafından incelenmiř ve onların grřleri dođrultusunda deđiřiklikler yapılarak yeniden dzenlenmiřtir.
- c. Gzlem formları 3 đrenciye uygulanmıřtır.
- d. n uygulamanın ardından gzlem formlarında gerekli dzenlemeler yapılmıř ve formlara son biimi verilmiřtir.

Odak Grup Görüşmesi Formu

Odak grup görüşmesi önceden belirlenmiş bir konu hakkında, önceden belirlenmiş bir grup katılımcının düşüncelerini öğrenmek için planlanmış bir tartışmalar serisidir (Baş, Çamır, & Özmaldar, 2008). Çokluk, Yılmaz ve Oğuz (2011) çalışmasında odak grup görüşmelerinin planlanmasında ilk yapılması gerekeni araştırmanın amacının belirlenmesi, sonraki aşamaları ise soruların belirlenmesi, zaman planlaması, katılımcıların belirlenmesi, moderatörün ve görevlerinin belirlenmesi olarak açıklamıştır. Araştırmada odak grup görüşmesinin amacı uygulamada elde edilen verileri desteklemek, öğrencilerden derinlemesine bilgi almayı sağlamak, yapılan uygulamayla ilgili ne düşündüklerini ne yaşadıklarını ne hissettiklerini anlamaya çalışmak olarak belirlenmiştir. Yarı yapılandırılmış görüşme tercih edilerek soru listesine bağlı almak şartıyla görüşme esnasında akışa göre sorulara yer verilmiştir. Öğrencilerin yaşlarının küçük olması nedeniyle görüşmelerin her grup için 10-15 dakikayı geçmemesine dikkat edilmiştir. Gruplar, araştırmanın amacına uygun olarak Problem Çözme Becerisi Ölçeği puanı düşük ve yüksek olan öğrenciler gruplara yerleştirilerek belirlenmiştir. Araştırmacı bu görüşmenin moderatörü olarak yerini almış ve görüşme boyunca sorular çerçevesinde grubu yönlendirerek amaçlanan bilgileri toplamayı hedeflemiştir.

Veriler, 2016-2017 eğitim öğretim yılı 2.döneminde toplanmıştır. Odak grup görüşmesi algoritma eğitimi alan ve almayan öğrencilerden Problem Çözme Becerisi Ölçeği puanı düşük ve yüksek öğrenciler baz alınarak gerçekleştirilmiştir. Odak grup görüşmeleri 15 dakikayı geçmemiştir. Her bir odak görüşmesine 6-8 arası öğrenci katılmıştır. Yapılan planlama Tablo 4'te belirtilmiştir, ancak o gün okula gelmeyen öğrenciler olması nedeniyle odak grup görüşmeleri Tablo 5'te belirtildiği öğrenci sayıları ile gerçekleştirilmiştir.

Tablo 4. Planlanan Odak Grup Görüşmesi Tablosu

Grup	PÇBÖ Puanı Yüksek (>36)	PÇBÖ Puanı Düşük (<36)
Algoritma eğitimi alan	7	8
Algoritma eğitimi almayan	7	8

Tablo 5. Gerçekleştirilen Odak Grup Grup Görüşmesi Tablosu

Grup	PÇBÖ Puanı Yüksek (>36)	PÇBÖ Puanı Düşük (<36)
Algoritma eğitimi alan	6	6
Algoritma eğitimi almayan	7	7

Araştırmacı, görüşmenin başlangıcında öğrencilere OSMO uygulaması konusunda görüşlerini ve düşüncelerini almak istediğini belirtmiştir. Aşağıdaki soruları yönelterek görüşmeyi başlatmış ve sürdürmüştür:

- OSMO deyince aklınıza ne geliyor?
- OSMO kullanarak neler yapabilirsiniz?
- OSMO kullanmak size ne kazandırdı?
- OSMO'yu derslerde kullanmak ister misiniz?
- Kodlama deyince aklınıza ne / neler geliyor?

Verilerin analizinde içerik analizinin en uygun yöntem olduğu belirtilmektedir (Kitzinger ve Farquhar, 1999). Veri analizinde en iyi yolun, kayıt sırasında ya da sonrasında verilerin çözümlenmesi yapılırken, belli başlıklar altında belirlenen anahtar temaların olması gerektiği belirtilmektedir (Çokluk, Yılmaz ve Oğuz, 2011). Bu çalışmada da görüşme esnasında kayıt altına alınan verilerin içerik analizi yapılabilmesi için araştırmacı ve danışman tarafından ses kayıtları dinlenmiş, hem araştırmacı hem danışman tarafından ayrı metinler haline getirilmiştir. Araştırmacı ve danışman tarafından hazırlanan görüşmelere ait metinler karşılaştırılmış, düzenlenmiş, temalar belirlenerek belirlenen temalara göre bulgular raporlaştırılmıştır.

Problem Çözme Becerisi Ölçeği (PÇBÖ)

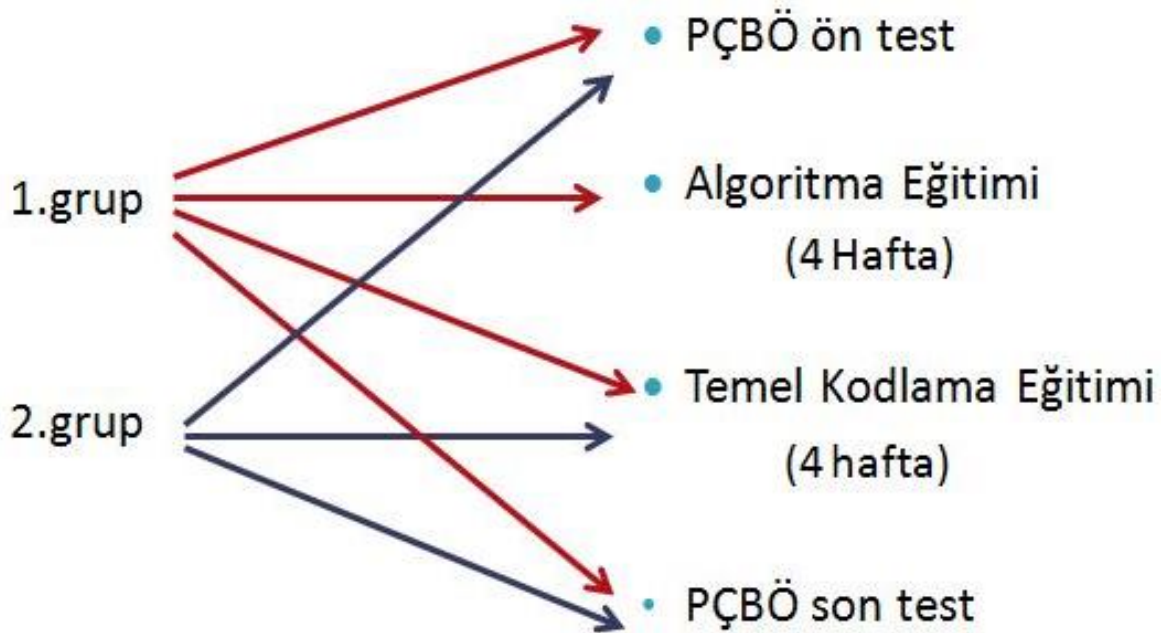
Çalışmada öğrencilerin problem çözme becerilerini ölçmek amacıyla Oğuz (2012) tarafından geliştirilen on sekiz problem durumuna ait çizimlerden ve değerlendirme formundan oluşan beşli likert tipi bir ölçek olan “Problem Çözme Becerisi Ölçeği (PÇBÖ) kullanılmıştır. Ölçeğin geçerlik çalışması için Kapsam Geçerlik İndeksi ve Açıklayıcı Faktör Analizi; güvenirlik çalışması için Cronbach Alfa

İç Güvenirlik Katsayısı ve Test – Tekrar Test Kararlılık Güvenirlik Katsayısına bakılmıştır. Kapsam geçerliği indeksi değerlerinin hesaplanması sonucunda, maddelerin uygunluk düzeyi için kapsam geçerliği indeksi 0.99; maddelerin çizimlere uygunluk düzeyi için ise kapsam geçerliği indeksi 0.96 olarak hesaplanmıştır. Açıklayıcı faktör analizi sonucunda ölçeğin tek faktörlü olduğu sonucuna varılmıştır. Bu faktör, ölçeğe ilişkin toplam varyansın % 30.68’ini açıklamaktadır. Güvenirlik katsayısı .86, test tekrar test sonrası korelasyon katsayısı .60 olarak bulunmuştur. PÇBÖ’den elde edilen ilk uygulama ve ikinci uygulama puanlarının istatistiksel anlamda farklılaşmadığı bulunmuştur. Geçerlik-güvenirlik çalışmaları sonucunda, 60-72 aylık çocuklar için ölçeğin uygun bir ölçme aracı olduğu sonucuna ulaşılmıştır (Oğuz ve Akyol,2015).

Öğrencilere eğitimlere başlamadan ön test olarak eğitimler sonrasında son test olarak Problem Çözme Becerisi Ölçeği uygulanmış alınan puanlar not edilmiştir.

Uygulama

Uygulama, PÇBÖ uygulanması ve algoritma eğitimi 4 hafta, kodlama eğitimi 4 hafta olacak şekilde toplam 8 haftada tamamlanmıştır.



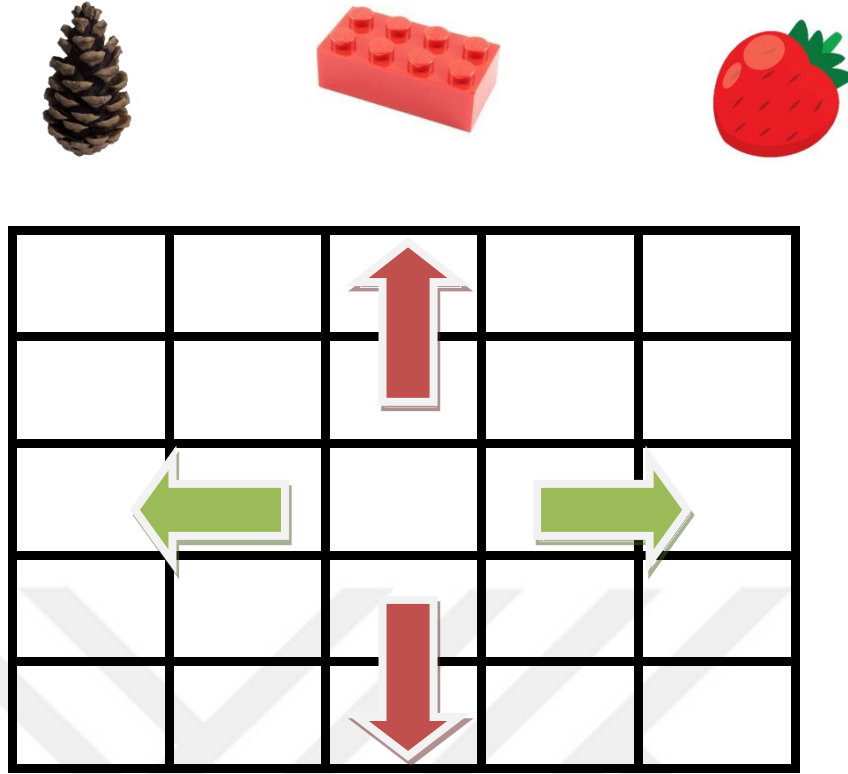
İlk hafta, ilk gün PÇBÖ iki gruba da uygulanmış, ön test verileri olarak kaydedilmiştir. Aynı hafta ilk gruba verilmesi planlanan 4 haftalık algoritma eğitimine geçilmiştir. Algoritma eğitimi verilirken, eğitim öncesi planlandığı şekilde algoritma eğitimine giriş, yön kavramları, çileklerle yön kavramları ve hazine avı etkinliği başlıkları takip edilmiştir.

Algoritma eğitimine giriş bölümünde, öğrencilerle günlük hayatta izlemiş oldukları sıralı işlemlerden, algoritma adımlarından bahsedilmiştir. Örneğin, sabah uyandıktan sonra evden çıkana kadar yapmış oldukları işlemleri düşünmeleri, anlatmaları, sıralı bir yol izlemeleri beklenmiştir. Daha sonra bir öğrenci seçilerek oturduğu yerden kalkıp sınıftan dışarı çıkana kadar yaptığı hareketleri sırayla söylemesi istenmiştir.

Yön Kavramları bölümünde, öğrencilere sağ-sol, ileri, geri komutları verildiğinde neler yaptıkları gözlenmiş ve birbirlerini incelemeleri, takip etmeleri beklenmiş, sonrasında yön komutlarının pekiştirilmesi için etkinlikler yapılmıştır.

Çileklerle Yön Kavramları bölümünde, etkinlik öncesinde çocukların sınıflarının zeminine 5x5 ölçüde zemine yapışacak bantlardan bir çalışma ortamı hazırlanmıştır (Şekil 1). Öğrencilerin zemin üzerinde ileri geri sağa sola dönebilecekleri belirtilerek ne yapacakları hakkında bilgi verilmiştir. Bu zeminde öğrencilerin her birine teker teker uygulamak için hem sınıfta kullandıkları eğitim materyalleri hem de araştırmacı tarafından hazırlanan "çilek" baskısı kullanılarak çalışmalar yürütülmüştür. Öğrencilere verilen problemi çözmek (nesnelere sırayla ulaşmak) için izleyecekleri yolu önce sözel olarak tanımlamaları istenmiş, sonrasında bu yolu fiziksel olarak izlemeleri beklenmiştir. Aynı hafta birkaç tane nesne konularak hepsini almaları için gerekli adımları önce sözel olarak tanımlamaları, sonra uygulamaları beklenmiştir. Sonrasında yine birden fazla nesneyi almaları için uygulayacakları gereken en kısa adımı sözel olarak tanımlamaları ve sonra uygulamaları beklenmiştir.

Hiçbir aşamada öğrencilerin yanlış komutları düzeltilmemiş, öğrencilerin hatalarını kendilerinin bulmalarını sağlamak için hata yaptıkları durumda öğrenciler ilk başlangıç karesine gönderilmiştir.



Şekil 1. Zemin Çalışma Ortamı

Hazine Avı bölümünde, okulun çeşitli yerlerine saklanan "Çilek" resimlerinin arkasında yer alan görevlere göre tüm çilekleri toplamaları beklenmiştir.













İlk gruba verilen 4 haftalık algoritma eğitiminden sonra her iki grupta da OSMO Koding Uygulamasına geçilmiştir. Uygulama aşamasında OSMO Koding seti kullanılmıştır (Şekil 2).



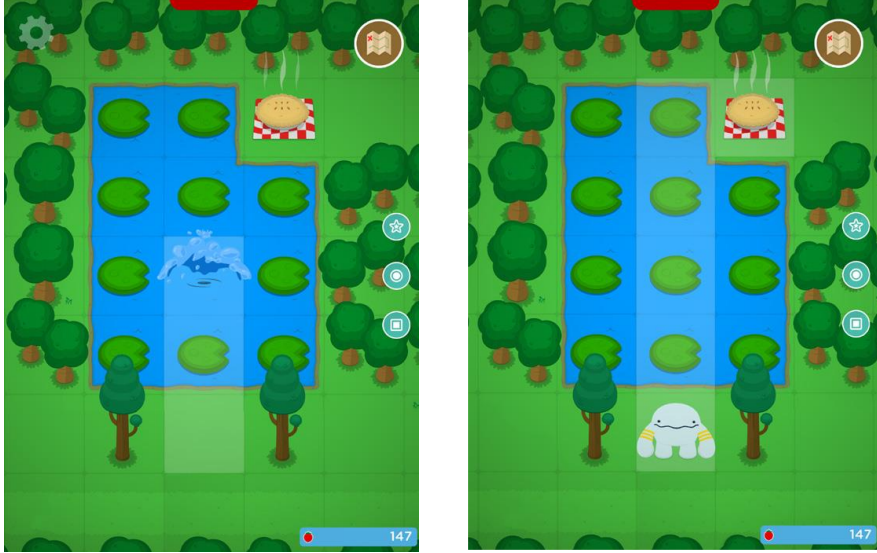
Şekil 2. OSMO Koding Seti

OSMO Koding Seti 5-12 yaş grubu için tasarlanmış ve öğrencilerin çocukların problem çözme ve mantık becerilerini öğreten, dijital dünyada onların başarılı olmasına yardımcı olmayı planlayan, kodlama yapmalarını sağlayan kolay bir settir. Bu set OSMO iPad tabanlı ve OSMO kod bloklarından oluşmaktadır. Her set “Awbie” adında çilek yemeyi çok seven ve eğlenceli bir karakteri kontrol etmek için kullanılan 19 manyetik bloktan oluşmaktadır (Tablo 6).

Tablo 6. Program Blokları

<i>Program Bloğu</i>	<i>Görevi</i>	<i>Program Bloğu</i>	<i>Görevi</i>
	Yürümek		Dikkat
	Zıplamak		1 adım
	El uzatmak		2 adım
	Sihir (Magic)		3 adım
	Döngü kurmak		4 adım
	Çalıştırmak		5 adım

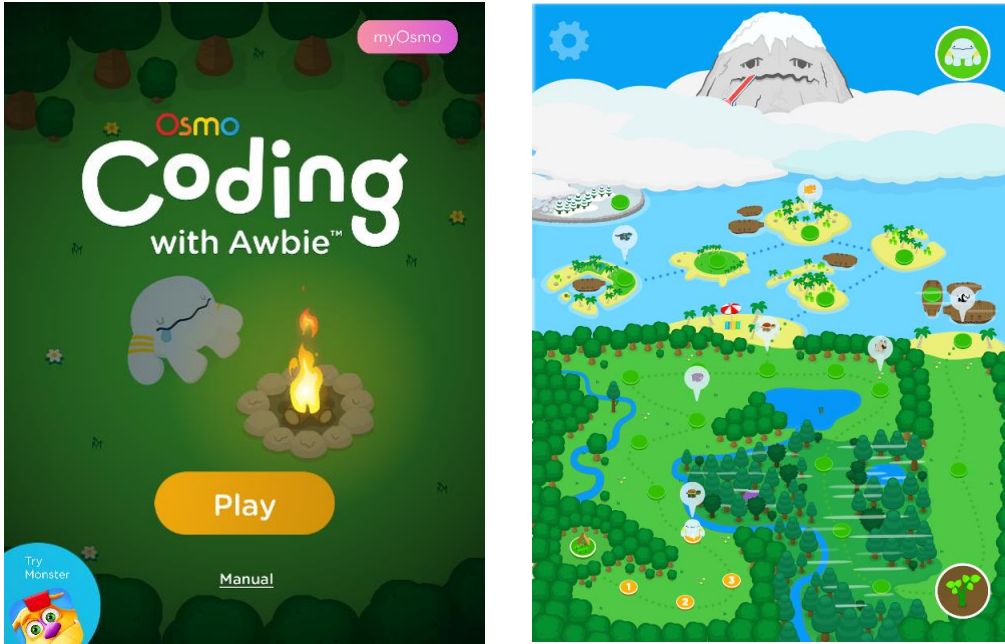
Uygulamanın kendi içinde belirli kuralları bulunmaktadır. Örneğin; nilüfer çiçeği uzun süre bir ağırlık taşıyamaz (Şekil 3). Eğer karakterimiz uzun süre üzerinde durursa batacak ve en yakın güvenli kareden yeniden başlayacaktır.



Şekil 3. Nilüfer Çiçeği Örneği

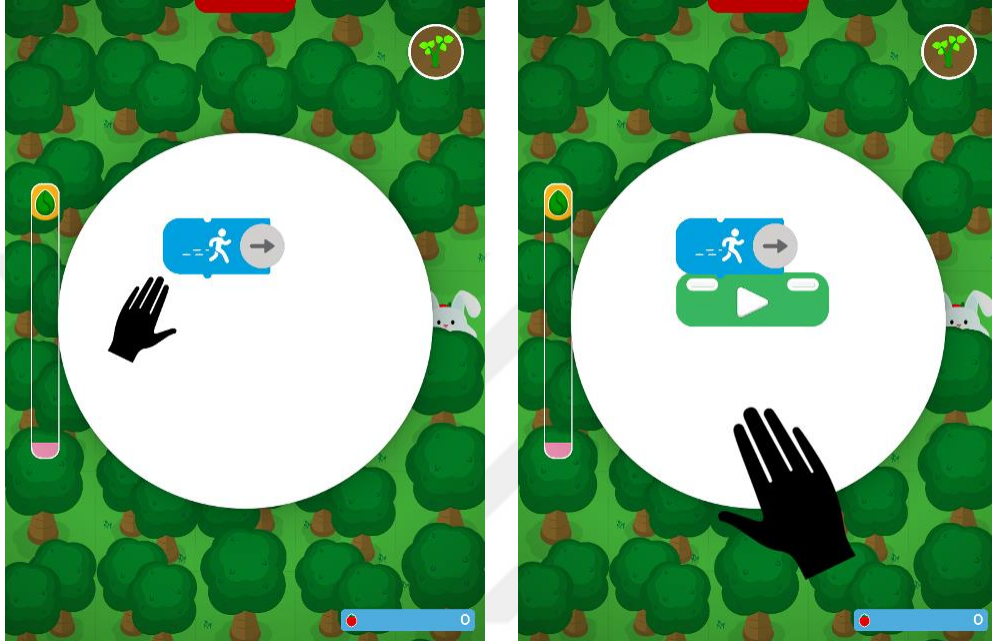
Öğrencilere her hafta belirli kod bloklarını kullanacakları seviyeler kullanılmıştır.

Uygulamanın başlangıç ekranı ve aşamaları gösteren ekran görüntüleri verilmiştir (Şekil 4). Burada uygulama "Play" komutu ile başlatıldığında daha önceden kayıtlı bir seviye varsa oradan devam edilir ya da ilk seviyeden başlanır.

