



T.C.

ULUDAĞ ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM
TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI

**ALGORİTMA ÖĞRETİMİNDE SCRATCH
KULLANIMININ ORTAOKUL ÖĞRENCİLERİ
ÜZERİNDEKİ BAŞARIYA ETKİSİ VE ÖĞRENCİ
GÖRÜŞLERİ**

Yüksek Lisans Tezi

SEVİM AYDIN

BURSA, 2019



T.C.

ULUDAĞ ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM
TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI

ALGORİTMA ÖĞRETİMİNDE SCRATCH
KULLANIMININ ORTAOKUL ÖĞRENCİLERİ
ÜZERİNDEKİ BAŞARIYA ETKİSİ VE ÖĞRENCİ
GÖRÜŞLERİ

Yüksek Lisans Tezi

SEVİM AYDIN

TEZ DANIŞMAN

DOÇ.DR. ERHAN ŞENGEL

BURSA, 2019

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim.

SEVİM AYDIN

10/07/2019



EĞİTİM BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA İNTİHAL YAZILIM RAPORU

ULUDAĞ ÜNİVERSİTESİ
EĞİTİM BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih:21/10/2019

Tez Başlığı / Konusu: Algoritma Öğretiminde Scratch Kullanımının Ortaokul Öğrencileri Üzerindeki Başarıya Etkisi Ve Öğrenci Görüşleri

Yukarıda başlığı gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam 73 sayfalık kısmına ilişkin, 11/11/2019 tarihinde şahsım tarafından Turnitin adlı intihal tespit programından (Turnitin)* aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı % 15 'tür.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar hariç/dahil
- 3- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.


Tarih ve İmza
15.12.2019

Adı Soyadı: Sevim AYDIN

Öğrenci No: 801490003

Anabilim Dalı: Bilgisayar ve Öğretim Teknolojileri

Programı:

Statüsü: Y.Lisans Doktora

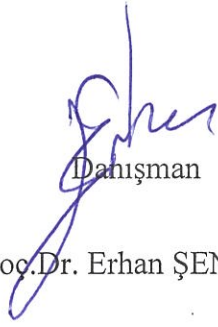

Danışman
(Doç. Dr. Erhan Şengel, 18.12.2019)

YÖNERGEYE UYGUNLUK ONAYI

“Algoritma Öğretiminde Scratch Kullanımının Ortaokul Öğrencileri Üzerindeki Başarıya Etkisi Ve Öğrenci Görüşleri” adlı Yüksek Lisans tezi, Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlanmıştır.

Tezi Hazırlayan

Sevim AYDIN

Danışman

Doç.Dr. Erhan ŞENGEL

Bilgisayar ve Öğretim Teknolojileri Eğitimi ABD Başkanı

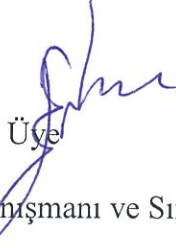

Prof. Dr. Aysan ŞENTÜRK

T.C.

ULUDAĞ ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE,

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı'nda 801490003 numara ile kayıtlı Sevim AYDIN'ın hazırladığı "Algoritma Öğretiminde Scratch Kullanımının Ortaokul Öğrencileri Üzerindeki Başarıya Etkisi ve Öğrenci Görüşleri" konulu Yüksek Lisans çalışması ile ilgili tez savunma sınavı, 22/11/2019 günü 14.00-16.00 saatleri arasında yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin/çalışmasının (başarılı/başarısız) olduğuna (oybirliği/oy çokluğu) ile karar verilmiştir.

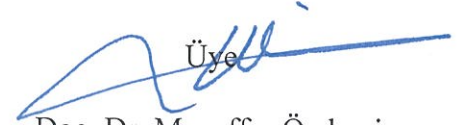


Üye

(Tez Danışmanı ve Sınav Komisyonu Başkanı)

Doç. Dr. Erhan ŞENGEL

Uludağ Üniversitesi



Üye

Doç. Dr. Muzaffer Özdemir

Çanakkale 18 Mart Üniversitesi



Üye

Prof. Dr. Aysan ŞENTÜRK

Uludağ Üniversitesi

Önsöz

Değerli bilgilerini benimle paylaşan, zamanını ayıran ve gerçekten bitirmemi sabırla bekleyen, her zaman ne zaman bitirmeyi düşünüyorsun diye soran sevgili danışmanıma çok teşekkür ediyorum. Eğitimin yanı sıra kendi çocuğu gibi ilgileyen değerli hocama ne kadar teşekkür etsem az olacağına eminim.

Rahmetli anne ve babama beni her zaman kolladıkları ve her konuda arkamda oldukları, beni yetiştirdiği ve bu günlere getirdiği için çok teşekkür ediyorum.

Sevim AYDIN

Özet

Yazar	: Sevim AYDIN
Üniversite	: Uludağ Üniversitesi
Ana Bilim Dalı	: Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı
Bilim Dalı	: -
Tezin Niteliği	: Yüksek Lisans Tezi
Sayfa Sayısı	: xii + 73
Mezuniyet Tarihi	: 2.09.2019
Tez	: Algoritma Öğretiminde Scratch Kullanımının Ortaokul Öğrencileri Üzerindeki Başarıya Etkisi Ve Öğrenci Görüşleri
Danışmanı	: Doç. Dr. Erhan ŞENGEL

ALGORİTMA ÖĞRETİMİNDE SCRATCH KULLANIMININ ORTAOKUL ÖĞRENCİLERİ ÜZERİNDEKİ BAŞARIYA ETKİSİ VE ÖĞRENCİ GÖRÜŞLERİ

Zaman içerisinde değişen ve gelişen öğretim sistemlerinde teknolojinin eğitim müfredatları içerisindeki etkisi oldukça önem kazanmaktadır. Scratch ve diğer öğretim teknolojileri, günümüzde birçok kazanımı öğrencilere kazandırdığı ve birden fazla duyu organına hitap ederek öğrenmeyi kolaylaştırdığı için eğitim müfredatlarına entegre edilmeye çalışılmaktadır. Çalışmada algoritma öğretiminin, geleneksel yöntemle kıyasla, Scratch ile kullanımın başarıya etkisi araştırılmış, öğrenciler detaylı şekilde gözlemlenmiş ve görüşlerine başvurulmuştur. Araştırmada karma model kullanılmış, nicel ve nitel verilerin toplanmış, analiz edilmiş ve yorumlanmıştır. Nitel ve nicel verilerin toplam süreci eş zamanlı olarak uygulandığından araştırmanın deseni ‘yakınsayan paralel desen’ olarak belirlenmiştir. Var olan iki sınıftan rasgele şekilde deney ve kontrol grubu oluşturulmuştur. Kontrol grubunda 14 öğrenci ve deney grubunda 19 öğrenci bulunmaktadır. Başarı testi araştırmacı tarafından geliştirilmiş ve kapsam geçerliliği bir uzman tarafından gerçekleştirilmiştir. Başarı testi 103 kişiye uygulanmış ve güvenilirlik analizine tabi tutularak alfa katsayısı 0,78 olarak

bulunmuştur. Başarı testi, ön test ve son test şeklinde uygulanmıştır. Araştırmadaki her iki gruba da ön test uygulanmış, üç haftalık eğitim verilmiş ve son test uygulanmıştır. Deney grubunda Scratch ile algoritma örnek soruları çözülürken, kontrol gurubunda aynı örnek sorular, geleneksel yol ile çözülmüştür. Nicel araştırma, nitel veriler ile desteklenmiş ve katılımcılarla yarı yapılandırılmış mülakat uygulanmıştır. Bu görüşmede elde edilen nitel veriler gruplanarak anlamlı bulgular çıkarılmıştır. Deney grubu öğrencilerine, Scratch ile yapılan uygulamaları derinlemesine incelemek amacıyla uygulama gözlem formu oluşturulmuştur. Nicel araştırma sonucunda algoritma öğretiminde Scratch kullanımının, geleneksel öğretim yöntemine kıyasla öğrenci başarısı üzerinde istatistiksel olarak anlamlı bir etkisi olmadığı tespit edilmiştir. Ama her iki öğretim yönteminin de öğrenci başarısını arttırdığı, deney grubunda yer alan öğrencilerin başarı ortalamasının kontrol grubundakilerden daha fazla olduğu görülmüştür.

Anahtar Kelimeler: Scratch, eğitim teknolojileri, başarıya etki, algoritma.

Abstract

Author : Sevim AYDIN
University : Uludag University
Field : Computer Education and Instructional Technology
Branch :
Degree Awarded : Master Thesis
Page Number : xii + 73
Degree Date : 10.07.2019
Thesis : The Effect of Scratch Usage on The Success of Secondary Students
In Algorithm Teaching and Student Opinions
Supervisor : Doç. Dr. Erhan ŞENGEL

THE EFFECT OF SCRATCH USAGE ON THE SUCCESS OF SECONDARY STUDENTS IN ALGORITHM TEACHING AND STUDENT OPINIONS

The effect of technology in educational curricula is gaining importance in the changing and developing education systems. Today, Scratch and other instructional technologies are being integrated into educational curricula as it provides many gains to students and facilitates learning by addressing multiple sensory organs. In this study, the effect of algorithm teaching on the success of using Scratch compared to the traditional method was investigated, students were observed in detail and their opinions were consulted. Mixed model was used in the research, quantitative and qualitative data were collected, analyzed and interpreted. Since the total process of qualitative and quantitative data was applied simultaneously, the design of the study was determined as paralel convergent parallel pattern.. Experimental and control groups were randomly formed from two existing classes. There were 14 students in the control group and 19 students in the experimental group. The achievement test was developed by the researcher and the scope validity was performed by an expert. The success test was applied to 103 people and alpha coefficient was found to be 0.78 after reliability analysis. Achievement test was applied as pre-test and post-test. Both groups were given pre-test, three-week training and post-test. In the experimental group, the sample questions were solved by Scratch algorithm, while in the control group the same sample questions were solved in the conventional way. The quantitative research was supported by qualitative data and a semi-structured interview was conducted with the participants. The qualitative data obtained in this interview were grouped and significant findings were

obtained. Application observation form was created for the experimental group students in order to examine the applications made with Scratch in depth. As a result of the quantitative research, it has been found that the use of Scratch in algorithm teaching has no statistically significant effect on student achievement compared to traditional teaching method. However, it was seen that both teaching methods increased student achievement and the average of students in the experimental group was higher than the control group.

Keywords: Scratch, educational technologies, effect on achievement, algorithm

İçindekiler

	Sayfa No
Önsöz.....	iv
Özet	v
Abstract	vii
İçindekiler.....	ix
Tablolar Listesi.....	xi
Şekiller Listesi.....	xii
Simgeler ve Kısaltmalar Dizini	xiii
1.BÖLÜM	1
Giriş.....	1
1.1. Problem Durumu	2
1.2. Araştırmanın Amacı	7
1.3. Araştırmanın Önemi	8
1.4. Tanımlar.....	8
2. BÖLÜM	10
Kavramsal ve Kuramsal Çerçeve	10
2.1. Programlama Eğitimi ve Öğretimdeki Temel Yaklaşımlar	10
2.1.1. Programlama eğitimi.....	10
2.1.2. Yapılandırmacılık	15
2.1.3. İşbirlikli öğrenme.....	17
2.1.4. Öz- düzenlemeye dayalı öğrenme.....	19
2.2. Scratch Programı	20
2.2.1. Scratch programının özellikleri.	22
2.2.2. Scratch programının eğitime entegrasyonu.	24
2.2.3. Yapılan çalışmalar.	25
2.3. Bilişim Teknolojileri ve Yazılım Dersi	28
2.3.1. Programlama dilleri öğretiminde algoritma eğitimi.	30
2.3.2. Kodlama.....	36
3.BÖLÜM	38
Yöntem.....	38
3.1. Araştırmanın Modeli.....	38
3.2. Veri Toplama Araçları.....	39

3.2.1. Başarı testi.....	40
3.2.2. Görüşme formu.....	40
3.2.3. Uygulama gözlem formu.....	40
3.3. Çalışma Grubu.....	43
3.4. Uygulama Süreci	Hata! Yer işareti tanımlanmamış.
3.5. Verilerin Analizi.....	46
3.6. Varsayımlar	47
3.7. Sınırlılıklar.....	48
4. BÖLÜM	49
Bulgular.....	49
4.1. Nicel Araştırma Bulguları: Scratch Programının Algoritma Öğretimindeki Başarıya Etkisi.....	49
4.2. Nitel Araştırma Bulguları	51
4.4. Uygulama Gözlem Formu Verileri.....	56
5. BÖLÜM	58
Sonuç, Tartışma ve Öneriler.....	58
5.1. Sonuç ve Tartışma	58
5.2. Öneriler	64
Kaynakça.....	66
Ekler	74
Ek 1: Başarı Testi	74
Ek 2: Kazanımlar Listesi	74
Ek 3: Scratch Uygulamalarının Haftalık Kazanımlara Göre Örnekleri.....	76
Ek 4: Görüşme Formu	80
Ek 5: Sınıf Uygulamaları Örnekleri.....	81
EK 6: Uygulama Gözlem Formu.....	85
Özgeçmiş.....	86

Tablolar Listesi

<i>Tablo</i>		<i>Sayfa</i>
1	Geleneksel sınıf ile yapılandırmacı sınıfın karşılaştırması	17
2	İşbirlikli öğrenme ile geleneksel öğrenme grupları arasındaki farklar	19
3	İki sayıyı toplamak için verilen algoritmanın çözümün sağlanması	35
4	Girilen sayının kendine gelene kadarki sayıların toplamını veren algoritma çözümünün sağlanması.	36
5	Araştırmanın zaman çizelgesi.	43
6	Oluşturulan grupların cinsiyete göre dağılımları	49
7	Normallik testi sonucu	49
8	Deney ve kontrol gruplarındaki öğrencilerin son teste ilişkin betimsel bulguları ..	50
9	Son test puanlarının karşılaştırılmasına ilişkin bağımsız iki örnek t testi sonuçları	50
10	Birinci temaya ait kodlar	51
11	İkinci temaya ait kodlar	54

Şekiller Listesi

<i>Şekil</i>		<i>Sayfa</i>
1	Scratch programının giriş ekranı.....	21
2	Scratch programı örnek kod bloğu.....	22
3	SOLVEIT	31
4	VB Tutor	31
5	A Simple Karel World.....	32
6	Örnek1'in çözümü	46

Simgeler ve Kısaltmalar Dizini

BTY	: Bilişim Teknolojileri ve Yazılım
MEB	: Milli Eğitim Bakanlığı
N	: İstatistik veri sayısı
p	: Anlamlılık değeri
sd	: Serbestlik derecesi
ss	: Standart sapma
t	: Hesaplanan istatistik t değeri
\bar{x}	: Aritmetik Ortalama

1.BÖLÜM

Giriş

Bilgisayarın gelişimi, teknolojik hayatın bir parçası olarak gelişim ve üretim sürecine önemli katkılar sağlamaktadır. Bilgisayar teknolojileri toplumun her kesimine hitap eden, birbirinden farklı mesleklere hizmet eden hayatın vazgeçilmez bir parçası haline gelmektedir. Bilgisayar ve geliştirdiği teknolojiler günlük hayatı kolaylaştırmaktadır. Geçmişten günümüze pek çok alanda, insan yaşamına, olumlu katkıları olduğu bilinen bir gerçektir. Günlük yaşamın ve iş hayatının vazgeçilmezi haline gelen bilgisayar teknolojilerinin insanların gelişimi için eğitime de entegre edilmesi gerekmektedir. Gelişmekte olan ülkelerin eğitim sistemleri incelendiğinde, karşımıza çıkan önemli bulgulardan birisi de, eğitim sistemlerinin artık hayatımızın vazgeçilmezi haline gelen teknolojiye, uygun hale getirilmiş olmasıdır. Donanımlı okullarda eğitim alması beklenen öğrencilerin, kendilerini pek çok alanda ifade edebilmeleri için eğitim müfredatlarının da teknoloji sürecine ayak uydurarak, akılcı çözümler üretmesi beklenmektedir. Süreç içerisinde toplumda pek çok mesleğin gerekliliklerinden birisi de gelişen teknolojiyi takip ederek, mesleklerin içerisine entegre edilmesi ihtiyacıdır.

Bilgisayar teknolojilerinin, öğrencilere, eğitim sistemleri aracılığıyla, küçük yaşlardan itibaren öğrenmesi ve onların teknoloji geliştirip, bu teknolojiyi yönlendirebilecek donanıma sahip olması beklenmektedir. Eğitim, içinde bulunduğu koşullara ayak uydurmalı, teknolojiye ve çağa bağlı olarak değişen ve gelişen bakış açısına sahip olmalıdır.

Bilgi teknolojileri alanında, her öğrenciye teknolojiden yararlanma ve bilgi-işlemsel düşünme becerisi kazandırma fırsatı sağlamaya çalışılmaktadır. Farklı teknolojik araçları kullanarak ürünler tasarlama, geliştirme, yayınlama ve sunma konuları ele alınmıştır. Bu amaç çerçevesinde Talim Terbiye Kurulu Bilişim Teknolojileri ve Yazılım dersi öğretim programı; problem çözme, programlama alanında algoritma tasarımına ilişkin anlayış geliştirme ve problem çözme becerisi kazandırmayı hedeflemiştir.

Araştırma kapsamında giriş bölümünde araştırma ile ilgili bilgiler, araştırmanın problem durumu, problem cümleleri, araştırmanın amacı, alt problemler, araştırmanın önemi ve tanımlara yer verilmiştir.

1.1. Problem Durumu

Günümüzde bilim ve teknolojideki gelişmeler hızla değişmekte ve yenilenmektedir. Bu gelişmelerin eğitimde kullanılmasıyla eğitsel kavramlar değişmekte ve öğrenme-öğretme sürecini de beraberinde değiştirmektedir. Okur-yazarlık kavramı yerini bilgi ve/veya bilgisayar okur-yazarlığına, alışlagelmiş öğretim ise yerini bilgisayar destekli öğretime bırakmaya başlamıştır (Gültekin, 2006). Bilgisayar destekli eğitim öğrencilerin çoklu zeka kuramına göre matematiksel zekalarını çeşitli yazılım programlarının desteği ile geliştirmektedir. Gelişen teknoloji ile birlikte dünyanın farklı yerlerinde programlama eğitimi verilmekte ve programlama dili eğitiminin önemli olduğu birçok kaynakta vurgulanmaktadır. İnsanların ileri derece düşünme yetenekleri olan problem çözme, mantıksal-matematiksel düşünme, eleştirel düşünme ve yaratıcı düşünme becerilerinin gelişimine katkıda bulunur. Her yıl düzenli olarak ülkeler arasında yapılan belirli kriterlere dayanan eğitim karşılaştırmalarında ülkemizdeki öğrencilerin sayısal ve matematiksel zekalarının ön plana çıkmadığı ve verilen eğitimlerin eksik kaldığı görülmektedir. Bu süreçte programlama dilinin uygulamalı olarak öğretilmesinin disiplinlerarası uygulamalarının öğrenciler üzerinde etkili bir yöntem olacağı öngörülmektedir.

Bilgisayar programlama dili ve onun sistematığıne hakim olmak farklı becerilerin aynı anda kullanılmasını ve aktarılmasını sağlamaktadır. Özellikle öğrencilerin, uygulaması beklenen program yazılımları, problem çözme becerilerini geliştirmeye katkı sağlarken mantıksal ve matematiksel zekalarını çalıştırarak analitik düşünme becerilerinin artmasına katkı sağlamaktadır (Korkmaz, 2012). Özyurt ve Özyurt (2015)'a göre öğrenciler programlama dillerini öğrenmeye yatkındırlar. Bu yatkınlığın geliştirilmesi için öğrencilere

yaratıcılıklarını kullanabilecekleri, merak duygularını pekiştirebilecekleri, severek ve isteyerek yapabilecekleri etkinliklerin belirlendiği öğrenme ortamları sunmak oldukça önemlidir. Alanda kullanılan öğretim yöntem ve tekniklerinin problem çözmeye dayalı olması ve eleştirel düşünme becerilerini geliştirmesi beklenmektedir. Sınıf ortamında akranlarıyla sürekli iletişim halinde olan öğrenci pek çok duyu organına hitap eden etkinliklerle ve akranlarıyla kurduğu iletişim ile öğrenme sürecine daha kolay entegre olurken, öğrencilerin öğrenme becerisi de kolaylaşmaktadır (Coşar, 2013).

Bilgisayar programlama dilini öğrenmek tıpkı yabancı bir dil öğrenmek gibidir. Özellikle küçük yaşlardan itibaren bir programlama dili öğrenmek akıcılık ve akılda kalıcılık bağlamında daha etkili olmaktadır. Bu süreçte programlama teknikleri öğrencilerin eğitim sürecinde ilkökul- ortaokul sürecinde başlamalıdır. Erken yaşlarda başlayacak programlama eğitimi ilerleyen yıllarda analitik düşünme yeteneğinde pozitif farklılıklara neden olmaktadır. Analitik düşünme bir konuyu veya bir problemi genelden özele ayrıntılandırarak ilgili bilgileri değerlendirme durumudur. Konuların çözümlenerek genelden özele ayrıntılı hale getirilmesi ve sorgulanması parçaların tek tek ele alınarak bu sonuçların genele entegre edilmesi durumudur. Analitik düşünme yeteneğine sahip kişiler problemlere bakış açıları ve çözüm bulma yolları ile diğerlerinden ayrılacak yeteneklere sahiptirler (Demirer, 2016).

Özmen ve Altun (2014) yaptıkları bir araştırma sonucunda üniversite düzeyindeki öğrencilerin programlama dillerini geliştirmeyi ve öğrenmeyi zor buldukları tespit edilmiştir. Yapılan araştırmada nitel araştırma yöntemlerinden fenomenoloji kullanılmıştır. Programlamanın mantığı ve hataları ayıklama konularında zorluk yaşayan üniversite öğrencilerinin daha küçük yaşlardan itibaren eğitim ortamlarında programlama dillerini öğrenmeleri gerekmektedir. Üniversite düzeyinde programlama dilinin öğrenilmesi, süreçte pek çok sıkıntıyı ortaya çıkarmaktadır. Bu süreçte en büyük zorluklar öğrencilerin pratik

eksikliklerinden ve algoritma oluşturmamalarından kaynaklanmaktadır (Özmen & Altun, 2014).

Algoritma, verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların hiçbir yoruma yer vermeksizin açık, düzenli ve sıralı bir şekilde söz ve yazı ile ifadesidir (MEB, 2013). Programlama eğitimi verilmeden önce algoritma kavramı anlatılmalıdır. Çünkü tüm programlama dillerinin temeli algoritmaya dayanır. Bir problemin çözümünde kullanılan algoritma adımları ile öğrencilere programlama becerisi, mantıksal düşünme ve problem çözme becerisi kazandırıldığı düşünülmektedir.

