



HACETTEPE ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ

Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

KODLAMA EĞİTİMİNDE ROBOT KULLANIMININ
ORTAOKUL ÖĞRENCİLERİNİN ERİŞİ, MOTİVASYON VE
PROBLEM ÇÖZME BECERİLERİNE ETKİSİ

Fatih ÖZER

Yüksek Lisans Tezi

Ankara, 2019



Liderlik, arařtırma, inovasyon, kaliteli eęitim ve deęiřim ile

Daha ileriye... En İyiyeye...



HACETTEPE ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ

Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

KODLAMA EĞİTİMİNDE ROBOT KULLANIMININ
ORTAOKUL ÖĞRENCİLERİNİN ERİŞİ, MOTİVASYON VE
PROBLEM ÇÖZME BECERİLERİNE ETKİSİ

EFFECT OF USING ROBOTICS IN TEACHING CODING ON
ACHIEVEMENT, MOTIVATION AND PROBLEM SOLVING SKILLS
OF MIDDLE SCHOOL STUDENTS

Fatih ÖZER

Yüksek Lisans Tezi

Ankara, 2019

Kabul ve Onay

Eđitim Bilimleri Enstitüsü M¼d¼rl¼đ¼ne,
Fatih ÖZER'in hazırladığı "Kodlama Eđitiminde Robot Kullanımının Ortaokul Öđrencilerinin Eriři, Motivasyon ve Problem Çözme Becerilerine Etkisi" bařlıklı bu çalıřma j¼rimiz tarafından **Bilgisayar ve Öđretim Teknolojileri Ana Bilim Dalı, Bilgisayar ve Öđretim Teknolojileri Bilim Dalında Yüksek Lisans Tezi** olarak kabul edilmiřtir.

J¼ri Bařkanı

Prof. Dr. K¼rřat ÇAđILTAY

J¼ri Üyesi (Danıřman)

Prof. Dr. Hakan TÜZÜN

J¼ri Üyesi

Prof. Dr. Süleyman Sadi SEFEROđLU

J¼ri Üyesi

Doç. Dr. Hasan ÇAKIR

J¼ri Üyesi

Dr. Öđr. Üyesi G¼knur KAPLAN

Enstit¼ Y¼netim Kurulunun
..... Tarihi ve
sayılı kararı.

Bu tez Hacettepe Üniversitesi Lisans¼st¼ Eđitim, Öđretim ve Sınav Y¼netmeliđi'nin ilgili maddeleri uyarınca yukarıdaki j¼ri üyeleri tarafından 10 / 07 / 2019 tarihinde uygun gör¼lm¼ř ve Enstit¼ Y¼netim Kurulunca / / tarihinde kabul edilmiřtir.

Prof. Dr. Ali Ekber řAHİN
Eđitim Bilimleri Enstitüsü M¼d¼r¼

Öz

Bu araştırmanın amacı, kodlama eğitiminde robot kullanımının 5. ve 6. sınıf öğrencilerinin erişimi, motivasyon ve problem çözme becerilerine etkisini belirlemektir. Araştırma Bolu ili Merkez ilçesinde 87 öğrenciyle yürütülmüştür. Yarı deneysel desenlerden kontrol grubu ön test – son test deney modeline göre hazırlanan araştırmada, altı haftalık temel programlama eğitiminin ardından, deney grubu öğrencileri altı hafta süresince robot kullanarak kodlama öğrenirken kontrol grubu öğrencileri aynı süre boyunca robot kullanmadan kodlama öğrenmişlerdir. Araştırma kapsamında veri toplama aracı olarak “Kişisel Bilgi Formu”, “Programlama Erişim Testleri”, “Çocuklar İçin Problem Çözme Envanteri” ve “Motivasyon Ölçeği” kullanılmıştır. Verilerin çözümlenmesinde bağımsız örneklem t-testi, eşleştirilmiş örneklem t-testi ve kovaryans analizi kullanılmıştır. Araştırma sonucunda programlama erişim testi sonuçlarına göre her iki grupta da ön test - son test puanları arasında istatistiksel olarak anlamlı bir farklılık bulunmuştur. Grupların son test puanları karşılaştırıldığında deney grubu lehine anlamlı bir artış olduğu tespit edilmiş, ön test puanları kontrol altına alınarak yapılan analiz sonucunda ise deney grubu lehine anlamlı bir fark ortaya çıkmıştır. Problem çözme envanteri sonuçlarına göre deney grubu ön test - son test puanları arasında istatistiksel olarak anlamlı bir farklılık bulunmuş, kontrol grubu ön test - son test puanları arasında anlamlı bir farklılık bulunmamıştır. Grupların son test puanları karşılaştırıldığında gruplar arasında anlamlı bir farklılık ortaya çıkmamış, ön test puanları kontrol altına alınarak yapılan analiz sonucunda ise deney grubu lehine anlamlı bir farklılık ortaya çıkmıştır. Motivasyon son test sonuçlarına göre ise robotik kodlama faaliyetlerinin öğrencilerin motivasyonlarını arttırdığı ancak bu artışın istatistiksel olarak anlamlı bir fark yaratmadığı belirlenmiştir.