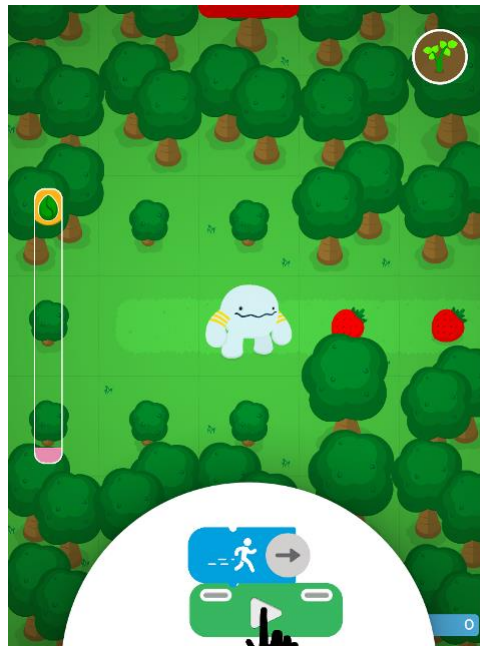


Şekil 4. OSMO Başlangıç Ekranı

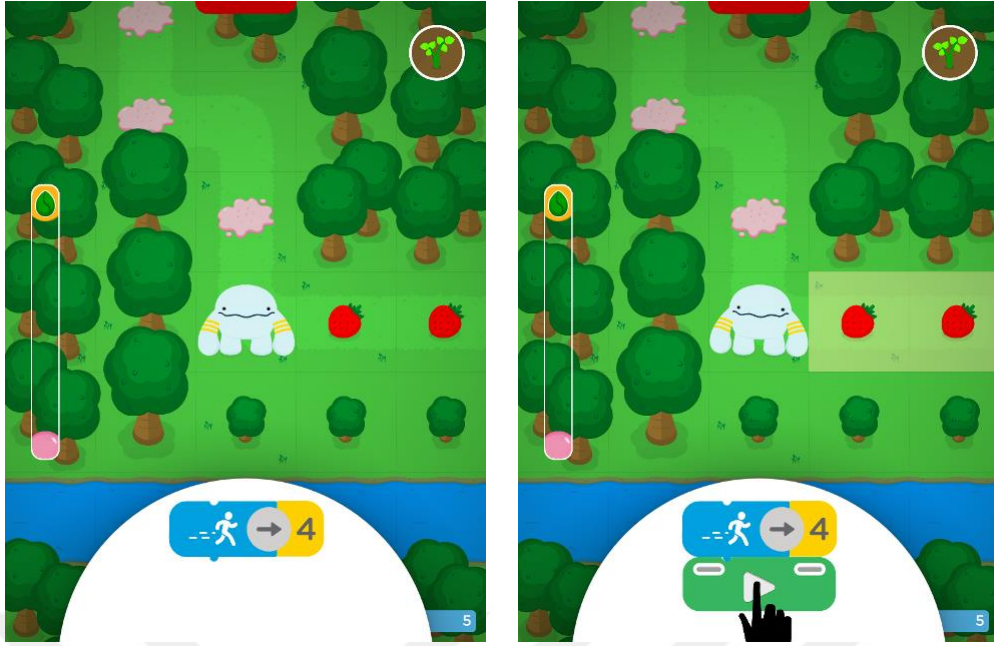
Asıl uygulamaya geçmeden önce bazı blokların ve bloklar sonucunda işlem ile karakterin ne yapılacağı gösterilir (Şekil 5). Oluşturulan kod bloklarına göre karakterin ne yapacağı farklı bir renkle gösterilir böylece adımın doğru olup olmadığı da kontrol edilebilir. Adım adım neler yapılacağı konusunda yönlendirme yapılır. Öğrencilere ilk hafta uygulamasına başlamadan önce bu bölüm gösterilmiş ve sonrasında devam edilmiştir (Şekil 6, Şekil7, Şekil 8).



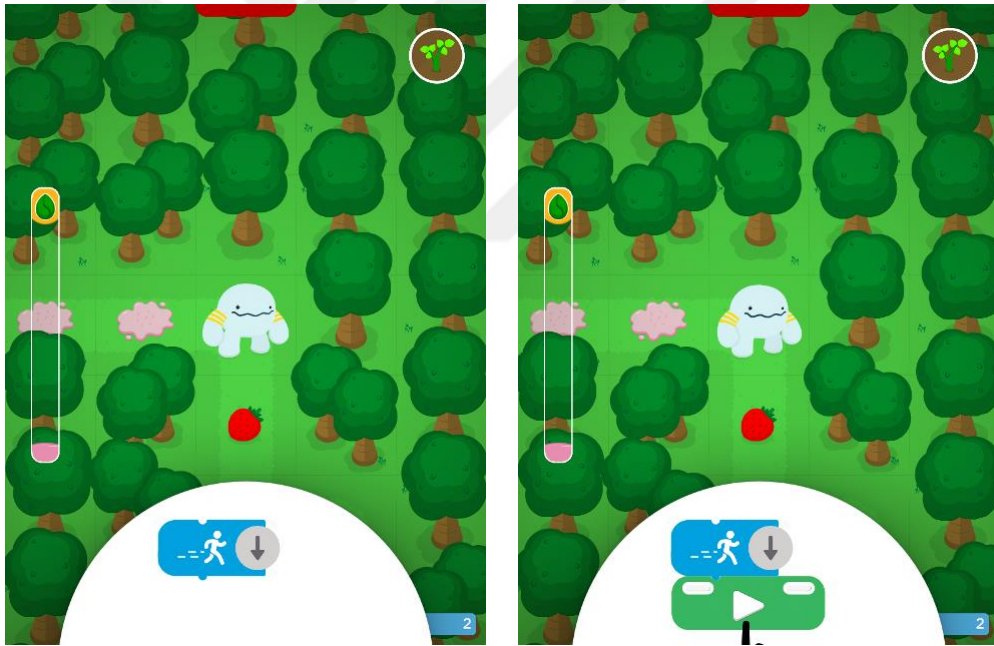
Şekil 5. Örnek Uygulama Adımları



Şekil 6. Örnek Uygulama Adımları - 2

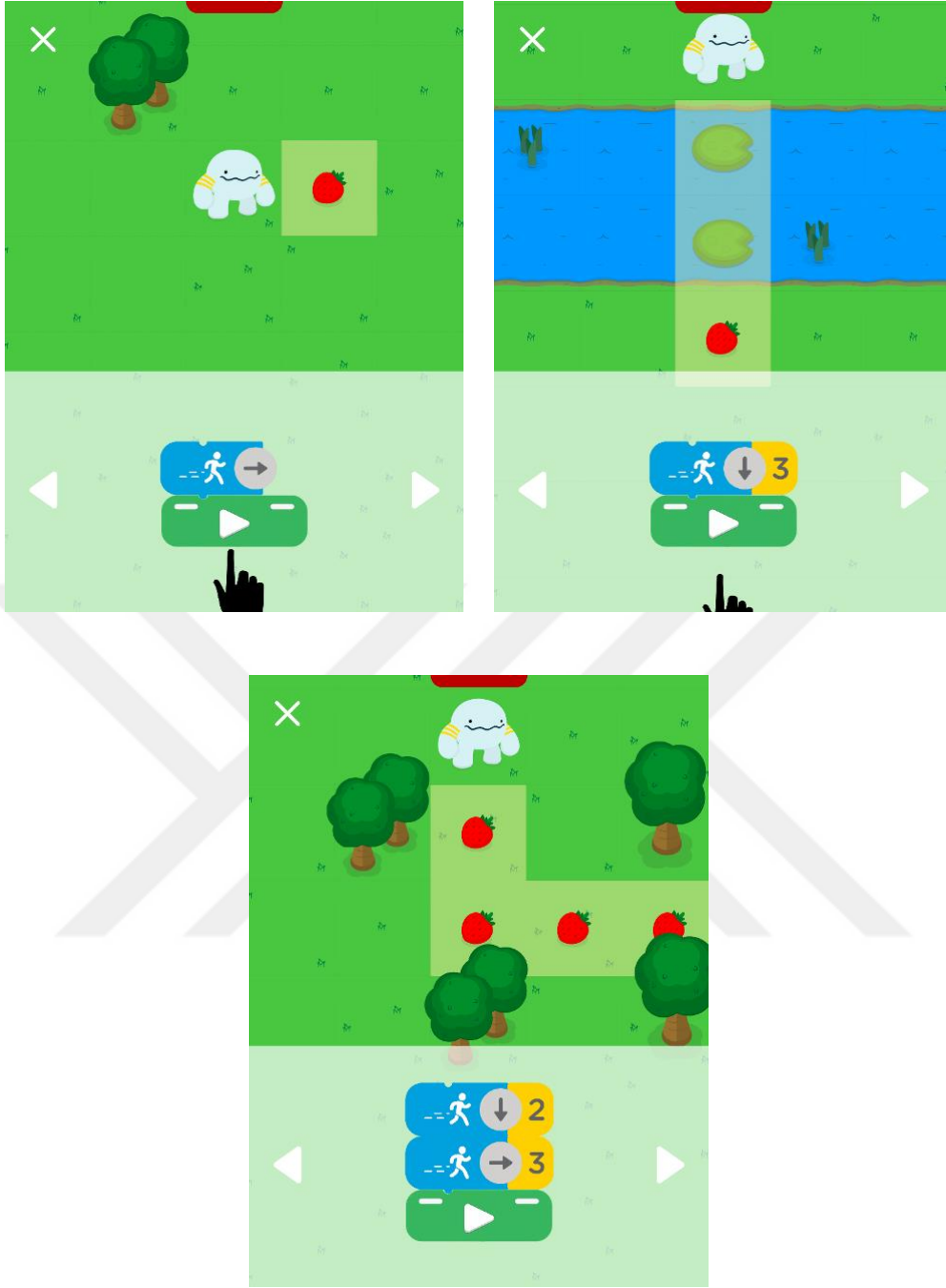


Şekil 7. Örnek Uygulama Adımları - 3



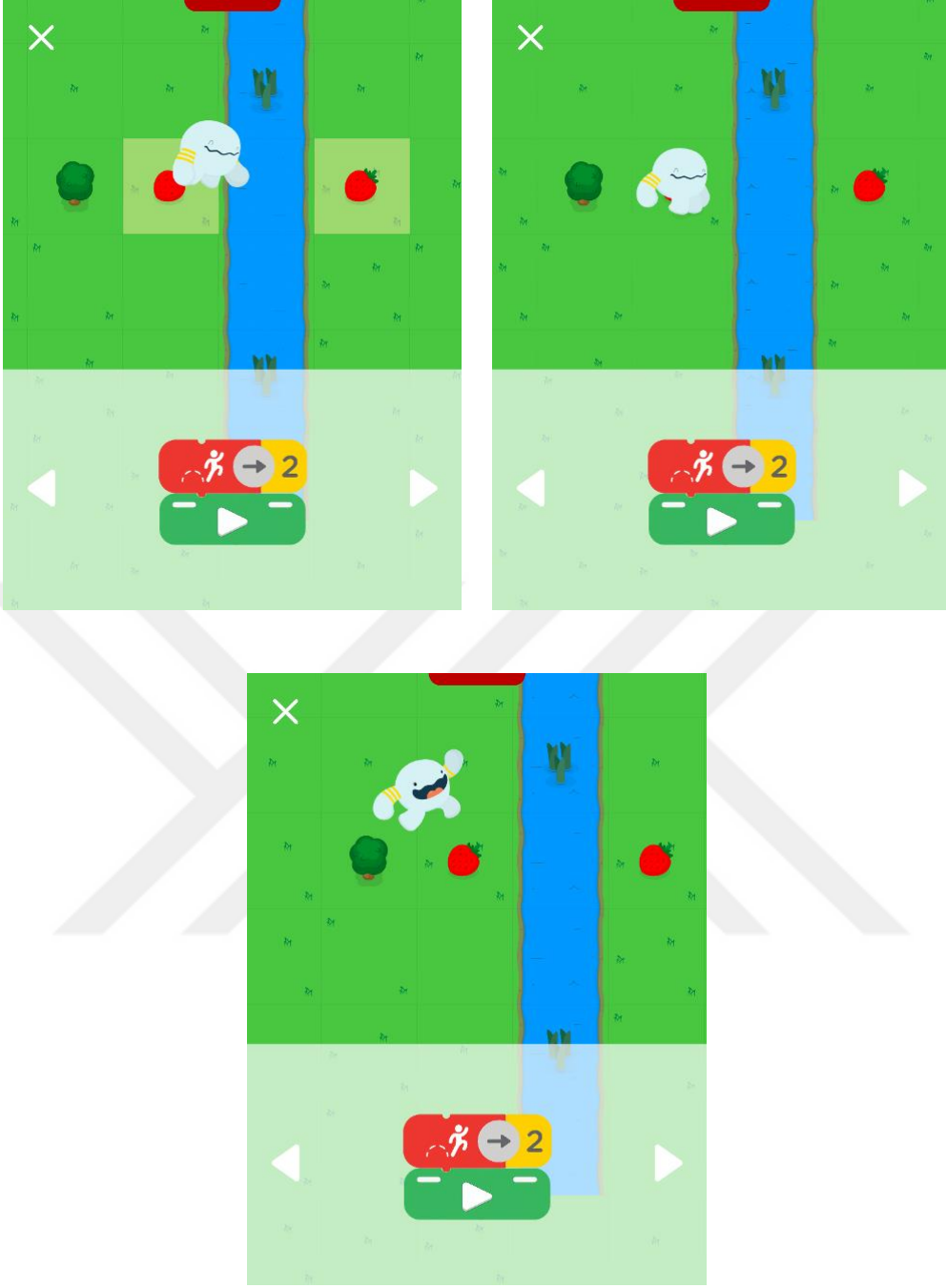
Şekil 8. Örnek Uygulama Adımları - 4

İlk hafta yön tuşlarının görevlerini algılamaları (ileri-geri-sağ-sol) için bir ve iki aşamadan oluşan görevleri tamamlamaları beklenmiştir (Şekil 9).

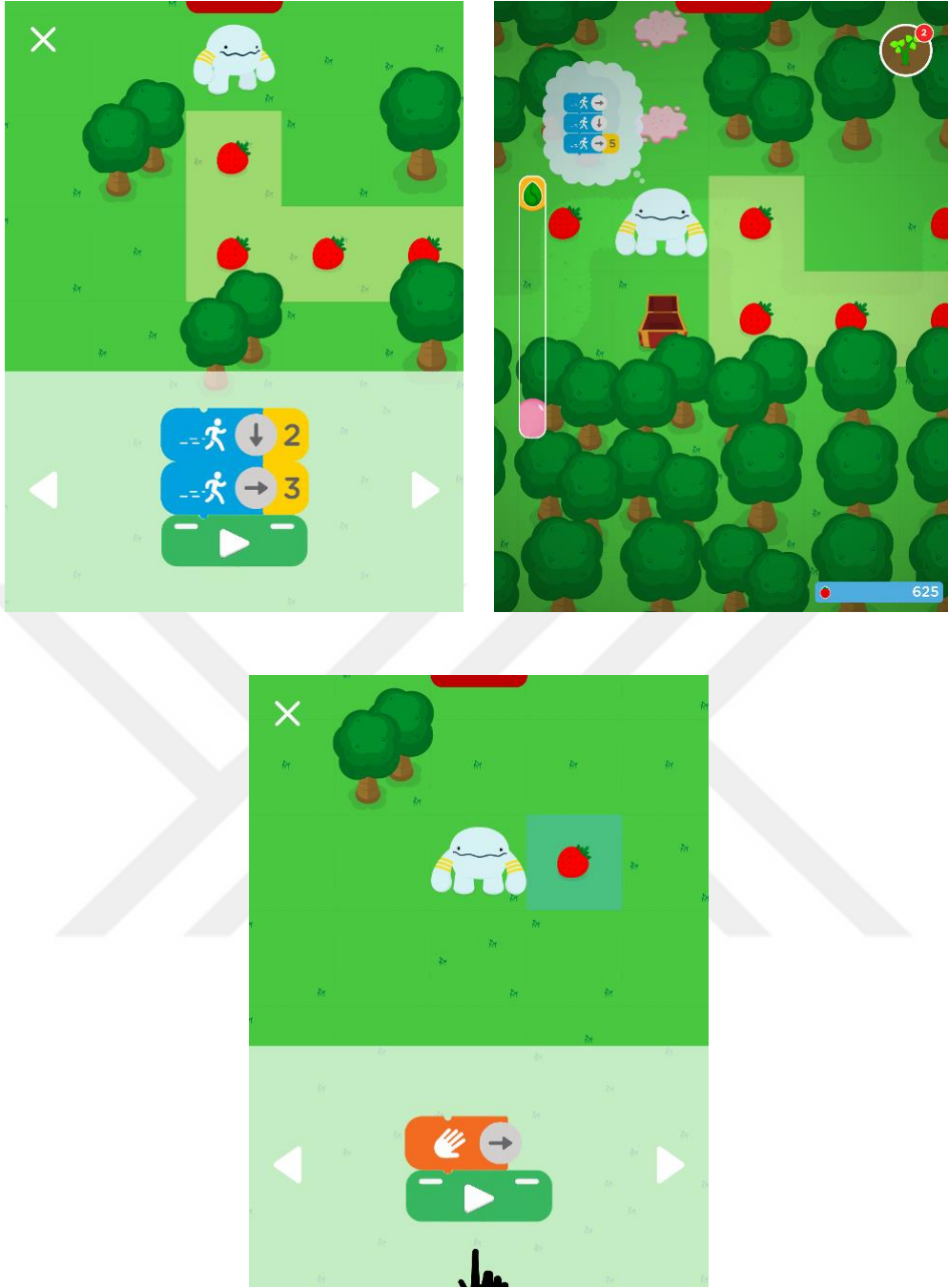


Şekil 9. İlk Hafta Örnekler

İkinci hafta öncelikle zıplama kod bloğu gösterilmiş sonrasında iki ve daha fazla aşamadan oluşan, aynı kod satırında ileri-geri-sağa-sola kod bloklarından en az iki tanesini, zıplama ve elle uzanma bloğunu kullanmalarını sağlayacak şekilde görevleri tamamlamaları beklenmiştir (Şekil 10-11).