Programlama dillerinin öğretimi ve eğitime entegre edilmesi tıpkı lisan öğrenimi gibi küçük yaşlardan itibaren başlamalıdır. İlkokul ve ortaokul öğrencilerine yönelik programlama dillerinin öğretilmesi, tasarım yapabilecekleri ortamların hazırlanması, bilgisayarın etkin ve verimli kullanımının öğretilmesi gelecek süreçlerde öğrencilerin birey olma yolunda çeşitli bilişsel beceriler kazanmalarına katkı sağlamaktadır (Akpınar & Altun, 2014). Bu amaçlar doğrultusunda Milli Eğitim Bakanlığı, eğitim-öğretim yılı başında bilgisayar ve programcılık öğrenilmesini temel alan dersin içeriğini ve niteliğini büyük ölçüde değiştirmektedir. Bilişim Teknolojileri ve Yazılım dersi kapsamında yayınlanan çerçeve programın öğrenme alanlarının dördüncü basamağında, “Problem Çözme, Programlama ve Özgün Ürün Geliştirme” yer almaktadır (Talim ve Terbiye Kurulu, 2013). Çerçeve programda yer alan bu basamakta, problem çözme yaklaşımları, programlama, algoritma ve strateji geliştirme, yazılım projesi geliştirme, uygulama ve yaygınlaştırma standartları ve bu standartlar içerisinde düzeyler belirlenmiştir (TTKB, 2013). 2016 yılında Talim ve Terbiye Kurulu 5. ve 6. sınıflarda zorunlu, 7. ve 8. sınıflarda seçmeli ders olarak kodlama eğitimini algoritmik ve bilişimsel düşünme becerilerini arttırmaya yönelik olarak Bilişim Teknolojileri ve Yazılım Dersi içerisine eklenmiştir. Bu ders zorunlu hale getirilerek 2018-2019 yılından itibaren 5.6.7. ve 8. sınıflarda uygulanmaya başlanmıştır. Talim ve Terbiye Kurulu, 2018 yılında yayınladığı

Bilişim Teknolojileri ve Yazılım Dersi programı kapsamında algoritma tasarlama ve öğretimine 1, 2, 3 ve 4. sınıflarda, kazandırılması hedeflenen becerilere yer vermektedir. İlkokul öğrencilerinin algoritma tasarımına ilişkin anlayış geliştirmeleri programın özel amaçları arasında yer almaktadır (Talim ve Terbiye Kurulu, 2018). Eğitim sistemimizde programlama alanında alınan kararlara rağmen uygulama konusunda zorluklar yaşanmaktadır. Yenilikçi yaklaşımlar yerine algoritma ve strateji geliştirme standardında öğretim geleneksel yöntem ile öğrencilere öğretilmektedir.

Belirli eğitim düzeylerinde ve yaş gruplarında öğrencilerin öğrenme biçimleri ve alışkanlıkları farklılık gösterebilmektedir. Üniversite düzeyinde bir erişkin genç ile ilkökul düzeyinde eğitim alan bir çocuk arasında algı farkındalık seviyesi farklıdır. Eğitim programları tasarlanırken özellikle küçük yaş grupların da somut kavramların kullanılması, beş duyu organıyla algılayabilecekleri düzeyde programın planlanması oldukça önemlidir. Ersoy ve diğerleri (2011), programlamayı öğrenme sürecinin daha anlaşılabilir hale gelebilmesi adına soyut kavramların somutlaştırılmasının olanaklı olduğunu ifade etmiştir. Ayrıca yazarlar öğretim sürecinde soyut kavramlar ile çalışılmasının, öğrencilerin yazdıkları kodun çalışır halini fiziksel olarak gözlemleyememelerine neden olduğunu dile getirmişlerdir.

Özyurt ve Özyurt (2015), yapılan çalışma kapsamında programlama dillerinin öğretiminin oyunlara dayandırılabilceğini öne sürmüştür. Kazanılması istenilen kazanımlar doğrultusunda gelişime açık bir öğrenme ortamı oluşturarak, özellikle küçük yaş gruplarındaki öğrencilerin programlamaya karşı tutumlarını ve bilişsel etkilerini araştıran çalışmalar yapılabileceğini dile getirmiştir. Bu oyunla öğretim ihtiyacı dünyanın farklı ülkelerine ait eğitim sistemlerinde de ortaya çıkmıştır. Oyun temelli programlama ile öğrenme ve eğitimi bir araya getiren ise Scratch olmuştur. Oyun temelli eğitim süreci, teknik bilgilerden arındırılarak, öğrencilerin programlama kazanımlarını gerçekleştirebilecekleri hale gelmesini sağlamaktadır (Doğan & Kert, 2016). Scratch, Massachusetts Institute of

Technology (MIT) tarafından geliştirilen görsel bir programlama dilidir. Bu dili kullanarak öğrenciler oyun programlayabilir, kendi animasyonlarını oluşturabilir ve yaratıcılıkları dâhilinde farklı çalışmalar üretebilirler. Scratch ortamı aracılığıyla programlama temellerini öğrenerek, kendi oyunlarını geliştirebilirler. Bu kapsamda Fal ve Çağıltay (2013), Scratch'in, programlamaya başlarken programlama mantığı ve algoritmik düşünme yeteneği kazandırdığını ortaya çıkarmıştır. Scratch'in, çok karmaşık ilişkileri analiz edip, karşılaşılan problemleri küçük parçalara bölerek sorunu çözmeye ve programlama dili ile kullandığımız temel yapıları en etkili şekilde öğrenmeye yardımcı olacağı düşünülmektedir.

Disiplinlerarası çeşitli uygulama alanları yaratan Scratch öğrencilere özgür bir sanal uygulama ortamı oluşturmaktadır. Bu uygulama ortamları sadece öğrencinin sayısal zekasını değil aynı zamanda uygulama alanına göre görsel sanatlar, müzik gibi pek çok sözel-görsel alanla ilgili yaratıcılığının gelişimine de katkı sağlamaktadır. Maloney, Burd, Kafai, Rusk, Silverman ve Resnick (2010), öğrencilerin bireysel olarak, animasyonlu hikayeler, oyunlar ve interaktif sanat gibi Scratch projeleri oluşturabilmeleri, onlara teknolojik akıcılık, matematiksel ve problem çözme becerileri geliştirmelerine katkı sağlamaktadır.

Scratch ile öğrenciler oyun, animasyon vb. tasarımında karşılarına çıkacak yapılandırılmamış problemleri çözmek için çaba harcayacaklardır. Yapılandırılmamış problem ortamının Scratch ile sağlanabileceği düşünülmektedir. Yapılandırılmamış problemlerin çözümü biraz zor olabilir, çünkü birden fazla disiplindeki bilgilerin kullanılması gerekmektedir. Dolayısıyla, Scratch programı farklı disiplinlerdeki (matematik, bilgisayar bilimi, dil bilimi, sosyal bilgiler gibi) bilgilerin öğrenimini de desteklemektedir (Scratch, 2018).

Oluk, Korkmaz ve Oluk (2018), algoritma ve bilgi-işlemsel düşünme becerilerini geliştirmede Scratch kullanımının etkisinin olup olmadığını inceleyen çalışmalarında, deney grubuna Scratch ile algoritma öğretimi, kontrol grubuna ise normal yol ile algoritma eğitimi

gerçekleştirmişlerdir. Araştırma sonucunda, Scratch kullanımının algoritma geliştirmek için kullanılabilir bir öğrenme aracı olduğu sonucunu ortaya çıkarmışlardır.

Konuyla ilgili araştırmalara bakıldığında, çocuklara programlama becerileri öğretmek için farklı görsel programlar geliştirilmiştir. Çocuklar ve programlama öğrenmek isteyenler için hazırlanmış görsel açıdan zengin ve kullanımı kolay Scratch, Alice, Microsoft Small Basic gibi programlama araçları bulunmaktadır (Ward ve diğerleri, 2010).

Bireylerin karşılaşılan problemler için farklı çözüm yolları bulmaları ve en kısa olanını seçmeleri programlama becerileri açısından önemlidir (Yükseltürk ve Altıok, 2015). Çözümün en kısa yolunu bulabilmek için algoritma mantığının iyi anlaşılması gereklidir. Bu çalışma ile ortaokul öğrencilerine Scratch kullanılarak algoritma eğitiminin verilmesinin, öğrencilerin algoritma çözümü üzerindeki başarısına etkisinin olup olmadığının incelenmesi ve öğrencilerin konuyla ilgili görüşlerinin alınması amaçlanmıştır. Scratch programı ile farklı bir eğitim deneyimi alan öğrencilerin süreçte kodlama eğitimini de bu uygulama üzerinden almasının bilgisayar okur-yazarlığı açısından manidar bir etki yaratması beklenmektedir.

Scratch programının uygulamaları ve yapılan akademik çalışmalar göz önünde bulundurularak çalışmanın problem cümlesi şu şekilde oluşturulmuştur; ‘Ortaokul öğrencilerine algoritma öğretiminde, Scratch kullanmanın başarıya etkisi nedir? ve öğrencilerin Scratch ile algoritma öğrenimine yönelik görüşleri nelerdir?’ olarak belirlenmektedir.

1.2. Araştırmanın Amacı

Araştırmanın amacı; Scratch programı ile algoritma öğreniminin öğrencilerin başarılarına etkisini tespit etmek ve öğrencilerin genel görüşlerini ortaya çıkarmaya çalışmaktır. Bu amaç doğrultusunda aşağıdaki hipotezler ortaya atılmıştır;

- Algoritma öğretiminde, Scratch kullanmanın başarıya katkısı vardır.
- Algoritma öğretiminde Scratch kullanmak, başarının artırılmasında etkilidir.

- Scratch, algoritma öğretimini kolaylaştırır.

1.3. Araştırmanın Önemi

Günümüzde algoritma eğitimi, bireylere geleneksel yöntemler ile öğretilmekte, soyuttan somuta bir öğretim anlayışı benimsenmektedir. Arabacıoğlu ve diğerleri (2007), yaptıkları bir çalışmada; algoritma eğitimini somutlaştırmak için bir uygulama dili tasarlamışlar ve bu uygulama dilini kullanılarak programlamanın daha ekili öğretilmesi için teoride verilen bilgilerin ve yaklaşımların uygulanarak etkililiğine bakmışlardır. Oluşturulan algoritmaların çözümünü ise doğruluk tablosu adı verilen yöntem ile görebilmeyi sağlamışlardır. Bu yöntem öğrencinin tek başına etkili bir şekilde çalışabilmesine katkı sağlamıştır. Bu çalışma kapsamında bilgi teoriden pratiğe aktararak sağlanmış ve etkili bir öğretim süreci uygulanmıştır. Araştırmada, etkili öğretim sürecinin başarısının artırılması açısından önemli olduğu vurgulanmıştır (Arabacıoğlu ve diğerleri, 2007).

Erken yaşta algoritma eğitimi, programlama öğrenimini kolaylaştırmakta ve bu eğitimin animasyon ve oyun yoluyla verilmesi de öğrenmeyi olumlu yönde etkilemektedir. Literatürde, hem animasyon hem de oyun yolu ile programlama öğretimine katkı sunduğu belirtilen Scratch programının algoritma öğretiminde etkililiğinin araştırıldığı bu çalışmanın hem öğretmenlere hem de araştırmacılara ilerleyen dönemlerde katkı sağlayacak bir rehber olması hedeflenmektedir.

1.4. Tanımlar

Scratch: Massachusetts Institute of Technology (MIT) tarafından geliştirilen görsel bir programlama dilidir. Bu dil özellikle 8-16 yaş grubu öğrenciler için oluşturulmuştur. Bu dili kullanarak öğrenciler oyun programlayabilir, kendi animasyonlarını oluşturabilir ve yaratıcılıkları dâhilinde farklı çalışmalar üretebilirler (Collobert ve diğerleri, 2011).

Yapılandırmacı Öğrenme Yaklaşımı: Yapılandırmacılık, bilginin doğası, nasıl bildiğimiz ve bilginin yapılandırılma sürecidir (Demirel, 2002).

Kodlama: Bir sistemden/makineden/bilgisayardan yapılması istenen şeylerin sistematik bir şekilde bu araçlara anlatılması yöntemidir. Kodlama belirli programlar aracılığıyla bilgisayar üzerinden komutlar verilerek bu komutların belirli bir düzende uygulanabilir olması durumudur (Sayın ve Seferođlu, 2016).

Algoritma: Belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yoldur (MEGEP, 2014).

Programlama: Belirli komutların sıralanmasına ve bilgisayar ortamında modellenmesi işine programlana denmektedir (Kesici ve Kocabaş, 2007).

2. BÖLÜM

Kavramsal ve Kuramsal Çerçeve

Günümüzde teknolojinin hızlı gelişmesi ve bu teknolojilerin eğitim ortamındaki yansımalarına ve uygulamalarına olan ihtiyaç her geçen gün artmaktadır. Süreç içerisinde yeni programların yazılması ve geliştirilmesi ihtiyaç haline gelmektedir. Araştırmanın bu bölümünde gelişen teknolojiyle birlikte artan eğitim ihtiyaçları ve teknolojinin eğitim ortamındaki yansımaların üzerinde durularak kavramsal çerçeve oluşturulmuş ve çalışmanın alt yapısı ile ilgili kuramsal bilgilere yer verilmiştir.

2.1. Programlama Eğitimi ve Öğretimdeki Temel Yaklaşımlar

Programlama eğitiminin, öğrenciler tarafından öğrenilmesi için alanın yapılandırıcılık, işbirliğine dayalı öğrenme ve öz-düzenlemeye dayalı öğrenme yaklaşımları açılardan ele alınıp değerlendirilmesi yapılmıştır.

2.1.1. Programlama eğitimi.

Program, belirli komutların bilgisayar ortamında yazılarak işleyişinin sağlanmasına dayanan yapıdır (Coşar, 2013). Gelişmekte olan ülkelerin eğitim sistemlerinde yeni nesillerin halihazırda bulunan programları kullanmak yerine özellikle yeni programlar üretmeye yatkın olmaları gerekmektedir. Gelişen teknolojiye ayak uydurabilmenin en iyi yolu onu kullanmaktan çok teknoloji üretkenliğini sağlamaktır. Son dönemlerde yazılım alanında kendini geliştirmiş, eğitimlerini tamamlamış ve üretken insanların sayısı her geçen gün artmaktadır. Yaratıcı ve üretken sürece dahil olan kişiler özellikle gelişmişlik seviyesini artırarak ülke geleceğinde söz söyleme hakkına sahip bireyler olmaktadır. Bu nedenler göz önünde bulundurulduğunda eğitimini bu alanda tamamlamış bireyler dışında farklı alanlarda görev ve sorumlulukları olan kişilerin de özel ilgi alanlarına yazılım konusu dahil olmaktadır ve bu süreçte onların da temelden eğitim alması önemli görülmektedir. Programlama eğitimi,

yazılım çalışmalarının uygulanmasını ve sürdürülebilmesini sağlayan önemli bir eğitim olarak karşımıza çıkmaktadır (Kert ve Uğraş, 2009).

Bilgisayarın işlem yapabilmesi için belirli komutlar serisine ihtiyaç vardır. Belirli komutların sıralanmasına ve bilgisayar ortamındaki modellenmesi işine programlama denilmektedir (Kesici ve Kocabaş, 2007). Bilgisayar programları hazırlanırken belirli aşamalardan geçmektedir. Öncelikli olarak problemin tanımının yapılması, probleme dair çözümlerin üretilmesi, programların yapım aşamasının sonunda yorumlanması ve düzenlenmesi gibi aşamalar önemli görülmektedir. Bu programlama aşamalarının açıklamalarına aşağıda yer verilmektedir (Kesici ve Kocabaş, 2007).

- *Problemin tanımı:* Problem durumunun ortaya çıkış aşamasından itibaren tanımlanıp anlaşılabilir hale getirilmesi gerekmektedir. Bu kısımda araştırmalar sonucunda problemin ortaya koyulması gerekmektedir.
- *Çözüm yolunun belirlenmesi:* Tanımlama aşamasından sonra çözüme ulaşmak adına gerekli hazırlıklar yapılması ve çözüme gidecek işlemlerin algoritmasının hazırlanması gerekmektedir. Bu algoritmalar sonucunda da akış şeması oluşturularak harekete geçilmesi beklenir.
- *Programın kodlanması:* Bu aşama problemin, çözüm yolundan sonraki durumdur. Programın yazılması için, programlama dilinin kurallarına dikkat edilmeye başlanmaktadır. Programlama kelime anlamıyla belirli şartlara ve düzene göre yapılması öngörülen işlemlerin bütünü anlamına gelir. Programlama bilgisayara ya da elektronik devre ve mekanik sistemlerden oluşan düzeneklere bir işlemi yaptırmak için yazılan komutlar dizisinin bütünü veya bir kısmı olarak tanımlanır.
- *Programın yorumlanması ve derlenmesi:* Bu aşamada yazılan kodlar makine diline çevrilerek kullanıma hazır hale getirilmektedir.

- *Programdaki hataların belirlenmesi ve giderilmesi:* Programı kullanmaya başlamadan önce içindeki yazım ve mantık hatalarının tespit edilip ve giderildiği aşamadır. Son olarak program tekrar makine diline çevrilir ve çalışması denir.

Öğrenciler programların mantığını kavradıkları zaman tasarım sürecine dahil olarak yeni fikirler geliştirebilirler. Programlama sürecinde hatalarını fark ederek onları telafi edebilme güdüsü geliştirmeleri oldukça önemlidir. Programlama eğitiminin öğrenciler üzerindeki etkileri ve faydaları oldukça fazladır. Yapılan çalışmalarda saptanan farklılıkları şu şekilde sıralandırabiliriz (Karabak ve Güneş, 2013; Akpınar ve Altun, 2014);

- Öğrenciler okul içerisinde kodlama ve program yapabilecekleri alanları daha fazla kullanabilmektedirler.
- Geniş hayal güçlerini açığa çıkarabilecekleri bir ortam oluşabilir.
- Problemlerin çözümünde ve başarıya ulaşılmasında planlama ve sonuç belirleme yetenekleri gelişebilir.
- Okulda diğer alanlara karşı motivasyonları ve güdülenmeleri artabilir.
- Hafızalarını daha fazla kullanabilmelerine katkı sağlayabilir.
- Öğrencilere analitik düşünme becerileri kazandırırken, pek çok alanda başarı seviyelerinin artmasını destekler.
- Ürüne dönük büyük projeler yapma, küçük projelerin entegrasyonu ile karmaşık problemlere çözüm üretme alışkanlığı edinmelerini sağlar.
- İşbirlikli çalışma, öğrenme becerileri, yaparak öğrenme ve bilgisayara öğretmek öğrenme alışkanlıkları ve kültürü geliştirilebilir

Programlama eğitimi, öğrencilerin bilgisayar üzerinde mantık yürütmelerini sağlamanın yanında, disiplinlerarası öğrenme ortamını da sağlayarak, fen ve matematik alanlarına katkıda bulunmakta ve geliştirmelerini sağlamaktadır. Öğrencilerin en büyük başarı

ölçütlerinden birisi de öğrenilen problem çözme becerisinin günlük hayata entegre ediliyor olmasıdır.

Programlama eğitimi günümüz dünya ülkelerinin eğitim politikalarında oldukça önemli bir yere sahiptir. Eğitim programlarına yazılım ağırlıklı müfredatlar ekleyen ülkelerin sayısı her geçen gün artmaktadır. Estonya, gelişen eğitim sistemiyle 2012 yılının eğitim-öğretim döneminden itibaren yedi yaşından itibaren programlama eğitimine önem vererek müfredatlarına eklemiştir (Demirer & Nurcan, 2016). Güney Kore, dünya ülkeleri arasında bilime ve teknolojiye verdikleri önem doğrultusunda Bilim ve Gelecek Planlama Bakanlığı tarafından ilkokuldan itibaren eğitim sistemi içerisinde yazılım derslerinin zorunlu olacağını, 2017 yılında ilkokulların, 2018 de ise liselerin yazılım eğitimi almaya başladıklarını açıklamıştır (Özçakmak, 2014). İngiltere, Milli Eğitim Bakanlığı okullarda uygulanan bilgi ve iletişim teknolojileri dersinin müfredatının teknolojik gelişmeleri ön plana çıkarılarak çocukların yaratıcılıklarını geliştirebilecekleri ve eğlenerek öğrenebilecekleri bir ortamda olacağını söylemiştir (Burns, 2012). Bunların dışında İngiltere, kodlamayı eğitim sistemi içinde öncelikli öğrenme alanlarından biri olarak kabul edecek bir takım çalışmalar yapmaktadır. Bu ülkede 2014 yılı ‘kodlama yılı’ kabul edilmiş ve düzenli olarak Avrupa Kod Haftası kutlanılıp bu eğitime farkındalık yaratılmak istenmiştir. Bu etkinlikler çocukların ve gençlerin kodlama öğrenmesini teşvik ederken girişimcilik ruhlarını da güdülemektedir. Gelecekte yazılımcı ve programcı ihtiyacının artacağına dair araştırma verileri bulunmaktadır. Bu süreçte gelişen teknolojiye ayak uydurmak isteyen ülkelerin eğitim politikalarında kodlama ve yazılım eğitimi öncelik kazanmaktadır. Dünya ülkelerinde gelişmelere yetişebilmek ve ihtiyacı karşılamak adına lisans ve lisansüstü eğitimlerin dışında daha alt yaş gruplarına hitap eden kodlama konseptli okullar hizmet vermeye başlamıştır. Okulların ve hükümetlerin dışında kodlama eğitimlerinin gelişmesi için bireysel gelişmelerin örneklerine de rastlanmaktadır. Uygulanmakta olan müfredat programında vakit kaldığı ve imkanlar

oluşturulduğu takdirde bireysel farkındalık yaratan öğretmenler görmek de mümkündür. Örneğin Fransa’da, iki öğretmen tarafından kodlamanın öneminin fark edilmesiyle okullarında Scratch programının kullanılmaya başlanması buna bir örnektir (Madda, 2013). Yapılan çalışmalar programlama dilinin tıpkı lisan öğrenmek gibi küçük yaşlardan itibaren öğrenilmesi gerektiğine dikkat çekmektedir (Özçınar ve diğerleri, 2016).

Dünya genelinde oluşan kodlama ve yazılım furçasına Türkiye de gerekli koşulları hazırlayarak katılım göstermektedir. Bilişim ve teknoloji alanındaki firmalar bu anlamda gerekli iş gücü konusunda sorunlar yaşamaktadırlar. Bu konuda yetiştirilmiş eleman sayısı azdır. Bilişim ve Teknolojileri Dersi bu süreçte okullardaki ihtiyacı karşılamak amacıyla içerik ve müfredat olarak yeniden düzenlenmektedir. 2012 yılında yayınlanan 69 sayılı karar ile Bilişim Teknolojileri dersinin adı değişerek “Bilişim Teknolojileri ve Yazılım” dersi olmuş ve ders müfredatına yazılımla alakalı konular eklenerek 5. sınıftan itibaren öğrencilere uygulamalı bir ortamda bu dersin öğretilmesine karar verilmiştir (TTKB, 2012). Son dönemlerde gelişen dünya ülkelerinin eğitim sistemlerine fazlaca yer verdikleri kodlama ve yazılım programlarının müfredata dahil edilmesiyle birlikte gelişen teknolojiyi takip eden bireyler yetiştirilmesi hedeflenmektedir (TTKB, 2012). Okullarda yapılan bilişim teknolojileri derslerinin dışında, sivil toplum kuruluşları ve özel sektördeki firmalara bağlı projeler ve etkinlikler ile kodlama eğitimlerine dikkat çekmek amaçlanmıştır. Ayrıca özel yetenekli bireylere eğitim veren Bilim ve Sanat Merkezlerinde (BİLSEM) de kodlama eğitimleri arttırılarak, özel yetenekleri olan öğrencilere yönelik fazladan çalışmalar planlanmaktadır. Bu çalışmalar için BİLSEM laboratuvarları ve öğretmenleri donanımlı hale getirilmektedir (Çoban, 2018).