Anahtar sözcükler: kodlama, Scratch, robot, erişim, motivasyon, problem çözme

Abstract

The aim of this study is to determine the effect of using robots in coding on achievement, motivation and problem solving skills of 5th-6th grade students. The study was conducted with 87 students in the central district of Bolu. The research design of the study was quasi-experimental type that included a control group. After six weeks of basic programming training, the experimental group students learned coding using robots for six weeks while the control group students learned coding without using robots for same period. "Personal Information Form", "Achievement Tests", "Problem Solving Inventory for Children" and "Motivation Scale" were used as data collection tools. Independent samples t-test, paired samples t-test and covariance analysis were used for data analysis. According to programming achievement scores, significant difference was found between the pretest-posttest scores of both groups. When the posttest scores of groups were compared, significant increase was found in favor of the experimental group. After the pretest scores were taken under control, a significant difference was found in favor of the experimental group. According to problem solving inventory scores, significant difference was found between the pretest-posttest scores of experimental group, whereas no significant difference was found in the control group. When the posttest scores of groups were compared, no significant difference was found between groups. When the pretest scores were taken under control, a significant difference was found in favor of the experimental group. According to motivation posttest results, robotic coding activities increased the students' motivation but this increase was not significant.

Keywords: coding, Scratch, robot, achievement, motivation, problem solving

Teşekkür

Çalışmanın ilk gününden bu zamana kadar değerli görüş ve önerileriyle desteğini her daim hissettiğim, umutsuzluğa kapıldığımda tekrar çalışmama odaklanmamı sağlayan, tecrübesizliğimden kaynaklanan hatalarıma hoşgörülle yaklaşan saygıdeğer danışmanım Prof. Dr. Hakan TÜZÜN'e,

Yüksek lisans eğitimim boyunca vermiş olduğu katkılardan dolayı değerli hocalarım Prof. Dr. Süleyman Sadi SEFEROĞLU, Prof. Dr. Halil YURDUGÜL, Prof. Dr. Yasemin KOÇAK USLUEL, Prof. Dr. Buket AKKOYUNLU'ya,

Çalışmamı inceleyen ve değerli katkılarını eksik etmeyen hocalarım Prof. Dr. Kürşat ÇAĞILTAY, Doç. Dr. Hasan ÇAKIR ve Dr. Öğr. Üyesi Gökür KAPLAN'a,

Anlayışlı tavırlarıyla yardımlarını ve desteklerini benden esirgemeyen değerli okul müdürüm Cemil AYDIN'a,

Çalışmama katılan öğrencilere,

Çalışmama verdiği destekten dolayı meslektaşım Umut Efe TÜRKER'e,

Ayrıca bu günlere gelmemde büyük pay sahibi olan rahmetli babam İbrahim ÖZER'e, annem Müberra ÖZER'e ve kardeşlerim Arzu GÖKTAŞ ve Dilek ÖZER'e sonsuz sevgilerimi ve teşekkürlerimi sunarım.

İçindekiler

Öz.....	ii
Abstract.....	iii
Teşekkür.....	iv
Tablolar Dizini.....	vii
Şekiller Dizini.....	ix
Simgeler ve Kısaltmalar Dizini.....	x
Bölüm 1 Giriş.....	1
Problem Durumu.....	1
Araştırma Problemi.....	5
Sayıtlılar.....	6
Sınırlılıklar.....	6
Tanımlar.....	7
Bölüm 2 Araştırmanın Kuramsal Temeli ve İlgili Araştırmalar.....	8
Programlama ve Programlama Eğitimi.....	8
Programlama Öğreniminde Gerekli Bilgiler.....	11
Programlama Öğretiminde Yaşanan Zorluklar.....	13
Programlama Öğretiminde Gerekli Olan Beceriler.....	16
Problem ve Problem Çözme Becerileri.....	17
Öğrenme Ortamları ve Motivasyon.....	21
Blok Tabanlı Görsel Programlama.....	22
Blok Tabanlı Programlama Öğretiminin Katkıları ile İlgili Alanyazın.....	24
Eğitimde Robotik Kullanımı.....	27
Blok Tabanlı Robotik Programlama Öğretiminin Katkıları ile İlgili Alanyazın.....	31
Bölüm 3 Yöntem.....	35
Çalışma Grupları.....	36
Veri Toplama Araçları.....	39