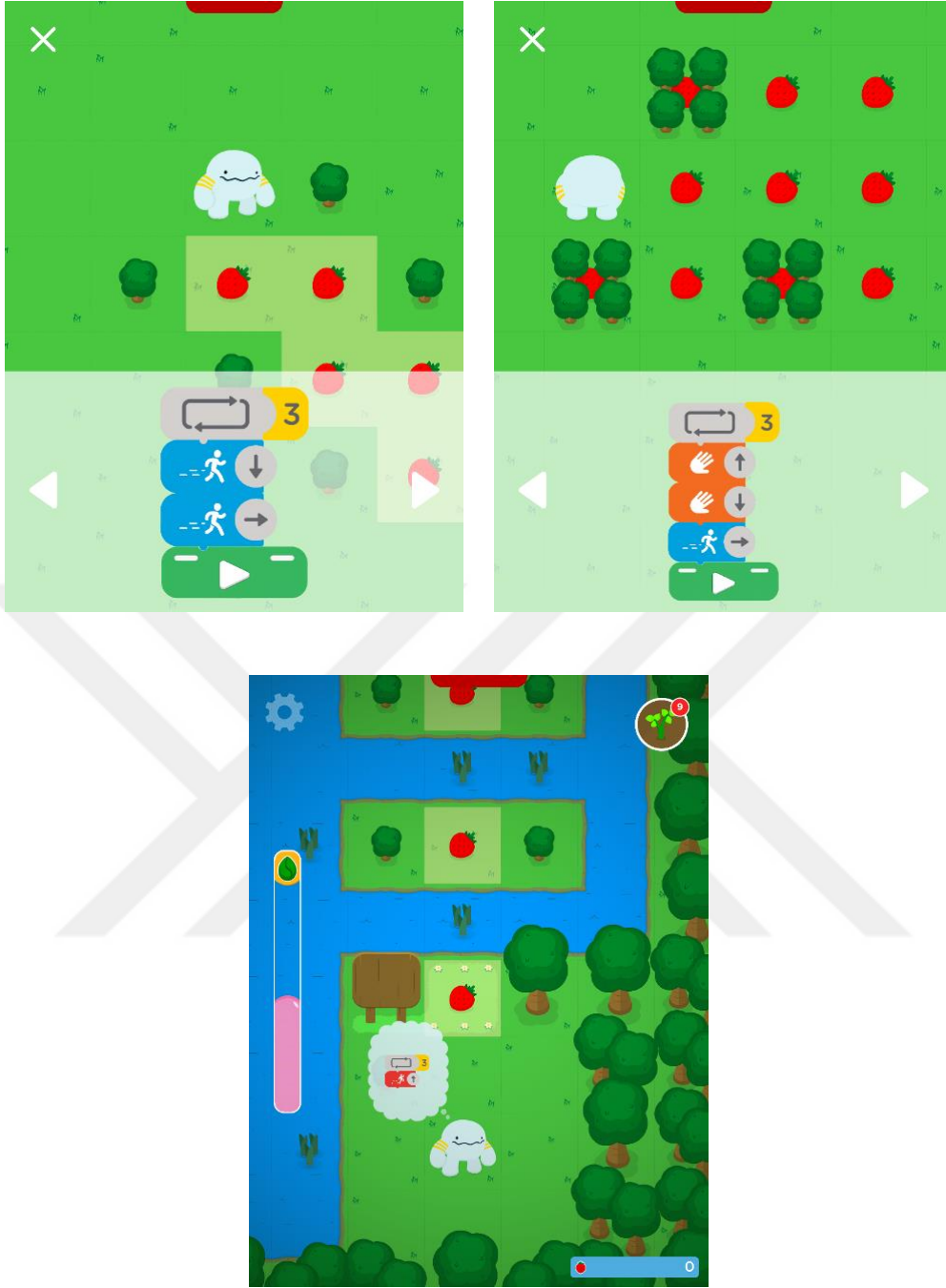


Şekil 10. İkinci Hafta Örnekleri



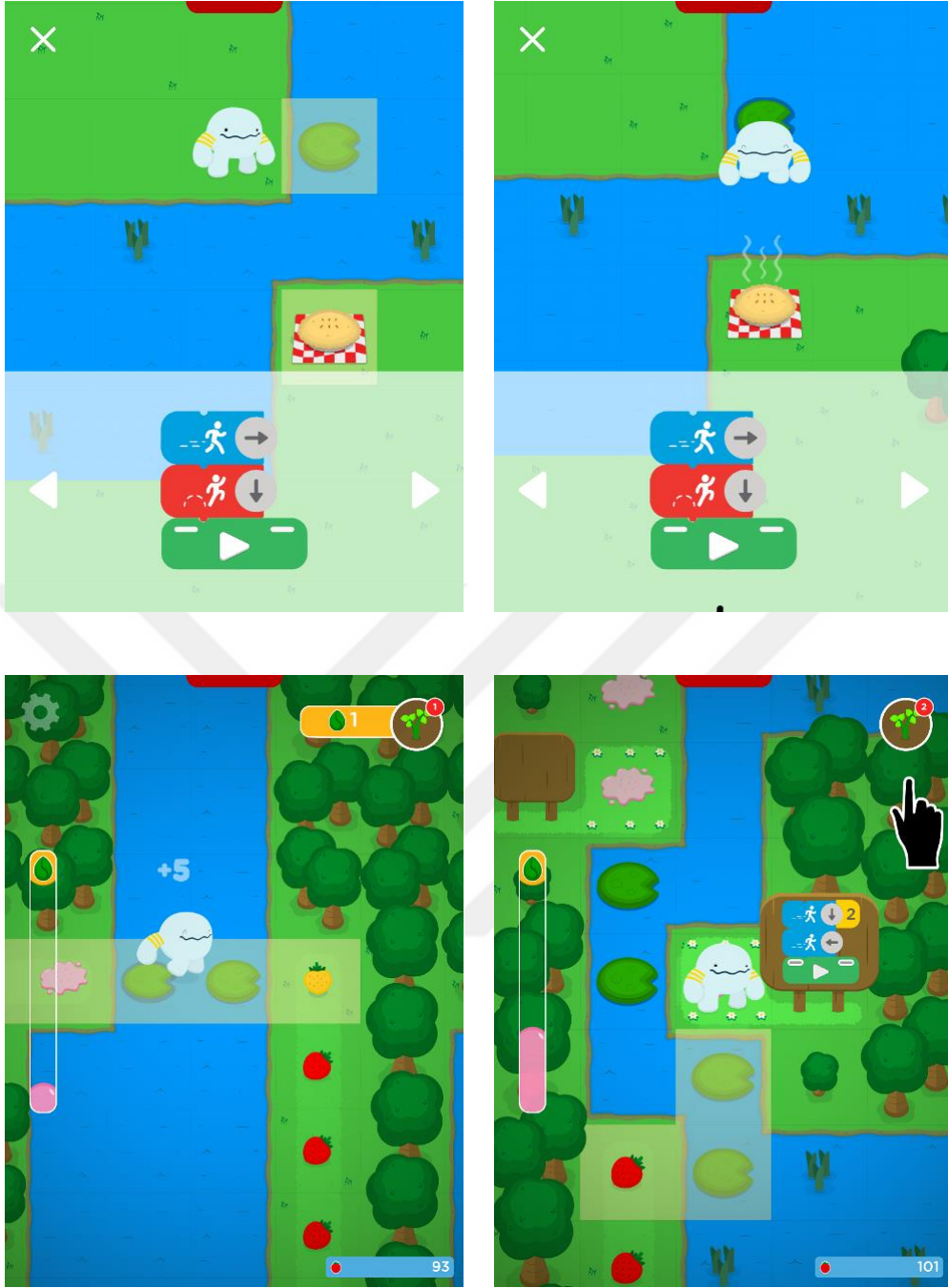
Şekil 11. İkinci Hafta Örnekleri-2

Üçüncü hafta döngü kod bloğunun kullanımı bir örnekle önce gösterilmiş sonrasında kullanmaları beklenmiştir (Şekil 12).



Şekil 12. Üçüncü Hafta Örnekleri

Dördüncü hafta şimdiye kadar uygulanan tüm blokların yanında nilüfer çiçeklerinin bulunduğu bölüme geçilerek zıplamaları ve uzun süre çiçek üzerinde durmamaları gerektiğini anlamaları beklenmiştir (Şekil 12).



Şekil 13. Dördüncü Hafta Örnekleri

4 hafta süren OSMO Kodlama eğitimi esnasında OSMO uygulamasını kullanırken ilgili aşamada istenen görevleri yerine getirerek, uygulayarak, sonuçları görerek blokların işlevlerini anlamaları sağlanmaya çalışılmıştır. Her hafta yapılan uygulamada ilgili haftada yapmaları beklenen bloklar masanın üzerine bırakılarak kendilerinin seçim yaparak problemi çözmeleri beklenmiştir. Süreç boyunca öğrenciler istenen görevleri yerine getirmişler, yaptıkları hataları görüp yeniden denemişlerdir. İlk haftalardaki

uygulamalarda hızla hareket etmek, alıřtırma blođuna hemen basmak isterlerken, 4.hafta uygulamasında adımlarını kontrol ettikten sonra alıřtırma blođunu kullanmayı tercih etmiřlerdir.

Verilerin Analizi

Arařtırmada elde edilen nicel verilerin istatistiksel analizinde SPSS programı kullanılmıřtır. Verilerin analizinde grup sayısının kk olmasından dolayı parametrik mi non-parametrik mi testlerin kullanılacağına karar vermek iin ilk olarak dađılımların homojenlik ve normalliđi test edilmiřtir.

Tablo 7. Dađılımların Test Edilmesi

Grup	Kolmogorov-Smirnova			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Deney	,152	14	,200 *	,946	14	,502
Kontrol	,127	15	,200 *	,927	15	,250

Tabloya gre $p > 0.05$ dzeyinde anlamlı olduđundan dađılımların normal olduđu grlmř, verilere ait homojenlik testleri de yapılmıř $p > 0.05$ olduđundan homojen olduđu sonucuna ulařılmıřtır. Bu nedenle parametrik testlerle hesaplamalarda veri setinin tmnn kullanılması parametrik olmayan testlere gre daha stn olmalarını sađladıđından, verilerin analizinde parametrik testler kullanılmıřtır.

Verilerin analizinde bađımsız rneklemeler iin t-testi ve bađımlı rneklemeler iin t-testi kullanılmıřtır. Bađımsız rneklemeler iin t-testi, bađımsız iki gruba ait ortalama arasındaki farkın manidarlıđını test etmek iin, bađımlı rneklemeler iin t-testi iki bađımlı puan setinden elde edilen ortalamalar arasındaki farkı test eder (Bykztrk, okluk, Kkl, 2010). Sonular % 95 gven aralıđında, anlamlılık $p < 0,05$ dzeyinde ift ynl olarak deđerlendirilmiřtir.

Gzlem formu ve ses kaydından elde edilen nitel verilerin analizinde ierik analizi yntemi kullanılmıřtır. İerik analizi szel, yazılı ve diđer materyallerin nesnel

ve sistematik bir şekilde incelenmesine olanak tanıyan bilimsel bir yaklaşımdır (Tavşancıl ve Aslan, 2001). İçerik analizinde, dokümanlardan elde edilen nitel araştırma verilerinin işlenmesi, verilerin kodlanması, temaların bulunması, kodların ve temaların düzenlenmesi, bulguların tanımlanması ve yorumlanması şeklinde dört aşama kullanılır (Yıldırım ve Şimşek, 2008). Elde edilen nicel çözümler, nitel veriler desteklenerek yorumlanmıştır.



BÖLÜM III

KURAMSAL ÇERÇEVE VE İLGİLİ ARAŞTIRMALAR

Bu bölümde bilişimsel düşünme, problem çözme, eğitimde algoritma öğretimi, eğitimde programlama, okul öncesinde programlama uygulamalarına yönelik kavramlarla ilgili bilgilere yer verilmekte, okul öncesinde kullanılan kodlama siteleri ve bazı uygulamalardan bahsedilmektedir. Son olarak da bu konularda yapılan araştırma ve yayınların bazıları aktarılmaktadır.

Bilişimsel Düşünme ve Problem Çözme

Bilişimsel düşünme ile ilgili birbirinden farklı Türkçe karşılıklar ulusal alan yazında yer almıştır (Kert, Yeni, Şahiner, 2017). Bu karşılıklar arasında; “Bilgi-İşlemsel Düşünme” (Kalelioğlu ve Gülbahar, 2015), “Bilgisayarca Düşünme” (Korkmaz, Çakır ve Özden, 2015), “Bilgisayımsal Düşünme” (Çınar ve Tüzün, 2017), “Bilişimsel Düşünme” (Patan vd., 2015), “Hesaplamalı Düşünme” (Özçınar, Yecan ve Tanyeri, 2016), “Komputasyonel Düşünme” (Şahiner ve Kert, 2016) sayılabilir. Farklı başlıklar ile olsa da, temel de araştırmacılar tarafından benzer tanımlamalarla çerçevesi çizilmektedir.

Patan (2006), bilişimsel düşünmeyi (Computational Thinking), teknolojinin kullanımı ile çözülebilecek bir problemi tanımlayarak çözüm yollarını algoritmik düşünme ile belirleyip, en etkili ve verimli çözümü uygulayarak değerlendirmeyi ve benzer durumlara bu süreci transfer etmeyi içeren bir beceri kümesini ifade etmektedir, şeklinde tanımlamıştır. Wing (2006) bilişimsel düşünmeyi, bilgisayar bilimi temelindeki kavramları kullanarak, problem çözme, sistem tasarlama ve insan davranışını anlama olarak tanımlamıştır (Yıldız, Çiftçi, Karal,2017). ISTE (2015) tarafından bilişimsel düşünme yeteneğinin yaratıcı düşünme, algoritmik düşünme, problem çözme ve işbirlikli öğrenme gibi becerileri içerdiği ifade edilmiştir.

Wing (2006) kavramın alt başlıklarını Communications of the ACM’de yayınlanan “Computational thinking” çalışmasında ele almış ve yazısı içerisinde bilişimsel düşünme becerisini şu alt kavramlarla açıklamaya çalışmıştır:

- Problem çözüme
- Soyutlama
- Ayırıştırma
- Sezgisel akıl yürütme
- Matematik ve mühendislik temelli düşünme

Bilişimsel düşünmenin bir problem çözüme süreci olduğunu ve şu özellikleri içerdiği belirtilmiştir (Barr, Harrison ve Conery (2011):

- Problemi bilgisayar veya diğer araçların etkin olabileceği şekilde formülize etme,
- Verileri mantıksal olarak düzenleme ve analiz etme,
- Modeller ve benzetimler gibi soyutlamalar yoluyla verilerin sunulması,
- Algoritmik düşünerek çözüm yollarını otomatikleştirme (bir dizi adımlarla),
- Tanımlama, analiz etme, adımları ve kaynakları en verimli ve etkili şekilde bir araya getirilmesi ile olası çözümleri uygulama,
- Bu problem çözüme sürecini çeşitli problemlere aktarma ve yaygınlaştırma.

Bütün bunlarla beraber, Yıldız, Çiftçi ve Karal (2007)'in aktardığı gibi bu tanımlamaların yanı sıra Uluslararası Eğitim Teknolojileri Topluluğu (ISTE) ve Bilgisayar Bilimi Öğretmenleri Derneği (CSTA), yükseköğretim, sanayi ve K-12 eğitim liderleri ile işbirliği yaparak işlevsel bir bilişimsel düşünce tanımı ortaya koymuşlardır (CTSA & ISTE, 2011). Bilişimsel düşünceyi, aşağıdaki özellikleri içeren (sadece bunlarla sınırlı olmayan) bir problem çözüme süreci olarak tanımlamışlardır:

- Problemlerin çözümüne yardımcı olması için bilgisayar ve diğer araçları kullanmayı sağlayan problemleri formüle etme
- Verileri mantıksal olarak organize etmek ve analiz etme
- Modeller ve simülasyonlar gibi soyutlamalar yoluyla verileri temsil etme
- Çözümleri algoritmik düşünce yoluyla otomatikleştirme (sıralı adımlar dizisi)

- Aşamaların ve kaynakların en verimli ve etkili bir kombinasyonunu gerçekleştirmek amacıyla olası çözümleri belirleme, analiz etme ve uygulama
- Bu problem çözme sürecini genelleme ve çeşitli problemlere transfer etme

Bilişimsel düşünme kavramında karşımıza çıkan "problem" kavramını John Dewey; öğrencinin öğrenmesini sağlayan, öğrenciyi başarıya götüren çözülmesi gereken güçlükler olarak tanımlamaktadır. Problemler, kişisel örgütsel, toplumsal, ekonomik veya teknolojik olabilir. Problem çözme ise hayata ve topluma uyum süreci olarak düşünülebilir. Problem çözme davranışları problem çözme sürecindeki zihinsel faaliyetlerin gözlenebilir ürünüdür (Erden,1984). Aynı zamanda da duyuşsal ve psikomotor becerileri gerektiren karmaşık bir süreçtir. Problem çözme her şeyden önce belli bir amaca ya da amaçlara ulaşmak için karşılaşılan güçlükleri ortadan kaldırmaya yönelik bilgileri düzenlemeyi, esnek olmayı ve bilişsel kaynakları etkili bir biçimde kullanmayı gerektiren bir dizi çabayı içermektedir (Akkaya, 2006).

Problemleri çözme becerisinin en önemli amacı ise (Şahin,2004);

1. Öğrenciyi edilgen bilgi alıcısı olmaktan çıkarıp aktif, özgür ve kendi kendine öğrenen ve problem çözen kişi yaparak eğitim programının vurgusunu öğretmekten öğrenmeye kaydırabilme,
2. Ezberlenmesi gereken bilgiyi sınırlandırıp, öğrenciyi yüklü içerikler yerine çözülecek problemlerle karşı karşıya bırakarak yeni bilgiler edinmesini sağlayacak beceri ve tutumları geliştirmesini sağlayabilme,
3. Öğretmenin, öğrencilere problemlerini çözmeleri için kolaylaştırıcı ortamlar hazırlaması ve işbirliği yapabilmesidir.

Problem çözme, belli aşamalardan oluşmaktadır. Polya (1957) problem çözme sürecini ana hatlarıyla şu basamaklardan oluştuğunu ifade etmektedir:

- Problemi anlama,
- Plan yapma,
- Planı uygulama,
- Geriye bakış (Polya, 1957).

Okulöncesi dönem ve her yaş grubunda kullanılabilir problem çözme aşamalarını şu şekilde sıralamak olanaklıdır (Şahin, 1998, Akt. Akkaya, S.):

Problemin anlaşılması: Bilimsel problem çözme yönteminde ilk aşama, tahmin edilmesi gereken isteğin ne olduğunun, ne gibi zorlukların bulunduğu veya bulunabileceğinin ve çözümü neyin sağlayabileceğinin saptanabilmesidir. Bu basamağın gereklerinin dikkatli bir biçimde ve tam olarak yerine getirilmesi ileride harcanacak gereksiz çabaları da önleyecektir.

Daha önce kazanılmış bilgi ve verilerin toplanması: Problem çözümüne akıllıca bir yaklaşımda, önceden aynı problem ya da benzer problemler karşısında başkalarının neler yaptığını öğrenmek vardır. Bu sayede belki de yeniden keşfedilmesi uzun zaman alacak anahtar bilgiler bulunabilir, öğrenildiği takdirde kolayca kaçınılabilecek hatalar işlenmemiş olur.

Olası çözümlerin ya da hipotezlerin ortaya konması: Bir problemle ilgili bilgi toplanırken problemin nasıl çözülebileceğine ilişkin fikirler ve olasılıklar da ortaya çıkar. Tüm bunlar hipotezler halinde ifade edilebilir ve daha sonra bu hipotezlerden biri doğru çözüm olarak ortaya çıkabilir.

Olası çözümlerin değerlendirilmesi: Bu aşama aslında bir önceki aşamanın devamı niteliğindedir. Olası çözümler öne sürülürken bu çözümler aynı zamanda değerlendirmeye de alınırlar. Bu bakımdan bu aşama bir yargılama ve tahmin evresidir.

Olası çözümlerin denenmesi ve sınanması: Bu aşamada çocuk, problemi bir kez tanımladıktan, olası çözüm yollarını düşündükten ve ipuçlarını değerlendirdikten sonra, en uygun gördüklerini seçip dener. Eğer çözüm uygunsa sorun yoktur. Ancak, çözüm uygun değilse başka çözüm yolları deneyebilir.

Sonuçların çıkartılması: Elde edilen yeni bilgilerin olgular halinde çözümlenip saklanması ileride karşılaşılabilecek yeni problemlere de ışık tutacaktır. Bu nedenle problem çözme işleminden elde edilen sonuçların tam ve doğru olarak belirlenmesi şarttır.

Stevens (1998) problem çözme sürecinin aşamalarını aşağıdaki gibi sıralamaktadır:

- Problemin anlaşılması,

- Gerekli bilgilerin toplanması,
- Problemin köküne inilmesi,
- Çözüm yollarının ortaya konulması,
- En iyi çözüm yolunun seçilmesi,
- Problemin çözülmesi.

Etkili problem çözme becerisi elde etmiş bireylerin bağımsız ve yaratıcı düşündükleri, sosyal yeterlilikleri olduğu, kendilerine güvenen, belirsizliklerin üstesinden gelebilen kişiler olduğu belirtilmektedir (Dow ve Mayer, 2004, Akt: Serin, 2010). Gelbal (1991), problem çözmede, bireylerin eğitilmesinin de önemli rol oynadığını, öğrencilere problemi vermeden önce o konu ile ilgili bir eğitimin verilmesinin onların problemi çözmelerini kolaylaştırabileceği görüşünü belirtmektedir.

Çocuk çok küçük yaşlardan itibaren çeşitli problemlerle (Örneğin, “koltuğun arkasına yuvarlanan topumu nasıl alabilirim?”, yağmur nereden gelir?” vb.) karşılaşmaktadır. Okul öncesi dönem çocuğuna çeşitli problem çözme durumlarında deneyim edinebilmesi için bu deneyimlerin sistematik bir biçimde kendisine sağlanması gerektiği belirtilmektedir (Kişisel ve Yıldırım,1983).

Çocuğun, problem çözme sürecinde neden-sonuç ilişkisi kurallarını anlaması gerekmektedir. Çocuk neden-sonuç ilişkisini anladıktan sonra problem çözmede belirli durumlarda neler olabileceğini önceden keşfetme ve geçmiş olayların nedenlerini anlayabilme olmak üzere iki yeni beceri kazanmaktadır (Kişisel ve Yıldırım, 1983).

Genel tanımlara bakıldığında problem çözme kavramının bilişimsel düşünme tanımlarının içerisinde yer aldığı, aralarındaki ilişkinin öne çıktığı görülmektedir (Kert, Yeni ve Şahiner,2017). Wing (2006), bilişimsel düşünmenin “bilgisayar bilimleri temelindeki kavramların altını çizerek problem çözmeyi, sistem tasarlamayı ve insan davranışını anlamayı gerektirdiğini” ifade etmiştir (Wing, 2006). Wing’ in 2011 yılında yaptığı tanımlamada ise bilişimsel düşünme “problemleri ve çözümlerini formüle ederken kullanılan düşünce süreçleri, çözümlerin bir bilgi-işlem ajanı tarafından etkili bir biçimde yürütülebilecek şekilde temsil edilebilmesi” olarak açıklamıştır (Wing, 2011).

Algoritma Eğitimi

Algoritma, belirli bir problemi çözmek veya belli bir amaca ulaşmak için takip edilmesi gereken adım adım yönergelerden oluşan bir yapıdır. Bu yapının oluşturulmasında kullanılan komutların mümkün olduğunda basit ve kısa olması tercih edilir. Algoritma eğitimi ise, programlama öğretiminden önce çözülmesi istenen probleme yönelik olarak verilmesi gereken bir eğitim olarak tasarlanmaktadır. Bu tasarım öğrencinin yaşına, hazır bulunuşluk düzeyine, ilgisine göre hazırlanmaktadır.

Algoritmik düşünce kavramı problem çözme becerisi, problem çözme süreci gibi kavramlar düşünüldüğünde karşımıza çıkan disiplinlerarası alanda değerlendirilmesi gereken bir kavramdır. Bu kavramı sadece bilgisayar bilimi ya da programlama ile kısıtlamak doğru değildir. Bilgisayar bilimi sadece bilgisayarı öğretmek değil bilgisayardaki uygulamaların nasıl çalıştığı konusunda bilgi edinmeyi kapsamaktadır. Okullardaki bilgisayar bilimi dersinin sadece programlamayı öğretmekten ziyade bilişimsel düşünme, algoritmik düşünme, algoritmik öğrenme hakkında olması gerektiği düşünülmektedir (Gibson, 2012).

Algoritmik düşünme, algoritmaların oluşturulması ve anlaşılmasıyla bağlantılı bir tür yetenekler havuzudur (Futschek,2006):

- Verilen problemleri analiz etme yeteneği
- Bir problemi tam olarak belirleme yeteneği
- Verilen problem için uygun olan temel eylemleri bulma yeteneği
- Temel eylemleri kullanarak verilen probleme yönelik doğru bir algoritma oluşturma yeteneği
- Bir problemin tüm özel ve normal durumlarını düşünebilme yeteneği
- Bir algoritmanın verimliliğini artırma becerisi.

Algoritmik düşünmeyi öğrenmenin en başında öğrenme senaryosuna dahil olması gereken çok temel kavramları gözlemleyebiliriz (Futschek ve Moschitz,2011):

- Temel komutlar - Temel eylemler
- Komutların sırası
- Komutların alternatifleri (if)
- Komutların yinelemesi (loop)
- Soyutlama komutu (method)

Şahin ve Namlı (2017) tarafından yapılan araştırmada ilk kez algoritma eğitimi alan ortaokul öğrencilerine BDE ve drama yöntemleriyle verilen algoritma eğitimi sonrası problem çözme becerileri incelenmiş ve her iki yöntemde de eğitim sonucunda problem çözme becerilerinde olumlu yönde anlamlı farklılık bulunmuştur.

Özmen ve Altun (2014) yapmış oldukları araştırmada öğrencilerin programlama sürecinde yaşadıkları zorlukları programlama bilgisi, programlama becerisi, programın mantığını kavrama ve hata ayıklama olarak belirlemişlerdir ve öğrencilerin, programlamadaki başarısızlıklarının en büyük nedenlerini pratik eksikliği, algoritma oluşturmama ve bilgi eksikliği olduğunu vurgulamışlardır. Aynı araştırmada programlama deneyimi yüksek öğrenciler programlama başarısının algoritma oluşturmaya bağlı olduğunu belirtmişlerdir. Algoritma oluşturma programlama öğrenmeye başlamadan önce ilk olarak öğrenilmesi gereken bölümdür. Algoritma oluşturma süreci aslında vakit alan ve düşünmenin, tartışmanın en yoğun olduğu dolayısıyla programlama dili öğrenmeye başlayanların zor olarak nitelendirdikleri aşamadır.