Genel hatları itibariyle programlama dili günümüz eğitim sisteminin bir parçası olduğu gibi etkileri de hayatımızın pek çok yönünde karşımıza çıkmaktadır. Program yazabilmek bilgisayar üzerinde komutlar vererek işlem yapmaktan çok daha fazla şey

kazandırmaktadır. Programlama, kişilere problemi tespit etme, problemin çözüm yollarını araştırma ve sistematik düşünme yeteneği kazandırmanın yanı sıra olaylar arasındaki bağlantıları ilişkilendirme gibi pek çok bilişsel özellikleri kazandırmaktadır (Fesakis & Serafeim, 2009).

2.1.2. Yapılandırmacılık.

Yapılandırmacılık literatürde, yapısalcılık, bütünleştirmecilik ve kurmacılık gibi değişik isimlerle karşımıza çıkmaktadır. Günümüzde ise yapılandırmacılığı, bilginin doğası, nasıl bildiğimiz ve bilginin yapılandırma süreci şeklinde açıklayabiliriz (Demirel, 2002).

Öğrenme ortamlarının etkililiğini arttırmak için bu ortamların verimli kullanılması ve öğrenciler açısından bu verimin nasıl arttırılabileceğinin bilinmesi gerekmektedir. Öğretim uygulamalarının iyileştirilmesi ve içeriğinin doldurulması için öğretim kuramları oldukça önemlidir. Geçmişten günümüze eğitime davranışçı yaklaşımın hakim olması öğrencilerin yaratıcılıklarını geliştirmeye engel olmuştur. Bu bağlamda günümüzde yaratıcılıkları ortaya çıkarmayı hedefleyen, öğrencinin merkezde olduğu, öğretmenin rehber konumda olduğu yapılandırmacılık ön plana çıkmıştır. Yapılandırmacı yaklaşımın kökleri 1960'lı yıllarda Kant'a dayanmaktadır ve Bruner tarafından sistematikleştirilmiştir. Yapılandırmacı eğitim kuramında bilgi ve birey birbiri ile içe içedir (Demirel, 2011). Yapılandırmacı anlayışta bireyin bilgiyi içselleştirmesi için öğrenme ortamının da ona göre düzenlenmesi gerekmektedir.

Yapılandırmacı yaklaşımın uygulamalarında öğretmen ve öğrencilerin önemli olduğu kadar sınıf ortamının da öğrenmeye hazırlanması oldukça önemlidir. Bilgisayarla öğrenme ortamlarında ve kodlama eğitiminde, yapılandırmacı yaklaşımdan yola çıkılarak edinilen bilginin gerçek hayattaki durumlarla bağdaştırılması oldukça önemlidir. Etkili öğrenme durumlarında öğretmen, rehber konumda olarak öğrencileri istenilen bilgiye ulaştırmakla sorumludur. Bu yaklaşım tarzında öğretmen bilgiyi hazır olarak sunmaz ve öğrencinin kendi

imkanlarıyla bilgiye ulaşmasına öncülük eder. Söz konusu yaklaşım, sınıf ortamında da öğrencilere programda belirtilen hedefleri kazandırmakla birlikte günlük hayatta yaşanan problemlere çözüm getirilmesi de hedeflemektedir. Öğretmen bu rehberlik sürecinde öğrencilerinin açıklarını kapatarak hatalarını telafi etmeleri noktasında onlara yardımcı olmalıdır. Yapılandırmacı yaklaşımda öğrenci etkinliklerde ve dersin gidişatında etkin olan konumdadır. Tablo 1’de yapılandırmacı sınıf ile geleneksel sınıfın karşılaştırması verilmiştir (Brooks ve Brooks, 1993; Akt. Deryakulu, 2001).

Tablo 1

Yapılandırmacı sınıf ile geleneksel sınıfın karşılaştırması (Brooks & Brooks, 1993).

<u>Yapılandırmacı Sınıflar</u>	<u>Geleneksel Sınıflar</u>
Eğitim programı dahilinde bilginin önemli kavramlar üzerinden yenilenmesi esastır.	Eğitim programı dahilinde temel bilgi ve becerilerin kazanılması önemli görülür.
Öğrencilerin öğrenme problemlerinin öncelikli durum olması önemlidir.	Yapılan programın harfiyen uygulanması önemlidir.
Programdaki etkinlikler öğrencilerin materyalleri üzerinde bilgi kaynaklarını pekiştirmelerine dayanır.	Programdaki etkinlikler yazılı yayınların çerçevesiyle sınırlıdır.
Öğretmenler öğrencilerin görüşlerini göz önünde bulundurur ve yönlendirirler.	Öğretmenler genellikle sunuş yoluyla öğretim tekniğini benimserler.
Öğrenmenin değerlendirilmesi yapması öğrenme işinin bir parçasıdır. Bu süreçte öğrencinin çalışmaları gözlemlenir.	Ölçme ve değerlendirme sistemi aslında yazılı sınavlara dayanmaktadır.
Öğrenciler genellikle grup ve akran çalışmasına yatkın olurlar.	Öğrenciler bireyseldir ve grup çalışmasına yatkın değildirler.
Öğrenciler, gerçek dünyaya ilişkin kuramlar oluşturabilir.	Öğrenciler, öğretmenin, verdiği bilgileri yazacağı boş bir levha olarak görülür.

Yapılandırmacı yaklaşımda öğrenciler, kendilerini doğru bir şekilde ifade ederek etkili öğrenme ortamında gerekli kazanımlara sahip olmaktadır. Hedeflere ulaşılması istenilen durumlarda çalışma grupları oluşturularak işbirliğine dayalı öğrenme ortamının gereklerini yerine getirilmektedir. Gelenekselci öğrenme ortamıyla yapılandırmacı yaklaşım kıyaslandığında öğrenme ortamlarında yaratıcılığın ve özgünlüğün ön plana çıktığı gözlemlenmektedir. Geleneksel sınıflarda öğrenme ortamlarında özgünlük gerektiren kodlama eğitiminin gerçekleşmesi uygun görülmemektedir. Kodlama ve programlama eğitimi yaratıcılığı tetikleyen yapılandırmacı yaklaşıma daha uygundur (Fesakis ve Serafein, 2009)

2.1.3. İşbirlikli öğrenme.

İşbirlikli öğrenme, öğrencilerin kazanımlarının ortak olmasından yola çıkılarak grup arkadaşlarının da öğrenmelerine katkı sağlamayı amaçlar. Bu öğrenme modelinde grup

çalışmasının gereklerinin önemini kavrayan öğrenciler, sadece kendileri için değil grup çıkarları adına hareket etmektedirler ve sürece katılmaktadırlar. İşbirlikli öğrenme, öğrencilerin aynı ortamda ve koşullarda motivasyonlarını artırarak saygı ve birlikte çalıştıkları arkadaşlarının fikirlerine ve yaptıklarına saygılı ve hoşgörülü olarak öğrenmelerine yardımcı olan (Saban, 2004; Senemoğlu, 2001), yaşlılarıyla birlikte daha eğlenceli bir öğrenme ortamı (Tan ve diğerleri, 2002) olarak ifade edilebilir. İşbirlikli öğrenme; öğrencilerin sınıfta ve atölyelerde karma çalışma grupları oluşturularak ortak kazanımlar doğrultusunda birlikte öğrenmelerine kolaylık sağladığı, iletişim becerilerini geliştirdikleri ve süreç boyunca öğrencilerin aktif oldukları bir öğrenme yöntemidir (Doymuş, 2005). İşbirlikli öğrenmenin geleneksel öğrenmeden farklılaştığı en önemli nokta öğrencilerin bu sürece sadece kendileri için değil grup üyeleri ve akranları için de dahil olmalarıdır. Grup içerisindeki her bireyin sorumluluğu hedefe ulaşma noktasında artmakta ve önem kazanmaktadır. Gömleksiz (1993), Miller (1989) ve Johnson, Johnson ve Holubec (1990, 1994)'in işbirlikli öğrenme ve geleneksel öğrenme grupları arasındaki farklara ilişkin görüşleri Tablo 2'de verilmiştir.

Tablo 2

Geleneksel öğrenme ve işbirlikli öğrenme grupları arasındaki farklar (Miller, 1989; Johnson, Johnson ve Holubec, 1990, 1994; Gömleksiz, 1993).

<u>Geleneksel Öğrenme Grupları</u>	<u>İşbirlikli Öğrenme Grupları</u>
Grup elemanları arasında birbirlerine karşı olumlu bir bağlılık söz konusu değildir.	Grup üyeleri arasında olumlu ve işbirliğine dayalı bir tutum söz konusudur. Grup üyelerindeki çocuklar başarılarına aynı anda ulaşırlar.
Homojen ve birbirine yakın özelliklerde grup elemanlarından oluşur.	Heterojen gruplar oluşturulur. Karma ve farklılıkları barındıran gruplar sınıf içerisinde etkileşimli bir öğrenme ortamı oluşturur. Yaratıcılıkları ve bakış açıları gelişir.
Grup lideri bir tanedir ve elemanları yönlendirir.	Liderlik grup üyeleri arasında paylaşılmaktadır. Her üye birbirinin açıklarını kapatarak ilerler.
Üyelerin sorumlulukları arasında grup elemanlarının öğrenmesine katkı sağlamak yoktur.	Üyeler birbirlerinin öğrenmesi için kendilerini sorumlu hissederler.
Bireysel sorumluluk var.	Grup içerisinde sorumluluk bilinci söz konusudur.
Tek başına çalışmalar ve bireysel ürünler söz konusudur.	Her grup elemanının verimli olması için üyeler birbirinin çalışmalarına katkı sağlar. Gruptan çıkmış tek bir ürüne vurgu yapılır.
Sosyal becerilere daha az önem verilir.	Sosyal beceriler doğrudan öğretilir.
Öğretmen gruplara nadiren karışır öğrencilerin bireysel çalışmaları ön plana çıkar.	Öğretmen grup elemanlarını gözlemleyerek sorunların çözülmesinde yardımcı konumundadır.
Uygulamada gerekli işlemlerin yapılandırılmasına dikkat edilmez.	Öğretmen, grupların etkin ve verimli hale gelmesi için yapılandırmaya dikkat eder.
Grup çalışmasında bireysel sorumluluklar söz konusu değildir.	Grup üyelerinin bireysel görev ve sorumlulukları bütün bir ürün çıkaracak biçimde bölünmektedir.

2.1.4. Öz-düzenlemeye dayalı öğrenme.

Her konunun ve dersin içeriğinde uygun bulunan ve uyarlanması daha kolay ve etkili olan öğrenme modelleri mevcuttur. Programlama dillerinin öğrenilmesinde işbirlikçi ve yapılandırmacı öğrenme modellerinin dışında üzerine sınırlı sayıda çalışma yapılmış olan ve farklı öğrenme modellerinin içerisinde yer verilen öz-düzenlemeye dayalı öğrenme modeli de öne çıkmaktadır. Öz-düzenleme yaklaşımının temeli, bireysel çalışmalarda bireyin öğrenme sürecinde kendi kontrolünü kendisinin yapmasına dayanmaktadır. Birey olabilme süreçlerinde hedeflerine ulaşabilmelerinde öz-düzenlemeye dayalı öğrenme tekniği öğrencilerin bilişüstü

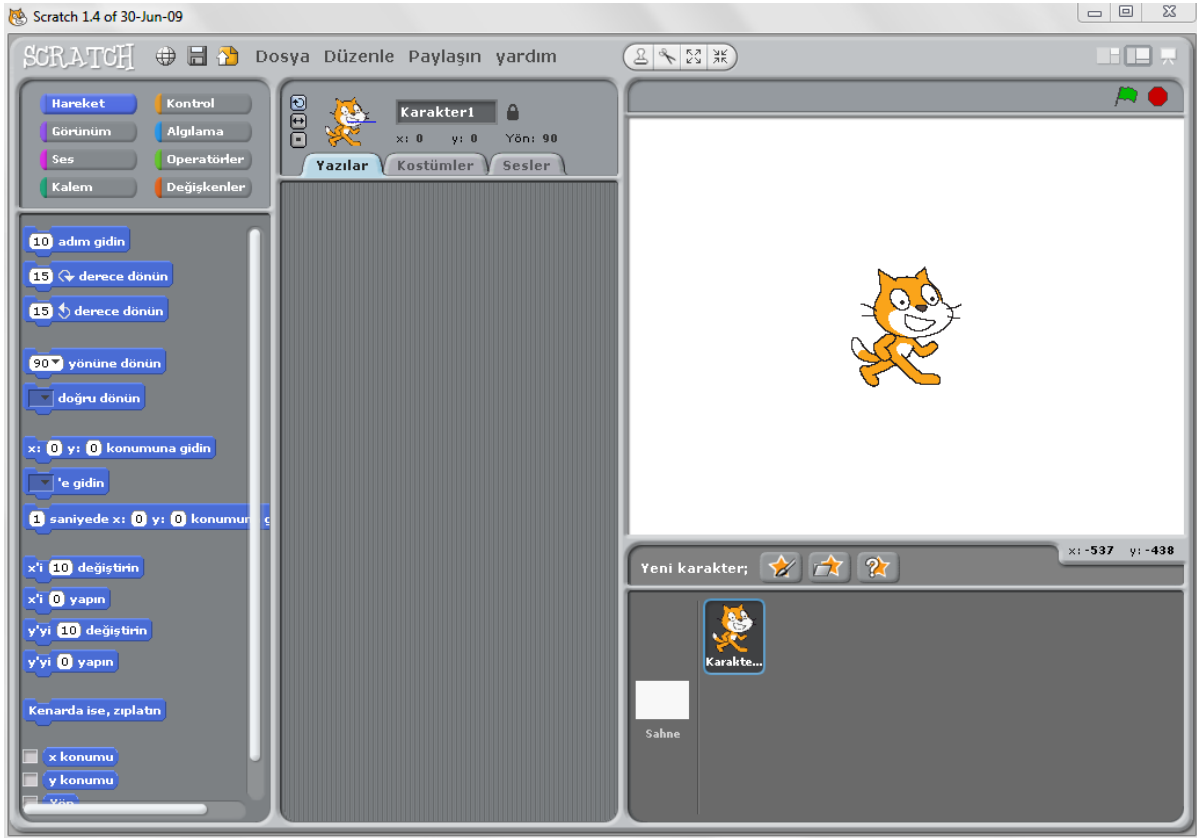
yeterliliklerini arttırdığı gibi, davranışlarını düzenlediği ve motivasyonlarını arttırdığı ifade edilmektedir (Bandura, 1994). Programlama dillerinin öğrenme aşamalarında analiz yeteneği ve düşünmenin sistematik olması, öz-düzenleme becerilerinin geliştirilmesi bağlamında önem kazanmaktadır. Öz-düzenleme becerilerinin gelişimi kendi kendini gözlem, kendi kendini yargılama, kendi kendine eyleme geçirme gibi alt boyutlardan oluşmaktadır (Schunk, 2001). Öz-düzenlemeye dayalı öğrenme yaklaşımında insanların şahsi çabaları ve bu çabaların karşılığında da başarının sağlanması adına problem çözme yeteneğinin gelişmesi ve kendi eksikliklerini belirleyebilmesi oldukça önemlidir.

2.2. Scratch

Scratch, MIT Media Lab tarafından 8 yaş ve üzeri bireylerin kullanabileceği grafik arayüze sahip kodlama aracıdır. Programın kullanımı basittir ve ilgi çekici bir tasarımdan oluşmaktadır. Scratch programı basit kullanım tarzı ve eğlenceli arayüzünün tasarımı sayesinde çocukların ve gençlerin ilgisini çekerek kodlama yapmaları için onları teşvik etmektedir. Kod alanına sürüklenerek oluşturulan komutlar görsel olarak komutun uygulanmış halini görmemize katkı sağlamaktadır. Program 8 yaşın üzerindeki çocukların animasyon oluşturmasını, oyun ortamında simülasyon tasarımlarını olanaklı hale getirmektedir. Programa ücretsiz ulaşılabilir olması eğitim temelli kullanılarak daha fazla kitleye ulaşmasını sağlamaktadır.

Şekil 1

Scratch programının giriş ekranı (Scratch, 2018).



Scratch programı açıldığında Şekil 1.'deki görüntü ile karşılaşılır. Bu arayüzde sol tarafta kod blokları, orta kısımda kodların sürüklendiği program alanı ve sağ tarafta da animasyonun uygulamasının görüldüğü kutu bulunmaktadır. Bu uygulama hataların telafi edilmesi açısından diğer kodlama programlarına kıyasla daha kolay kullanılabilir. Kod blokları sürüklenip bırak mantığıyla çalışmaktadır. Ses bloğu da karakterleri seslendirmeye imkan veren kodların bulunduğu kısımdır. Hareket bloğunda 15 kod bulunmaktadır.

Scratch programında karmaşık programlama mantığı yerini blok temelli sürüklenip bırak mantığına bırakmıştır. Programı tasarlamak ve uygun kod bloklarını alt alta sıralamak yeterlidir. Kodlama mantığının anlaşılabilirliğini artırmak adına her kod bloğu farklı renkte ifade edilmektedir. Her kod bloğunun altında o blokla ilgili komutlar görülmektedir. Bu şekilde bir

tasarım yapısı kodlamayı basitleştirerek öğrencileri motive etmektedir. Şekil 2’de görüldüğü gibi kod bloklarında farklı renkler farklı kod parçalarını temsil etmektedir.

Şekil 2

Scratch programı örnek kod bloğu.



Scratch alanında yapılan bir takım akademik çalışmalar (örn. Malan ve Leitner, 2007 ve Calder, 2010) Scratch programının öğrenciler üzerinde kodlama programının özünü anlamalarını destekleyen, heyecanlandıran, ilgilerini çeken ve motivasyonlarının artmasına katkıda bulunan özelliklere sahip olduğunu belirtmektedir.

Ülkemizde ise son dönemlerde Scratch ile kodlama eğitimi hakkında yapılan çalışmalar artış göstermektedir. Örneğin Keçeci ve Alan (2016) yaptıkları çalışmada Scratch programının eğitsel oyun tasarımında kullanımı üzerine 28 madde ve 3 faktörden oluşan tutum ölçeği geliştirmişlerdir. Çalışmalarında Scratch’in, yazılımın öğretilmesi konusunda daha verimli olduğunu ifade etmişlerdir.

2.2.1. Scratch programının özellikleri.

Scratch görsel programlama dillerinden birisidir ve öğrencilerin keşfetme duygularını güdülemektedir. Bu süreçte problemlerin çözümünde algoritma tasarlamaya başlamada öğrencilerin yaratıcılıklarının gelişmesine katkı sağlamaktadır. Burada renkli ve eğlenceli bir tasarım olmasından kaynaklanan avantajlar söz konusudur. Öğrencilerin ilgisini çeken

arayüzü sayesinde oyunla öğrenmelerine sebep olmaktadır. Görsel yönden kullanıcı dostu bir arayüze sahip Scratch programının temel özellikleri şunlardır (Pepler ve Kafai, 2007) :

- *Bloklar Kullanılarak Programlama:* Kullanıcıların program parçalarını sürükle-bırak mantığıyla birleştirerek yeni programlar oluşturmasını sağlar. Farklı renklerdeki kod blokları kullanıcıya grafik arayüzü sunmakta ve herhangi bir hatada yazım hatası gibi sıkıcı uyarılar vermemektedir.
- *Medya Araçları Kullanılarak Programlama:* Scratch kodlama aracına resim, ses ve animasyon gibi medya araçları kolayca eklenebilir. Bu eklentiler kullanıcıların hayal güçlerini geliştirerek ilgi çekici programlar yapmalarına imkân vermektedir.
- *Yapılan Programların Paylaşımı:* Kullanıcılar yaptıkları programları internet üzerinden paylaşabilir, başka uygulamaları inceleyebilirler. Geliştirdikleri program parçalarını işbirliği yaparak birbirlerine aktarabilirler.
- *Çoklu Dil Desteği:* Programı kullanmasını sağlayan içerisinde 40 üzeri dil desteği sağlamaktadır. Böylece kullanıcılar dil engeline takılmadan işbirliği içinde farklı projeler üretebilmektedir.

Scratch, öğrencilerin akranları ile iş birliğine girerek bireyin kendi kendine öğrenmesini desteklemektedir. Ayrıca gençleri proje paylaşımına, araştırmaya ve diğer programlama dillerine göre daha az öğretime odaklayarak öğrenmeye teşvik etmektedir. Günde ortalama 1500 yeni proje <http://scratch.mit.edu> sitesine yüklenmektedir. Scratch'in sosyal paylaşım sitesinde, kullanıcılar projelerini paylaşarak geri dönütler alabilirler, diğer kullanıcılar tarafından teşvik edilirler ve diğer projelerden öğrenirler. Scratch programının özellikleri şu şekilde sıralanabilir (Maloney ve diğerleri, 2010):

- Tek pencereci kullanıcı arayüzü,
- Canlılık,
- Kod bloklarını canlandırma,

- Hata mesajı olmaması,
- Komut setinin en aza indirilmesi.

Bu özellikler bir arada bulunduğu sistemde küçük yaş gruplarının kodlama eğitimine dahil olmaları için avantajlı hale gelmektedir.

2.2.2. Scratch programının eğitime entegrasyonu.

Bilgi çağını yakalayabilmek adına, hedeflere ulaşılması noktasında bilişim eğitiminin süresi arttırılarak içeriğinde kodlama eğitimine daha fazla yer verilmesi önemlidir. Temel eğitimin içeriğinde bilgisayar kullanımı ve kodlama konusunda yaratıcı sürece dahil olmak oldukça önemlidir. Bilgisayar eğitimi ve programlama dilini öğrenmek kişisel gelişim sürecinde sistematik düşünmeyi, problem çözme yeteneği kazanmayı sağlamaktadır. Scratch programının üç temel ilkesi şunlardır; daha fazla düşünebilme, daha fazla anlam katma ve diğer programlama dillerine göre daha fazla sosyallik sağlamaktır.

Scratch eğitimi görsel yapısı sebebiyle soyut olan kavramların somutlaştırılarak anlık ulaşılmasını sağlar ve böylece animasyon ve oyun tasarımı yapılmasına imkan verir. Scratch becerikli öğrencilere, hayal güçlerini sahneye yansıtma imkânı sağlamaktadır. Ayrıca yazılım, öğrencilere koordinat sistemi gibi matematiksel kavramları öğretir (Lee, 2011). Scratch programının öğrenciler ve öğretmenler için tasarlanan farklı sürümleri bulunmaktadır. Eğitimciler için tasarlanan versiyonunda öğrencilere yönelik geribildirimlerin oluşması için yeterli altyapı bulunmaktadır.

Yazılım eğitiminin küçük yaşlardan itibaren öğrenilmesi doğru bir karar olmakla birlikte öğrencilerin hazır bulunuşluk seviyesinin de göz önünde bulundurulması gerekmektedir. Örneğin ilkokul seviyesindeki öğrencilerin yaş grupları göz önünde bulundurulduğunda soyut ve karmaşık kavramları algılayabilmeleri için program içerisindeki bu kavramların somut hale getirilmesi önem kazanmaktadır. Algoritma çocukların algılaması için soyut bir düzeydedir fakat bu sürecin sorunsuz aşılmasında bazı özel şirketler resim, ses

ve mzik yardımıyla ocukların kendi projelerini gerekleştirebileceđi grsel programlama aralarını geliştirmişlerdir (Gen ve Karakuş, 2011) Bu uygulamaların eđitime entegre edilmesi srecinde devlet kurumlarına destek olan pek ok yazılım ve teknoloji firması da bulunmaktadır.

2.2.3. Yapılan alıřmalar.

đrencilerin psikososyal, bilişsel ve duyuşsal algılamalarına katkı sađlaması sebebiyle Scratch programının okullarda ve eđitim sisteminin ierisinde yođun olarak kullanılmaya başlanması arařtırmacıların da ilgi odađı olmasını sađlamaktadır. Bu gelişim ve yaygınlaşma srecinde yapılan alıřmalar uygulamalı bir řekilde yapılarak analiz edilmiştir.