Köse ve Tüfekçi (2015) yapmış oldukları araştırmada değerlendirme süreçleri sonucunda elde edilen bilgiler ışığında algoritma ve akış şeması kavramlarının öğretimi bağlamında etkili bir araç olarak kullanabilecekleri bir akıllı yazılım sistemi geliştirmişlerdir. Bu sistemin kullanılması ile öğrencilerin ileri düzeyde programlama konularına daha hazır düzeyde olacakları sonucuna varmışlardır.

Durak (2009), öğrencilere programlama mantığı ve algoritmaları öğretmede kolaylık sağlayacak bir öğretim materyali hazırlayarak bu materyalin öğrencilerin başarıları üzerindeki etkisini araştırdığı çalışmasında, programın kullanımının kolay, faydalı ve öğrenmede etkili olduğu sonuçlarına varmıştır.

Oluk ve diğerleri(2018) tarafından yapılan araştırmada Scratch kullanılarak deney grubuna verilen algoritma eğitimi sonucunda öğrencilerin algoritma geliştirme ve bilgi işlemsel düşünme becerilerinin, mevcut öğretim programı kullanılarak kontrol grubuna verilen algoritma eğitimi sonuçlarına göre anlamlı derecede daha fazla yükseldiği görülmüştür.

Grover, Pea ve Cooper (2015) tarafından yapılan çalışmada, ortaokul bilgisayar bilimi dersi ile ilgili “Bilişimsel Düşünmeyi Geliştirmek İçin Temeller (FACT)” isimli

bir kurs oluşturulmuş, bu kurs ile öğrencilerin algoritmik problem çözme ile gelecekteki etkileşim için hazırlanmaları ve motive edilmeleri amaçlanmıştır. Hazırlanan kurs 7 hafta boyunca Kuzey Kaliforniya bölgesinde yer alan 54 ortaokul (6,7 ve 8.sınıf) öğrencisine uygulanmıştır. Araştırmanın sonucunda, öğrencilerin algoritmik düşünme becerilerinde önemli kazanımlar elde ettikleri ve Stratch programından elde ettikleri öğrenmelerini metin tabanlı bir programlamaya transfer edebildikleri görülmüştür.

Eğitimde Programlama

Programlama, bilgisayarın donanıma nasıl davranacağını anlatan, bilgisayara yön veren komutlar ve işlemler bütünüdür (Gülbahar, Kalelioğlu ve Karataş,2017). Eğitimde programlama kavramı son yıllarda tüm ülkelerin gündeminde olan bir konu olmakla birlikte, küçük yaşlardan itibaren her seviyede zorunlu bir ders olarak eğitim öğretim programlarına dahil edilmeye başlanmıştır. Programlamanın öğrenciler için karmaşık ve zor bir süreç olarak görülmesi nedeniyle öğrencilere programlamanın temelini, genel mantık kurallarını, algoritmasını öğretmeye yönelik görsel blok tabanlı programlama araçları geliştirilmiştir. Özellikle yeni başlayan öğrencilerde için sürece dair olumlu izlenimler yaratmıştır (Çatlak, Tekdal ve Baz ,2015; Calder 2010; Lai ve Yang, 2012; Ching-San ve Ming-Horng, 2012).

Programlama ile program çözme süreçleri birbirine benzer özellikler taşımaktadır. Bu konuda Polya'nın (1957) problem çözme adımları Pea and Kurland (1987) tarafından programlamanın aşamalarıyla aşağıdaki gibi eşleştirilmiştir (Çakıroğlu, Sarı, Akkan, 2011):

<u>Problem Çözme</u>		<u>Programlama</u>
Problemleri anlama	—————→	Problemi anlama
Çözüm için plan yapma	—————→	Programı yazma
Planı uygulama	—————→	Programı koşturma
Çözümü değerlendirme	—————→	Programı düzeltme ve tamamlama

Çalışmalarla önemi kavranmaya başlanan programlama eğitiminin faydaları şu şekilde listelenmiştir (Akpınar ve Altun, 2014):

- Öğrenciler okulda sürekli olarak bu araçları kullanıp dijital okur yazarlıklarını geliştirebilirler,
- Okula ve derslere olan motivasyonları arttırılabilir,
- Öğrencilerin problem çözme ve analitik düşünme becerileri,
- Uzamsal düşünme becerileri,
- Ürüne dönük büyük projeler yapma, küçük projelerin entegrasyonu ile karmaşık problemlere çözüm üretme alışkanlığı,
- İşbirlikli çalışma ve öğrenme becerileri
- Yapararak öğrenme ve bilgisayara öğretmek öğrenme alışkanlıkları ve kültürü, geliştirilebilir.

Liao ve Bright (1991) programlamanın bilişsel sonuçlarını inceleyen altmış beş çalışmayı değerlendirerek bir meta analiz çalışması yapmışlardır. İncelenen çalışmaların büyük bir kısmında programlamanın problem çözme yeteneğini geliştirmeye olumlu katkılar sağladığını belirtmişlerdir (Çakıroğlu, Sarı, Akkan, 2011).

Akçay ve Çoklar (2016), yapmış oldukları çalışmada programlama becerilerinin, eleştirel düşünme, algoritmik düşünme, analitik düşünme, problem çözme, çok yönlü düşünme, sorgulama gibi becerileri geliştirdiğini belirtmişlerdir.

Ramalingam, LaBelle ve Wiedenbeck (2004), öğrencilerin öz yeterlilik ve zihinsel modellerinin programlama öğrenmedeki etkisini araştırmışlardır. Araştırmada, programlama için öz yeterliliğin önceki programlama deneyimlerinden etkilendiği ve öğrencinin devam ettiği bir programlama kursu aracılığıyla geliştiği, öğrencinin zihinsel modelinin öz yeterliliği etkilediği, güçlü bir zihinsel model geliştirmenin öz yeterlik duygularını arttırdığı ve hem zihinsel model hem de öz yeterliliğin öğrencinin ders performansını etkilediği sonucuna varmışlardır.

Kurt ve Erol (2017) BÖTE bölümü öğrencilerinin programlamaya ilişkin tutumlarını inceledikleri araştırmalarında öğrencileri programlama sürecinde zorlayan problem analizinde yaşanan sıkıntılar, kod yazma zorluğu ve yapıları nerede kullanacağını karıştırma gibi etmenler olduğuna ilişkin bir sonuca ulaşmışlardır. Öğrencilerin, programlama mantığını kavramakla ilgili yaşadıkları sıkıntının ise

algoritma öğretimi sürecinde uygulama olmaması, kısa süre ayrılması ve hazırbulunuşluk düzeyine dikkat edilmemesinden kaynaklı olduğu görülmüştür. Katılımcılar algoritma öğretiminden beklentilerini ise öğretim sürecinin daha görsel ve uygulamalı olması şeklinde belirtmişlerdir.

Nowaczyk (1983), çalışmasında matematik ve ingilizce derslerindeki geçmiş akademik başarıları, önceki bilgisayar kullanım deneyimleri, derste beklenen not, mantık ve cebirsel kelime problemleri üzerine performansları ile programlama dersi performansları arasında anlamlı bir ilişki olduğunu belirtmiştir. Programlama dillerinin İngilizce olması, bulanık kaynakların dilinin ingilizce olması gibi nedenlerle ingilizce dersindeki akademik başarı ile programlama başarısının birbirini etkileyebileceği düşünülmektedir.

Programlamanın özellikle görsel programlamanın öğrencilerin problem çözme becerisine katkı sağlayıp motivasyonlarını arttırdığı gözlemlenmiştir (Calder 2010; Lai ve Yang, 2012, Akt:Şahin ve Namlı,2017). Örneğin, Calder (2010) öğrencilere programlama dili olarak Scratch kullanarak gözlem ve görüşmeler yapmış, öğrencilerin yazdığı ilerleme ve yansımalarını dile getiren blog yazılarını incelemiştir. Araştırma sonucunda, Scratch yazılımının problem çözme için ilgili çekici ve kullanımı nispeten daha kolay bir alan olduğu, matematiksel kavramların anlaşılmasında verimli ve motive edici bir programlama ortamı sağladığı görülmüştür.

Çatlak, Tekdal ve Baz (2015), tarafından yapılan çalışmada Scratch programlama dili ile gerçekleştirilen bilimsel çalışmalar incelenmiştir. İncelenen çalışmalarda Scratch programının algoritma ve programlama eğitime başlangıç aşamasında verilebileceği, çoklu ortam öğeleri ile desteklenerek soyut kavramların daha kolay öğrenilmesini kavramlar arasındaki ilişkisel bağların daha güçlü kurulmasını sağlayabileceğine dair bulgulara ulaşıldığı belirtilmiştir.

Akyüz (2018), programlamanın basit parçaları bir araya getirerek karmaşık problemlere çözümler üretmeyi sağlayan yap-boz benzeri faydalı zihinsel bir etkinlik olduğunu ve bu yönüyle programlamanın çocuklarımızın parça-bütün ilişkilerini öğrenmelerine, detaylara özen göstermelerine, düşüncelerini organize ve ifade etmelerine ve genel olarak zekâlarının gelişmesine katkıda bulunacağını ifade etmiştir. Çocukluk döneminden itibaren verilecek programlama eğitimi, düşünme becerilerini olumlu yönde etkileyebileceği gibi, süreç ve ürün boyutlarında ayrı ayrı ele alındığında

öğrenme sürecinde öz-düzenlemeye, işbirlikçi öğrenmeye ve keşfetmeye dayalı yapılandırmacı yaklaşıma paralel bir öğrenme altyapısı oluşturulmasına katkıda bulunabilecektir (Kert ve Uğraş, 2009). Erken yaşta kod yazmaya başlayan bireyler, ilerleyen yaşlarda kendi proje ve uygulamaları tasarlayabilecek ve hayata geçirebileceklerdir (Demirer ve Sak,2016).

Okul Öncesinde Programlama

Günümüzde çocukların bilişim teknolojileri ile karşılaşma, tanışma, kullanma yaşı çoğu zaman okul hayatına başlamadan gerçekleşmektedir. Amerikan pediatri akademisi (APA), 2-5 yaş arası çocuklar için ekran kullanımını günde 1 saat ile sınırlandırılmasını, ebeveynlerin çocukların neler gördüklerini anlamalarına yardımcı olmak için çocuklarıyla birlikte medyayı görmelerini tavsiye etmektedir. Ayrıca, 6 yaş ve üstü çocuklar için medya türlerine ve medya kullanarak harcadıkları süreye bağlı olarak limitler konulması ve medyanın sağlık için gerekli olan yeterli uyku, fiziksel aktivite ve diğer davranışların yerini almadığından emin olunması ile ilgili uyarılarda bulunmaktadır. Okul öncesi ve okul çağı çocuklarında teknolojik araçların kullanımı, fiziksel, zihinsel ve psikolojik ve sağlık problemlerini ortaya çıkarmaktadır. Bunlar, obezite, tembellik, uykuculuk, saldırganlık ve dikkat eksikliği olarak karşılaşılan problemlerdir (akt.Sayan,2016; Nunez-Smith, Wolf, Huang, Emanuel, & Gross, 2008). NAEYCF ise teknolojik araçların çocuklardaki olumlu etkilerinin dil ve kelime hazinesinde gelişmeler, mantık matematiksel anlayış, problem çözme becerileri, kendini yönetme, ve sosyal becerilerde gelişmeyi sağlama olarak belirlemektedir (NAEYCF, 2012).

Okul öncesi eğitim hayatı başladığında çocukları bu teknolojilerle “doğru” tanıştırmak, bilinçli, güvenli, etkili ve verimli bir şekilde kullanmalarını sağlamak eğitim öğretim alanında yer alan herkesin sorumluluğu olarak düşünülmelidir. Bu durumu okul öncesi dönemde kazandırılan alışkanlıklar, bilişsel düşünme, problem çözme, işbirlikli öğrenme, yaratıcı düşünme becerilerinin hayatlarını etkileyeceği düşüncesi ile bir araya getirerek uygun yöntem ve teknikler uygulanabilmektedir.

Logo dilinin çeşitli uygulamaları, ToonTalk, Squeak Etoys, Stagecast Creator, Microworlds JR ve Scratch gibi özellikle küçük çocuklar için tasarlanmış çeşitli

programlama ortamları vardır, bu programların özellikleri şunlardır (Fessakis, Gouli, Mavroudi,2013):

- Basitleştirilmiş söz dizimi
- Çocukların programları oluşturmak için sürükleyip bırakabilecekleri komut sembollerinin kullanımıyla programlama
- Komutların hemen yürütülmesi (derleme-bağlama aşamaları olmadan)
- Programlama paradigmaları metaforlar kullanırken (örneğin Logo'da çocuk doğrudan işlemciye değil bir kaplumbağaya komutları verir) ama Scratch programında yer alan sahneye çocuklar nesnelere, aktörler ve kuklaların her birini yerleştirip ayrı ayrı kodlaması.

Yazılım programlama ortamlarına ek olarak, çocukların fiziksel nesnelere hareket ettirmelerine izin veren sistemler de önerilmiştir. Bir ekran yerine fiziksel arayüzler kullanan somut tasarımlar küçük yaşlarda daha yaygın hale gelmektedir (Manches & O'Malley, 2012). Bu sistemlerden bazıları bir bilgisayarın kullanımını gerektirirken, diğerleri TORTIS sistemi (Perlman, 1976) ve Valiant Roamer Robot (Amethyst Consultancy Ltd, 2003) gibi bağımsız elektro mekanik aygıtlardır. Robotlar, mühendislik ve tasarım süreci, akış, döngüler ve parametreler, sensörler ve dallar Bers tarafından erken öğrenme için güçlü fikirler olarak tanımlanmaktadır (Bers,2010). Robotik, öğrencilere anlamlı bir yaratım tasarlamak, inşa etmek ve programlamak için eşsiz bir öğrenme fırsatı sunmaktadır (Cejka, Rogers, & Portsmore, 2006). Popüler sınıf kaynakları yaklaşık 30 yıl önceki orijinal Logo zemin kaplumbağasının yinelemelerini temsil eden zemin robotlarını içermekte (Beebot, BigTrack ve Pixie gibi) ancak daha yenilikçi yaklaşımlar ortaya çıkmaya başlamaktadır. Örneğin, Cubetto, kesilmiş şekiller içeren bir tahtaya yerleştirilmiş blokları kullanmaktadır. Dash ve Dot, 4 yaşından büyük çocukların basit talimatları girme etkilerini keşfetmelerini sağlayan oyuncak robotlardır. Robot Turtles, bir tahta etrafında bir kaplumbağanın gezinmesi için bir sosyal bağlam sağlayarak talimatlar oluşturmak için kullanılan 3 yaşından itibaren çocuklara programlama temellerini oyun kartları ile öğreten dijital olmayan bir tahta oyunudur (Manches and Plowman,2017). TangibleK Robotik programı, çocukların robot yapmak ve davranışlarını programlamak için kullanabilecekleri araştırma tabanlı, gelişimsel olarak uygun bir robot seti içeren, robotları çocukların sayısal düşünme becerilerini geliştirmek ve mühendislik tasarım süreci hakkında bilgi edinmek için bir araç olarak kullanmaktadır (Bers,2010). Son araştırmalar, 4 yaşındaki çocukların bilgisayar

programlamanın temel kavramlarını anlayabileceklerini ve basit robotlar oluşturabildiğini göstermiştir (akt.Bers,2010; Bers, Ponte, Juelich, Viera, & Schenker, 2002; Cejka, Rogers, & Portsmore, 2006).

Okul öncesinde programlamanın sistematik bir şekilde kullanılması ile ilgili kritik bir nokta, geliştirmeye uygun bilgisayar programlama ortamlarının mevcudiyeti değil, iyi bilgilendirilmiş ve hazır öğretmenler tarafından uygulanmış ve doğrulanmış ve her gün okul uygulamasına kolayca entegre edilebilecek uygun şekilde tasarlanmış öğrenme aktiviteleri ve destekleyici materyallerin geliştirilmesi olarak belirtilmiştir (Fessakis, Gouli, Mavroudi,2013).

Hamada (1986), erken yaşlarda programlama eğitimi alan öğrencilerin matematiksel bilgi ve problem çözme becerilerinde daha yüksek başarı gösterdiklerini belirtmektedir. 5-6 yaş arası anaokulu çocuklarının bilgisayar programlamasını kullanarak problem çözme boyutlarına ilişkin bir araştırmada, çocuklar, etkileşimli tahtada Logo tabanlı bir ortam kullanarak bir dizi benzer programlama problemini çözmeye yer almışlardır (Fessakis, Gouli, Mavroudi,2013). Araştırma sonucu ile, çocukların ilgi çekici öğrenme etkinliklerinden zevk aldıkları ve matematiksel kavramları, problem çözme ve sosyal becerileri geliştirme fırsatlarına sahip oldukları görüşünü desteklenmiştir.

Clement ve Gullo(1984), programlamayı öğrenmenin çocukların bilişsel tarzı, üstbilişsel yeteneği, bilişsel gelişim ve yönergeleri tanımlama yeteneği üzerindeki etkisini değerlendirmiştir. Bir gruba bilgisayar destekli öğretim, bir gruba programlama eğitimi verilmiş, programlama grubunun üstbilişsel yeteneği ve yönerge tanımlama yetenekleri açısından daha yüksek performans sergiledikleri sonucuna ulaşmıştır.

Baz (2018) tarafından yapılan araştırmada günümüzde tasarlanan kodlama yazılımlarının küçük yaştaki çocukların programlama dillerinin karmaşık kod yapılarını öğrenmelerinin önüne geçerek, eğlenceli şekilde uygulamalar yapabilmelerini sağlamakta, kodlamayı doğrudan öğretmek yerine, kodlama araçlarını kullanan çocukların motivasyonlarını arttırmaya yönelik olduğu belirtilmiştir.

Okul Öncesinde Kullanılan Kodlama Siteleri ve Uygulamaları

Baz (2018), yapmış olduđu çalışmada çocuklar için kodlama yazılımlarının karşılaştırılmasına ilişkin bulguları paylaştığı tabloda yaş gruplarına göre kodlama yazılımlarını vermiştir. Bu bölümde tabloda yer alan 5-7 yaş aralığı için uygun olan kodlama uygulamaları incelenmiştir.



	<i>Kodlama Yazılımı</i>	<i>Web Adresi</i>	<i>Desteklediği İşletim Sistemi</i>	<i>Yaş Aralığı</i>	<i>Ücret Durumu</i>	<i>Dil Desteği</i>	<i>Mobil Uyum</i>	<i>Örnek Projeler</i>	<i>Sosyal Medya /Blog</i>	<i>Yardım Desteği</i>
1	Scratch	www.scratchjr.org/	Mac, Linux, Windows	5-7, 7-11	Ücretsiz	✓	✓	✓	✓	✓
2	code.org	https://code.org/	Mac, Linux, Windows	5-7, 7-11, 11-üzeri	Ücretsiz	✓	✓	✓	✓	✓
3	Kodable	www.kodable.com	Mac, Linux, Windows	5-7	Ücretli	✗	✓	✓	✓	✓
4	The Foos	http://thefoos.com/	Mac, Linux, Windows	5-7	Ücretli	✓	✓	✓	✓	✓
5	Tynker	https://www.tynker.com/	Mac, Linux, Windows	5-7	Ücretli	..	✓	✓	✓	..
6	Box Island	https://boxisland.io/	Mac, Linux, Windows	5-7	Ücretli	✓	✓	✓	✓	✓
7	Cargo Bot	https://itunes.apple.com/tr/app/cargo-bot	iOS	5-7	Ücretsiz	✗	✓	✗	✗	✗
8	Daisy Dinosaur	https://itunes.apple.com/us/app/daisy-the-dinosaur/	iOS	5-7	Ücretsiz	✗	✓	✗	✗	✗
9	Blockly	https://developers.google.com/blockly/	Mac, Linux, Windows, iOS, Android	5-7	Ücretsiz	✓	✓	✓	✓	✓
10	Move the Turtle	http://movetheturtle.com/	iOS	5-7	Ücretli	✓	✓	✗	✗	✗

Tablo 8. Baz(2018), Çocuklar için kodlama yazılımlarının karşılaştırmalı incelemesi

ScratchJr

5-7 yaş arası çocuklar için geliştirilmiş uygulamanın kendi etkileşimli öykülerini ve oyunlarını programlarken yeni becerileri öğrendiği belirtilmektedir. Blok programlama mantığı ile devam eden blokların birbirine eklenmesi ile karakterlere hareket, dans, zıplama ve şarkı söyleme gibi eylemlerin yaptırıldığı bir uygulamadır. Scratch dilinden esinlenerek daha küçük yaştaki çocuklar için yeniden tasarlandığı belirtilmektedir. İnternet üzerinden online kullanılabilceği gibi telefon tablet uygulamaları ile de ücretsiz kullanılabilir (Şekil 14).



Şekil 14. ScratchJr



Şekil 15. ScratchJr Ekran Görüntüleri

Code.org

Her yaş grubunda farklı kurs planları tasarlanmış online bir kodlama sistemidir. Yaş grubuna uygun kursu seçtikten sonra o kursta yer alan dersler ve yapılacak etkinlikler listelenir (Şekil 16). 4-6 yaş grubunda önce bilgisayarsız etkinliklerle dersler basit algoritmalar ile başlamakta daha sonra sürükle-bırak, parça birleştirme ve kod bloklarına geçiş şeklinde dersler devam etmektedir (Şekil 17). Bunun dışında hesap oluşturarak hem kendi ilerlemenizi görebilir hem de ücretsiz olarak derslere ulaşabilirsiniz.

Code Studio'da Öğrenin
28 milyon öğrenci tarafından 21,955,669,726 satır kod yazıldı.
İlerlemenizi ve projelerinizi kaydetmek için bir hesap oluşturun. Ya da herhangi bir hesaba gereksinim duymadan direkt programlamaya başlayabilirsiniz. Bütün dersler ücretsiz olarak sunulmaktadır.

Hesap oluştur

Bilgisayar Bilimi Temelleri
Bilgisayar Bilimi öğrenmeye her yaş için olan bu 20 saatlik kurslarla başlayın.

Son kursunumu görüntüle >

Kurs 1
Okumaya yeni başlayanlar için 1. Dersten başlayın.
4-6 yaş arası

Kurs 2
Kurs 2 Bilgisayar Bilimlerine Giriş, okuma bilen öğrenciler için tasarlanmıştır.
6 yaş ve üstü (okuma becerisi gerektirir)

Kurs 3
Kendi oyunlarınızı kodlayıp, etkileşimli hikayeler oluştururken programlamanın derinlerine inin. Kurs 3, Kurs 2'ye devamdır.
8 yaş ve üstü (Ders 2 sonrası)

Kurs 4
Sunulan say döngüleri ve parametrel fonksiyonlar kullanarak daha karmaşık programlar oluşturun. Bu Kurs 3'ün devamıdır.
10 Yaş ve Üstü (Ders 3 sonrası)

Şekil 16. Kurslar ve etkinlikler listeleri

Kurs 1

Yeni okuyucular için Kurs 1 ile başlayın. Yeni okuyucuların bilgisayar programı oluşturmalarında yardımcı olması, problem çözmedeki becerilerini geliştirmeleri, zor görevleri yapabilmeleri ve başkaları ile birlikte çalışabilmeleri için tasarlanmıştır. Dersin sonunda, öğrenciler, paylaşabilecekleri kendi farklı hikayelerini veya oyunlarını oluşturabilecekler. Anaokulu ve 1. sınıfları için önerilir.

Sıradaki denet | Yardım Al

Ders adı	İlerleme Durumu
1. Mutlu Haritalar	Bilgisayarsız Etkinlik 1
2. Oynat, Oynat	Bilgisayarsız Etkinlik 1
3. Yapboz: Nasıl sürükleyip bira...	1 2 3 4 5 6 7 8 9 10 11 12
4. Labirent: Sıra	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
5. Labirent: Hata Ayıklama	1 2 3 4 5 6 7 8 9 10 11 12
6. Gerçek-yaşam Algoritmaları: ...	Bilgisayarsız Etkinlik 1 2

Şekil 17. Kurs 1 Dersleri

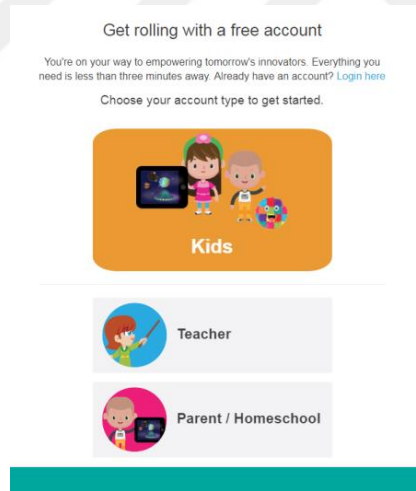
Kodable

Kodable, tüm kademeler için Bilgisayar Bilimi Standartları;

- Bilişimsel Düşünme,
- Programlama ve İletişim,
- Programlama Etkisi,
- Sosyal Duygusal Öğrenme,
- Günlük Bağlantılar,
- Programlama ve Ben, olmak üzere altı grupta belirlenerek ve ilerlemeler bu standartlar doğrultusunda gerçekleştirilmektedir.

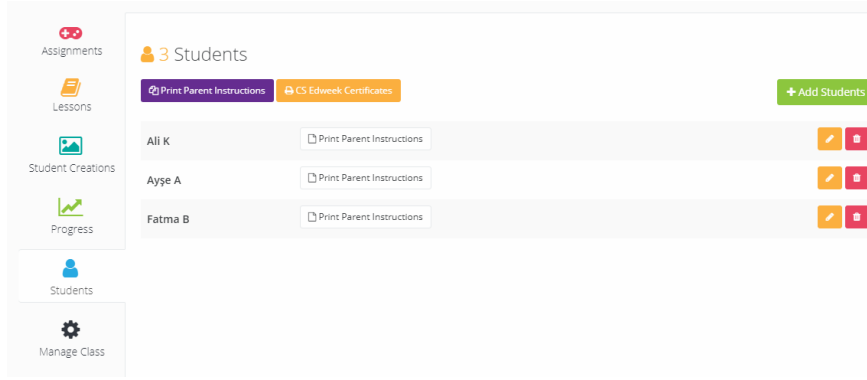
Okul Öncesi Bilgisayar Bilimi Standartlarını “Makineleri kontrol etmek için kullanıcıların kullandıkları dili tanımlama” olarak genel bir cümle ile açıklamaktadır.

Öğrenci, öğretmen ya da aile olarak ücretsiz hesap oluşturup, oturum açtıktan sonra sürece dahil olunan ve ilerlemelerin takip edilebileceği bir sistemleri bulunmaktadır (Şekil 18).



Şekil 18. Kodable Kullanıcı Seçimi

Öğretmen olarak hesap açıldıktan sonra öğrenciler kaydedilerek, dersler kademeli olarak atanarak süreç ilerleyebilmektedir. Okulda ders içinde kullanılabileceği gibi okul dışında da kullanılabilecek ve öğretmen tarafından kontrol edilebilecek bir sistemleri bulunmaktadır (Şekil 19-20).

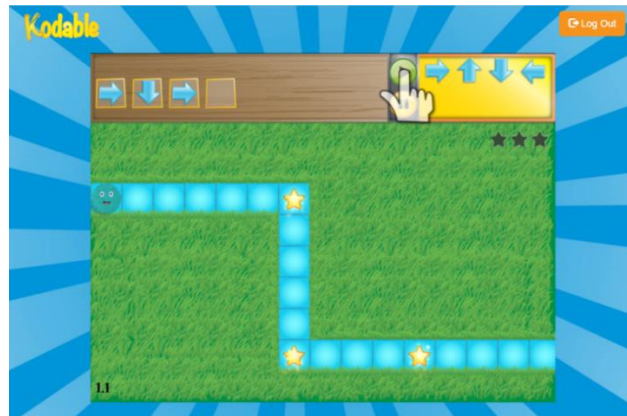


Şekil 19. Kodable Öğrenci Ekleme



Şekil 20. Kodable Öğretmen Kontrol Ekranı

Öğrenci girişi yapıldıktan sonra ilk aşama ile program başlamaktadır. Belirtilen yolu izlemesi için komutları sıralaması ve çalıştır düğmesine tıklaması yeterlidir (Şekil 21). Öğrencilerin sayıları bilmiyor olmaması sebebiyle sadece oklarda yön tayin etmeleri beklenmektedir.



Şekil 21. Kodable Komut Sıralama ve Çalıştırma Ekranı

Kids Koding with The Foes

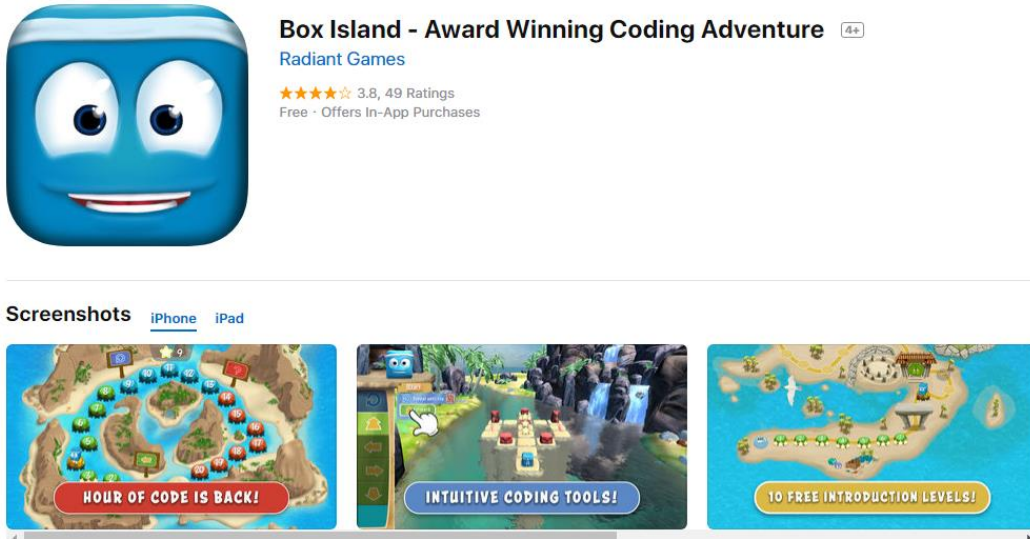
4-9 yaş için codeSpark Academy tarafından hazırlanan tamamı ebeveynler tarafından izlenebilir, her biri bilgisayar programlamanın temeli olan yapbozlar, bulmacalar, oyunlar, adım adım yaratıcı projeler, oyun tasarımı gibi çeşitli interaktif öğrenme etkinlikleri sunan “kelimesiz” denilen bir arayüz kullanmaktadır. codeSpark Academy, Scratch ve Alice gibi görsel programlama dillerinden, Logo, SmallTalk ve Squeak gibi diğer başlangıç dillerinden esinlenmiştir. Uygulama ile örüntü tanıma, problem çözme, sıralama, algoritmik düşünme, hata ayıklama, döngüler ve şartlar gibi temel bilgisayar bilimi kavramlarına hakim olunacağı belirtilmektedir. (codeSpark,2018).



Şekil 22. Kids Koding with The Foes Ekran Görüntüleri

Box Island

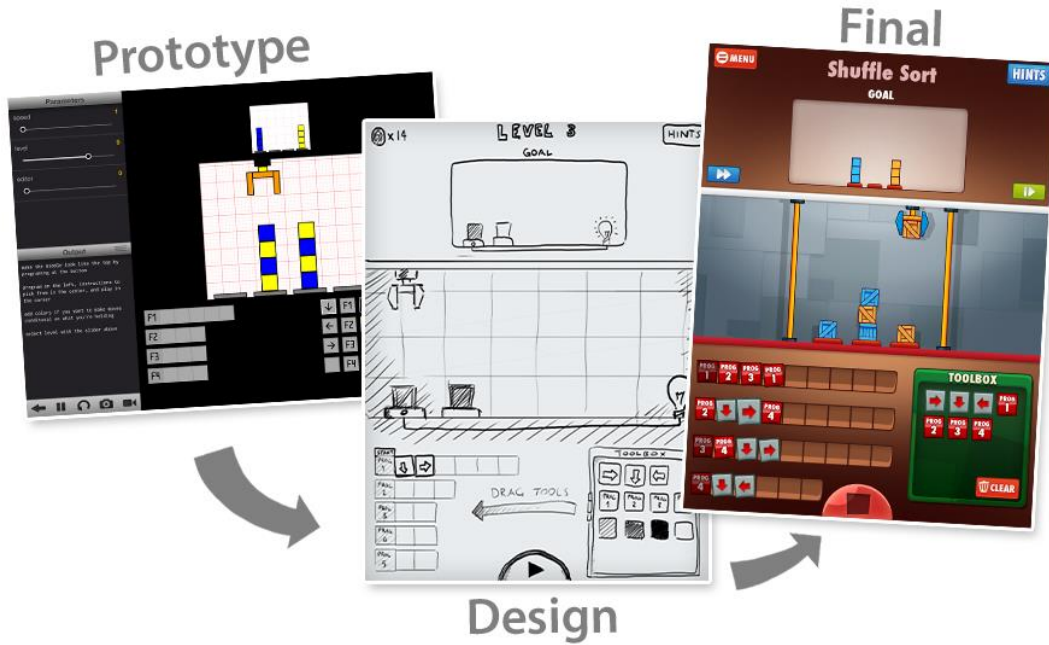
Kodlamanın temellerini öğretirken çocukları heyecan verici bir maceraya götüren ilk on seviyesi ücretsiz toplamda yüz seviyeden oluşan bir mobil oyun olarak tanımlanmaktadır. Çocuklar döngüler ve şartlar gibi temel bilgilerini kullanarak oyuna devam etmektedirler. Oyunun genel tanımlamaları kısmında 4 yaş ve üzeri denilmesine rağmen ayrıntılarda 6 yaş ve üzeri için uygun olduğu belirtilmektedir.



Şekil 23. Box Island Ekran Görüntüleri

Cargo-Bot

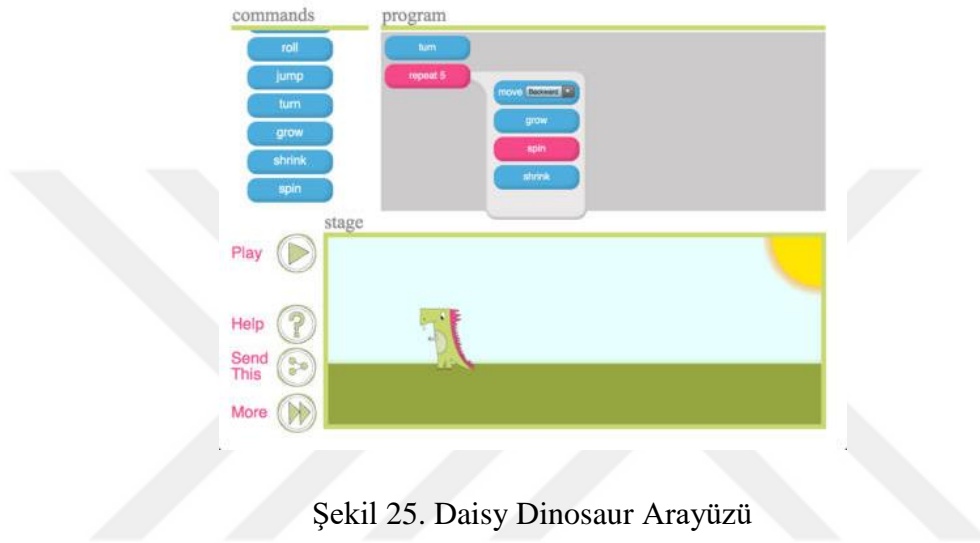
Bir robotun sandıkları nasıl taşıyacağını öğrettiği bir bulmaca oyunu olarak Codea kullanılarak tamamen iPad üzerinde tasarlanmıştır (Şekil 24).



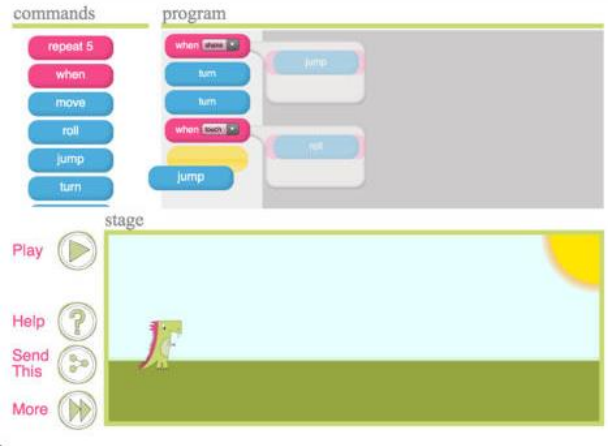
Şekil 24.Cargo Bot Tasarım Aşaması

Daisy Dinosaur

Daisy adındaki dinozoru sürükle bırak yöntemi ile kullanılan komutlar aracılığıyla hareket ettirebilen her yaştan çocuğun kullanabileceği bir ara yüze sahip olduğunu belirtmektedir. Bu uygulama 5-7 yaş arası çocuklar için kodlamanın temellerini tanıtmaktadır. Çocukların bu uygulamanın zorluklarını çözerek, nesnelerin, sıralamaların, döngülerin, olayların temellerini sezgisel olarak kavrayacaklarını belirtmektedir.



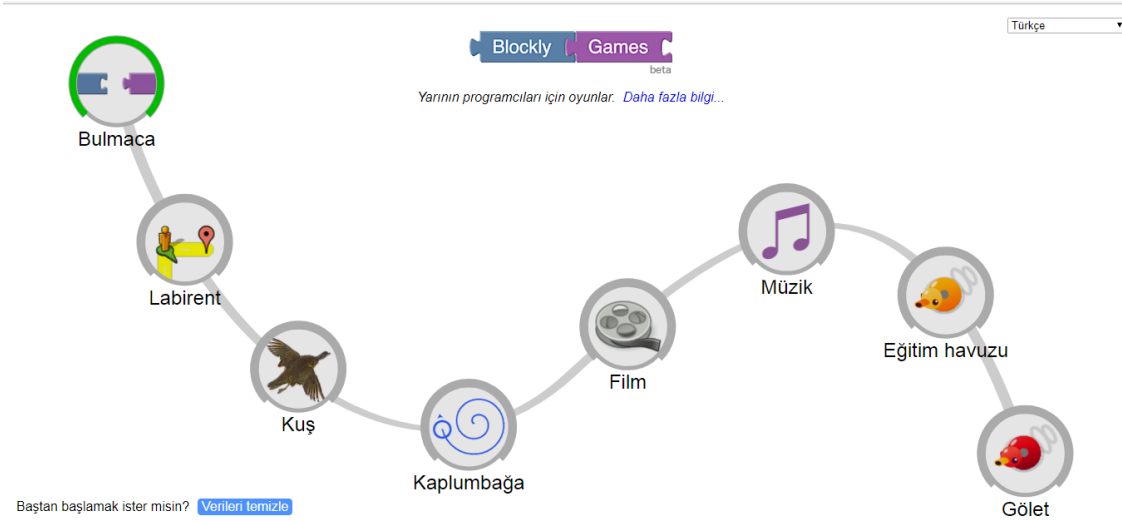
Şekil 25. Daisy Dinosaur Arayüzü



Şekil 26. Daisy Dinosaur Sürükle-Bırak

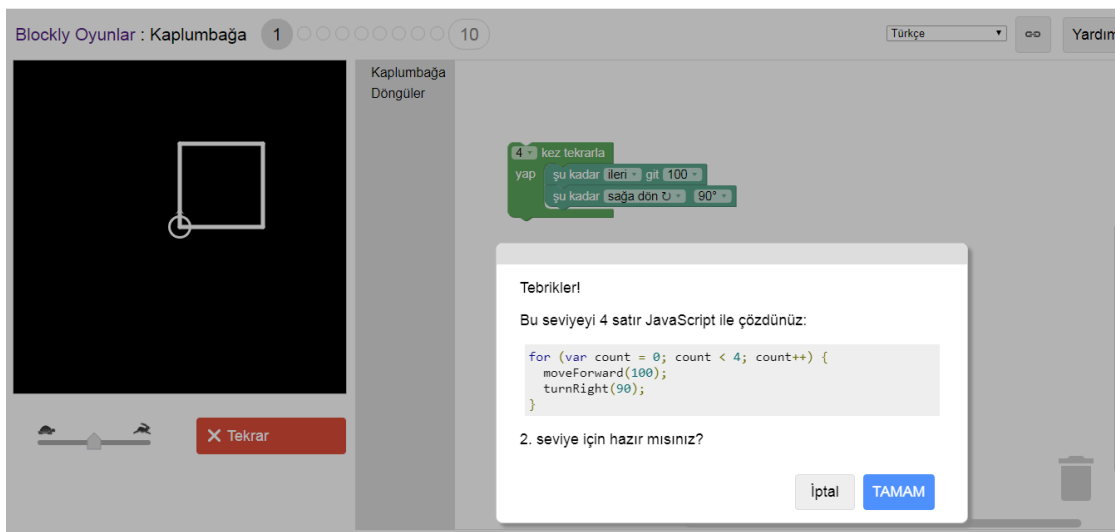
Blockly

Blockly geliştiriciler için, Blockly uygulamaları öğrencilerin kullanımı için uygun olduğu belirtilmektedir. Blockly, deneyimli geliştiricilere yönelik karmaşık bir kütüphanedir.



Şekil 27. Blockly Games Ekranı

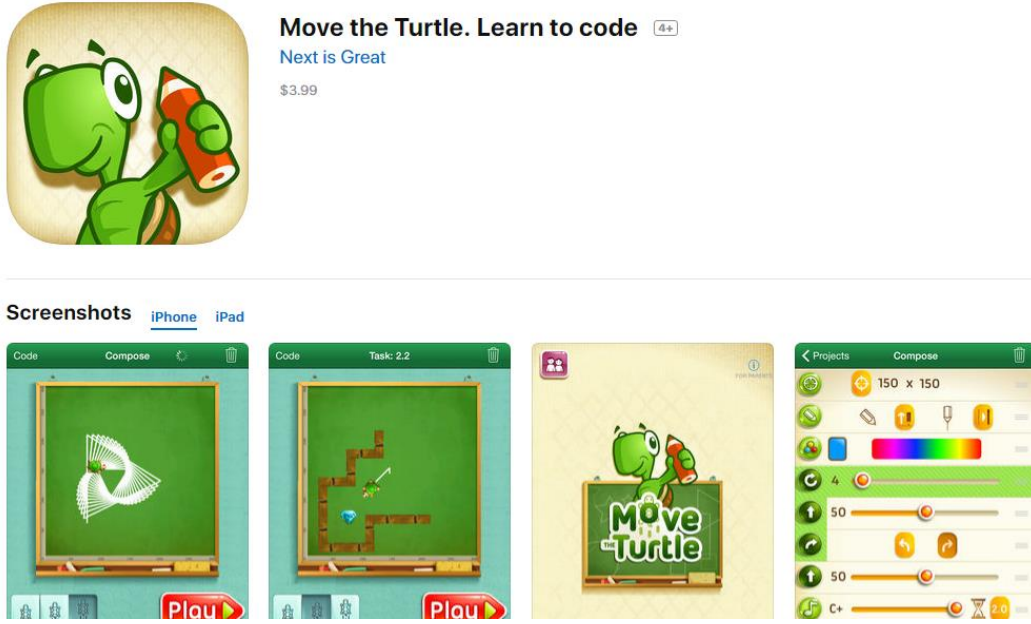
Verilen bir problemi nasıl çözebileceği düşündürülüp uygulamak için kodlama bloklarından yararlanması beklenmektedir. Şekil 28'de olduğu gibi kodlama blokları ile problemin çözümü doğru bir şekilde tamamlandığında programlama dillerinden karşılıkları da verilerek sonraki adıma geçebilecekleri bildirilmektedir. Hatalı bir çözüm yaptığında tekrar denemesi beklenmektedir.