Fesakis ve Serafein (2009), kurs mantıđı zerinden eđitim veren bir yerde, đrencilerin grřlerini belirlemek adına alıřma yapmışlardır. Bu alıřmaya katılan đrencilerin %65'i programın kolay kullanıldıđını, %85'i ara yzn basit ve anlaşılır olduđunu, %80'i ise tasarımınnın estetik olduđunu ifade etmiştir. đrencilerin %80'i, Scratch'ın basit bir programlama dili olduđunu, %90'ı ise grafik kod bloklarını faydalı bulduklarını belirtmişlerdir. đrencilerin %80'i Scratch programına katılmadan nce, eđitimde teknoloji kullanımının ilgi ekici olduđunu ifade etmişlerdir. Bu oran eđitimden sonra %92 olmuştur.

Gen ve Karakuş (2012)'un alıřmasında, bilgisayar oyunları tasarımında Scratch kullanımı zerinde geliştirdikleri dersin ieriđinde, oyun teması esas alınarak uygulamalar yapılmıştır. Nicel ve nitel arařtırma verileri toplanarak 109 ikinci sınıf đrencisi ile uygulamalı alıřmalara yer verilmiş ve elde edilen bulgular analiz edilmiştir. alıřma sonucunda đrenciler Scratch programının basit, anlaşılır, kolay ve eđlenceli olduđunu belirtmişlerdir.

Adams ve Webster (2012) tarafından yapılan bir alıřmada Scratch kampına katılan đrenciler video zelliđini kullanmak yerine oyun ile đrenmeyi tercih etmişlerdir. Kampın

genel amacı ortaokul ve lise düzeyindeki öğrencilere programlama kavramlarının öğretilmesidir. Çalışmada, 22 ortaokul ve lise öğrencisinin 300 farklı Scratch projesindeki kod bloklarının istatistiği çıkarılmış ve elde edilen veriler, t-testi ile analiz etmişlerdir. Değişkenler, döngü deyimleri, koşul ifadeleri vb. temel öğelerin nasıl kullanılacağı öğrencilere gösterilmiştir. Yaz kampında 209 Alice projesi, 103 Scratch oyun projesi, 10 Scratch video projesi yapılmıştır. Scratch oyun projelerinde daha fazla değişken, kod bloğu, koşullu ifade, döngü deyimleri kullanıldığı görülmüştür. Hikâyelerde en çok diyaloglar kullanılmaktadır. Videolarda neredeyse oyunlar kadar döngüler kullanılmış ama değişkenler ve koşullu ifadeler çok daha az kullanılmıştır. Scratch video projelerinin oluşturulması sırasında daha az komutların paylaşılması sebebiyle öğrenme aşaması oyun projelerine göre daha basit olmuştur ve öğrencilerin daha fazla dikkatini çekmiştir. Yapılan bu araştırma öğrencilerin oluşturacakları komutların nasıl belirlenebileceği konusunda fikir vermiştir.

Meerbaum-Salant, Armoni ve Ben-Ari (2013) Scratch programının öğrencilere bilgisayar bilimi kavramlarını öğretmede başarılı olup olmadığını araştırmışlardır. 20 haftalık sürede haftada iki ders saati, iki eğitim öğretim yılı süresince 204 ortaokul öğrencisi ve sekiz öğretmen ile araştırma gerçekleştirilmiştir. Gözlem ve görüşmeler ile nitel veriler toplanmıştır. Öğrencilerin tekrarlı yapılar, değişkenler gibi önemli bilgisayar bilimi kavramlarını öğrendikleri tespit edilmiştir.

Özyurt ve Özyurt (2015) tarafından üniversite öğrencilerinin programlamaya karşı tutum ve programlama öz-yeterliklerinin belirlenmesine yönelik bir çalışma yapılmıştır. Çalışma 2013-2014 bahar yarıyılında 325 üniversite öğrencisinin katılımıyla yapılmıştır. Veri toplama aracı olarak Bilgisayar Programlamaya Karşı Tutum Ölçeği (BPKTÖ) ve Bilgisayar Programlama Öz-yeterlik Ölçeği (BPÖYÖ) kullanılmıştır. Araştırma betimsel nitelikli tarama modelinde yürütülmüştür. Elde edilen sonuçlar incelendiğinde çalışmada öğrencilerin programlamaya karşı tutumlarının olumlu olduğu görülmüş olmasına karşın bu tutumun daha

da olumlu hale getirilmesi için programlama derslerinde kullanılan öğrenme/öğretme teknik ve yöntemlerinin gözden geçirilmesi, derslerde öğrencilerin problem çözme ve eleştirel düşünme becerilerini geliştirecek etkinliklere yer verilmesi önerilmiştir.

Maloney ve arkadaşları (2010) tarafından yapılan çalışma kapsamında Scratch programının, günlük yaşamlarının büyük bir bölümünde bilgisayar olan gençler arasında popüler olan etkinliklerden biri olmasını sağlamak amaçlanmıştır. Öğrencilerin bireysel olarak geliştirdikleri, animasyonlu hikayeler, oyunlar ve interaktif sanat gibi Scratch projelerinin, onların teknolojik akıcılık, matematiksel ve problem çözme gibi becerilerini geliştirdiği tespit edilmiştir. Bu çalışma öğrencilere uygulanan animasyonlu uygulamaların, oyunların hangi düzeyde olması gerektiği konusunda yol gösterici nitelikte olmuştur.

Kalelioğlu ve Gülbahar (2014), 49 ilköğretim 5. sınıf öğrencilerinin problem çözme becerileri üzerinde Scratch programının etkisini araştırmışlar ve nicel sonuçlarına göre, Scratch platformda programlama yapılmasının, ilköğretim öğrencilerinin problem çözme becerileri üzerinde anlamlı bir farklılığa neden olmadığını tespit etmişlerdir. Ayrıca tüm öğrencilerin programlamayı sevdikleri ve programlarını geliştirmek istedikleri de ifade edilmiştir.

Fal ve Çağıltay'a (2013) göre, Scratch, programlamaya başlarken programlama mantığı ve algoritmik düşünme yeteneği kazandırır. Scratch ile çok karmaşık ilişkileri kolaylıkla analiz edilip, karşılaşılan problemleri küçük parçalara bölerek sorunu çözmemize ve programlama dili ile kullandığımız temel yapıları en etkili şekilde öğrenmemize yardımcı olmaktadır.

Konu ile ilgili ortaya çıkan alan yazın taramasındaki çalışmalar farklı araştırma modellerinde nitel ve nicel araştırma yöntemleriyle oluşturulmuştur. Araştırma süreçlerinde öğretmenler ve öğrencilerin katılımcı olduğu farklı örneklem gruplarıyla gerçekleştirilen çeşitli çalışmalar bulunmaktadır. Ülkemizde kodlama öğreniminin müfredat içerisinde

yaygınlaştırılmasına ve öğrencilerin bu öğrenim sürecini kolaylaştırarak daha etkili hale getirilmesine çalışılmaktadır.

Bilişim teknolojileri ve yazılım dersi kapsamında öğrencilerin uygulamaları olarak kodlama öğrenimi 21.yy. becerileri içerisinde önemli bir yer tutmaktadır. Bu durum sonraki bölümde ayrıntılı olarak ele alınmaktadır.

2.3. Bilişim Teknolojileri ve Yazılım Dersi

Bilişim teknolojileri ve yazılım dersi kapsamında yapılandırmacı ve öğrenci merkezli yaklaşımın oldukça önemli olduğu ve bu yaklaşımlara ek olarak işbirliğine dayalı öğrenme kuramı ve öz- düzenlemeye bağlı öğrenme kuramlarıyla da desteklenmesi gerektiği göze çarpmaktadır. Bu yaklaşım modelinde rehber ve yol gösterici rolünde olan öğretmen her proje için ön hazırlığını yapan, hazırbulunuşluk düzeyi yüksek öğrenciyi yönlendirmektedir.

Yapılandırmacı yaklaşımın temelinde yaparak ve yaşayarak öğrenmeye dayalı bir ortam söz konusudur. Ders içeriğinde ve genel hatlarına bakıldığında öğrencilere kazandırılması beklenen bilgi, beceri ve değerlere ilişkin yeterlikler öğretim programında bilişim okur yazarlığı, bilişim teknolojilerinin katkılarıyla iletişim kurma, bilgiyi yapılandırma, işbirlikli çalışma, problem çözme şeklinde belirtilmiştir. Bilişimde okur yazarlık yeterliliği, yaşam boyu öğrenme konusunda sorumluluk alma gibi kazanımları ifade etmektedir (TTK, 2015).

Teknolojinin gelişmesiyle birlikte bilişim teknolojileri 1996 yılından itibaren hayatın her alanında ve eğitim sektörünün bir parçası olmuştur. Seçmeli ders olarak başlayan bilgisayar dersi 2000'li yılların başında bilişim teknolojileri ismini alarak ortaokullarda seçmeli ders olarak okutulmaya başlanmıştır. Ders eğitim programının içerisinde 2010 yılından sonra ortaokullarda zorunlu hale getirilmiştir. Dersin kapsamında Word, Paint, Office gibi programlar temel bilgisayar kullanma becerisi olarak ele alınmıştır.

Teknolojinin ilerlemesiyle birlikte günlük işleri yapma konusunda öne çıkan robotlar ve diğer teknolojik aletler ders kapsamında ele alınmaya başlanmıştır. Bilişim teknolojileri

dersi programı kapsamında 2013 yılında Office programlarının yanında teknolojik aletlerin ve robotların programlanmasına katkı sağlayan algoritma ve kodlama eğitim programlarının içerisine dahil edilmiştir.

Günümüzde bilgi birikimine paralel olarak bilişim teknolojileri gelişmektedir. Yazılım eğitimleri ülkelerin eğitim programlarının içeriğinde önem kazanmaktadır. Bu eğitim özellikle Güney Kore ve İngiltere gibi ülkelerde önem kazanmaktadır (Burns, 2012, akt: Demirer, 2016). Programlama öğretimi zor gibi görünebilir. Programlama eğitiminin ortaokul hatta isteğe bağlı olarak ilkokulda başlaması algoritma mantığını öğrenebilme sorunu olarak ortaya çıkarmıştır. Algoritma mantığını görselleştirip müzik ve ses gibi medya araçları ekleyerek kodlama mantığını zenginleştirmiş ve görsel programlama araçlarını oluşturmuşlardır (Demirer, 2016). Çocuklar ve programlama öğrenmek isteyenler için hazırlanmış görsel açıdan zengin ve kullanımı kolay Scratch, Alice, Microsoft Small Basic, gibi programlama araçları bulunmaktadır. Kodlama mantığını öğrenmek isteyen bireyler bu sayede, eğlenceli ve kolay bir şekilde kod yazabilme imkânına sahip olmaktadır.

Scratch yazılımının görsellik açısından zengin olması ve kullanım kolaylığı gibi özellikleri bakımından avantajlı konumda olduğu ve tercih edildiği gözlemlenmektedir. Yapılan literatür taramalarında 2012-2016 yılları arasında Scratch programının problem çözme, yazılım, kodlama ve algoritma öğretiminin uygulanabilirliği konusunda yapılan çalışmalar hız kazanmıştır. Çatlak, Tekdal ve Baz (2015) tarafından Scratch programının yazılım ve kodlama öğretiminde kullanılması ile ilgili olarak 32 dökümanlık bir alan yazın taraması yapılmıştır. Bu alan yazın taramasının sonuçlarına göre program farklı yaş ve eğitim seviyelerinde en fazla ortaokul düzeyinde (%35) ve 8-16 yaş aralığına uygulanmıştır. Bu uygulamalara ve çalışmalara göre en belirgin sonuçlar Scratch programının algoritma ve programlama üzerinde olumlu yönde, öğrencilerin ilgisini çeken problem çözme sürecine katkı sağlayan, yaratıcı düşünme becerilerini geliştiren etkileri vardır (Keçeci, 2016).

Çalışmanın içeriğinde elde edilen veriler öğrencilerin yapılandırmacı yaklaşımla problemleri çözüme becerilerinin artırılmasına yöneliktir. Gelişen eğitim programları içerisinde okul öncesi dönemden itibaren etkili şekilde verilmesi beklenen kodlama eğitiminin farklı yaş grupları ve sınıf düzeylerindeki etkilerini incelemek gerekir.

2.3.1. Programlama dilleri öğretiminde algoritma eğitimi.

Programlama dillerinin öğrenilmesi ve aşamaları bir döngüye benzetilmektedir. Garner (2003), programlama öğretiminde, yazılımın yaşam döngüsünü dört aşamada tanımlar. Bunlar; problemin analizi, tasarım ve bir çözüm/algoritma geliştirme, algoritma uygulamak, test ve algoritmayı gözden geçirmedir.

Problem Analizi: Problem öğretimi geçtiğimiz yıllardan bu yana bir değişim göstermemiştir. Öğretmenler genellikle kontrol ve veri yapılarını sunmakta, birkaç örnek gösterip ve bireylerin yapmasını beklemektedirler. Bireylerin yakındıkları konu, öğretmenin anlattıklarını anlayıp yeni bir problemle karşılaştıklarında bunu yapamamalarıdır (Garner, 2003). Bu süreç içerisinde geleneksel bir yaklaşım benimseyen öğretmenler öğrencilere hazır bilgiler vererek bu bilgilerin uygulanma aşamasında sürekli yönlendirme yaparak tek başına alınması gereken kararları üstlenmişlerdir. Öğrenme deneyiminde komutları olduğu gibi uygulayan öğrenci gerekli yerlerde özgün görüşlerini kullanamamaktadır. Deek ve McHugh tarafından 2000 yılında geliştirilen Solvelt programı, problem çözme, planlama, tasarım ve test aşamalarını içermektedir. Şekil 3’ de görülen bu program video klipler ve mini dünyalar içermektedir (Garner, 2003).

Şekil 3

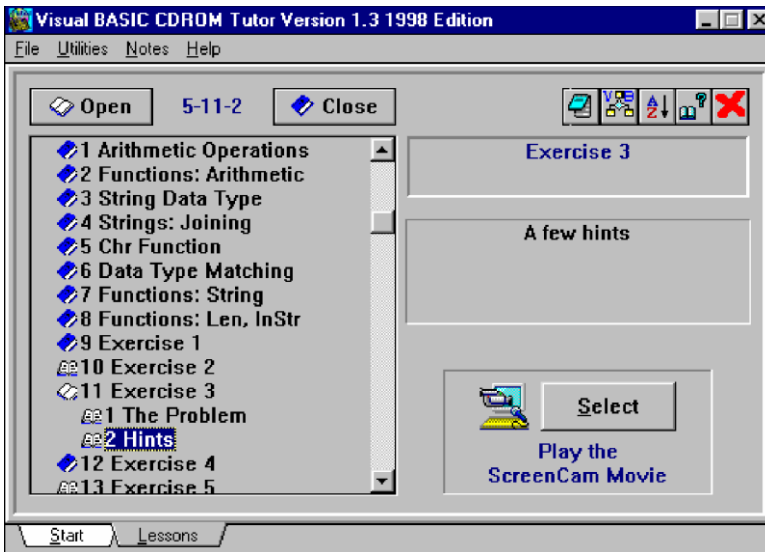
SOLVEIT (Deek & McHugh, 2000.)



Garner'ın 1997 yılında geliştirdiği "Multimedia Command Centre" programı ise içerisinde bulunan programlama problem setleri ve internet üzerinden verilen çeşitli ipuçları ve filmlerle birlikte bireylerin problemi analiz etmesini sağlamaktadır. Şekil 4'de görüntüsüne yer verilen bu program, Visual Basic (VB) programlama dili için geliştirilmiştir (Garner, 2003).

Şekil 4

VB Tutor



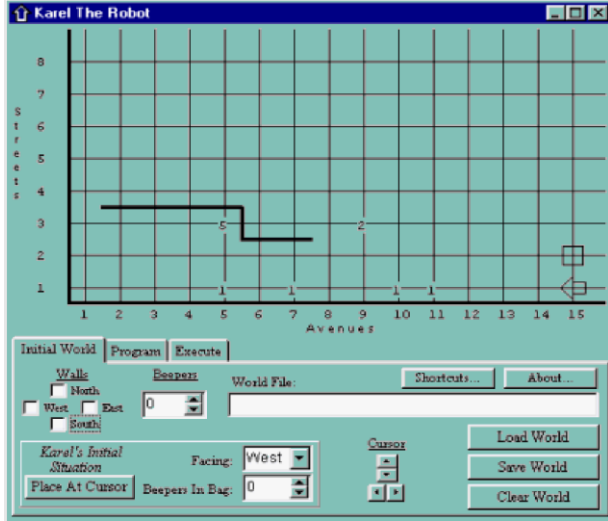
Tasarım ve bir çözüm/algorithm geliştirme: Problem analizinden sonra bir çözüm tasarımı gereklidir. Geçmişte bu aşama için kullanılan birkaç program yer almaktadır. Şekil 5.

örnek bir görüntüsü verilen Karel ve Robot bu aşamada yardımcı olabilecek bir programdır.

Altta yer alan komutlara göre üstte değişimler gözlenir (Garner, 2003).

Şekil 5

A Simple Karel World



Algoritma uygulama ve test etme: Programlama derslerini anlatan öğretmenler var olan görselleştirilmiş algoritma araçları yerine, geçmişte öğrendikleri -alışıl gelmiş- yola yani tahta ve kalem/tebeşir kullanma tekniğine başvururlar.

Algoritmayı gözden geçirme: Üç aşama sonucunda problem çözülmediyse algoritma tekrardan test edilir.

Mesleki Eğitim ve Öğretim Sistemini Güçlendirme Projesi (MEGEP, 2014) kapsamında hazırlanan yeterliliğe dayalı program modülleri liselerde öğretmenler tarafından kullanılmakta, modül içeriği anlatılmaktadır. Programlama temelleri ders modülleri incelendiğinde algoritma tanımı ve aşamaları verilmektedir. Algoritma, verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların hiçbir yoruma yer vermeksizin açık, düzenli ve sıralı bir şekilde söz ve yazı ile ifadesidir. Algoritma oluşturulurken belirli aşamalara göre oluşturulur (MEGEP, 2014).

- Problemi Tanımlama: Problem bu aşamada tam olarak anlaşılmalı, problem cümlesi oluşturulmalı, anlamlar ve birbirleriyle ilişkileri de tanımlanmalıdır.

- **Problemi Geliştirme:** Problemin özü ve tanımı gerçekleştirildiğinde çözüm yolları aranmalıdır. Genellikle bir problemin birden fazla çözüm yolu olabileceği unutulmamalı ve çözüm yollarından en uygun olanı seçilmelidir.
- **Sisteme Uygunluğunu Tespit Etme:** Sonuçların dış ortama, dolayısıyla insana aktarımı düzgün bir biçimde yapılmalıdır. Programcı program çıktısı olarak almak istediği dökümün biçimini tasarlar.
- **Çözümü Kağıt Üzerinde Gösterme (Prototip Oluşturma):** Algoritmanın geliştirilmesinden sonra ilgili kişiler tarafından anlaşılabilir olmasında ve aktarımında daha kolay olması nedeniyle prototip oluşturulur. Oluşturulan model kapsamında algoritmada veya yazılımda olası hatalar tespit edilerek yanlışlıkların düzeltilmesi ve telafi edilmesi sağlanmış olur.
- **Çözümü Derleme:** Algoritma yazıldıktan sonra elde edilen sonuçlar ile sorunun çözümlenip çözümlenmediği yani algoritmanın problemlerinin çözümlenmiş olması tespit edilir.
- **Çözümü Geliştirme:** Bu aşamada doğruluğu kontrol edilmiş çözümün tekrar edilen blokların daha kısa yoldan sonuca ulaşması durumları gözden geçirilir.
- **Oluşabilecek Hatalar:** Hazırlanan algoritmada en sık karşılaşılan hatalarda, problemin çözüm adımları belirlenirken yapılan yanlışlıklardan kaynaklanan mantık hatalarının olduğu görülmektedir (MEGEP, 2014).

Algoritmayı basit bir örnekle açıklamamız gerekirse; aşağıdaki adımlar bir çocuğun resim yapmasında izleyeceği yolu tanımlamaktadır.

Adım1: Masaya yürü.

Adım2: Sandalyeye otur.

Adım3: Boya kutusunu aç.

Adım4: Defterde boş sayfayı aç.

Adım5: Boyaları masanın üzerine çıkar.

Adım6: Mavi boyayı sağ eline al.

Adım7: Bulut resmi çiz.

Algoritma eğitimi verilirken sorular yavaş yavaş zorlaşmaya ve öğrenciler tarafından anlaşılmanmaya başlamaktadır. Öğretmenler, algoritma öğretimini verirken öğretmenlerinden öğrendikleri şekilde devam ettirmekte ve alışlagelmiş algoritma öğretimini sürdürmektedirler. Alışlagelmiş algoritma öğretiminde; problem verilir, adımlar yazılır, çözümün sağlanması tahtada yapılarak öğretim gerçekleştirilir. Bu öğretimi bir örnek üzerinden gösterecek olursak aşağıdaki programın çözümü Tablo 3.'de olduğu gibidir.

Örnek: İki sayıyı toplamak için gerekli programa ait algoritma adımlarının yazılması

Adım1: Başla

Adım2: A sayısını gir.

Adım3: B sayısını gir.

Adım4: Toplam= A+B işlemini yap

Adım5: Toplam değerini ekrana yaz

Adım6: Bitir

Tablo 3

İki sayıyı toplamak için verilen algoritmanın çözümün sağlanması.

<u>A</u>	<u>B</u>	<u>Toplam=A+B</u>	<u>Toplam</u>
10	20	30	30
5	8	13	13

Yukarıda verilen basit bir örnektir. Dolayısıyla çözümünün tablo şeklinde verilmesinin sonucunda bireyler algoritmanın çalışma mantığını anlayabilmektedirler. Örneğimizi biraz zorlaştırdığımızda Tablo 4'te görüldüğü gibi çözüm aşaması da zorlaşmaktadır.

Örnek: Kullanıcı tarafından girilen bir sayının (N), kendine gelene kadarki sayıların toplamını (T) veren programa ait algoritma adımlarının yazılması (X sayaç)

Adım1: Başla

Adım2: N'yi OKU

Adım3: T=0

Adım4: X =1

Adım5: T=T+X

Adım6: X=X+1

Adım7: EĞER X<=N İSE 5. ADIMA GİT

Adım6: T'yi yaz.

Adım7: Bitir

Tablo 4

Girilen sayının kendine gelene kadarki sayıların toplamını veren algoritma çözümünün sağlanması.

<u>N</u>	<u>T</u>	<u>X</u>	<u>T = T+ X</u>	<u>X=X+1</u>	<u>EĞER X<=N</u> <u>İSE</u>	<u>T</u>
5	0	1	1	2	2<=5	
			3	3	3<=5	
			6	4	4<=5	
			10	5	5<=5	
			15	6	6<=5	15

Geleneksel algoritma öğretiminde örnekler bu şekilde tahtaya yazılarak, çözüme ulaştırılmaktadır.

2.3.2. Kodlama.

Bilgisayar sisteminin yönetim aracı ve bilgisayarı anlamak ve yönlendiren, programlama dili eğitimi, bilişim teknolojileri gelişim sürecinde önemli bir yere sahiptir. Tüketici toplumdan, üretici topluma geçmenin, bir ürün ortaya koyabilmenin önemini kavramış devletler, öğrencilere küçük yaştan itibaren programlama eğitimi vermenin çalışmalarını yapmaktadır.