Şekil 28. Blockly Problem Çözümü

Move the Turtle. Learn to Code

Adından da anlaşıldığı üzere amacı kaplumbağa karakterini hareket ettirmek olarak tasarlanmıştır. Çocuklara sezgisel grafik komutları kullanarak bilgisayar programları oluşturmanın temellerini öğreten eğitici bir uygulama olarak tanımlanmaktadır. Aynı zamanda bu uygulama ile çocukların döngüler, prosedürler, değişkenler ve koşullu talimatlar gibi kavramlara aşina olacaklarını belirtilmektedir.



Şekil 29. Move the Turtle. Learn to Code

BÖLÜM IV

BULGULAR VE YORUMLAR

Okul öncesi eğitime devam eden beş yaş grubu öğrencilerine verilecek algoritma ve kodlama eğitiminin çocukların problem çözme becerilerine etkisinin saptanması amacı ile yapılan araştırmada elde edilen bulgular ve bu bulgulara ait yorumlara bu bölümde yer verilmiştir. Kullanılan veri toplama araçlarına ait bulgular gruplandırılarak verilmiştir.

Problem Çözme Becerisi Ölçeği Analizinden Elde Edilen Bulgular

Deney ve kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test puanlarına ilişkin t testi sonuçları Tablo 8'de verilmiştir.

Tablo 9. Deney ve kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test puanlarına ilişkin t testi sonuçları

Grup	N	X	Ss	Sd	t	p
Deney	14	38,29	7,23	27	0,228	0,687
Kontrol	15	37,60	8,81			

Tablo 9'da görüldüğü gibi, deney grubundaki öğrencilerin ön test puan ortalamalarının kontrol grubu öğrencilerinin ön test puan ortalamalarından yüksek olduğu bulunmuştur. Ancak grupların ön test puanları arasındaki ilişkinin $p > 0.05$ düzeyinde anlamlı olmadığı ($p = 0,687$) gözlenmiştir.. Bu bulgulara göre iki grup arasında anlamlı bir fark olmadığı, rastgele seçilen bu grupların aynı seviyede olduğu söylenebilir.

Kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test ve son test puanlarına ait t-testi sonuçları Tablo 10'da verilmiştir.

Tablo 10. Kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları ön test ve son test puanlarının t-testi sonuçları

Ölçüm	N	X	Ss	Sd	t	p
Ön test	15	37,60	8,81	14	-0,91	0,378
Son test	15	39,00	8,57			

Tablo 10’da görüldüğü gibi, kontrol grubundaki öğrencilerin son test puan ortalamalarının ön test puan ortalamalarından yüksek olduğu ancak anlamlılık değeri ($p=0,378$) $p>0.05$ olduğundan ön test ve son test puanları arasındaki farkın istatistiksel olarak anlamlı olmadığı söylenebilir.

Deney grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği’nden aldıkları ön test ve son test puanlarına ait t-testi sonuçları Tablo 11’de verilmiştir.

Tablo 11. Deney grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği’nden aldıkları ön test ve son test puanlarının t-testi sonuçları

Ölçüm	N	X	Ss	Sd	t	p
Ön test	14	38,29	7,23	13	-2,55	0,024
Son test	14	42,21	6,57			

Tablo 11’de görüldüğü gibi, deney grubundaki öğrencilerin son test puan ortalamalarının ön test puan ortalamalarından yüksek olduğunu, anlamlılık değeri ($p=0,024$) $p<0.05$ olduğundan ön test ve son test puanları arasındaki fark istatistiksel olarak anlamlıdır. Algoritma eğitimi alan öğrencilerin problem çözme becerileri, eğitim verilmeyen kontrol grubuna göre anlamlı düzeyde artmıştır. Bu durum, algoritma eğitiminin deney grubundaki öğrencilerin problem çözme becerileri üzerinde etkili olduğunu ve araştırmanın amacına ulaştığını göstermektedir.

Alanyazında bu bulguları destekleyen birçok çalışma bulunmaktadır. Mittermeier (2013), okuma yazmaya başvurmadan basit algoritmaları keşfetmesi için anaokulu gruplarıyla yaptığı çalışmada, 4-6 yaş aralığındaki çocukların algoritma geliştirme kapasitesine sahip olduklarını, geliştirdikleri algoritmayı sözlü olarak ifade edebildiklerini ve farklı çözümler ile algoritmalarını verimli bir şekilde iyileştirme veya revize etmek için ipuçlarını uygulayabildiklerini belirtmiştir.

Gibson (2012), 5-17 yaşlarındaki okul çocuklarına grafik teorisi ve algoritmalar öğretme konusundaki deneyimlerini bildirdiği çalışmada algoritmaların oyunlara veya bulmacalara olası çözümler sunduğu probleme dayalı öğrenmenin önemine değinerek 5-11 yaş arasındaki çocukların algoritma ve hesaplamayı öğrenme konusunda oldukça yüksek potansiyele sahip olduklarını belirtmiştir.

Özmen ve Altun (2014) yapmış oldukları araştırmada öğrencilerin programlamadaki başarısızlıklarının en büyük nedenlerini pratik eksikliği, algoritma oluşturmama ve bilgi eksikliği olduğunu vurgulamışlardır. Dolayısıyla bu çalışmada verilen algoritma eğitimi programlama eğitimi öncesinde atılan başarılı bir adım olarak değerlendirilebilir.

Tablo 12. Deney ve kontrol grubunda yer alan öğrencilerin Problem Çözme Becerisi Ölçeği'nden aldıkları son test puanlarına ilişkin t testi sonuçları

Grup	N	X	Ss	Sd	t	p
Deney	14	42,00	6,64	27	1,048	0,327
Kontrol	15	39,00	8,56			

Tablo 12'de görüldüğü gibi, deney grubundaki öğrencilerin son test puan ortalamalarının kontrol grubu öğrencilerinin son test puan ortalamalarından yüksek olduğu bulunmuştur. Ön test puanları ile karşılaştırıldığında her iki grupta da yükselme görülmüş olmakla birlikte deney grubundaki yükselişin daha yüksek olduğu görülmüştür. Bununla birlikte grupların son test puanları arasındaki ilişkinin $p>0.05$ düzeyinde anlamlı olmadığını ($p=0,327$) söyleyebiliriz. Bu bulgulara göre iki grup arasında son test puanları açısından anlamlı bir fark olmadığı, dolayısıyla verilen temel kodlama eğitimi ile öğrencilerin problem çözme becerilerine olumlu katkı sağladığını söyleyebiliriz. Alanyazında bu bulguları destekleyen birçok çalışma bulunmaktadır. DiSessa ve Abelson(1986), programlama ortamlarını, problem çözme süreci boyunca öğrencilerin düşüncelerini net ve özlü bir şekilde analiz etmelerini, organize etmelerini, ifade etmelerini ve değerlendirmelerini destekleyen güçlü yeniden yapılandırılabilir bilgi işleme ortamı olarak görmektedir (akt. Fessakis, Gouli ve Mavroudi,2013). Hamada (1986), erken yaşlarda programlama eğitimi alan öğrencilerin matematiksel bilgi ve problem çözme becerilerinde daha yüksek başarı gösterdiklerini belirtmektedir.

5-6 yaş arası anaokulu çocuklarının bilgisayar programlamasını kullanarak problem çözme boyutlarına ilişkin yapılan bir araştırmada çocukların ilgi çekici öğrenme etkinliklerinden zevk aldıkları ve matematiksel kavramları, problem çözme ve sosyal becerileri geliştirme fırsatlarına sahip oldukları görüşü desteklenmiştir (Fessakis, Gouli, Mavroudi,2013).

Eğer hatalı adım varsa çözüm yollarını kontrol eder.	Hatalı adımı hemen farkeder ve düzeltir.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	14	93
	Hatalı adımı fark eder ama yine aynı hatayı yapar.																	0	0
	Hatalı adım/adımları fark etmez.													X				1	7
Farklı çözüm yollarını denemede esnekler.	Farklı yolları dener.			X		X		X	X	X					X	X	7	47	
	Öğretmenden yardım isteyerek farklı yolları dener.	X	X								X	X	X	X			6	40	
	Farklı yolları denemekten çekinir.			X		X											2	13	
Problemi çözerken süreyi dikkatli kullanır.	Problemi en kısa şekilde çözümler.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	15	100	
	Problemi uzun yoldan çözümler.																0	0	
	Kısa yolu bilir ancak problemi uzun yoldan çözümler ister.																0	0	
Problemlerle uğraşırken duygusal tepkileri vardır.	Sakindir.		X	X		X	X	X	X	X	X		X	X	X	X	12	80	
	Heyecanlıdır.	X			X							X					3	20	
	Sinirlidir.																0	0	

Tablo 14. Gözlem Formu (Deney Grubu)

Okul Öncesi Öğrencilerine Yönelik Problem Çözme Gözlem Formu (Deney Grubu)																		
Maddeler		Öğrenciler															f	%
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Problemin ne olduğunu anlamaya çalışır.	Soru sorar			X	X			X			X					X	5	33
	Göz kontağı kurar.	X	X	X		X	X		X	X	X	X	X	X	X	12	80	
	Hiç ilgilenmez.															0	0	
Problem çözmeye isteklidir.	Problem çözmeye hemen başlar.	X	X	X		X	X		X	X	X		X	X	X	11	73	
	Yardım bekler.				X			X				X				4	27	
	Problemi çözmek istemez.															0	0	

Çözüm yolları konusunda yorum yapar.	<i>Kendi kendine düşünür.</i>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	15	100
	<i>Fikrini arkadaşlarıyla paylaşır.</i>																0	0
	<i>Fikrini öğretmeni ile paylaşır.</i>								X	X					X		3	20
Çözüm yoluna karar verir.	<i>Çözümü doğru uygular.</i>		X		X			X	X		X			X	X	X	8	53
	<i>Çözüm için öğretmeninden destek alır.</i>	X		X		X	X			X		X	X				7	47
	<i>Çözümü hatalı uygular.</i>																0	0
Eğer hatalı adım varsa çözüm yollarını kontrol eder.	<i>Hatalı adımı hemen farkeder ve düzeltir.</i>	X	X		X	X	X		X		X		X	X	X	X	11	73
	<i>Hatalı adımı fark eder ama yine aynı hatayı yapar.</i>			X													1	7
	<i>Hatalı adım/adımları fark etmez.</i>								X		X						2	13
Farklı çözüm yollarını denemede esnekler.	<i>Farklı yolları dener.</i>	X				X				X		X		X			5	33
	<i>Öğretmeninden yardım isteyerek farklı yolları dener.</i>				X		X	X	X	X		X		X			7	47
	<i>Farklı yolları denemekten çekinir.</i>	X	X													X	3	20
Problemi çözerken süreyi dikkatli kullanır.	<i>Problemi en kısa şekilde çözümler.</i>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	15	100
	<i>Problemi uzun yoldan çözümler.</i>																0	0
	<i>Kısa yolu bilir ancak problemi uzun yoldan çözümler ister.</i>																0	0
Problemlerle uğraşırken duygusal tepkileri vardır.	<i>Sakindir.</i>			X		X	X		X				X		X		6	40
	<i>Heyecanlıdır.</i>	X	X		X			X	X	X		X		X		X	9	60
	<i>Sinirlidir.</i>																0	0

Kontrol grubunda “Problemin ne olduğunu anlamaya çalışır” maddesine yönelik olarak 3 öğrencinin (%20) soru sorduğu, 7 öğrencinin (%47) göz kontağı kurduğu, 5 öğrencinin (%33) de hem soru sorduğu hem de göz kontağı kurduğu; deney grubunda ise 3 öğrencinin (%20) soru sorduğu, 10 öğrencinin (%67) göz kontağı kurduğu, 2

öğrencinin (%13) de hem soru sorduğu hem de göz kontağı kurduğu görülmüştür. Her iki grupta da problemle ilgilenmeyen öğrenci olmamıştır. Bu bulgulara göre öğrencilerin hem soru sorarak hem göz kontağı kurarak problemin ne olduğunu anlamaya çalıştıkları söylenebilir.

Kontrol grubunda “Problem çözmeye isteklidir” maddesine yönelik olarak 12 öğrencinin (%80) problem çözmeye hemen başladığı, 3 öğrencinin (%20) yardım beklediği; deney grubunda ise 11 öğrencinin (%73) problem çözmeye hemen başladığı, 4 öğrencinin (%27) yardım beklediği görülmüştür. Her iki grupta da problemi çözmek istememe durumu oluşmamıştır. Bu bulgulara göre öğrencilerin problem çözmeye istekli olduklarını söylenebilir. Clements ve Sarama'da (2003) belirtildiği gibi 1984'te 3 ve 4 yaşındaki çocuklarla yapılan bir çalışma sırasında Logo ortamında yaptıkları çizimler esnasında kağıt üzerine yaptıkları çizimlere göre daha fazla etkileşimde buldukları görülmüştür. Bu çalışmada da kullanılan kodlama ortamında öğrencilerin daha fazla etkileşimde bulunmak istemelerinden dolayı problem çözmeye istekli oldukları söylenebilir.

Kontrol grubunda “Çözüm yolları konusunda yorum yapar” maddesine yönelik olarak 11 öğrencinin (%73) kendine kendine düşündüğü, 3 öğrencinin (%20) fikrini öğretmeni ile paylaştığı, 1 öğrencinin (%7) hem kendi kendine düşündüğü hem de fikrini öğretmeni ile paylaştığı; deney grubunda ise 12 öğrencinin (%80) kendi kendine düşündüğü, 3 öğrencinin (%20) hem kendi kendine düşündüğü hem de fikrini öğretmeni ile paylaştığı görülmüştür. Her iki grupta da fikrini arkadaşlarıyla paylaşan öğrenci olmamıştır. Bu bulgulara göre öğrencilerin çözüm yolları konusunda kendi kendine düşündüğü ya/ya da öğretmeni ile fikir paylaşımında buldukları söylenebilir. Anaokulları ve küçük çocukları karşılaştıran bir araştırmada, Logo'nun anaokulları öğretmenleri arasındaki işbirliğine dayalı davranışı kolaylaştırdığı ve sosyal ve dil becerilerini geliştirdiği, aynı zamanda daha genç yaştaki çocukların çalışmalarına odaklandıkları da belirtilmiştir (Strand, 1986). Bu çalışmada da etkileşimde buldukları kişi/kişilerle sosyal ve dil becerilerini geliştirdikleri söylenebilir.

Kontrol grubunda “Çözüm yoluna karar verir” maddesine yönelik olarak 10 öğrencinin (%67) çözümü doğru uyguladığını, 5 öğrencinin (%33) çözüm için öğretmeninden destek aldığı; deney grubunda ise 8 öğrencinin (%53) çözümü doğru uyguladığını, 7 öğrencinin (%47) çözüm için öğretmeninden destek aldığı görülmüştür.

Her iki grupta da öğrencilerin çözümü hatalı uygulama davranışı gözlenmemiştir. Bu bulgulara göre öğrencinin çözüm yoluna karar verip doğru uyguladığı ya da öğretmeninden destek alarak çözümü uyguladığı söylenebilir. Akyüz (2018), programlamanın basit parçaları bir araya getirerek karmaşık problemlere çözümler üretmeyi sağlayan yap-boz benzeri faydalı zihinsel bir etkinlik olduğunu ve bu yönüyle programlamanın çocuklarımızın parça-bütün ilişkilerini öğrenmelerine, detaylara özen göstermelerine, düşüncelerini organize ve ifade etmelerine ve genel olarak zekâlarının gelişmesine katkıda bulunacağını ifade etmiştir. Bu çalışmada da çözüm yoluna karar verirken bilişimsel düşünme ve problem çözme süreçlerini takip ettiği söylenebilir.

Kontrol grubunda “Eğer hatalı adım varsa çözüm yollarını kontrol eder” maddesine yönelik olarak 14 öğrencinin (%93) hatalı adımı hemen fark edip düzelttiği, 1 öğrencinin (%7) hatalı adım/adımları fark etmediği; deney grubunda ise 11 öğrencinin (%73) hatalı adımı hemen fark edip düzelttiği, 1 öğrencinin hatalı adımı fark edip ancak yine aynı hatayı yaptığı, 2 öğrencinin (%13) hatalı adım/adımları fark etmediği görülmüştür. Bu bulgulara göre öğrencilerin hatalı adım varsa çözüm yollarını hemen fark edip düzeltmeye çalıştıkları söylenebilir. Sullivan ve Bers (2016) tarafından okul öncesinden 2.sınıfa kadar olan öğrencilerle yapılan çalışma, anaokulundan itibaren bir robotun programlanması ile ilgili temel kavramlara hakim olabildiğini ortaya koymaktadır.

Kontrol grubunda “Farklı çözüm yollarını denemede esnektir” maddesine yönelik olarak 7 öğrencinin (%47) farklı yolları denediği, 6 öğrencinin (%40) öğretmenden yardım isteyerek farklı yolları denediği, 2 öğrencinin (%13) farklı yolları denemekten çekindiği; deney grubunda ise 5 öğrencinin (33) farklı yolları denediği, 7 öğrencinin (%47) öğretmenden yardım isteyerek farklı yolları denediği, 3 öğrencinin (%20) farklı yolları denemekten çekindiği görülmüştür. Bu bulgulara göre kontrol grubunun farklı yolları denemekte daha esnek olduğu, deney grubunun ise öğretmenden yardım isteyerek farklı yolları denediği söylenebilir.

Kontrol grubunda ve deney grubunda “Problem çözerken süreyi dikkatli kullanır” maddesine yönelik olarak tüm öğrencilerin problemi en kısa şekilde çözümlendiği görülmüştür. Bu bulgulara göre öğrencilerin bir sonraki aşamaya geçmek adına problemi çözerken en kısa şekilde çözmek istedikleri söylenebilir.

Kontrol grubunda “Problemlerle uğraşırken duygusal tepkileri vardır” maddesine yönelik olarak 12 öğrencinin (%80) sakin olduğu, 3 öğrencinin (%20) heyecanlı olduğu; deney grubunda ise 6 öğrencinin (%40) sakin olduğu, 9 öğrencinin (%60) heyecanlı olduğu görülmüştür. Bu bulgulara göre problemle uğraşırken verdikleri duygusal tepkilere bakıldığında kontrol grubundaki öğrencilerin daha sakin deney grubundaki öğrencilerin daha heyecanlı olduğu söylenebilir.

Odak Grup Görüşmesi Analizinden Elde Edilen Bulgular

Temal:Osmo Teması

Araştırmaya katılan öğrenciler “OSMO deyince aklınıza ne geliyor?” sorusuna Osmo çikolatasının reklamı ya da kullanılan OSMO uygulaması cevaplarını vermişlerdir. OSMO ile neler yapabilecekleri konusunda uygulamadaki geçtikleri aşamaları anlatan durumlardan, ileriki aşamalarda olabileceğine inandıkları durumlardan bahsetmişlerdir.

"Uzaya gideriz"

"Pasta yiyebiliriz"

"Çilek yiyebiliriz"

"Ormanda yolculuk edebiliriz" gibi cevaplar verilmiştir.

Osmo kullanarak en son aşamaya geçmeyi ve tamamlamayı, en büyük çileğe ulaşmayı, oyunun en zor kısmını geçmeyi planladıkları görülmektedir. Öğrencilerin bunları düşünürken Polyo (1957) tarafından problemi anlama, plan yapma, planı uygulama ve geriye bakış olarak tanımlanan problem çözme süreci aşamalarından geçtikleri söylenebilir. Çocukların hayal dünyasının nasıl geniş, biz yetişkinlerden ne kadar farklı çalıştığı da şu cümleyle görülmektedir:

"...Aklımızı kullanarak aklımızda olan şeyleri serbest bırakmak!..."

Bu çalışmada, öğrencilerin son derece ilgisini çeken, günlük hayatta karşılaşılabilecekleri örneklerle bağdaştırdıkları durumlar içeren OSMO uygulamasını severek kullandıkları seslerindeki coşkudan, soruları cevaplarken heyecanlanma durumlarından görülmüştür. Benzer şekilde, Baz(2018) tarafından yapılan küçük

yaştaki çocuklar için tasarlanan kodlama yazılımlarının çocukların motivasyonlarını arttırmaya yönelik olduğu belirtilmiştir.

Tema2: Kodlama Teması

“Osmo’nun çilek yemesine nasıl yardım ettiniz?” sorusuna uygulamada yapmış oldukları adımlardan örnekler vermişlerdir.

“Zıplatma”

“İleri gitme”

“Elini uzatma”

“Zıplama”

“2 adım ileri”

“2 adım zıplama”

Çilek yemeye nasıl yardım ettikleri konusunda uygulamada kullandıkları bloklara yönelik olarak yaptıkları işlemleri sözel olarak dile getirdikleri görülmektedir. Akyüz (2018), düşüncelerini organize ve ifade etmelerine katkıda bulunacağını ifade etmiştir.

“Osmo kullanmak size ne kazandırdı?” sorusuna uygulamada karşılaştıkları durumlar ve nesnelere ilgili “Ödül”, “Şans”, “Çilek”, “Yürümek”, “Turta” gibi cevaplar vermişlerdir. Bunun yanında,

“Düşündürdü”,

“İyi kodlama” gibi aslında yaptıklarının bir düşünme durumu olduğunu ve bunun sonunda bir kodlama etkinliği yaptıklarını fark ettikleri görülmektedir.

“Kodlama deyince aklınıza ne geliyor?” sorusuna uygulamada kullandıkları aşamalardan, karşılaştıkları durumlardan örnekler vermişlerdir.

“Çilek toplamak geliyor”

“Osmoya komut verdik”

“4 adım ileri git” gibi cevaplar vermişlerdir.

Her öğrenci fikrini söyledikten sonra bir öğrenci kodlama konusunda aklına gelen başka şeyler olduğunu söylemiş ve şunları eklemiştir:

“İnsanlara bir şey olup uyguluyorlar”

“O şeyleri ayarlıyoruz basıyoruz istediğimiz yere hayal gücümüzle götürüyoruz.”

Kodlama teması içerisinde ele aldığımız son sorunun cevaplarına baktığımızda yapılan uygulama sonucunda öğrencilerin kodlama kavramıyla tanışarak kodlamaya yönelik olumlu tutumda buldukları söylenebilir. Patan(2016), tarafından yapılan çalışmada okul öncesinde kodlama öğretim programı uygulanan öğrencilerin kodlamaya karşı olumlu tutum geliştirdikleri görülmüştür. Öğrencilerin kodlamayla tanışma yaşlarının erken olmasının olumlu tutum durumlarını etkileyebileceği söylenebilir.

Hem deney hem kontrol grubunda PÇBÖ puanları düşük ve yüksek olan öğrencilerin kodlamayla problem çözmeye yönelik olarak odak grup görüşmelerinde verdikleri yanıtlar incelenmiştir.

Deney grubunda PÇBÖ puanları yüksek olan öğrencilerin odak grup görüşmesinde kodlamayla problem çözmeye yönelik olumlu cevaplar verdikleri, PÇBÖ puanları düşük olan öğrencilere göre detaylı görüş bildirdikleri görülmüştür. Verdikleri cevaplara baktığımızda kodlama ile ilgili bir takım fikirler edindikleri görülmüştür:

"Zihnimizi güçlendirdi"

"Burası benim yerim siz mesela öne git dediniz iki adım sola dediniz, düşünmüş olduk"

"Evde yaptığım faaliyetlerde de zihnimi güçlendirdi"

"Sizin dediğiniz şeylerle sağ sol ön ve arkayı öğrendik."

"Kodlama demek, yönlendirme demek"

"O bize örnek adımlar gösteriyor ama bazen yanlış olabiliyor"

"Hayal gücümüzü kullanarak sağı solu öğreniyoruz bu da bize çok iyi bir çalışma oluyor"

"Osmo'yu yönlendirerek, parçaları birleştirerek Osmo'yu yönlendirdik"

"Hafızamız güçlendi"

Deney grubunda PÇBÖ puanları düşük olan öğrencilerin işin daha çok oyun kısmına odaklandıkları, ancak sorular detaylandıkça onların da kodlamayla problem çözmeye yönelik olumlu cevaplar verdikleri görülmüştür.

"Osmo bize oyun kazandırdı"

"Osmo 3 adım ileri deriz"

"Osmo 1 adım çileğin üstünden atlar, 1 adım ileri git"

"Asansör mesela elektrikle çalışıyor ayarını değiştirirsin yukarı çıkar"

Kontrol grubunda PÇBÖ puanları yüksek olan öğrencilerin odak grup görüşmesinde verdikleri cevaplara baktığımızda kodlamayla problem çözmeye yönelik olumlu cevaplar verdiklerini görülmüştür:

"Osmo'yu sağa sola yöneterek"

"Kodlama deyince, insanlar bir şey uyguluyorlar kendileri yapıp da yazmak"

"O şeyleri ayarlıyoruz basınca istediğimiz kodlamayı yapıyor, hayal gücümüze göre kodlamayı yapıyor "

"Osmo kullanmak daha iyi kodlama öğrenmemizi sağladı"

"Beynimizi çalıştırdı"

Kontrol grubunda PÇBÖ puanları düşük olan öğrencilerin işin daha çok oyun kısmına odaklandıkları, ancak sorular detaylandıkça onların da kodlamayla problem çözmeye yönelik olumlu cevaplar verdikleri görülmüştür

"İlk kodlamanın niye geldiğini merak etmişim"

"Kodlama deyince çilek toplama geliyor aklıma"

"Düşünmem gerek çünkü "

"Çileği almak için düşünmemiz gerekiyor 1 adım ileri sağa sola"

Kodlama teması içerisinde yer verdiğimiz sorulara verilen cevaplara ve odak grup görüşmesinde elde edilen verilere baktığımızda, öğrencilerin kodlama kavramıyla tanışarak, karşılıklarına çıkan problemi çözmeye çalıştıklarını ve bunu uygulama esnasında uyguladıkları adımlarla, görüntülerle bağdaştırarak yorum yaptıkları görülmüştür. Alanyazında bu bulguları destekleyen çalışmalar vardır. Akçay ve Çoklar(2016), tarafından yapılan programlama becerilerinin, eleştirel düşünme, algoritmik düşünme, analitik düşünme, problem çözme, çok yönlü düşünebilme, sorgulama gibi becerileri geliştirmektedir. Calder (2010) tarafından özellikle görsel programlamanın öğrencilerin problem çözme becerisine katkı sağlayıp motivasyonlarını arttırdığı gözlemlendiği belirtilmiştir.

Tema3: Problem Çözme Teması

Araştırmaya katılan öğrencilere “problem” olarak bir kavramdan bahsedilmemiştir. OSMO kullanmak problem çözme sürecine katkı sağladı mı sorusuna,

“Düşünmemizi sağladı.”

“Çilekleri toplamamızı sağladı.”

“OSMO canavarını zıplatmayı sağladı.” gibi cevaplar verilmiştir.

Problem çözme teması ile kodlama teması aslında içiçe geçmiş sorularla donatılmıştır. Öğrencilerin verilen ya da karşılaştıkları, problem durumunda özellikle "düşünmek", "karar vermek", "uygulamak" ifadesini kullandıkları görülmüştür. Yapılan

bu uygulama ile öğrencilerde problem durumuna bir bakış açısı kazandıkları, sistematik düşünce kavramı doğrultusunda düşünmeye başladıkları, sıralı adımları takip etmeleri gerektiğini keşfettiler, bunu hem sözlü olarak ifade edebildikleri, hem uygulamada kullandıkları görülmüştür. Gelbal (1991), problem çözmede, bireylerin eğitilmesinin de önemli rol oynadığını, öğrencilere problemi vermeden önce o konu ile ilgili bir eğitimin verilmesinin onların problemi çözmelerini kolaylaştırabileceği görüşünü belirtmektedir. Okul öncesi dönem çocuğuna çeşitli problem çözme durumlarında deneyim edinebilmesi için bu deneyimlerin sistematik bir biçimde kendisine sağlanması gerektiği belirtilmektedir (Kişisel ve Yıldırım,1983). Bu çalışmada da öğrencilerin problem çözme durumlarını tanımaları, bu durumlar karşısında problemi anlamaları, tanımlamaları, plan yapıp çözüm aşamalarını düşünmeleri kısaca problem çözme süreçlerini işletebilmeleri için imkan verilmiştir.

“Osmo’yu derslerde kullanmak ister misiniz?” sorusuna tüm öğrenciler coşkulu bir “evet” cevabı vermişlerdir. Öğrencilerin dikkatini çeken, öğrenmelerini kolaylaştıran, daha çok duyu organına hitap etmesini sağlayan uygulamaların öğretim sürecinde kullanılması öğrencilerin daha istekli olmasını sağladığı düşünülmektedir. Ayrıca öğrencilerin uygulama esnasındaki heyecan ve coşkusundan öğrenme sürecine katılmak istedikleri görülmüştür. Alanyazında bu bulguları destekleyen çalışmalar vardır. Fessakis, Gouli ve Mavroudi(2013) tarafından 5-6 yaş arası anaokulu çocuklarının bilgisayar programlamasını kullanarak problem çözme boyutlarına ilişkin yapılan araştırmada çocukların ilgi çekici öğrenme etkinliklerinden zevk aldıkları ve matematiksel kavramları, problem çözme ve sosyal becerileri geliştirme fırsatlarına sahip oldukları görüşü desteklenmiştir.

BÖLÜM V

SONUÇLAR VE ÖNERİLER

Bu bölümde amaçlar ve alt amaçlara ait verilerin analiz edilmesi ile ulaşılan bulgulara dayalı olarak ortaya çıkan sonuçlar, bu sonuçların alanyazındaki araştırmalar ile tartışılması ve bundan sonra yapılabilecek araştırmalara yönelik önerilere yer verilmiştir.

Sonuçlar

Okul öncesi eğitime devam eden beş yaş grubu öğrencilerine verilecek algoritma ve kodlama eğitiminin çocukların problem çözme becerilerine etkisinin saptanması amacı ile yapılan araştırmada kodlama eğitimi öncesinde verilecek algoritma eğitiminin öğrencilerin problem çözme becerilerini olumlu etkilediği, PÇBÖ öntest ve sontest puanlarında anlamlı bir farklılık olduğu bulunmuştur. Şahin ve Namlı (2017) tarafından yapılan araştırmada ilk kez algoritma eğitimi alan ortaokul öğrencilerine BDE ve drama yöntemleriyle verilen algoritma eğitimi sonrası problem çözme becerileri incelenmiş ve her iki yöntemde de eğitim sonucunda problem çözme becerilerinde olumlu yönde anlamlı farklılık bulunmuştur. Araştırmanın sonuçlarına ve ilgili araştırmalara bakıldığında algoritma eğitiminin öğrencilerin problem çözme becerilerini olumlu etkilediği söylenebilir.

Grupların hangisinin deney hangisinin kontrol olacağı rastgele seçilmiştir. PÇBÖ ön test puanlarına bakıldığında kontrol grubunun puanlarının deney grubunun ön test puanlarından daha yüksek olduğu görülmüştür. Uygulama sonrasında uygulanan son test puanları karşılaştırıldığında ise deney grubunun puanlarının kontrol grubunun puanlarından yüksek olduğu görülmüştür. Bu durumda verilen algoritma eğitiminin çocukların problem çözme becerilerine olumlu katkı sağladığı söylenebilir.

Algoritma ve kodlama eğitimi alan öğrencilerin son test puanlarının ön test puanlarının ortalamalarından yüksek olduğu ve aradaki farkın istatistiksel olarak anlamlı olduğu görülmüştür. Verilen algoritma eğitiminin öğrencilerin problem çözme becerileri üzerinde olumlu etkisi olduğu söylenebilir.

Algoritma eğitimi almayan ancak kodlama eğitimi alan öğrencilerin son test puan ortalamalarının ön test puan ortalamalarından yüksek olduğu görülmüş ancak istatistiksel olarak fark anlamlı değildir. Calder (2010) öğrencilere programlama dili olarak Scratch kullandırıp gözlem ve görüşmeler yapmış, öğrencilerin yazdığı ilerleme ve yansımalarını dile getiren blog yazılarını incelemiştir. Araştırma sonucunda, Scratch yazılımının problem çözme için ilgili çekici ve kullanımı nispeten daha kolay bir alan olduğu, matematiksel kavramların anlaşılmasında verimli ve motive edici bir programlama ortamı sağladığı görülmüştür. Bu çalışmada verilen kodlama eğitiminin bu anlamda sürece katkı sağladığı ve problem çözme becerilerini geliştirdiği söylenebilir.

Algoritma eğitimi alan ve almayan PÇBÖ puanı yüksek olan öğrencilerin problem çözmeye yönelik görüşlerine bakıldığında PÇBÖ puanı düşük olan öğrencilere göre olaya daha uyumlu oldukları, özellikle algoritma eğitimi alan grubun gerçek hayatlarından örneklerle durumu destekledikleri, daha ayrıntılı, daha kapsamlı görüşler bildirdikleri görülmüştür. Akyüz (2018), programlamanın basit parçaları bir araya getirerek karmaşık problemlere çözümler üretmeyi sağlayan yap-boz benzeri faydalı zihinsel bir etkinlik olduğunu ve bu yönüyle programlamanın çocuklarımızın parça-bütün ilişkilerini öğrenmelerine, detaylara özen göstermelerine, düşüncelerini organize ve ifade etmelerine ve genel olarak zekâlarının gelişmesine katkıda bulunacağını ifade etmiştir. Hamada (1986), erken yaşlarda programlama eğitimi alan öğrencilerin matematiksel bilgi ve problem çözme becerilerinde daha yüksek başarı gösterdiklerini belirtmektedir. Bu araştırmalara da bakıldığında erken yaşta verilecek programlama eğitiminin öğrencilerin problem çözme becerilerini olumlu etkilediği söylenebilir.

Öneriler

1. Bu çalışmada elde edilen veriler 8 haftalık sınırlı bir çalışma grubunda uygulama sonucunda elde edilmiştir. Böyle bir çalışmanın daha geniş bir çalışma grubu ile daha uzun sürecek bir uygulama ile yapılması elde edilecek sonuçların daha genellenebilir olmasını sağlayabilir.
2. Bu çalışmada elde edilen veriler sınırlı sayıdaki OSMO Kodung setiyle elde edilmiştir. Benzer bir çalışmanın her öğrencinin kendi çalışabileceği bir set ile yapılması daha etkili ve verimli sonuçlar verebilir.

3. Okul öncesi eğitime devam eden beş yaş grubu öğrencilerine verilecek algoritma ve kodlama eğitiminin çocukların problem çözme becerilerine etkisinin saptanması amacıyla yapılan bu çalışmada kodlama eğitimi öncesinde verilecek algoritma eğitiminin problem çözme becerilerine olumlu katkı sağladığı, erken yaşta verilecek iyi planlanmış bir algoritma eğitimi ile kodlama eğitimi öncesinde sağlam bir temel oluşturup çocukların günlük yaşantılarını da kolaylaştırması sağlanabilir.
4. Bütün dünyada kabul görmüş durumda olan kodlama eğitimi ülkemizde de okul öncesi eğitimde verilmeye başlanmalıdır bu eğitimin ilk bölümü algoritma eğitimi olmalıdır. Hangi kademedede olursa olsun algoritma eğitimi verildikten sonra kodlama eğitimine geçilmesi verimi arttıracaktır bu şekilde bir uygulama üst düzeyde fayda sağlayacaktır. Öğrencilerin bu araştırmada kullanılan bir materyali kullanarak kodlama eğitimi verilmesi şart değildir. Okul türü, öğrenci hazırbulunuşluğu, materyal desteği dikkate alınarak ücretsiz olarak kullanılan uygulamalar ya da öne çıkan başka uygulamalardan da seçimler yapılabilir.
5. "Her Kademedede Bilişim" sloganından yola çıkarak doğru ve iyi yapılandırılmış bir kodlama eğitimi, bu alanda uzman kişiler olan "Bilişim Teknolojileri Öğretmenleri" tarafından her yaş grubuna ve seviyeye zorunlu ders kategorisinde verilmelidir.
6. Bu çalışma, uygulamayı yapan bireylerin araştırma sürecine doğrudan katılımını sağlayarak öğrenmeyi, doğrudan sistemin içinde olarak ortaya çıkan sorunlara çözüm üretmeyi aslında uygulamayı iyileştirmek için sistemli, sürekli ve uzun süreli yürütülen bir eylem araştırması olarak yapılabilir.

KAYNAKLAR

- Agogi, E., Rossis, D. ve Stylianidou, F. (2014). Creative Little Scientist: D 6.6 Set of Recommendations to Policy Makers and Stakeholders, 64pp.
- Akçay, A. , Çoklar, A.N. (2016). Bilişsel Becerilerin Gelişimine Yönelik Bir Öneri: Programlama Eğitimi. https://www.researchgate.net/publication/320517601_Bilissel_Becerilerin_Gelisimine_Yonelik_Bir_Oneri_Programlama_Egitimi. Erişim Tarihi: 01.05.2018
- Akkaya, S. (2006). Okul öncesi eğitim kurumlarında uygulanan fen ve doğa etkinliklerinin çocukların problem çözme becerilerine etkisi konusunda öğretmen görüşleri. Unpublished master's thesis, Anadolu Üniversitesi, Eskişehir.
- Akkuş, S., Yılmaz, Y., Şahinöz, A., & Sucaklı, İ. (2015). 3-60 ay arası çocukların televizyon izleme alışkanlıklarının incelenmesi. Hacettepe Üniversitesi Sağlık Bilimleri Fakültesi Dergisi.
- Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. Elementary Education Online, 13(1), 1-4.
- Akyüz, A.O.(2018). Çocuklara Programlama Nasıl Öğretilir?. "Prof.Dr.Emine Akyüz'e Armağan Akademisyenlikte 50.Yıl, Pegem Akademi, Nisan 2018, Ankara.
- American Academy of Pediatrics.(2016). American Academy of Pediatrics Announces New Recommendations for Children's Media Use. [Çevrim-içi:<https://www.aap.org/en-us/about-the-aap/aap-press-room/Pages/American-Academy-of-Pediatrics-Announces-New-Recommendations-for-Childrens-Media-Use.aspx>]. Erişim Tarihi: 12.08.2018.
- Amethyst Consultancy (2003). Roamer. London, UK: Valiant Technology, [Çevrim-içi: http://www.valiant-technology.com/uk/pages/roamer_home.php], Erişim Tarihi: 15.03.2017.
- Arabacıoğlu, T. (2006). İnternet destekli programlama mantığı öğretimi. Yüksek Lisans Tezi, Gazi Üniversitesi, Ankara, Türkiye.
- Arabacıoğlu, T., Bülbül H.İ.,ve Filiz, A. (2007). Bilgisayar Programlama Öğretiminde Yeni Bir Yaklaşım, IX. Akademik Bilişim Konferansı içinde (s.193-197), Dumlupınar Üniversitesi, Kütahya.
- Baki, A., ve Gökçek, T. (2012). Karma yöntem araştırmalarına genel bir bakış. Elektronik Sosyal Bilimler Dergisi, 11(42), 1-21.
- Balanskat, A., Engelhardt, K. (2015). Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe, European Schoolnet. [Çevrim-içi: <https://www.researchgate.net/publication/284139559>], Erişim tarihi: 11.08.2018.

- Baş T., Çamır, M., ve Özmalda, B. (2008). Nitel Araştırma Yöntemleri. Ankara: Seçkin Yayınevi
- Baz, F.Ç. (2018). Çocuklar İçin Kodlama Yazılımları Üzerine Karşılaştırmalı Bir İnceleme. *Curr Res Educ* (2018), 4(1) · 36-47
- Benzer, A.İ., Erümit A.K. (2017). Programlama Öğretimine Yönelik Lisansüstü Tezlerin İncelenmesi, *Journal of Instructional Technologies & Teacher Education* Vol 6 No 3 (2017), 99-110.
- Berksoy, İ., Sözcü Ö.F., Armağan, E., Arslan, A. (2016). Algoritma ve Programlama Eğitiminde Scratch Programı Üzerine Yapılan Çalışmaların Analizi. 18. Akademik Bilişim Konferansı, Aydın. [Çevrim-içi: <http://ab.org.tr/ab16/bildiri/74.docx>], Erişim tarihi: 22.02.2017.
- Bers, M. U. (2010). The TangibleK robotics program: applied computational thinking for young children. *Early Childhood Research & Practice*, 12, 2, n2.
- Billings, K. (1983). Research on school computing. In M. T. Grady & J.D. Gwaronski (Eds), *Computers in Curriculum and Instruction*. VA. APCD.
- Borusan. (2016). Gelecek Kodlamada, <http://www.borusan.com.tr/tr/haberler/gelecek-kodlama>, Erişim Tarihi: 17.03.2017.
- BTE Derneği. (2013). Türkiye’de ilk ve ortaokullarda (ilköğretim) okutulan bilişim teknolojileri derslerinin tarihi. [Çevrim-içi: <http://bte.org.tr/bte-derneği/bt-derslerinin-tarihcesi/>], Erişim Tarihi: 14.03.2017
- Business Wire (English). [Çevrim-içi: <http://www.businesswire.com/news/home/20131014006133/en/Code.org-Introduces-%E2%80%9CHour-Code%E2%80%9DCampaign-Inspire-10>], Erişim Tarihi: 15.03.2017.
- Büyüköztürk, Ş. (2002). Sosyal bilimler için veri analizi el kitabı. Ankara: Pegem Akademi.
- Büyüköztürk, Ş., Çokluk, Ö. ve Köklü, N. (2010). Sosyal Bilimler İçin İstatistik. Ankara: Pegem Akademi.
- Calao, L. A., Moreno-León, J., Correa, H. E., ve Robles, G. (2015). Developing Mathematical Thinking with Scratch: an Experiment with 6th grade students. *Design for Teaching and Learning in a Networked World* (s. 17-27). Springer International Publishing.
- Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education*, 22(4), 711.

- Chang, C. K. (2014). Effects of Using Alice and Scratch in an Introductory Programming Course for Corrective Instruction. *Journal of Educational Computing Research*, 51(2), 185-204.
- Clements, D. H. ve Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*. 76(6). 1051-1058.
- Clements, D., Nastasi, B., ve Swaminathan, S. (1993). Young children and computers: crossroads and directions from research. *Young Children*, 48(2), 56-64, Washington DC, USA.
- Clements, D., ve Sarama, J. (2003). Strip mining for gold: research and policy in educational technology – a response to “fool's gold”. *Association for the Advancement of Computing in Education, Norfolk*, 11(1), 7-69.
- Code.org. (2013). Code.org introduces “Hour of code” campaign to inspire 10 million students to learn to code.
- Code.org. (2015). The Hour of Code is here. [Çevrim-içi: <https://code.org/>], Erişim tarihi: 11.11.2015.
- Code.org. (2016). tr.code.org, Erişim Tarihi: 10.02.2017
- “Coding - the 21st century skill”, Digital Agenda for Europe - European Commission. (n.d.). [Çevrim-içi: <https://ec.europa.eu/digital-single-market/coding-21st-century-skill>], Erişim tarihi: 04.04.2018, (2018)
- Creswell, J.W. (2013). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. London: Sage.
- Curriculum and Instruction: A 21st Century Skills Implementation Guide (2009). The Partnership for 21st Century Skill, [Çevrim-içi: <http://goo.gl/0na2jN>], Erişim tarihi: 21.03.2017.
- Çakıroğlu, Ü., Sarı, E., Akkan, Y. (2011). Üstün yetenekli öğrencilere programlama öğretiminin problem çözmeye katkısı konusunda öğretmen görüşleri, 5th International Computer & Instructional Technologies Symposium, 22-24 September 2011, Fırat University, Elazığ.
- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal Of Instructional Technologies & Teacher Education*, 4(3).
- Çetin, E. (2012). Bilgisayar Programlama Eğitiminin Çocukların Problem Çözme Becerileri Üzerine Etkisi. Yüksek Lisans Tezi. Gazi Üniversitesi, Ankara.
- Çırpılı, A.S. (2016). Erken Yaş Gruplarında Algoritma Eğitimi Süreci. 25. Ulusal Eğitim Bilimleri Kongresi, Antalya, 21-22.
- Çokluk, Ö., Yılmaz, K., Oğuz, E. (2011). Nitel Bir Görüşme Yöntemi: Odak Grup Görüşmesi, *Kuramsal Eğitimbilim*, 4 (1), 95-107.