Problem çözme becerisi, öğretmenlerin öğrencilere geleneksel eğitimde kazandırması gereken en önemli anahtar becerilerden biridir. İnsanların otantik sorunlarla başa çıkmaları, olası çözümleri tespit etmeleri, kendi bilgi, beceri ve deneyimlerini kullanabilmeleri için karmaşık zihinsel bir süreçtir (Lai & Yang, 2011). Günlük yaşantımızda da problem çözme kavramı ve programlama kavramları birbirlerini etkileyen ve tetikleyen kavramlardır.

Problemin olduğu yerde bu problemin çözümü için planlı bir program yapmak gerekmektedir.

Lai ve Yang (2011), görselleştirilmiş programlamanın, öğrencilerin problem çözme yetenekleri ve mantıksal akıl yürütme becerileri üzerindeki etkisini araştırmak için bir çalışma yapmışlardır. Çalışmada 6. sınıf öğrencilerine Scratch programlama kursu verilmiştir. Araştırma sonucunda Scratch'in problem çözme yetenekleri üzerindeki etkisinin önemli olduğunu bulmuşlardır. Ayrıca öğrencilerin kodlama eğitiminde edindikleri problemin basamaklarını öğrenme ve irdeleme aşamalarını günlük yaşantılarına uygulamaları durumunda kişisel gelişimlerine katkı sağladıklarını tespit etmişlerdir. Calder (2010) de benzer bir çalışma ile öğrencilerin kodlama öğretiminde motivasyonlarının ve iş disiplinlerinin arttığına dair gelişimsel durumlarını ortaya koymuştur. Fessakis, Gouli ve Mavroudi (2013) literatür taraması kapsamında kodlama eğitiminin sadece ortaokul öğrencileri üzerinde değil ilkokul öğrencileri üzerinde de problem çözme faaliyetlerini geliştirdiğini vurgulayarak, 5-6 yaş grubundaki öğrencilerin, kodlama uygulamalarından sonra problem çözme becerilerinin geliştiğini kanıtlamıştır.

Yapılan araştırmalar kapsamında incelenen alan yazında önceki bölümlerde de görülmektedir ki kodlama eğitiminin ve algoritmanın öneminden her çalışmada özenle söz edilmiştir. Programlama becerilerinin küçük yaştan itibaren bireylerin sosyal yaşamlarında problem çözme becerisi olarak yansıtacak bir süreç olduğu bilinmektedir. Araştırmanın ortaokul düzeyindeki öğrencilere yönelik uygulamaları ve algoritma öğretimi konusundaki görüşlerini aktarabilecekleri, bu eğitimin kendileri için faydalarını gözlemleyebilecekleri bir alan oluşmasını sağlamak için araştırmanın farklılık yaratması beklenmektedir. Yapılan çalışmalar içerisinde kodlama eğitimi günümüzde ve gelecekte meslek seçimleri konusunda onlara kolaylıklar sağlayacak niteliktedir. Kodlama eğitimi alan bireyin bilişsel ve duyuşsal alanda çeşitli becerilere sahip olduğu literatür kapsamında ele alınmaktadır.

3.BÖLÜM

Yöntem

Yöntem bölümünde araştırmanın modeli, örneklem (çalışma grubu), veri toplama araçları ile verilerin toplanması ve çözümlenmesi süreçleri tanımlanmıştır.

3.1. Araştırmanın Modeli

Araştırmada ‘karma araştırma modeli kullanılmıştır. Bu model; nitel ve nicel araştırma yöntemlerini kullanarak verileri toplama, analiz etme ve bulguları bütünleştirmeye olanak veren araştırma olarak tanımlanmaktadır (Creswell ve Clark,2017). Uşun (2012), karma araştırma modelini; tek bir çalışma ya da çalışmalar dizisindeki aynı temel olgulara ilişkin nitel ve nitel verilerin toplanması, analiz edilmesi ve yorumlanması olarak tanımlanmaktadır. Araştırmanın deseni ‘yakınsayan paralel desen’ olarak belirlenmiştir. Bu desen araştırmacının nitel ve nicel aşamaları araştırma sürecinde eş zamanlı olarak uygulamasıdır. Bu desen kapsamında kullanılan yöntemler eşit önceliklidir. Çözümleme sırasında yöntemler ayrı tutulur ve sonuç-tartışma kısmında birleştirilir (Creswell,2013).

Araştırmanın nicel kısmında, deneysel araştırma yöntemlerinden biri olan yarı deneysel desen kullanılmıştır. Bu desen araştırmacının oluşturduğu deney ve kontrol gruplarına katılımcıların rastgele atanmadığı, sadece oluşturulan grupların deney ve kontrol grubunda olma koşulunun rastgele gerçekleştirebildiği durumlarda tercih edilen bir desendir (Büyüköztürk ve diğerleri, 2017).

Nitel araştırma; bir olayı derinlemesine incelemeyi amaçlayan, olayı; anında, yerinde ve bütüncül ele alan, gözlem, görüşme ve doküman analizi gibi tekniklerinde tek başına ya da birlikte yaygın olarak kullanıldığı nitel işlemler sürecini kapsamaktadır (Yıldırım ve Şimşek, 2016). Araştırmada nitel verilerin toplanması amacıyla öğrencilere yarı yapılandırılmış görüşme soruları yöneltilmiştir. Görüşme soruları doğrultusunda toplanan veriler araştırmacı ve bilişim teknolojileri konusunda uzman bir kişi tarafından kontrol edilmiştir. Ortaya

çıkarılan ve ortak olarak belirlenen kodlar ve anekdotlar kapsamında bulgular yorumlanarak ele alınmıştır.

Deneysel uygulama süreci Tablo 5’de verilmiştir. Her iki gruba başarı testi, öğretim sürecinde ele alınacak kazanımlara yönelik ön test ve son test olarak uygulanmıştır.

Tablo 5

Araştırmanın zaman çizelgesi

<u>Gruplar</u>	<u>Ön Test</u>	<u>Yapılacak işlemler</u>	<u>Son Test</u>
Deney(N=19)	Başarı Testi	Scratch ile Algoritma Öğretimi	Başarı Testi
Kontrol(N=14)	Başarı Testi	Geleneksel Yol ile Algoritma Eğitimi	Başarı Testi

Deney ve kontrol grubuna öncelikle ön test yapılmış, sonrasında her iki grup için algoritma eğitimi vermeye başlanmıştır. Algoritma eğitim 3 haftayı kapsamış ve toplamda 240 dakika sürmüştür. Algoritma eğitimi sonucunda başarı testi son test olarak tekrar uygulanmıştır. Nicel verileri desteklemek amacıyla deney ve kontrol gruplarına mülakat yapılmış ve deney grubunun öğrenme süreci takip etmek amacıyla bir uygulama gözlem formu doldurulmuştur. Araştırma da nicel veriler, nitel veriler ile desteklenerek araştırmanın güvenilirliğini artırmak ve öğrencilerin görüşlerini derinlemesine incelemek hedeflenmiştir.

3.2. Veri Toplama Araçları

Nitel ve nicel veri toplama araçlarının kullanıldığı bu çalışma kapsamında her iki araştırma yöntemi ile ilgili veri toplanmak amacıyla mülakat ve uygulama gözlem formu geliştirilmiştir. Bu araçlara ait açıklamalar aşağıda verilmiştir.

3.2.1. Başarı testi.

Çalışmaya katılan öğrencilerin düzeyini belirlemek için kullanılan başarı testi, bir uzman ile birlikte hazırlanmıştır (EK-1). Başarı testini oluşturmak için kazanım listesi hazırlanmış ve bu listeye göre sorular oluşturulmuştur. Başarı testinin kazanımlarına ait liste, Ek-2’de verilmektedir. Kazanım listesi kapsamında, yapılan uygulamalarının çoğunlukla analiz basamağında olduğu görülmektedir. Başlangıçta 10 tane sorudan oluşacak olan başarı testi, bir uzman tarafından kontrol edildikten sonra 8’e indirilmiştir.

Başarı testi, öğrencilerin sınıf ortamında yaptıkları uygulamalara yönelik açık uçlu sorulardan oluşmaktadır. Eğitim programı içerisinde uygulama yapılan haftaların kazanımları göz önünde bulundurulmuştur (Bkz. Ek 3). Uygulamalar sırasında ve yapılan pilot çalışmada sorularla ilgili bir problem yaşanmamıştır.

Başarı testinin güvenilirliğini arttırmak amacıyla bilişim teknolojileri alanında uzman kişi tarafından verilen cevaplar tekrar değerlendirilmiştir. Başarı testi 103 kişiye uygulanmış ve Cronbach’s alpha güvenirlik katsayısı 0.78 bulunmuştur. Hesaplanan Cronbach’s alpha güvenirlik katsayısı 0,70 ve üzerinde ise ölçek güvenilir kabul edilir (Yaşar, 2014).

Başarı testi, ön test ve son test şeklinde uygulanmıştır. Her iki grup katılımcıları algoritma kavramıyla yeni karşılaşmaktadır. Başarı testi temel algoritma sorularını içermektedir. Deney grubuna Scratch üzerinden örnek algoritma soruları çözülürken, kontrol grubunda aynı örnek sorular geleneksel yollarla çözülmüştür. Her iki guruba, üç haftalık uygulama sonrasında başarı testi uygulanmıştır. Başarı testi ile algoritma eğitiminin öğrenciler tarafından öğrenilip öğrenilmediği kontrol edilmeye çalışılmıştır.

3.2.2. Görüşme formu.

Nitel veri toplamak amacıyla yarı yapılandırılmış olarak hazırlanan görüşme formu kullanılmıştır. Eğitim bilimlerinde görüşme tekniği kullanılması, yapılandırılmış,

yapılandırılmamış ve yarı yapılandırılmış görüşme formları kullanılarak yapılmaktadır (Yıldırım ve Şimşek, 2016). Görüşme formu hazırlanırken sorular taslak formatta hazırlanıp, görüşmenin akışına göre sorulma seyri değişiklik göstermektedir. Kullanılan görüşme sırasında araştırmacı esnek bir bakış açısında olmalı ve görüşme yapılan kişi için rahat bir ortam sağlanmalıdır (Karasar, 2011).

Bir uzman tarafından soruların uygulanabilirliği kontrol edilmiştir. Sorular oluşturulurken alan yazından yararlanılmıştır. Oluşturulan üç soru, araştırmanın alt problemlerine ve öğrencilerin görüşlerine yönelik olmasına özen gösterilmiştir.

Yapılan görüşmeler sırasında araştırmanın amacı, kaç dakika görüşme yapılacağı, araştırma sonuçlarının kullanım amacı, kişisel bilgilerin gizliliği durumlarıyla ilgili katılımcılara ayrıntılı bilgi verilmiştir. Görüşme sorularının sorulması sırasında değişiklik yapılmamasına dikkat edilmiştir. Görüşmenin gidişatına uygun olarak bu durum şekillendirilmiştir. Konu bütünlüğünün sağlanabilmesi için sorulan soruların dışında verilen cevaplar o soru için dikkate alınmamıştır. Bu süreç ek sorularla desteklenerek konudan ayrılması sağlanmıştır. Görüşmeler kapsamında araştırmacı, katılımcılara yönelik kod isimleri kullanma konusunda özen göstermiştir. Görüşmelere başlanmadan önce kişilerin sözlü ve yazılı onayı alınmıştır. Öğrencilere sorulan sorular aldıkları eğitimlerle ilgilidir olan Görüşme Formu (Bkz. Ek 4) aşağıdaki soruları içermektedir :

- Aldığınız bu eğitimle ilgili olarak genel görüşlerin nelerdir?
- Bu eğitimin ardından bilgisayarla ne yapılabileceğine dair düşüncelerinde değişiklik meydana geldi mi?
- Bu eğitim sizlere neler kattı?

Araştırma sürecinde soruların ve sonuçların açık ve tutarlı olması beklenirken, her kesimden araştırmacının okuduğunda anlayabilmesi ve teyit edilebilir olması da istenir.

Lincoln ve Guba (1985) inandırıcılığın başarılabilmesi için uzun süreli etkileşim, derinlik

odaklı veri toplama, çeşitleme, uzman incelemesi ve katılımcı teyidini önermektedir (aktaran Yıldırım ve Şimşek, 2016, s. 265).

Görüşmeler doğrudan yazıya aktarıldıktan sonra konuşma metinleri üzerinden analizler Excel programına aktarılmıştır. Konuşma metinleri tekrar kontrol edilerek verilen cevaplar katılımcılarla birlikte kontrol edilmiştir. Görüşme formundaki sorular katılımcılara son test yapıldıktan sonraki derste ve haftada yöneltilmiştir. Görüşme ortalama 10 dakika sürmüştür.

3.2.3 Uygulama gözlem formu.

Eğitim içerisinde Scratch ile yapılan uygulamaları incelemek amacıyla “Uygulama Gözlem Formu” oluşturulmuştur. Gözlem formu hazırlanırken alanında nitel araştırma yapmış iki uzman tarafından soruların uygulanabilirliği kontrol edilmiştir. Araştırmanın amacına uygun araştırmacı ve uzman kişi tarafından “Uygulama Gözlem Formu” oluşturmuştur (bkz. Ek 5). Nitel araştırmada gözlem, araştırmaya konu olan olay, olgu ve duruma ilişkin derinlemesine ve ayrıntılı açıklamalar yapmaya yöneltilmektedir (Yıldırım ve Şimşek, 2003). Uygulama Gözlem formu sadece deney grubunda yer alan 19 kişiye uygulanmıştır. Öğrencilerden beklenen 9 temel kazanımdan oluşmaktadır. Bu kazanımlar eğitim sürecinde işlenen örnekler algoritma sorularını kapsamaktadır (bkz. Ek 3). Bunun yanında uygulama örneklerinden arta kalan zamanda oluşturulan ve Ek-5 de yer alan sınıf uygulamaları örneklerini de içermektedir. Uygulama Gözlem Formu Ek-6’da verilmiştir. Uygulama Gözlem Formu, üç kategoriden oluşmaktadır. Bunlar;

Kategori 1: “Algoritma adımlarını başlatabiliyor”. Öğrencilerin algoritma adımlarını başlatıp başlatamamaları ders esnasında gözlemlenerek kaydedilmiştir. Bu kategori algoritmanın başladığını veya sona erdiğini belirleyen temel kategoridir.

Kategori 2: “Yaptığı uygulamayı görebilmesi ve hataları düzeltebilmesi”. Ders esnasında öğrenciler gözlemlenerek ve onlara soru sorularak verdikleri cevaplara göre değerlendirilmiştir. Algoritmayı geliştirmesini, farklı çözüm yolları aramasını, çözüm yollarının doğru olup olmadığı görebilmesini sağlayan kategoridir.

Kategori 3: “Kullanıcıdan veri girmesini sağlayabiliyor ve o veriyi kontrol edebiliyor”. Yapılan uygulamalar öğrencilerden mail yoluyla alınmış ve sonradan değerlendirilmiştir. Algoritmada, kullanıcıdan verinin girişini alabilme, hesaplama ve değişken ataması yapabilme, aritmetiksel ve mantıksal ifadeler için karar verme ve karşılaştırma işlemi yapabilmesini sağlayan üst düzey bir kategoridir.

3.3. Çalışma Grubu

Araştırma 2015-2016 Eğitim-Öğretim yılı Özel Ortaokulun da yer alan 33 öğrenci üzerinde gerçekleştirilmiştir. Öğrenciler dönem içerisinde normal eğitimlerine devam etmekte olan, öncesinde ders kapsamında algoritma ve Scratch eğitimi almamış olan, beşinci ve yedinci sınıf öğrencilerinden oluşmaktadır. Araştırmaya 19 erkek öğrenci deney grubundan, 4 kız öğrenci kontrol grubundan ve 10 erkek öğrenci kontrol grubundan katılım sağlamıştır. Tablo 6’da deney ve kontrol grubunun cinsiyetlerine göre dağılımları verilmiştir. Araştırmaya toplam 33 öğrenci katılmıştır.

Tablo 6

Oluşturulan grubun cinsiyete göre dağılım

		<u>Frekans(f)</u>	<u>Yüzde (%)</u>
Deney Grubu	Erkek	19	100
Kontrol Grubu	Kız	4	28,57
	Erkek	10	71,42

Arařtırmacı, girdiđi iki sınıftan kontrol ve deney grubunu rasgele řekilde seęmiřtir. Seęilen iki sınıf arasındaki kız ve erkek öđrencilerin Tablo 6’da görölmektedir. Deney grubu katılımcıların tamamının erkek katılımcıların olduđu, kız öđrencilerin bulunmadıđı görölmektedir. Kontrol grubunda, 4 kız öđrenci arařtırmanın %28’ni oluřtururken, erkek öđrenciler %71’ini oluřturduđu görölmektedir.

3.4. Uygulama Süreci

Her iki gruba ön test uygulandıktan sonra 3 hafta boyunca algoritma eđitimi verilmiřtir. 2015-2016 öđretim yılının 14-18 Aralık haftasında uygulama süreci bařlamıřtır. Her grubun haftadaki toplam eđitim süresi 80 dakikadır. Uygulama süreci detayları hafta 1-2-3 řeklinde ařađıda verilmiřtir.

Hafta 1

‘Karřılařtıđınız bir durumu nasıl çözersiniz? Örneđ olarak vereceđiniz problem söyler misiniz? Peki bu problemi çözmek için ne yaptığınızı anlatır mısınız? Algoritma nedir?’ soruları sırayla kontrol ve deney grubuna sorulmuřtur. Algoritma ile ilgili programlama, örneđleri verilerek etkinliklere devam edilmiř, kontrol grubu beyaz tahtada iřlemi çözüme ulařtırırken, deney grubu çözümleri Scratch üzerinden yapmıřtır. Örneđler ařađıda verilmektedir.

Örneđ1: Kullanıcının girdiđi ismi ekrana yazdıran programın algoritmasını yazınız.

Örneđ2: Kullanıcının girdiđi sayısı ekrana yazdıran programın algoritmasını yazınız.

Örneđ3: Kullanıcının girdiđi sayının 10 fazlasını ekrana yazdıran programın algoritmasını yazınız.

Örneđ4: Kullanıcının girdiđi sayının pozitif veya negatif olduđunu ekrana yazdıran programın algoritmasını yazınız.

Hafta 2

Ders içeriđi iřlenmeye bařlanmıř, kontrol grubuna algoritma eđitimi MEGEP (2014) modüllerinde yer alan örnekler kullanılarak, beyaz tahta üzerinde önce adımlar yazılmıř sonra algoritmanın kontrolü sađlanmıřtır . Deney grubu ise aynı örnek soruların Scratch üzerinden çözüme ulařmıřtır. Örnekler çözüme ulařtıktan sonra animasyon ve oyun yapılmaya bařlanmıřtır (EK-5). Örnekler ařađıda verilmektedir.

Örnek1: Kullanıcının girdiđi ismi ekrana yazdıran programın algoritmasını yazınız.

Örnek2: Sorulan bir soruya verilen evet-hayır durumuna göre deđiřen programın algoritmasını yazınız.

Hafta 3

Örnek sorular ařađıdaki verilmektedir.

Örnek1: İki sayının ortalamasını bulan ve ekrana yazdıran programın algoritmasını yazınız.

Örnek2: Girilen beř sayının toplamını veren programın algoritmasını yazınız.

Deney grubunun Scratch ile hazırladıđı 8 tane uygulama örnekleri öđrencilerden alınarak kaydedilmiřtir. Aynı zamanda örnekler yapıldıktan sonra kalan zaman diliminde elde edilen ve Ek-5 de sunulan 7 animasyon veya oyundan veriler derlenmiř ve uygulama gözlem formuna göre deđerlendirilmiřtir. Uygulamaların yapılması sürecinde belirlenen kazanımlara yönelik uygulamaların sečilmesine ve bu uygulama örneklerinin öđrencilerin ilgilerini çekmeye yönelik problemlerden oluřmasına dikkat edilmiřtir. Uygulanmalar tamamlandıktan sonra öđrencilere bařarı testi uygulanarak, görüřme formlarındaki yarı yapılandırılmıř sorular yöneltilmiřtir.

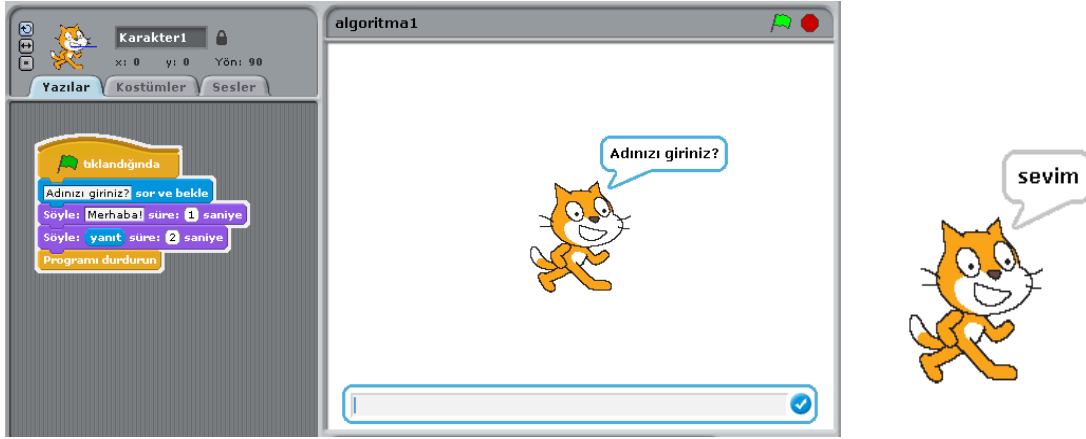
Scratch programı kullanılarak hafta 1'e ait örnek sorunun çözümü Şekil 6 da sizlere verilmiştir. Uygulama sürecinde örneklerin, Scratch ile çözümü EK 6'da gösterilmiştir.

1. Hafta

Örnek1: Kullanıcının girdiği ismi ekrana yazdıran programın algoritmasını yazınız.

Şekil 6

Örnek1'in çözümü



3.5. Verilerin Analizi

3.5.1. Nicel verilerin analizi.

Başarı Testi aracılığıyla uygulamalar sonucunda, toplanan ön test ve son test verileri SPSS (Statistical Package for Social Sciences for Windows) programı yardımıyla nicel olarak analiz edilmiştir. Akademik başarı ortalamalarının analizi bağımsız iki örnek t testi (Independent Samples Test) yöntemi ile yapılmıştır. Çalışmada bağımsız iki grup üzerinde karşılaştırma yapılmasından ve son test verilerinin normal dağılmasından dolayı parametrik olan t-testi tercih edilmiştir (Alpar, 2010).

3.5.2. Nitel verilerin analizi.

Mülakat, sözlü olarak sürdürülen, en az iki kişi arasında gerçekleşen, bir iletişim süreci olarak tanımlanmaktadır. Bu araştırmada yarı yapılandırılmış mülakatlardan yararlanılmıştır ve görüşmeler bütün öğrencilerle gerçekleştirilmiştir. Görüşme dökümanları kontrol edilip hazır hale getirildikten sonra Excel üzerinden her bir satıra numara verilmiştir.

Bu numaralar anekdotları ayırt etme aşamasında katkı sağlamıştır. Kategorilerin ve kodların oluşturulabilmesi için öncelikli olarak veriler bir bütün olarak değerlendirilmeye alınmıştır. Görüşme sorularına paralel doğrultuda temalar ve alt temalar oluşturulmuştur. Bu soruların altlarında belli başlı yer almasına olanak sağlanan kodlar belirlenmiştir (Yıldırım ve Şimşek, 2016). Görüşme kodlamaları sırasında elde edilen kategori başlıkları ve bu başlıklar altında belirlenen kodların düzenlenmesinde eğitim yönetimi alanında nitel araştırmaları olan bir alan uzmanı ile görüş birliği sağlanmıştır. İlgili kategori başlıkları ele alınarak, tema ve kodlar yeniden oluşturulmuştur.

Gözlem formu kullanılarak öğrencilerin uygulamalar sırasındaki durumları hakkında notlar alınmıştır. Analizler sırasında nitel veri bulgularında bu görüşme tutanaklarından faydalanmıştır. Bu süreç sonunda araştırmacı çalışmanın tutarlılığını incelemiştir. Sonrasında kategori ve temalar arasındaki güvenilirliğin belirlenmesi için elde edilen tablolar eğitim yönetimi alanında nitel çalışmaları olan bir alan uzmanına gönderilmiştir. Alan uzmanından alınan dönütler kapsamında yeniden düzenlemelere yer verilmiştir. Katılımcıların görüşmelerinden elde edilen önemli anekdotlar ve veriler incelenerek tema ve alt temalar altında uygun bulunan yerlere yerleştirilmiştir. Araştırma sürecinde elde edilen veriler, araştırmacının ulaştığı sonuçların ve yorumların katılımcılar ile teyit edilmesine özen gösterilmiştir. Teyit edilebilirlik nitel araştırmalar kapsamında oldukça önem taşımaktadır (Yıldırım ve Şimşek, 2016). Veri toplama aşamasında aynı kaynaklardan farklı sonuçlar çıkmasını engellemek için teyit edilebilirliğe önem verilerek bu amaç doğrultusunda veri toplama işleminin sonunda bu veriler özetlenerek katılımcıların onayı alınmıştır.

3.6. Varsayımlar

Çalışma sırasında;

- Deney ve kontrol grubundaki öğrencilerin derse olan ilgi ve tutumlarının aynı düzeyde olduğu,

- Kontrol grubunda yer alan bireylerin Scratch programını kullanmadığı,
- Çalışma süresince ve sonunda yapılan görüşmelerde sorulan sorulara samimiyetle cevap verdikleri,

varsayılmıştır.

3.7. Sınırlılıklar

Araştırma;

- 2015-2016 yılında Bursa ili, Özel Ortaokulda eğitim gören ortaokul öğrencilerinden oluşan 33 öğrenciyle sınırlıdır.
- Veri toplama araçları, mülakat soruları, ön test başarı testi, son test başarı testi ve uygulama gözlem formu verileri ile sınırlıdır.

4. BÖLÜM

Bulgular

Araştırmanın bulguları alt amaçlar doğrultusunda üç ana başlık altında sunulmuştur. Her bir başlık altında kodlara ilişkin bilgilere yer verilmiş, öğrencilerin görüşlerinden alıntılar yapılmış ve araştırmacının konuya ilişkin yorumlarına yer verilmiştir.

4.1. Nicel Araştırma Bulguları: Algoritma Öğretiminde, Scratch Kullanmanın Başarıya

Katkısı

Bu bölümde başarı testinin analizine yer verilerek, öğrenci görüşmelerinde yer alan ‘Bu eğitim sizlere neler kattı?’ sorusuna verilen yanıtların analizlerine yer verilmektedir.

Araştırmada kullanılan ve öğrencilerin başarılarını ölçen testteki 8 soru 103 kişiye uygulanmış ve testin güvenirlik katsayısı(alfa katsayısı) 0.78 olarak bulunmuştur. Başarı testi hem ön test ve son test şeklinde uygulanmıştır.

Elde edilen sayısal verilere testlerden hangisinin uygun olduğunu belirlemek için normallik testi yapılmıştır. Verilerin normal dağılıma sahip olduğu durumlarda parametrik testler, veriler normal dağılıma sahip olmadığı durumlarda ise parametrik olmayan testler uygun olmaktadır (Alpar, 2010). Yapılan son test normallik testi sonucu Tablo 7’de verilmiştir.

Tablo 7

Son Test Normallik testi sonucu

Gruplar	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	Df	Sig.	Statistic	Df	Sig.
Deney	,202	14	,127	,911	14	,164
Kontrol	,138	19	,200*	,941	19	,277

Tablo 7’de başarı puanlarının normal dağılım gösterip göstermediğini test etmek amacıyla "Kolmogorov-Smirnov" ve "Shapiro-Wilk" testi kullanılmıştır. Örneklem büyüklüğü 35’den küçük olduğu için test analizinde "Shapiro-Wilk" testinin sonuçları değerlendirilmiştir (Demir ve diğerleri ,2016). "Sig." değerleri 0.05' den büyük olduğu için tüm gruplar için "%95 güvenle veriler normal dağılımlıdır" denilebilir (Alpar, 2010).

Tablo 8

Deney ve kontrol gruplarındaki öğrencilerin ön test ve son teste ortalamaları

	<u>Gruplar</u>	<u>N</u>	<u>Ortalama</u>	<u>Std. Sapma</u>
Ön Test	Deney	19	0,0000	0,0000
	Kontrol	14	0,0000	0,0000
Son Test	Deney	19	57,0526	28,22818
	Kontrol	14	45,8571	29,59433

Tablo 8 de her iki grubun (deney-kontrol), başarı testine ilişkin ortalamaları verilmiştir.

Tablo 9

Son test puanlarının karşılaştırılmasına ilişkin bağımsız iki örnek t testi sonuçları

	<u>F</u>	<u>Sig.</u>	<u>T</u>	<u>df</u>	<u>Sig. (2-tailed)</u>
Puan	,163	,690	1,103	31	,278

Deney ve kontrol grubunun son test başarı puanları arasında istatistiksel olarak bir farklılığın olup olmadığını tespit etmek amacıyla t-testi uygulanmıştır (Tablo 9). Sonuçlar, iki farklı grubun son test başarı puanları arasında istatistiksel olarak anlamlı bir farklılık olmadığını göstermiştir ($p>0.05$). Başka bir ifade ile algoritma öğretiminde Scratch kullanmanın başarıya etkisinin bulunmadığı söylenebilir.

4.2. Nitel Araştırma Bulguları

Nitel araştırma yöntemlerinin kullanılmasıyla üç alt probleme karşılık gelen iki farklı tema kapsamında kodlar bulunmuştur. Bu bölümde birinci alt problem (Scratch programının algoritma öğretimindeki başarıya katkısı nedir?) ile ilgili öğrenci görüşlerine yer verilmiştir. Görüşmelerde elde edilen verilere ilişkin kodlar ve temalar Tablo 10'da verilmektedir.

Tablo 10

Birinci Alt Probleme İlişkin Tema ve Bu Temayı Ortaya Çıkaran Kodlar

<u>Alt Problem</u>	<u>Tema</u>	<u>Kodlar ve tekrar sayıları</u>
Scratch'ın algoritma öğretimindeki başarıya katkısı nedir?	• Kolay kodlama	• Kolay (16)
	• imkanı sunma	• Anında sonuç (15)
	• Kodlama yaparken sonucu anında görebilme	• Etkileşim(14)
	• Karşıdan girilen veriyi istenilen şekilde yönetme	• Eğlence (19) Oyun Animasyon
	• Eğlenceli öğretim imkanı sunma	• Yönetme (16)
	• Günlük hayata uyarlama	• Günlük yaşam (15)

Öğrencilerin aldıkları eğitim boyunca öğrenmiş oldukları bilgiler, başarı testinin haricinde anekdotlarla desteklenmektedir. Sorulan soru kapsamında ortaya çıkan kodlar

öğrencilerin uygulama sürecinde rahatlıkla sorulara cevaplar aradıklarını ve eğlenerek, oyunlar oluşturarak öğrendiklerini göstermektedir. Birinci alt problem sonucunda çıkan temaya göre, Scratch'in algoritma öğretiminde, kolay kullanma imkanı sunması, problem çözerken sonucun anında görülmesi, karşıdan girilen verinin istenilen şekilde yönetilebilmesi, eğlenceli öğretim imkanı sunabilmesi ve günlük hayata uyarlayarak başarıya katkı sağlaması beklenen öğrenme ortamının soru ve cevaplar doğrultusunda ilerlemesini desteklemektedirler.

Katılımcıların temel algoritma yapılarının öğretildiği derse ilişkin görüşleri incelendiğinde, deney grubunda yer alan katılımcıların ifadeleri şu şekildedir:

'Scratch ile hem eğlendim, bir ürün ortaya koyabildim.' (D1)(Eğlence, Anında Sonuç)

'Scratch konusunda oldukça iyiydim. Eğlendim, kendi karakterlerimi oluşturdum, ses ekledim.' (D2)(Kolay, Eğlence, Anında Sonuç)

'Planlama, programlama yapmayı öğrendim.' (D1)(Günlük yaşam)

'İki sayının toplamını yapmayı öğrendim, oyun yapmayı öğrendim. Bir ürün geliştirdim' (D6)(Yönetme, Eğlence)

'Komutlarla bir şeyi yönetmeyi öğrendim.' (D14)(Yönetme)

'Karşıdaki kişinin ismini alıp, ona ismiyle hitap edebildim.' (D19)(Yönetme, Anında sonuç)

Programdaki uygulamalara artan ilginin öğrencilerin bilişsel gelişimlerini desteklediği de göz önünde bulundurulduğunda, kendilerini yapabilme becerileri konusunda geliştiren öğrenciler, bir sonraki uygulama aşamasını da kavrayabilmektedirler. Soru ve cevap oluşturma aşamalarında kendi başarısını fark eden öğrenci, oyunların arka planının nasıl çalıştığına dair genel ve önemli bir bilgiye de kendiliğinden çıkarımlar yaparak ulaşmaktadır.

Bununla ilgili olarak D15:

'Scratch te soru cevap oluřturmayı öğrendim. Oyunların arka planını öğrendim. Nasıl yapılıyor diye düşünüyordum onu öğrenmiş oldum.' (D15)(Eğlence, Yönetme)

Öğrencilerin genel anlamda ifadeleri incelendiğinde, algoritma adımlarına ilişkin başlangıç bilgilerini öğrendikleri ve Scratch programını kullanarak animasyon ve oyun yapmayı öğrenerek bir ürün de geliřtirdiklerini kavradıkları ortaya çıkmıştır. Bununla birlikte piyasada bulunan oyun veya animasyonların nasıl yapıldığının da farkına varabildikleri söylenebilir.

Algoritma eğitimi sürecine ait bulgular yapılan görüşmede elde edilen verilerden derlenmiştir. Uygulamanın sürdüğü üç hafta boyunca deney grubundaki öğrenciler, soruları Scratch kullanarak çözerken zevk almışlar, uygulamaları kolayca gerçekleştirme yoluna gitmişlerdir. Karmaşık soruları daha basite indirgeyerek yapmışlardır.

Bu yaş grubundaki öğrenciler uygulamalı alanlarda kendilerini daha iyi ifade etmeleri, uygulamaları ve komutları kendi öz becerileriyle uygulama yoluna gittikleri, algoritmanın kodlama etkinliklerinin program üzerinden daha kolay bir şekilde gerçekleştiği gözlemlenmektedir. Scratch programının ilgi çekici arayüzü ve uygulamalarına ek olarak ses ve müzik eklentileri öğrencilerin uygulamalar sırasında hem eğlenmelerini sağlamış hem de ilgilerini çekerek, başka çevresel faktörlerle ilgilenmeden, daha uzun süre program başında kalabilmişlerdir.

Deney ve kontrol grubunda yer alan katılımcıların üç haftalık ders süresince aldıkları algoritma eğitimi hakkındaki genel görüşlerini ve eğitimin katkısını içeren ikinci alt probleme ait tema ve temaya ait kodlar Tablo 11'de yer almaktadır.

Tablo 11

İkinci Alt Probleme İlişkin Tema Ve Bu Temayı Ortaya Çıkaran Kodlar

<u>Alt Problem</u>	<u>Tema</u>	<u>Kodlar ve tekrar sayıları</u>
<ul style="list-style-type: none"> Katılımcıların aldıkları algoritma eğitimi doğrultusunda genel görüşleri nelerdir? Algoritma eğitiminin katılımcılara katkıları nelerdir? 	<ul style="list-style-type: none"> Sıralı, Mantıksal Ve Algoritmik Düşünme Becerisi Kazandırma Kodlama Becerisi Kazandırma 	<ul style="list-style-type: none"> Sıralı İşlem (20) Algoritmik düşünme (18) Mantıksal düşünme (18) Kodlama (15)

Deney grubundaki öğrenciler animasyon, oyun ve soru-cevap oluşturmayı öğrendiklerini belirtmişlerdir. Deney Grubunda yer alan katılımcıların algoritma eğitimi konusundaki görüşleri şu şekildedir:

‘Verilen problemlerin çözümünü sırasıyla gerçekleştirebildim. Farklı çözüm yolları oluşturabileceğimi derste söylemiştim.’(D15)(Algoritmik düşünme, Mantıksal düşünme)

‘Scratch programını rahatlıkla kullanabiliyorum ve uygulamalarımı sırasıyla gerçekleştirebiliyorum.’ (D1)(Sıralı işlem, Kodlama)

‘Problemi çözmek için planlama, programlama yapmayı öğrendim.’(D1)(Algoritmik düşünme, Mantıksal düşünme)

Kontrol grubu katılımcılarının, algoritma eğitimi ile ilgili görüşleri şu şekildedir:

K3: ‘Rahatlıkla yapabildim, Öğretildiği gibi aşamaları sırasıyla gerçekleştirdim’ (Sıralı işlem)

K4: ‘Adım adım kağıda yazdım. Önce başladım, sonra soruya göre istenileni devam ettirdim. Son soruyu yaparken zorlandım’(Sıralı işlem)

Kontrol grubundaki katılımcıların cevapları değerlendirildiğinde, bilgisayarın farklı programlarında daha fazla zaman geçirmek isteyen bir çoğunluk olduğu gözlemlenmektedir. Kontrol grubundaki katılımcılardan K5 diğer grup elemanlarından farklı olarak programın uygulamasının kolay ve anlaşılır olduğunu anekdotta ifade ederken katılımcılar arasında K14, K15, K16 ve K17 dersin sıkıcı olduğunu ve ilgilerini çekmediğini belirtmektedirler. Bunlara ek olarak kontrol grubunda bulunan K1, K2, K3 ve K4 kodlu katılımcılar teorik olarak alınan eğitimin ilk aşamalarının daha basit ve anlaşılır olduğunu vurgularken, sonraki aşamalarda karmaşık hale geldiğini ve zorlandıklarını belirtmişlerdir.

Kontrol grubuna günlük hayatta algoritmanın kullanılması ve uygulanması ile ilgili sorulduğunda, algoritma kavramının işlevini tam anlamadıklarını dile getirmişlerdir. Eğitim uygulaması sürecinde kontrol grubu katılımcıları, ‘Bilgisayar kullanarak yapsak olmaz mı?’ gibi sorular ile geri dönüşte bulunmuşlardır. Uygulama olması gereken alanlarda gelenekselci yöntem ile ilerlendiğinde karşılaşılan sorunlar burada göze çarpmaktadır.

Kontrol grubunda yer alan üç öğrenci (K6, K1, K2) eğitime diğerlerine göre daha fazla katılmışlar ve söz hakkı almışlardır. Beş öğrenci (K10, K14, K15, K16, K17) dersin hemen bitmesini istemiş ve bilgisayara oturma isteğiyle eğitimi tamamlamışlardır. Bu durum süre ve okul imkanları doğrultusunda planlı olarak ilerletilmeye çalışılmıştır. Kontrol grubundaki gayretli ve istekli öğrencilerin sınıf ortamındaki geleneksel öğretim yaklaşımından dolayı verimli bir şekilde derse katılmadıkları gözlemlenmektedir.

Kodlar incelendiğinde kolay kullanım, sonucu anında görme, verileri yönetme, oyun ve animasyon yaparak eğlenme kodlarının deney ve kontrol grubu öğrencilerine göre farkındalık oluşturulduğu göze çarpmaktadır. Bunun yanında kontrol grubunda bilgisayar kullanılmaması istenilen uygulamalar nedeniyle tekrarlanmaktadır. Eğitimin son aşamasında yapılan son örnekte, öğrencilerin zorlandığı ve konudan uzaklaştığını ortaya çıkan bulgular arasında yer almaktadır. Bunlardan yola çıkılarak öğrenciler algoritma adımlarını Scratch

programı aracılığıyla kavrar duruma gelmektedirler. Bunun yanında algoritma mantığının deney grubu öğrencileri tarafından daha anlaşılır hale geldiği görülmektedir.

Bu sorunun hazırlanmasındaki genel amaç öğrencilerin geçirdikleri süreçten memnun kalıp kalmadıklarını belirlemek, eğitimin olumlu ve olumsuz yanlarını tespit etmek ve eğitimlerin geliştirilmesi için neler yapılabileceği hakkında öneriler elde etmektir. Elde edilen bulgular ve kodlar öğrencilerin eğitim aşamalarında daha fazla uygulamaya ve daha uzun zamana ihtiyaç duyduklarını tespit etme konusunda yardımcı olmaktadır. Bu süreçte deney ve kontrol grubu arasında gözle görülür bir görüş farklılığı bulunmaktadır.

4.4. Uygulama Gözlem Formunda Elde Edilen Bulgular

Uygulama gözlem formu, üç kategoriden oluşmaktadır. Kategoriler, (1) “algoritma adımlarını başlatabiliyor”, (2) “yaptığı uygulamaları görebilmesi ve hataları düzeltebilmesi” ve (3) “kullanıcıdan veri girmesini sağlayabiliyor ve o veriyi kontrol edebiliyor” şeklindedir. Bulgular şu şekildedir:

Kategori 1 incelendiğinde, katılımcıların %100’ünün, algoritma adımlarını başlatabildiği, algoritmanın başladığını veya farklı başlatma seçenekleri olduğunu ve sona erdiğini belirleyen kod bloklarını kullanabildikleri gözlemlenmektedir. Katılımcıların Scratch ile yaptıkları oyun ve animasyonlar incelendiğinde farklı yollar ile oyunu başlatabildikleri gözlemlenmiştir (bkz. Ek 5).

Kategori 2 incelendiğinde, katılımcıların %100’ünün, algoritmayı oluşturabildikleri ve çözümünü görebildikleri, algoritmayı geliştirip farklı çözüm yolu aradıkları gözlemlenmiştir.

Kategori 3 incelendiğinde, “Eğer ise yi kullanabiliyor” ve “Algılamayı kullanarak kullanıcının veri girişini operatörler yardımıyla kontrol edebilme (yanıt,*, &. =

operatörlerinin kullanımı)” ifadelerinin 4 katılımcı (%21,05) tarafından anlaşılmadığı, 15 katılımcı (%78,94) tarafından uygulamalarında kullandıkları gözlemlenmiştir.

“Algılamayı kullanarak kullanıcının veri giriş yapmasını sağlayabiliyor”, “ Algılamayı kullanarak kullanıcının veri girişini okuyabiliyor (yanıt bloğunu kullanması)” ve “Yaptığı algoritmaları başka projelerde uygulayabiliyor” ifadelerinin 5 katılımcı (%26,31) tarafından yapılamadığı ama 14 katılımcı (%73,68) tarafından yapıldığı gözlemlenmiştir.

Nitel ve nicel veriler ayrı ayrı incelenmiştir. Her bir araştırma yönteminin getirdiği sonuçlar sayısal ve sözel bulgularla verilmiştir. Elde edilen bulguların neticesinde çıkan sonuçlar, maddeler halinde son bölümde verilmektedir.

5. BÖLÜM

Sonuç, Tartışma ve Öneriler

5.1. Sonuç ve Tartışma

Programlama öğrenmeye yeni başlayanların bir çoğu programlamayı ileri düzey eğitim alan kişilerin yapabileceğini, kendileri için zor bir iş olduğunu düşünmektedirler (Genç ve Karakuş, 2011). Programlama öğretimi, yazılım çalışmalarının sürdürülebilirliğini sağlayan önemli bir eğitim alanı olarak karşımıza çıkmakta ve önemi giderek artmaktadır (Kert ve Uğraş, 2009). Programlama öğretimini kolaylaştırmak amacıyla görsel programlama araçları geliştirilmiştir (Çatlak, Tekdal ve Baz, 2015). Bireylerin programlama becerileri kazanabilmeleri için problemi iyi anlamaları, farklı çözüm yolları bulmaları ve çözümün en kısa yolunu keşfetmeleri gerekmektedir (Yükseltürk ve Altıok, 2015). Bireyin bu becerileri alabilmesi için, algoritma mantığını iyi bir şekilde anlaması gerekmektedir. Bu nedenle programlama öğretiminin kazandırılmasında, en önemli becerinin, algoritma kavramının öneminin olduğu söylenmektedir (Köse ve Tüfekçi, 2015).

Bilgisayar Teknolojileri ve Yazılım dersi kapsamında, Scratch programı üzerinden öğretimi gerçekleştirilmiş ve öğretim sonucunda ortaokul öğrencilerinin başarı durumlarının tespiti incelenmiştir. Scratch programının seçilmesinin nedeni, programın kullanılır ve anlaşılabilir arayüzü sayesinde farklı yaş gruplarını derse teşvik edecek ölçüde eğlenceli olmasıdır (Armoni, Meerbaum-Salant & Ben-Ari, 2015; Maloney, Resnick, Rusk, Silverman & Eastmond, 2010).

Çalışmada nitel ve nicel araştırma verileri toplanmıştır. Araştırmanın nicel kısmında deney ve kontrol gruplarına ‘başarı testi’ uygulanmış ve sonuçlar analiz edilerek yorumlanmıştır. Nitel bölümde ise, yarı yapılandırılmış görüşme soruları hazırlanarak katılımcı öğrencilerin görüş ve önerileri alınmıştır. Uygulama gözlem formu kullanarak,

deney grubunda yer alan katılımcı öğrencilerin gerçekleştirdiği uygulama sonuçları incelenerek analiz edilmiştir. Çalışmada elde edilen verilerin analiz sonuçları aşağıda sıralanmaktadır.

- a. Her iki grupta da öğrenci başarısının arttığı gözlemlenmiştir. Deney grubu öğrencilerinin son test puan farkının kontrol grubu öğrencilerinin puan farkından yüksek olduğu bulunmuştur.
- b. İki farklı grubun son test başarı puanları arasında istatistiksel olarak anlamlı bir farklılık olmadığı görülmüştür ($p>0.05$). Başka bir ifade ile algoritma öğretiminde Scratch kullanmanın başarıya etkisinin bulunmadığı ortaya çıkmaktadır.

Bağımsız t testi sonuçlarında anlamlı farklılıklar olmasa bile öğrencilerin Scratch programa hakim olmaları ve uygulamanın daha fazla sürmesi durumlarında farklılıklar oluşabileceğini, yapılandırmacı eğitim yaklaşımında uygulamalı konuların öğrencilerin akıllarında uzun süreli bilgi olarak kalacağı düşünülmektedir.

- c. Eğitim esnasında deney grubunda yer alan öğrencilerin üç hafta boyunca derse ve programa ilgi ve isteklerinde bir azalma görülmemiş fakat kontrol grubunda yer alan öğrenciler, bilgisayara oturmadıklarını dile getirerek sıkılmışlardır. Öğrenci sayısına göre donanımlı sınıfların bulunduğu kurumlarda bu durumlar gözlemlenmemektedir. Bir eğitim ortamında ders materyallerinin ve araç-gereç eksikliğinin hem öğrencinin hem de öğretmenin motivasyonunu olumsuz yönde etkileyebileceği söylenebilir.