- Dalton, D. W. (1986). A Comparison of the Effects of LOGO and Teacher-Directed Problem Solving Instruction on the Problem-Solving Skills, Achievement and Attitudes of Low, Average and High Achieving Junior High Learners. Proceedings of Selected Research Paper Presentations at the 1986 Annual Convention of the Association for Educational Communications and Technology, 119-152.
- Demirer, V. ve Sak, N.(2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. Eğitimde Kuram ve Uygulama, 12(3), 521-546.
- Department for Education (2013). The National Curriculum in England. Key stages 1 and 2 framework document. Reference DFE-00178-2013. London: Department for Education.
- diSessa, A. (2000). Changing minds: Computers, learning, and literacy. MIT Press.
- diSessa, A. A., ve Abelson, H. (1986). Boxer: a reconstructible computational medium. Communications of the ACM, 29(9), 859–868.
- Dow, G. T., & Mayer, R. E. (2004). Teaching students to solve insight problems: evidence for domain specificity in creativity training. Creativity Research Journal, 16 (4), 389-402.
- Durak, G. (2009). Algoritma konusunda geliştirilen programlama mantığı öğretisi yazılımının başarısına etkisi. Yüksek Lisans Tezi, Balıkesir Üniversitesi.
- Elkin, M., Sullivan, A ve Bers, M.U. (2016). Programming with the KIBO Robotics Kit in Preschool Classrooms, Computers in the Schools, 33:3, 169-186.
- Erden, M. (1984). “İlkokulların Birinci Devresine Devam Eden Öğrencilerin Dört İşleme Dayalı Problemleri Çözerken Gösterdikleri Davranışlar” (Yayımlanmamış Doktora Tezi) Ankara: Hacettepe Üniversitesi.
- Erdoğan, B. (2005). Programlama başarısı ile akademik başarı, genel yetenek, bilgisayara karşı tutum, cinsiyet ve lise türü arasındaki ilişkilerin incelenmesi . Yüksek Lisans Tezi. Marmara Üniversitesi.
- Erol, O., KURT, A.A. (2017). BÖTE Bölümü Öğrencilerinin Programlamaya Karşı Tutumlarının İncelenmesi. Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi. Sayı: 41.S.314-325
- Ertürk, Y.D., Gül, A.A. (2006). Çocuğunuzu televizyona teslim etmeyin. Ankara: Nobel Yayınları
- Fessakis, G., Gouli, E., Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study, Computers ve Education 63, 87–97.
- Fırat, M., Kabakçı Yurdakul, I., ve Ersoy, A. (2014). Bir eğitim teknolojisi araştırmasına dayalı karma yöntem araştırması deneyimi. Eğitimde Nitel Araştırmalar Dergisi - Journal of Qualitative Research in Education, 2(1), 65-86.

- Flannery, L. P., ve Bers, M. U. (2013). Let's dance the "robot hokey-pokey!": Children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education*, 6(1), 81-101.
- Furber, S. (2012). *Shut down or restart: the way forward for computing in UK schools*. London: Royal Society.
- Futschek, G. ve Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In *Proceedings of the 5th International Conference on Informatics in Schools: Situation, Evolution and Perspectives (ISSEP)*, 155–164.
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. In *Proceedings of the 2nd International Conference on Informatics in Secondary Schools: Evolution and Perspectives (ISSEP)*, 159–168.
- Geist, E. (2016). Robots, Programming and Coding, Oh My!, *Childhood Education*, 92:4, 298-304.
- Gelbal, Selahattin. "Problem Çözme", *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*. 6: s.167, 1991.
- General Electric. (2016). GE, 'Kızlar Kodluyor' eğitimi ile öğrencileri dijital dönüşüme hazırlıyor, <http://goo.gl/Yi0Y0O>, Erişim Tarihi: 14.03.2017.
- Gibson, J.P. (2012). Teaching graph algorithms to children of all ages. In *ITiCSE '12: 17th Annual Conference on Innovation and Technology in Computer Science Education*, 34-39.
- Gökoğlu, S. (2017). Programlama Eğitiminde Algoritma Algısı Bir Metefor Analizi, *Cumhuriyet International Journal of Education-CIJE*, Vol 6, 2017, 1-14.
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199-237
- Gülbahar, Y., Kalelioğlu F., Karataş, E. (2017). *Bilgisayar Bilimi Ders Kitabı Kurl*, MEB, Ankara.
- Hamada, R. M. (1986). *The relationship between learning Logo and proficiency in mathematics*. Doctoral dissertation, Columbia University.
- Howland, K. and J. Good. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers ve Education*, 80, 224-240.
- Kafai, Y. B., ve Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, 95(1), 61-65.
- Kahraman, B. (2015). Avustralya'da ilkokullarda programcılık eğitimi veriliyor. [Çevrim-içi: <http://www.webtekno.com/sektorel/avustralya-da-ilkokullarda-programcılık-egitimi-veriliyor-h10859.html>], Erişim Tarihi: 01.08.2016.

- Karabak, D. ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Journal of Research in Education and Teaching*, 2(3), (s.175-181).
- Karadeniz, Ş., Samur, Y. ve Özden, M.Y.(2014). Playing with Algorithms to Learn Programming: A Case Study on 5 Years Old Children. *Proceedings of 9th International Conference on Information Technology and Applications (ICITA 2014)*, 3p., 1-4 July 2014, Sydney. [<http://www.icita.org/2014/papers/tr-Karadeniz.pdf>]
- Kert, S. B., Uğraş, T. (2009). Programlama Eğitiminde Sadelik ve Eğlence: Scratch Örneği. *1st International Congress of Educational Research. 1st International Congress of Educational Research*.
- Kert, S.B.,Yeni,S.,Şahiner, A.(2017). Komputasyonel Düşünme İle İlişkilendirilen Alt Becerilerin İncelenmesi.
- Kişisel, E. ve Yıldırım,S.M. (1983). *Bilişsel Etkinlikler*. İstanbul: Milli Eğitim Basımevi.
- Köse, U. ve Tüfekçi, A. (2015), *Pegem Eğitim ve Öğretim Dergisi*, 2015, 569-586
- Lewis, C., Esper, S., Bhattacharyya, V., Fa-Kaji, N., Dominguez, N., ve Schlesinger, A. (2014). Children's perceptions of what counts as a programming language. *Journal of Computing Sciences in Colleges*, 29(4), 123-133.
- Liao, Y. C., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A metaanalysis. *Journal of Educational Computing Research*, 7, 251-268.
- Manches, A. ve O'Malley, C. (2012). Tangibles for learning: a representational analysis of physical manipulation. *Personal and Ubiquitous Computing*, 16, 4, 405-419.
- Manches,A ve Plowman,L. (2017). Computing education in children's early years: A call for debate, *British Journal of Educational Technology* 48(1), 191-201.
- Michael, K. A. ve Omoloye, E. A., "Improving Structural Designs with Computer Programming in Building Construction", *IOSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 16, Issue 3, Ver. VI, PP 10 -16,
- Mittermeir, R.T. (2013)."Algoritmics for preschoolers-A contradiction?" *Creative Education*, 4(9), 557-562.
- Monroy-Hernandez, A., Resnick, M. (2008). Empowering kids to create and share programmable media. *ACM Digital Library*, 15(2), (s.50-53).
- NAEYCF (National Association for the Education of Young Children and Fred Rogers Center for Early Learning and Children's Media) (2012). Technology and interactive media as tools in early childhood programs serving children from birth through age 8: A joint position statement

- Nowaczyk, R.H.(1983). Cognitive Skills Needed in Computer Programming. <https://eric.ed.gov/?id=ED236466>. Eriřim Tarihi: 01.05.2018
- OECD, Bir Bakıřta Eđitim 2014, OECD Gstergeleri. (2014). [evrim-ii: <http://goo.gl/AG8RNV>], Eriřim tarihi: 15.03.2017.
- Ođuz, V., Akyol, A.K. (2015). Problem zme Becerisi leđi (PB) Geerlik ve Gvenirlik alıřması, ukurova niversitesi Eđitim Fakltesi Dergisi, 44(1), 105-122.
- Oluk, A., Korkmaz, . ve Oluk, A.H. (2018). Scratch'ın 5.Sınıf đrencilerinin Algoritma Geliřtirme ve Bilgi-İřlemsel Dřnme Becerilerine Etkisi. Turkish Journal of Computer and Mathematics Education, Vol.9.No.1.,54-71
- alan, H.(2015). Avustralya'da kodlama dersi ilköđretim mfredatına girdi. [evrim-ii: <http://onedio.com/haber/avustralya-da-kodlama-dersi-ilkogretim-mufredatina-girdi-593639>], Eriřim Tarihi: 01.06.2016.
- ndeř, . (2016). İngiltere ve ABD'de kodlama eđitimi. [evrim-ii: <http://www.hurriyet.com.tr/ingiltere-ve-abdde-kodlama-egitimi-40061515>], Eriřim Tarihi: 02.08.2016.
- zkaya, M.(2016). Erken ocukluk Dneminde Kodlama Eđitimi: lkelerin Eđitim Programlarının Karřılařtırmalı İncelenmesi, 25. Ulusal Eđitim Bilimleri Kongresi, Antalya. [evrim-ii : http://www.uebk.org/UEBK_bildiri_ozetleri.pdf], Eriřim Tarihi : 14.11.2016.
- zmen,B., Altun,A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles, Turkish Online Journal of Qualitative Inquiry, July 2014, 5(3)
- Papert, S. (1980) Mindstorms. NY. Basic Books
- Papert, S. (1991). Mindstorms: Children, computers and powerful ideas. Athens: Odysseas Publications (in Greek).
- Patan, B. (2016). Okul ncesi Kodlama đretim Programının Geliřtirilmesi. Yksek Lisans Tezi, Baheřehir niversitesi,.
- Perlman, R. (1974). TORTIS – Toddler's Own Recursive Turtle Interpreter System. In MIT AI Memo 311, Logo Memo 9. Cambridge, MA, USA: Massachusetts Institute of Technology.
- Perlman, R. (1976). Using computer technology to provide a creative learning environment for preschool children. In MIT AI Lab Memo 360, Logo Memo 24. Cambridge, MA,USA: Massachusetts Institute of Technology.
- Piaget, J. (1973). To understand is to invent: The future of education. New York, NY: Grossman Publishers.

- Ramalingam,V. LaBelle, D. Wiedenbeck, S. “Self-Efficacy and Mental Models in Learning to Program”, Proceedings of The 9th Annual SIGCSE Conference On Innovation And Technology In Computer Science Education, Cit:36, Sayı:3, s.171-175, 2004.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., (2009). Scratch:Programming for all. Communications of the ACM 52(11), 60-67.
- Resnick, M. (2013). Learn to code, code to learn. How programming prepares kids formore than math. EdSurge 8.
- Robincode. (2016). www.robincode.org, Erişim tarihi: 10.02.2017
- Sayan, H. (2016). Okul Öncesi Eğitimde Teknoloji Kullanımı. 21. Yüzyılda Eğitim Ve Toplum Eğitim Bilimleri Ve Sosyal Araştırmalar Dergisi, 5(13).
- Saygıner, Ş., Tüzün, H. (2017) İlköğretim Düzeyinde Programlama Eğitimi: Yurt Dışı Ve Yurt İçi Perspektifinden Bir Bakış, 19.Akademik Bilişim Konferansı, Aksaray. [Çevrim-ıci: <http://ab.org.tr/ab17/bildiri/211.pdf>], Erişim tarihi: 22.02.2017.
- Sayın, Z., & Seferoğlu, S. S. (2016). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. Akademik Bilişim Konferansı, Adnan Menderes University, Aydın.
- Schwartz, J., Stagner, J., and Morrison, W. (2006, July). Kid's programming language (KPL). In ACM SIGGRAPH 2006 Educators program (p. 52). ACM.
- Sesame Workshop. (2009). Sesame workshop and the PNC Foundation join White House effort on STEM education
- Shin, S., Park, P., ve Bae, Y. (2013). The effects of an information-technology gifted program on friendship using scratch programming language and clutter. International Journal of Computer and Communication Engineering, 2(3), 246.
- Solomon, C., ve Papert, S. (1975). Teach: A step toward more interactive programming. Logo Working Paper 43. Cambridge, MA, USA: MIT AI Lab.
- Stevens, M. (1998). Sorun Çözümleme. (Çev. Ali Çimen). İstanbul: Timaş Yayınları.
- Stoeckelmayr,K., Tesar, M., Hofmann, A. (2011). Kindergarten children programming robots: A first attempt. In Proceedings of Robotics in Education, 185–192.
- Strand, E. (1986). A descriptive study comparing preschool and kindergarten LOGO interaction. In Annual meeting of the American Educational Research Association (70th, San Francisco, CA, April 16–20, 1986). Texas, USA, ED270213, ERIC – Education Resources Information Center, Computer Sciences Corporation, Lanham, MD, USA.

- Sullivan, A ve Bers, M.U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade, *International Journal of Technology ve Design Education*, 26, 3-20.
- Şabanoviç, A., Yannier, S. (2003). Robotlar: Sosyal Etkileşimli Makineler, TÜBİTAK Bilim Teknik Dergisi
- Şahin ve Namlı (2017), Algoritma Eğitiminin Problem Çözme Becerisi Üzerine Etkisi, *Recep Tayyip Erdoğan Üniversitesi Sosyal Bilimler Dergisi*, 135-153
- Şahin, Ç. (2004). Problem Çözme Becerisinin Temel Felsefesi, *Kazım Karabekir Eğitim Fakültesi Dergisi*, Sayı:10. S.160-171.
- Tavşancıl, E. & Aslan E. (2001). İçerik Analizi ve Uygulama Örnekleri. Epsilon Yayınları: İstanbul.
- TBD. (2014). Programlama Çocuk Oyunağı. [Çevrim-içi: http://egitimplatformu.aydin.edu.tr/gundem/haber_detay.asp?haberID=32], Erişim Tarihi: 01.08.2016.
- TTKB. (2012). Bilişim teknolojileri ve yazılım dersi (5, 6, 7 ve 8. sınıflar) öğretim programı. Ankara: MEB.
- TTKB. (2016). Ortaöğretim bilgisayar bilimi dersi (kur 1, kur 2) öğretim programı. Ankara: MEB.
- TTKB. (2017). Bilişim teknolojileri ve yazılım dersi öğretim programı. Ankara: MEB.
- Türkiye Vodafone Vakfı. (2016). <http://www.turkiyevodafonevakfi.org.tr/tr/haberler/turkiye-vodafone-vakfi-ve-habitat-yarini-kodlayan-nesiller-icin-harekete-gecti>, Erişim Tarihi: 15.03.2017.
- White House. (2011). Educate to innovate. [Çevrim – içi: <http://www.whitehouse.gov/issues/education/educate-innovate>]
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. 1 Haziran 2017 tarihinde [<https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>] adresinden alınmıştır.
- Yıldırım, A., ve Simsek, S. (2008). Durum Çalışması. Sosyal Bilimlerde Nitel Araştırma Yöntemleri. Ankara: Seçkin Yayıncılık.
- Yıldız, A. G. M., Çiftçi, E., & Karal, H.(2017). Bilişimsel Düşünme Ve Programlama.
- Zorlu Holding. (2016). Basın Bülteni, http://www.zorlu.com.tr/Assets/u/163171-ZH_Robincode_Hour_of_Code_Duyuru_BB_021216.docx

CodeSpark. (2018). iTunes. Mayıs 2018 tarihinde <https://itunes.apple.com/us/app/codespark-academy/id923441570> adresinden erişildi.

Blockly. (2018). Mayıs 2018 tarihinde <https://developers.google.com/blockly/> adresinden erişildi.



EKLER

EK 1: Problem Çözme Becerisi Ölçeği (PÇBÖ) Maddeler

1. Bu çocuk, boyama yaparken elleri boya olmuş.
2. Bu çocuk, yemeğine tuz atarken tuzluğun kapağı açılmış ve bütün tuz yemeğine dökülmüş.
3. Bu çocuğun bisikletinin tekerleği patlamış.
4. Bu çocuk, oyuncakıyla oynarken arkadaşı onun oyuncakını izinsiz almış.
5. Bu çocuk, gömleğinin düğmesini açamamış.
6. Bu çocuğun topunun havası inmiş.
7. Bu çocuk, yeleğinin düğmelerini iliklerken düğmelerinden biri kopmuş.
8. Bu çocuk, pasta tabağını düşürmüştür.
9. Bu çocuğun pantolonuna arkadaşı sulu boya fırçasını düşürmüştür.
10. Bu çocuk, hikaye kitabının sayfalarını çevirirken sayfalarından biri yırtılmış.
11. Bu çocuk yapbozla oynarken yapbozun parçalarından birinin eksik olduğunu fark etmiş.
12. Bu çocuk, oturduğu masadan kalkarken ayanı arkadaşının üzerine dökmüştür.
13. Bu çocuk, parkta salıncağa binmek istemiş. Ama başka bir çocuk salıncaktan inmek istememiş.
14. Bu çocuk, dolabın üstünde bulunan oyuncakını almak için uzanmış, ama oyuncakını alamamış.
15. Bu çocuk, koşarken saksıyı kırmış.
16. Bu çocuk yemek yiyecekmiş, ama çatal kaşık yokmuş.
17. Bu çocuk, dondurmasını yerken yere düşürmüştür.
18. Bu çocuk, oyuncakıyla oynarken oyuncakını kırmış.

EK 2: Problem Çözme Becerisi Ölçeği (PÇBÖ) Örnek Sorular ve Cevaplar

Problem Durumu: Bu çocuk, oyuncuğıyla oynarken arkadaşı onun oyuncuğını izinsiz almış.

Araştırmacı : Bu çocuk, oyuncuğını nasıl geri alabilir?

Çocuk : Öğretmenine söyler. (1 puan)

Arkadaşına verebilir misin der. (1 puan)

Araştırmacı : Başka neler söyleyebilirsin?

Çocuk : İkisi de oynayabilir, paylaşabilir. (1 puan)

Araştırmacı : Başka?

Çocuk : Başka aklıma bir şey gelmiyor.

TOPLAM: 3 PUAN

Problem Durumu: Bu çocuk, dolabın üstünde bulunan oyuncuğını almak için uzanmış, ama oyuncuğını alamamış.

Araştırmacı : Bu çocuk, oyuncuğını dolabın üzerinden nasıl alabilir?

Çocuk : Babasından isteyebilir. (1 puan)

Araştırmacı : Başka neler söyleyebilirsin?

Çocuk : Tabureye çıkıp onu alabilir. (1 puan)

Araştırmacı : Peki başka neler olabilir?

Çocuk : Abisinden isteyebilir. (1 puan)

Sandalyeye çıkıp alabilir. (1 puan)

TOPLAM: 4 PUAN

EK 3: Problem Çözme Becerisi Ölçeđi (PÇBÖ) Bir Örnek Çizim


Bu çocuđun pantolonuna arkadaşı sulu boya fırçasını düşürmüő.



EK 4: Problem Çözme Becerisi Ölçeği Kullanım İzni

Sayın: Canan AKYOL ALTUN

Araştırmanızda kullanmak amacıyla talep ettiğiniz “Çocuklar İçin Problem Çözme Becerisi Ölçeği’ni (PÇBÖ)” kullanmanızda hiçbir sakınca bulunmamaktadır.
İyi çalışmalar dileklerle, 13.04.2017


Yrd. Doç. Dr. Vuslat OĞUZ

ÖZGEÇMİŞ

Adı Soyadı : Canan AKYOL ALTUN

Doğum Yeri : Ankara

Doğum Yılı : 1985

İletişim Bilgileri : Durali Alıç Mah.928.Cad. 56/15 Mamak / Ankara

E-Posta Adresi : cananakyol@gmail.com

Eğitim Durumu :

Zübeyde Hanım Anadolu Kız Meslek Lisesi - Bilgisayar (1999-2003)

Ankara Üniversitesi - Eğitim Bilimleri Fakültesi - Bilgisayar ve Öğretim Teknolojileri Eğitimi (2003-2007)

Çalıştığı Kurumlar :

Kars - Sarıkamış Kazım Karabekir İlköğretim Okulu -Bilişim Teknolojileri Öğretmeni (2007-2010)

Sakarya - Kocaali Osman Salihoglu İlköğretim Okulu - Bilişim Teknolojileri Öğretmeni (2010-2011)

Ankara - Milli Piyango Anadolu Lisesi - Bilişim Teknolojileri Öğretmeni (2011-Devam ediyor)

Seminerler - Kurslar :

Özel Eğitim Semineri	10/04/2009 - 18/04/2009
Web Tabanlı İçerik Geliştirme Kursu (I. Kademe)	19/07/2010 - 30/07/2010
Web Tabanlı İçerik Geliştirme Kursu (II. Kademe)	01/11/2010 - 12/11/2010
Eğitimde FATİH Projesi (Teknoloji Kullanımı) Kursu	17/09/2012 - 21/09/2012

FATİH Projesi Eğitimde Teknoloji Kullanımı (Eğitim Yöneticisi ve Eğitim Görevlisi)	Kasım 2012 - Mart 2013
Fatih Projesi Bilişim Teknolojilerinin ve İnternetin Bilinçli, Güvenli Kullanımı	05/05/2014 - 06/05/2014
FATİH Projesi Eğitimci Eğitimi Güncelleme Kursu	23/03/2015 - 05/04/2015
Fatih Projesi BT'nin ve İnternetin Bilinçli ve Güvenli Kullanımı Kursu	05/12/2016 - 16/12/2016
Fatih Projesi Etkileşimli Sınıf Yönetimi Kursu	18/01/2016 - 19/01/2016
Oracle Java Fundamentals Eğitimi (Java, Alice, Greenfoot)	07/05/2018 - 11/05/2018
Yarımı İnşa Et Programı	5-12-13 Mayıs 2018

Yayınlar :

Alper, A., Öztürk, S., & Altun, C. A. (2014). Problem Based Learning Studies in Turkey. *Journal of Faculty of Educational Sciences*, 47(1).

Kitap :

Akyol, C., Ünal, E., Aydın, F., Debbağ, G., Öztürk S., Karademir T., Kemeriz, Z. (2012). Alper A.(Ed.). Sosyal Ağlar. Ankara : Pelikan.