Yapılan benzer uygulamalı çalışma incelendiğinde, Adams (2010), yaptığı araştırma da 30 lisans öğrencisi ile Scratch çalışması gerçekleştirmiş; öğrencilerin Scratch ile olumlu bir bilgisayar deneyimi yaşadıklarını, programlamada temel fikirleri öğrendiklerini, bilgisayar alanında kariyer yapma isteği duyduklarını ifade etmişlerdir.

Başer (2013)'de yaptığı çalışmada geleneksel yöntemlerle anlatılan programlama derslerinin, öğrencilerin ilgisini çekmediğini ve çoğunun bilgisayar programlamayı öğrenmesinin zor ve uzmanlık gerektiren, sadece ileri seviyede eğitim almış uzman kişilerin yapabileceği iş olarak gördüğünü ortaya çıkarmıştır.

d. Yapılan uygulama gözlem formu incelendiğinde,

- Katılımcıların hepsinin algoritma adımlarını başarılı bir şekilde oluşturabildikleri, başlatma bloğunu kullanabildikleri veya farklı başlatma seçeneklerini kullandıkları,
- Katılımcıların hepsinin farklı problemlere karşı algoritmayı oluşturdukları ve çözümü görerek algoritmayı geliştirdikleri,
- Katılımcıların %78,94'unun, eğer ise kod bloğunu kullanabildikleri ve operatörler yardımıyla kullanıcıdan gelen veriyi işleyebildikleri,
- Katılımcıların %73,68'in, algılama kod bloğunu kullanarak, veriyi etkileşimli bir şekilde yönettiği görülmüştür.

Ozoran ve arkadaşları (2012), üniversite öğrencilerine algoritma eğitimi ve programlamaya başlangıç için Scratch temelli bir eğitim düzenleyip bu becerilerin kazandırılmasında Scratch programının tüm karakterleri ve görsel ortamı yönetmenin, algoritma tasarım becerilerini geliştirmesinde olumlu bir etkisi olduğunu belirtmişlerdir.

e. Öğrenci görüşlerine bakıldığında deney grubunda yer alan öğrenciler için Scratch ile algoritma öğrenmenin eğlenceli olduğu, animasyon ve oyunsal olarak hoşlarına gittiği ve zorluk çekmedikleri anlaşılmaktadır. Kontrol grubunda ise öğrencilerin çoğu sıkılmış ve bilgisayarlara oturmadıkları için algoritma öğrenmede zorluk yaşadıklarını belirtmişlerdir. Yapılan uygulamaların pek çok açıdan değerlendirilmesi gerektiği ve

ortam koşullarının uygulamaların etki alanını doğrudan ilgilendirdiği göze çarpmaktadır.

Kalelioğlu ve Gülbahar (2014)'da yaptığı çalışmasında, Scratch ile programlama öğretiminin 5. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmışlardır. Nitel araştırma sonuçlarına göre öğrencilerin çoğunun Scratch problemleri çözerken farklı yolları kullandıklarını, karakterlere komut vermekten hoşlandıklarını, Scratch programının kullanımını kolay bulduklarını ve kendilerini programlama konusunda geliştirmek istediklerini ifade etmişlerdir.

Benzer bir uygulama olan Fesakis ve Serafeim (2009)'in yaptığı çalışmada Scratch kursuna katılan öğrencilerin %65'i Scratch programının kullanımının kolay olduğunu, %85'i ara yüzün basit ve anlaşılır olduğunu ifade etmişlerdir.

Bu çalışmada uygulama yapılan gruplara bakıldığında deney grubunun görüşlerinin (eğlenceli, animasyon ve oyunsal olması), Fesakis ve Serafeim (2009)'in yaptığı çalışmayla benzer olduğu gözlemlenirken, kontrol grubunun ortam koşulları ve gelenekselci tutum sebebiyle öğrenme güçlüğü çektikleri, bunun sonucunda olumlu görüş bildirmedikleri ortaya çıkmaktadır.

Maloney ve diğerleri (2010)'ne göre Scratch aracı, programlamayı eğlenceli ve görsel hale getirmesinin yanında algoritma kavramlarının daha kolay öğrenilmesine yardımcı olmaktadır.

- f. Öğrencilerin oyunlara karşı bakış açılarının ve bilgisayarda neler yapabileceklerine ilişkin görüşlerinin değiştiği görülmektedir.

Genç ve Karakuş (2012), öğrencilerin oyunlara bakış açılarını tespit etmek amacıyla gerçekleştirdikleri çalışmada, Scratch programını basit, kolay ve eğlenceli bulmuşlar, Scratch ile programlama yapılarının öğrenilmesinin kolay olduğunu belirtmişlerdir. Dersin kapsamında oyun temasının olması öğrencileri motive etmiş ve

öğrencilerin sürece dahil olduklarını ortaya çıkarmışlardır. Öğrencilerin oyun programlama süreçlerinde, oyunu öncesinde düşünceleriyle kurgulayıp hayal güçlerini de kullanarak planladıktan sonra gerçek koşullara uygun olarak tasarımlarını sırasında hem düşünme becerisini geliştirmiş hem de yapılandırılmış problem çözümlerine katkı sağlamaktadır.

Programlama dillerinin öğretilmesi ve bu dillerin öğretimi aşamasında verim alınması öğrencilerin problem çözme becerileriyle ilişkilidir. Bu aşamada başarı düzeylerinin düşük olması gözlemlenmektedir. Programlama dillerinin öğrenmekte zorlanılmasının pek çok nedeni vardır. Programlama eğitimi aldığı süre içerisinde uygun eğitim ortamı sunulmayan öğrenci, derse karşı olan ilgi ve merakını kaybetme sürecine girmektedir (Şahin & Namlı, 2017). Bunların dışında özellikle küçük yaş gruplarındaki öğrencilerin programlama dillerine karşı önyargıları bulunmaktadır. Bunların önüne geçilebilmesi adına programlama sürecinin her seviyeye uygun hale getirilmesi önem kazanmaktadır (Akpınar & Altun, 2014). Bu seviyeye uygulanabilirlik çalışmalarında özellikle kodlama programlarının öğrenilebilmesi adına animasyonlar, görsel ve işitsel materyaller, etkili oyunlar kullanılmaktadır. Bu programın içeriğinin oyun gibi eğlenceli bir tarafının olması sebebiyle ortaya çıkan problemlere, öğrenciler tarafından yaratıcı çözümler bulunması ve düşünme becerilerinin geliştirilmesine katkı sağladığı gözlemlenmektedir. Algoritma gibi anlamlandırması karışık soyut kavramlara dayalı çalışmaların Scratch programı üzerinden ilerlenerek öğretilmesi öğrenciler için akılda kalıcılığı arttırmaktadır (Çakıroğlu ve diğ., 2011).

Bilgisayar eğitiminin yaygınlaştırılması sürecinde teorik bilgiden çok pratik ve uygulamaların önem kazandığı bilinmektedir. Ortaokul düzeyindeki öğrencilere yönelik görsel programlama eğitiminin başlamasıyla birlikte, programlama dilinin daha kolay bir şekilde öğrendikleri yapılan araştırmalar sonucunda ortaya çıkmaktadır (Akpınar & Altun,

2014). Görsel programlama dillerinin öğrenilmesinde, öğrenciler, uygulamaları verilen komutların nasıl aktarmaları gerektiği konusunda kendilerini geliştirmektedirler.

Öğrenciler programlama yaparken, programlama aşamalarının bir gerekliliği olan problemin tespit edilmesini ve sonrasında çözüme ulaştırılmasını üretmek ve tasarlamakla sorumludurlar. Daha sonra bu çözümü bilgisayar programları yardımıyla aktarmanın ve uygulanabilir olmasının bir yolunu bulmak durumundadırlar. Bu döngünün eksiksiz sağlanması gerekmektedir. Söz konusu uygulama aşamaları öğrencilerin sistemi kavramasına katkıda bulunmaktadır (Şahin & Namlı, 2017). Programlamada bir problemin tespit edilmesi ve çözüm sürecinin belirlenmesi problemin analiz ve sentezinin yapılabilmesi demektir. Problemi alt ana başlıklara ayırmak ve bölerek çözüm bulmak büyük bir problemin çözülmeye çalışılmasından daha önemlidir. Bu süreçte sadece soyut ortamdaki problemlerin çözümlenmesi değil, gerçek hayatta da karşılaşılabilecek problem çözme becerilerinin gelişimi de hedeftir ki programlamanın problem çözme becerisi geliştirmeye dönük katkılarıyla ilgili bulgular yapılan çalışmalarda oldukça kuvvetlidir. Belli disiplinlerdeki kavramların uygulanabilmesi ve kavranabilmesi programlama yoluyla yapılabilmektedir. Örneğin Matematikteki değişken, fonksiyon, alt öğeler gibi kavramlar programlama yoluyla daha kolay anlaşılabilir. Programlama yoluyla farklı alanlardaki ve ders içeriklerindeki müfredatların daha iyi kavranabilmesi adına kullanılabilir. Farklı derslerin ve konuları içine alarak uygulanması, öğrencilerin bu alana karşı daha verimli ve istekli çalışmalarını güdülemektedir.

Uygulamalar sırasında kullanılan içeriklere bakıldığında (EK 5: Sınıf Uygulamaları Örnekleri); kullanılan koordinat, akvaryum, balerin gibi örnek uygulamalarda, öğrencilerin ilgilerinin ve dikkatlerinin etkinlikler üzerinde olması, ayrıca öğrencilerin bu etkinlikler üzerinden hayal güçlerini, ne ölçüde kullanabildikleri de gözlemlenmektedir. Bu hayal gücünün bilgisayar programları tarafından geliştirilmesinin sağlanması, üç boyutlu görsellerin

kurgulanması konusunda da katkı sağlamaktadır. Öğrenciler, Scratch programındaki kukla zenginliğinin ve fantezi kuklalar kullanılarak, fantezi eylemlerin gerçekleştirilmesinin, hayal güçlerini olumlu yönde etkilediği ifade edilmiştir (Burke & Kafai, 2010; Giannakos ve diğerleri, 2013; Wilson & Moffat, 2010).

Sadece algoritma ve kodlama öğretimini amaç almayarak temel eğitim içerisinde bilgisayar ve medya okuryazarlığının küçük yaşlardan itibaren doğru öğrenilmesi ve uygulanabilirliği oldukça önemlidir. Bilgisayarı ve interneti gelişimsel amaçlar çerçevesinde kullanabilen öğrencilerin ilerleyen dönemlerde programlamanın geliştirilmesi konusunda merakları ve ilgileri artmaktadır.

5.2. Öneriler

Yapılan analiz sonuçları, gözlem ve yarı yapılandırılmış mülakat sonrası gelecekte yapılacak benzer çalışmalar için şu önerilerde bulunulmuştur:

- a. Araştırmacılar tarafından, çalışma grubu fazla tutularak farklı sonuçların tespit edilmesi önerilebilir.
- b. Öğretim süresini üç haftadan daha geniş zaman diliminde uygulanması önerilebilir.
- c. Programlama eğitimleri ve kodlamanın öğretilmesi konusu uygulama aşamasında farklı öğrenme ve öğretme yaklaşımları kullanılarak karşılaştırılabileceği bir çalışma yapılabilir.
- d. Scratch ve benzeri yazılımların bilişsel ve sosyal beceriler üzerindeki etkisi araştırılabilir.
- e. Başarı testi kağıt üzerinde değil de uygulama şeklinde değerlendirme yapılabilir.
- f. Algoritma eğitimlerinde Microsoft Small Basic ile birlikte Scratch, Alice ve LEGO gibi programlar da bu eğitimler için kullanılabilir. Öğrencinin zorluk çekmeyeceği seviyeye göre metin ve blok temelli programlar kullanılabilir.

- g. Milli Eğitim Bakanlıđı'nın ve ilgili il ve ilçe müdürlüklerinin bu gibi bilimsel çalışmalarını destekleyerek, arařtırmacıların uygulama ařamalarında zemin hazırlamaları önerilmektedir.

Kaynakça

- Akpınar, Y. & Altun, Y. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1- 4.
- Alpar, R. (2010). *Spor, sağlık ve eğitim bilimlerinden örneklerle uygulamalı istatistik ve geçerlik-güvenirlilik*. Detay Yayıncılık. Ankara.
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. Akademik bilişim konferansı.193-197. Dumlupınar/Kütahya.
- Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From Scratch to "real" programing. *ACM Transactions on Computing Education*, 14(4), 10-25.
- Barendsen, P. Sloep & G. v. d. Veer (Ed.).(2014). Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research. 103-111. Heerlen: Open University
- Bandura, A. (1994). Self-efficacy. V.S. Ramachaudran (Ed.) *Encyclopedia of Human Behavior* içinde (71-81). Newyork: Academic Press.
- Brooks, J. G., & Brooks, M. G. (1993). In search of understanding. *The case for constructivist classrooms*, 101-118.
- Burke, Q., & Kafai, Y. B. (2010). Programming & storytelling: Opportunities for learning about coding & composition. N. Parés & M. Oliver (Ed.), *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 348-351). NewYork: ACM.
- Burns, J. (2012). School ICT to be replaced by computer science programme. *BBC News*, 11.
- Büyüköztürk, Ş., Akgün, Ö. E., Demirel, F., Karadeniz, Ş., & Çakmak, E. K. (2015). *Bilimsel araştırma yöntemleri: Nicel Araştırma Yöntemleri*. Pegem Akademi. Ankara.

- Çatlak, Ş., Tekdal, M. & Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3), 13-25.
- Chang, S. E. (2004). Computer anxiety and perception of task complexity in learning programming-related skills. *Computers in Human Behavior*, 21,713-728. Retrieved January 20, 2009, from ScienceDirect database.
- Claypool, M. (2013). Dragonfly: Strengthening programming skills by building a game engine from scratch. *Computer Science Education*, 23(2), 112-137.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug), 2493-2537.
- Coşar, M. (2013). Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri (Yayımlanmamış doktora tezi). *Gazi Üniversitesi*, Ankara.
- Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches*. London: Sage Publications.
- Creswell, J. W., & Clark, V. L. P. (2017). *Designing and conducting mixed methods research*. Thousand Oaks, California: Sage Publications.
- Çoban, G. (2018). *Bilim ve sanat merkezleri yöneticilerinin yaşadıkları zorluklar ve çözüm yollarına ilişkin görüşleri: İstanbul örneği*. (Yayımlanmamış Yüksek Lisans Tezi), Kültür Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.
- Dahotre, A., Zhang, Y., & Scaffidi, C. (2010). A qualitative study of animation programming in the wild. G. Succi, M. Morisio & N. Nagappan (Ed.), *Proceedings of the 2010 ACM*

IEEE International Symposium on Empirical Software Engineering and Measurement
(pp.275-284). New York: ACM.

Demir, E., Saatçiođlu, Ö., & İmrol, F. (2016). Uluslararası dergilerde yayımlanan eğitim arařtırmalarının normallik varsayımları aısından incelenmesi. *Current Research in Education*, 2(3), 130-148.

Demirci, N., (2003). *Bilgisayarla etkili öđretme stratejileri ve fizik öđretimi*, Ankara: Nobel Yayınları

Demirel, Ö. (2002). *Eđitimde program geliřtirme* (4. Baskı), Ankara: Pegem A Yayıncılık

Demirel, Ö. (2011). *Öđretim ilke ve yöntemleri: Öđretme sanatı*. Pegem Akademi.

Demirel, Ö. (2012). *Eđitim sözlüđü dictionary of education*. Pegem A Yayıncılık.

Demirer, V., & Nurcan, S. A. K. (2016). Programming education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimini ve yeni yaklaşımlar. *Eđitimde Kuram ve Uygulama*, 12(3), 521-546.

Doymuş, K., Şimşek, Ü., ve Şimşek, U. (2005). İşbirlikçi öğrenme yöntemi üzerine derleme: işbirlikçi öğrenme yöntemi ve yöntemle ilgili çalışmalar a review on cooperative learning method: i. cooperative learning method and studies. *Erzincan Üniversitesi Eğitim Fakültesi Dergisi*, 7(1),59-79

Erol, O., (2015). *Scratch İle Programlama Öğretiminin Bilişim Teknolojileri Öğretmen Adaylarının Motivasyon ve Başarılarına Etkisi* (Doktora Tezi). YÖK Tez Yönetim Merkeziden alınmıştır.

Ersoy, H., Madran, R.O., Gülbahar, Y. (2011). Programlama Dilleri Öğretimine Bir Model Önerisi: Robot Programlama. *Akademik Bilişim '11 - XIII. Akademik Bilişim Konferansı Bildirileri*, İnönü Üniversitesi Malatya.

Estonya'da bilgisayar dili 1'inci sınıfa girdi. (2012). Hürriyet Gazetesi.

<http://www.hurriyet.com.tr/planet/21405007.asp> adresinden elde edildi.

- Fal, M., & Cagiltayc, N. E. (2013). How scratch programming may enrich engineering education. In *Proceedings of the 2nd International Engineering Education Conference. Mourtos, NJ.*
- Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. P. Brézillon, I.
- Garner, S. (2003). Acemilerin programlamayı öğrenmesine yardımcı olacak kaynakları ve araçları öğrenmek. *Gelen bilim ve bilişim eğitimi ortak bir konferans (INSITE) Haber* (s. 213-222).
- Genç, Z. & Karakuş, S., (2011). Tasarımla Öğrenme: Eğitsel Bilgisayar Oyunları Tasarımında Scratch Kullanımı. *5 th International Computer & Instructional Technologies Symposium*, Fırat University Elazığ.
- Giannakos, M. N., Jaccheri, L., & Proto, R. (2013). Teaching computer science to young children through creativity: Lessons learned from the case of Norway. M. v. Eekelen, E.
- Hwang, W. Y., Wang, C. Y., Hwang, G. J., Huang, Y. M., & Huang, S. (2008). A web-based programming learning environment to support cognitive development. *Interacting with Computers*, 20(6), 524-534. <https://doi.org/10.1016/j.intcom.2008.07.002>
- Kalelioğlu, F. & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education*, 3(1), 33-50.
- Karabak, D., & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 163-169.

- Keçeci, G., Alan, B., & Zengin, F. K. (2016). Eğitsel bilgisayar oyunları destekli kodlama öğrenimine yönelik tutum ölçeği: geçerlilik ve güvenirlik çalışması. *Education Sciences, 11(4)*, 184-194.
- Kert, S. B. (2008). *Elektronik performans destek sisteminin öğrencilerin akademik başarılarına ve öz-düzenlemeye dayalı öğrenme becerilerine etkisi*. Yayınlanmamış doktora tezi. Anadolu Üniversitesi, Eskişehir.
- Kert, S.B., & Uğraş, T.(2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. I. *Uluslararası Eğitim Araştırmaları Kongresi, Çanakkale*.
- Kesici, T., & Kocabaş, Z. (2007). *Bilgisayar 2 Ders Kitabı* (2. Baskı). Ankara: Semih Ofset.
- Korkmaz, Ö. (2012). The Impact of Critical Thinking and Logical-Mathematical Intelligence on Algorithmic Design Skills. *Journal of Educational Computing Research, 46(2)*, 173-193.
- Köse, U. & Tüfekçi, A. (2015). Algoritma ve akış şeması kavramlarının öğretiminde akıllı bir yazılım sistemi kullanımı. *Pegem Eğitim ve Öğretim Dergisi, 5(5)*, 569-586.
- Kukul, V. & Gökçearslan, Ş.(2015). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. *8th International Computer & Instructional Technologies Symposium, Trakya University Edirne*.
- Lamb, A. & Johnson, L. (2011). Scratch: computer programming for 21st century learners. *Teacher Librarian, 38(4)*, 64–68.
- Lee, Y.-J. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today, 34(2)*, 26-31.
- Lewis, C. M. (2010). How programming environment shapes perception, learning and goals: logo vs. scratch. *In Proceedings of the 41st ACM technical symposium on Computer science education.346-350*.

- Madda, M. J. (2013). An International “Creative Computing” Movement, 26.09.2014, <https://www.edsurge.com/n/2013-12-10-aninternational-creative-computing-movement>
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 39(1), 223-227
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B. & Resnick, M. (2010). Scratch: a sneak preview. In: *Second International Conference on Creating, Connecting, and Collaborating through Computing*. Kyoto, Japan, 104–109.
- MEGEP (2014). *MEGEP Öğretim Programları ve Modüler Öğretim Uygulama Kılavuzu*. Ankara: Milli Eğitim Bakanlığı.
- Milli Eğitim Bakanlığı (2012). *Bilişim teknolojileri ve yazılım dersi (5, 6, 7 ve 8. Sınıflar öğretim programı*. <http://ttkb.meb.gov.tr/program2.aspx/program2.aspx?islem=1&kno=196>’den alınmıştır.
- Miyadera, Y., Kurasawa, K., Nakamura, S., Yonezawa, N., & Yokoyama, S. (2007). A Real-time Monitoring System for Programming Education using a Generator of Program Animation Systems. *JCP*, 2(3), 12-20.
- Oluk, A., Korkmaz, Ö., & Oluk, H. (2018). Scratch’ın 5. Sınıf Öğrencilerinin Algoritma Geliştirme ve Bilgi-İşlemsel Düşünme Becerilerine Etkisi. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 9 (1), 54-71.
- Ozoran, D., Cagiltay, N., & Topalli, D. (2012). Using scratch in introduction to programming course for engineering students. In *2nd International Engineering Education Conference (IEEC2012)*, Vol. 2, 125-132.
- Özçakmak, Ş. (2014). *Bilgisayar kullanımı çocukta bağımlılık yapar mı?*. 22.09.2014, <http://www.haberturk.com/polemik/haber/973204-bilgisayar-kullanimi-cocukta-bagimlilik-yapar-mi?>

- Özçınar, H., Yecan, E., & Tanyeri, T. (2016). *Öğretmen gözüyle görsel programlama öğretimi*, 3. Uluslararası Eğitimde Yeni Yönelimler Konferansı, *Proceeding Book*, 71-79.
- Özdiñç, F., & Altun, A. (2014). Bilişim Teknolojileri Öğretmeni Adaylarının Programlama. *İlköğretim Online*, Vol 13, 1534-1541.
- Özmen, B., & Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 1-27.
- Özyurt, Ö., & Özyurt, H. (2015). A study for determining computer programming students' attitudes towards programming and their programming self-efficacy/Bilgisayar programcılığı öğrencilerinin programlamaya karşı tutum ve programlama öz yeterliklerinin belirlenmesine yönelik bir çalı. *Eğitimde Kuram ve Uygulama*, 11(1). 51-67.
- Russell & J.-M. Labat (Ed.), *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*. 258-262. New York: ACM.
- Schunk, D. H. (2001). *Self-regulation through goal setting*. 12.03.2007, <http://www.tourettesyndrome.net/Files/Schunk.pdf>.
- Scratch, (2018). <https://scratch.mit.edu> internet adresinden 14.01.2018 tarihinde erişildi.
- Şahin M.C. & Namlı N.A. (2017). Algoritma Eğitiminin Problem Çözme Becerisi Üzerine Etkisi. *Recep Tayyip Erdoğan Üniversitesi Sosyal Bilimler Dergisi*, 5, 135-153.
- TTKB (Talim ve Terbiye Kurulu Başkanlığı). (2012). *Ortaokul ve İmam Hatip Ortaokulu Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı*. 01.11.2015 tarihinde TTKB: <http://ttkb.meb.gov.tr> adresinden alındı.

- Türnüklü, A. (2000). Eğitim bilim arařtırmalarında etkin olarak kullanılabilir nitelikte bir arařtırma tekniđi: görüřme. *Kuram ve Uygulamada Eğitim Yönetimi Dergisi*, 24, 543–559.
- Uřun, S. (2012) *Eđitimde Program Deđerlendirme Süreçler Yaklařımlar ve Modeller*, Anı Yayıncılık: Ankara.
- Yařar, M. (2014). Bilimsel arařtırma yöntemleri dersine yönelik tutum ölçeđi geliřtirme çalıřması: Geçerlik ve güvenilirlik. *Eđitim Bilimleri Arařtırmaları Dergisi*, 4(2), 109-129.
- Yıldırım, A., & řimřek, H. (2013). *Sosyal bilimlerde nitelikte arařtırma yöntemleri*. Ankara: Seçkin Yayıncılık.
- Yükseltürk, E. & Altıok, S. (2015). Biliřim teknolojileri öđretmen adaylarının bilgisayar programlama öđretimine yönelik görüřleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65.
- Ward, B., Marghitu, D., Bell, T., & Lambert, L. (2010). Teaching computer science concepts in Scratch and Alice. *Journal of computing Sciences in Colleges*, 26(2), 173-180.

Ekler

Ek 1: Başarı Testi

SORULAR

- 1. İki sayıyı çıkarmak için gerekli programa ait algoritmayı yazınız.**
- 2. İki sayının ortalamasını bulmak için gerekli programa ait algoritmayı yazınız.**
- 3. Klavyeden girilen 5 sayının toplamını veren programa ait algoritmayı yazınız.**
- 4. Klavyeden girilen bir sayının 10 fazlasını ekrana yazdıran programa ait algoritmayı yazınız.**
- 5. İki sayıyı çıkarmak için gerekli programa ait algoritmayı yazınız.**
- 6. İki sayının ortalamasını bulmak için gerekli programa ait algoritmayı yazınız.**
- 7. Klavyeden girilen 5 sayının toplamını veren programa ait algoritmayı yazınız.**
- 8. Klavyeden girilen bir sayının 10 fazlasını ekrana yazdıran programa ait algoritmayı yazınız.**

Ek 2: Kazanım Listesi

<u>Kazanım</u>	<u>Bilgi</u>	<u>Kavrama</u>	<u>Uygulama</u>	<u>Analiz</u>	<u>Sentez</u>	<u>Değerlendirme</u>
Verilen bir problemde ifade ve eşitlikleri kullanarak çözüm üretir.	X					
Problemi çözmek için gerekli değişken, sabit ve işlemleri açıklar.		X				
Problem çözümünde kullanılacak operatörlere örnek verir.			X			
Verilen bir problemin çözümünde operatörleri kullanır.			X			
Döngü yapısını ve işlevlerini açıklar.					X	
Karar yapısını ve işlevlerini açıklar.					X	
Doğrusal mantık yapısını kullanan algoritmalar geliştirir.					X	
Döngü yapısı içeren algoritmalar oluşturur.					X	

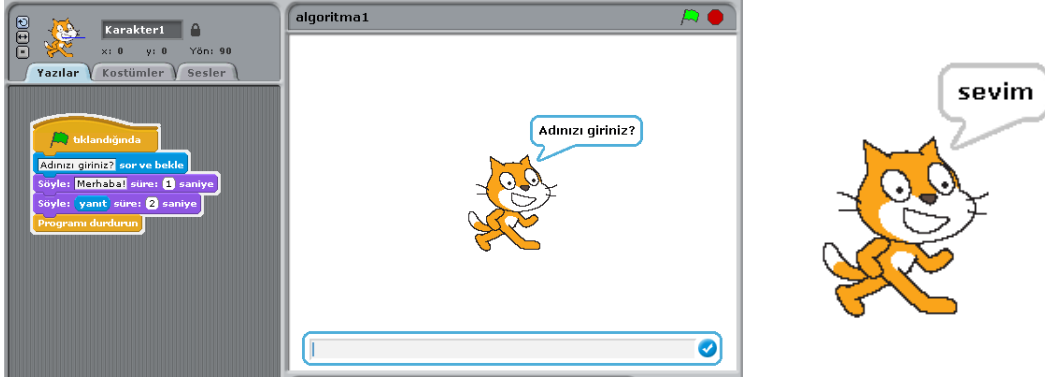
Ek 3: Scratch Uygulamalarının Haftalık Kazanımlara Göre Örnekleri

1. Hafta

Örnek1: Kullanıcının girdiği ismi ekrana yazdıran programın algoritmasını yazınız.

Şekil 7

Örnek1'in çözümü



Örnek 2: Kullanıcının girdiği sayıyı ekrana yazdıran programın algoritmasını yazınız.

Şekil 8

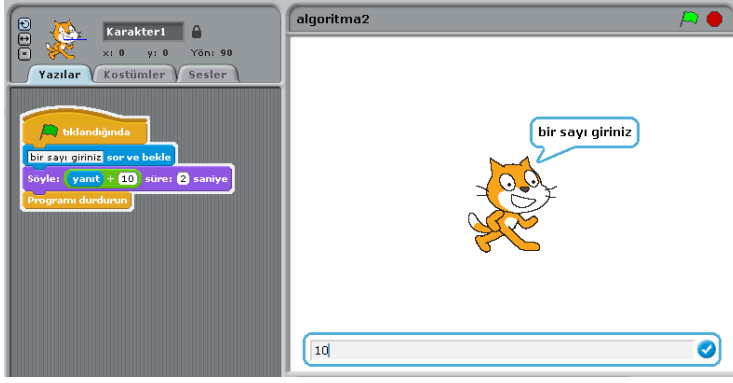
Örnek 2'in çözümü



Örnek 3: Kullanıcının girdiği sayının 10 fazlasını ekrana yazan programın algoritmasını yazınız.

Şekil 9

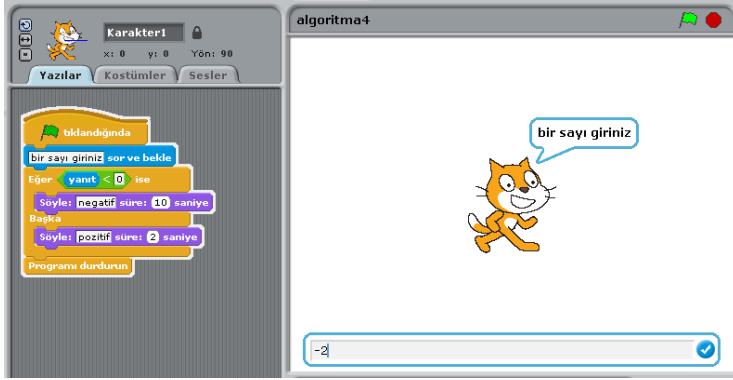
Örnek 3'in çözümü



Örnek 4: Kullanıcının girdiği sayının pozitif veya negatif olduğunu gösteren programa ait algoritmayı yazınız.

Şekil 10

Örnek 4'in çözümü

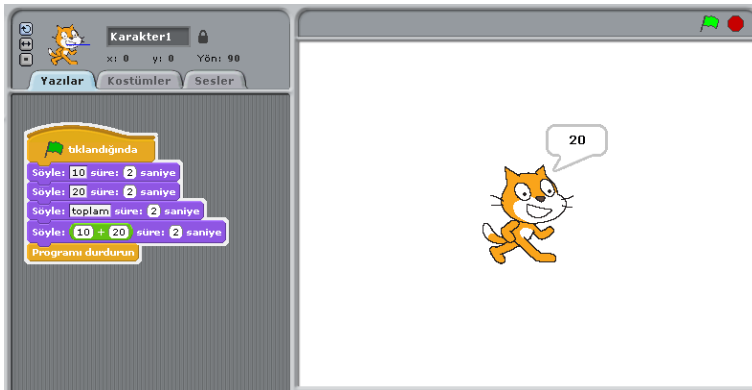


2. Hafta

Örnek 5: İki sayıyı toplamak için gerekli programa ait algoritmayı oluşturunuz.

Şekil 11

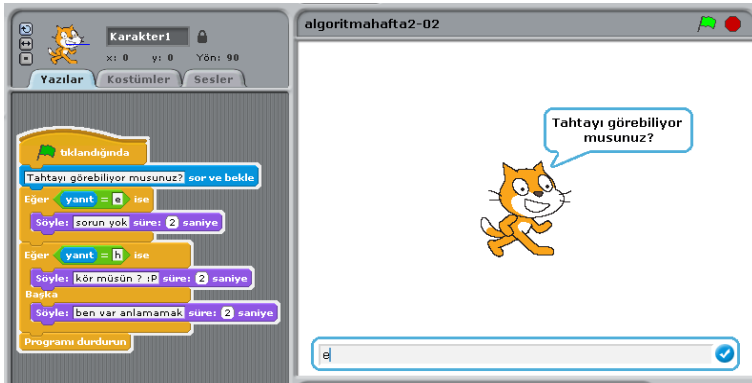
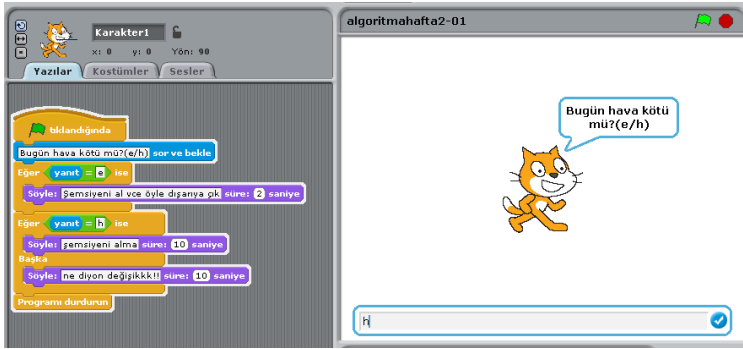
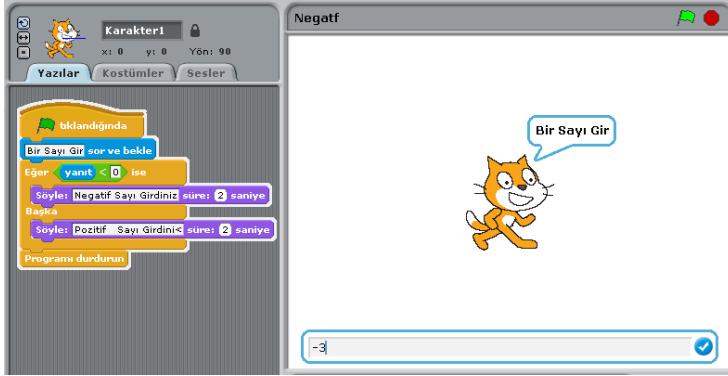
Örnek 5'in çözümü



Örnek 6: Sorulan bir soruya verilen evet – hayır durumuna göre değişen cevapların algoritmasını oluşturunuz. (Eğer kontrol deyimi kullanımı)

Şekil 12

Örnek 6'in çözümü:



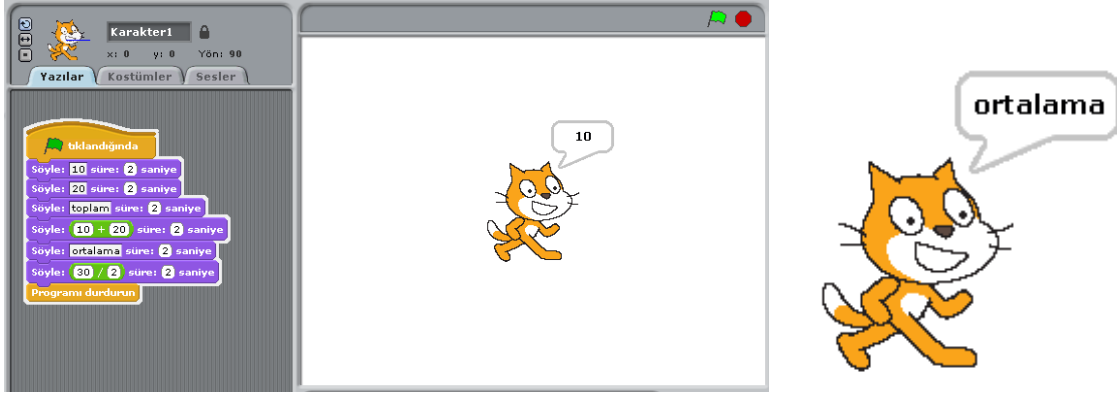
Bu ekran girilen sayının negatif veya pozitif olduğunu göstermektedir.

3. Hafta

Örnek 7: iki sayının ortalamasını bulan programa ait algoritmayı yazınız.

Şekil 13

Örnek 7'in çözümü



Örnek 8: Beş sayının toplamını veren programa ait algoritmayı yazınız (5 tane 5 girildi).

Şekil 14

Örnek 8'in çözümü



Ek 4: Görüşme Formu

ALGORİTMA ÖĞRETİMİNDE SCRATCH KULLANIMININ ORTAOKUL ÖĞRENCİLERİ ÜZERİNDEKİ BAŞARIYA ETKİSİ VE ÖĞRENCİ GÖRÜŞLERİ

Tarih: ___/___/___ **Saat(Başlangıç/Bitiş):** ____/____

GİRİŞ

Algoritma öğretiminde Scratch kullanımının ortaokul öğrencileri üzerindeki başarıya etkilerini incelemek amacıyla araştırmamı yürütmekteyim. Görüşmemize geçmeden önce, yaptığım tüm görüşmelerde verilen bilgiler sadece bu araştırmada kullanılacak ve kişisel bilgiler kesinlikle gizli tutulacaktır. Görüşmenin 10 dakika süreceğini tahmin ediyorum ve izin verirseniz görüşmeyi kaydetmek istiyorum. Bu şekilde hem zamanı daha etkili kullanabiliriz hem de sorulara vereceğiniz yanıtların kaydını daha ayrıntılı tutma fırsatı elde edebilirim.

Bu araştırmaya katılmayı kabul ettiğiniz için şimdiden teşekkür ederim. Eğer sizin bana görüşmeye başlamadan önce sormak istediğiniz bir soru varsa, önce bunu yanıtlamak istiyorum.

SORULAR

1. Aldığınız bu eğitimle ilgili olarak genel görüşlerin nelerdir?
2. Bu eğitimin ardından bilgisayarla ne yapılabileceğine dair düşüncelerinde değişiklik meydana geldi mi?
3. Bu eğitim sizlere neler kattı?

Ek 5: Sınıf Uygulamaları Örnekleri

The image displays two examples of Scratch projects. The top project, titled "KOORDİNAT-Scratch", features a character named "Karakter1" on a basketball court. The code includes a "bklandığında" (when clicked) event with a "x: 153 y: -37 konumuna gidin" (go to x: 153 y: -37 location) block, followed by three "1 saniyede x: -48 y: 52 konumuna gidin" (go to x: -48 y: 52 location in 1 second) blocks, and another "1 saniyede x: -78 y: -42 konumuna gidin" (go to x: -78 y: -42 location in 1 second) block. The bottom project, titled "Akvaryum-Scratch", features a character named "Turuncuk" in an aquarium. The code includes a "bklandığında" (when clicked) event with a "Sürekli" (forever) loop containing a "Kenarda ise, zıplattın" (if on edge, bounce) block and a "2 adım gidin" (move 2 steps) block. Another "bklandığında" (when clicked) event includes a "Mrs.Shark yaklaşıyor mu? ise sürekli" (if Mrs.Shark is approaching, then forever) loop with a "Gizleyin" (hide) block. A third "bklandığında" (when clicked) event includes a "Göster" (show) block. Both projects show the Scratch interface with the code editor, character selection, and stage view.

BALERİN- Scratch

SCRATCH Dosya Düzenle Paylaşın yardım

Hareket Görünüm Ses Kalem Kontrol Algılama Operatörler Değişkenler

Karakterler1 x: 9 y: -11 Yön: 90

Yazılar Kostümler Sesler

BALERİN

10 adım gidin
15 derece dönün
15 derece dönün
90 yönüne dönün
doğru dönün
x: 9 y: -11 konumuna gidin
'e gidin
1 saniyede x: 9 y: -11 konumuna gidin
x'i 10 değiştirin
x'i 0 yapın
y'yi 10 değiştirin
y'yi 0 yapın
Kenarda ise, zıplattın
x konumu
y konumu

tklandıığında

Sürekli

ballerina-a kostümüne geçin
0.5 saniye bekleyin
ballerina-b kostümüne geçin
0.5 saniye bekleyin
ballerina-c kostümüne geçin
0.5 saniye bekleyin
ballerina-d kostümüne geçin
0.5 saniye bekleyin

tklandıığında

Drum sesini çalın

Yeni karakter; Karakter Sahne



çiçek- Scratch

SCRATCH Dosya Düzenle Paylaşın yardım

Hareket Görünüm Ses Kalem Kontrol Algılama Operatörler Değişkenler

Karakterler1 x: 9 y: -30 Yön: 90

Yazılar Kostümler Sesler

çiçek


10 adım gidin
15 derece dönün
15 derece dönün
90 yönüne dönün
doğru dönün
x: 9 y: -30 konumuna gidin
'e gidin
1 saniyede x: 9 y: -30 konumuna gidin
x'i 10 değiştirin
x'i 0 yapın
y'yi 10 değiştirin
y'yi 0 yapın
Kenarda ise, zıplattın
x konumu
y konumu

tklandıığında

Sürekli

kostüm1 kostümüne geçin
Söyle: Ben çanak yaprak süre: 2 saniye
kostüm2 kostümüne geçin
Söyle: Ben taç yaprak süre: 2 saniye
kostüm3 kostümüne geçin
Söyle: Ben dişi organ süre: 2 saniye
kostüm4 kostümüne geçin
Söyle: Ben erkek organ süre: 2 saniye

Yeni karakter; Karakter Sahne



4-Diyalog-Yarisma- Scratch

SCRATCH Dosya Düzenle Paylaşın yardım

Hareket Kontrol Görünüm Algılama Ses Operatörler Kalem Değişkenler

Karakter4 x: -168 y: -91 Yön: 90 Yazılar Kostümler Sesler

Benim Adım Ne? sor ve bekle

Eğer yanıt = mustafa kemal atatürk ise söyleyin: Doğru

Başka söyleyin: Yanlış :>

1 saniye bekle

2+2 kac eder? sor ve bekle

Eğer yanıt = 4 ise söyleyin: Doğru

Başka söyleyin: Yanlış :>

1 saniye bekle

Atatürk kaç yılında öldü sor ve bekle

Eğer yanıt = 1938 ise söyleyin: Doğru

Başka söyleyin: Yanlış :>

1 saniye bekle

8+8 kac eder? sor ve bekle

Eğer yanıt = 16 ise söyleyin: Doğru

4-Diyalog-Yarisma

Yeni karakter; x: -337 y: -437

yarasa- Scratch

SCRATCH Dosya Düzenle Paylaşın yardım

Hareket Kontrol Görünüm Algılama Ses Operatörler Kalem Değişkenler

Karakter1 x: 159 y: -15 Yön: -74 Yazılar Kostümler Sesler

tıklandığında Sürekli 2 adım gidin Kenarda ise, zıplattın

yarasa

Yeni karakter; x: -367 y: -438

PACAMANA- Scratch

SCRATCH Dosya Düzenle Paylaşın yardım

Hareket Kontrol
Görünüm Algılama
Ses Operatörler
Kalem Değişkenler

Karakter1
x: -217 y: 141 Yön: 90
Yazılar Kostümler Sesler

sol ok tuşu basıldığında
10 adım gidin
15 derece dönün
15 derece dönün
90 yönüne dönün
doğru dönün
x: -217 y: 141 konumuna gidin
'e gidin
1 saniyede x: -217 y: 141 konumuna gidin
x'i 10 değiştirin
x'i 0 yapın
y'yi 10 değiştirin
y'yi 0 yapın
Kenarda ise, zıplattın
x konumu
y konumu

90 yönüne dönün
-90 yönüne dönün
5 adım gidin

tıklandığında
Sürekli
pengine yaklaşıyor mu?
x: -217 y: 141 konumuna gidin
kostüm1 kostümüne geçin

PACAMANA
x: -343 y: -436

Yeni karakter;
Karakte...
Sahne

EK 6: Uygulama Gözlem Formu

TEZ UYGULAMA GÖZLEM FORMU							
Öğrenci Adı-Soyadı: Gözlem Uygulamaları Analiz tarihi:	Başarı						Notlar
	Başarısız	Kısmen başarısız	Kısmen Başarılı	Başarılı	Çok Başarılı		
Değerlendirme Kriterleri							
Algoritma adımını başlatabiliyor.							
Algoritma adımını farklı yapıları kullanarak başlatabiliyor.(.. Tuşuna basıldığında)							
Algoritma yapısını kontrol edebiliyor.							
Algoritma yapısında neyin doğru çalışıp çalışmadığını göreberek düzeltebiliyor.							
Eğer ... İse yi kullanabiliyor.							
Algılamayı kullanarak kullanıcının veri giriş yapmasını sağlayabiliyor.							
Algılamayı kullanarak kullanıcının veri girişini okuyabiliyor.*yanıt							
Algılamayı kullanarak kullanıcının veri girişini operatörler yardımıyla kontrol edebiliyor.*yanıt & = operatörü							
Yaptığı algoritmaları başka projelerde uygulayabiliyor.							

Özgeçmiş

Doğum Yeri ve Yılı : MURGUL/1991

Öğr. Gördüğü Kurumlar:	Başlama Yılı	Bitirme Yılı	Kurum Adı
Lise	2005	2009	Arhavi Orhan Yücel Kız Meslek Lisesi
Lisans	2010	2014	Uludağ Üniversitesi
Yüksek Lisans	2014		Uludağ Üniversitesi

Bildiği Yabancı Diller ve Düzeyi : İngilizce - A1(TÖMER)

Çalıştığı Kurumlar:	Başlama ve Ayrılma Tarihleri	Kurum Adı
Uludağ Üniversitesi(part-time)	2013-2015	Sosyal Bilimler Enstitüsü
MEB (ücretli)	2014-2015	Görükle Mesleki ve Teknik Anadolu Lisesi
Özel Okul	2015-.....	Bursa Sınav Ortaokulu

Kullandığı Burslar: Başbakanlık Bursu

Aldığı ödüller : Uludağ Üniversitesi Proje Atölyesi “Encode your dreams” Projesi plaket

Uludağ Üniversitesi Proje Atölyesi “Üniversite Öğrenci toplulukları buluşuyor” Projesi plaket

Üye Olduğu Bilimsel ve Mesleki Topluluklar : U.Ü. Mezunlar Derneği Üyeliği

Yurt İçi ve Yurt Dışında Katıldığı Projeler : “Cultural Entrepreneurship in the Balkans” KA1 Erasmus+ projesi

Katıldığı Yurt içi ve Yurt Dışı Bilimsel Toplantılar :

7. Bilgisayar ve Öğretim Teknolojileri Eğitimi(BÖTE) Kurultayı Eskişehir Orhangazi Üniversitesi

8. Bilgisayar ve Öğretim Teknolojileri Eğitimi(BÖTE) Kurultayı İstanbul Bahçeşehir Üniversitesi

9. Bilgisayar ve Öğretim Teknolojileri Eğitimi(BÖTE) Kurultayı İzmir Ege Üniversitesi

Yayımlanan Çalışmalar :

Karadeniz Teknik Üniversitesi, Arttırılmış Gerçeklik ile Motivasyon ve Başarı: Deneysel Bir Çalışma(2016)

Aydın Menderes Üniversitesi, Okul Teknoloji Planı Hazırlama: Eşref Ergin Ortaokulu 2015-2018 Teknoloji Planı önerisi(2016)

Eskişehir Anadolu Üniversitesi, Ortaokul Öğrencilerine Programlama Becerileri Kazandırmada Scratch'in Etkililiği(2016)