Uygulama Süreci.....	48
Verilerin Analizi	55
Çalışmanın iç geçerliği	55
Çalışmanın dış geçerliği	56
Bölüm 4 Bulgular ve Yorumlar.....	57
Programlama Erişi Puanlarına İlişkin Bulgular	57
Problem Çözme Becerisi Puanlarına İlişkin Bulgular	60
Motivasyon Puanlarına İlişkin Bulgular.....	63
Bölüm 5 Sonuç, Tartışma ve Öneriler	64
Sonuç ve Tartışma	64
Öneriler	71
Kaynaklar	74
EK-A: Kişisel Bilgiler Formu.....	92
EK-B: Programlama Erişi Testi I.....	93
EK-C: Programlama Erişi Testi II.....	97
EK-Ç: Uygulama Basamakları Örnekleri (Kontrol Grubu).....	102
EK-D: Uygulama Basamakları Örnekleri (Deney Grubu).....	109
EK-E: Veri Toplama Araçları Kullanım İzinleri	116
EK-F: Etik Komisyonu Onay Bildirimi	117
EK-G: Etik Beyanı	118
EK-Ğ: Yüksek Lisans Tez Çalışması Orijinallik Raporu	119
EK-H: Thesis Originality Report.....	120
EK-I: Yayımlama ve Fikrî Mülkiyet Hakları Beyanı	121

Tablolar Dizini

Tablo 1 McGill ve Volet'in (1997) Programlama Dilleri Öğretim Tablosu	12
Tablo 2 Kodlama Eğitiminde Kullanılan Öğretici Yazılımlar	23
Tablo 4 Ön Test-Son Test Kontrol Gruplu Deney Modeli Tasarım Tablosu	35
Tablo 5 Deney ve Kontrol Gruplarının Cinsiyete Göre Dağılımı	36
Tablo 6 Grupların Yaş Ortalamaları	37
Tablo 7 Grupların Matematik Akademik Başarı Ortalamaları	37
Tablo 8 Grupların Matematik Akademik Başarı Ortalamalarının Karşılaştırılması	38
Tablo 9 Grupların Erişi Ön Test Puanlarının Karşılaştırılması	38
Tablo 10 Grupların Problem Çözme Becerileri Ön Test Puanları	39
Tablo 11 Ölçeğin Alt Faktörleri ve Cronbach Alfa Katsayıları	41
Tablo 12 Ölçeğin Alt Faktörleri ve Cronbach Alfa Katsayıları	43
Tablo 13 Uygulama Süreci ve Ders Kazanımları	49
Tablo 14 Uygulama Sürecinde Uygulanan Eğitim İçerikleri	51
Tablo 15 Deney ve Kontrol Gruplarının Erişi Puanlarına İlişkin (Ön Test - Son Test) Eşleştirilmiş Örneklem t-Testi Sonuçları	57
Tablo 16 Grupların Erişi Puanlarına İlişkin (Son Test) Bağımsız Örneklem t-Testi Sonuçları	58
Tablo 17 Grupların Ön Test Son Test Erişi Puan Ortalamaları, Standart Sapmaları ve Düzeltilmiş Son Test Ortalamaları	59
Tablo 18 Grupların Erişi Testinden Aldıkları Düzeltilmiş Son Test Puanlarıyla İlgili ANCOVA Tablosu	59
Tablo 19 Grupların Çocuklar İçin Problem Çözme Envanterine-ÇPÇE İlişkin (Ön Test - Son Test) Eşleştirilmiş Örneklem t-Testi Sonuçları	60
Tablo 20 Grupların Problem Çözme Envanterine-ÇPÇE İlişkin (Son Test) Bağımsız Örneklem t-Testi Sonuçları	61
Tablo 21 Grupların Problem Çözme Envanteri (ÇPÇE) Ön Test - Son Test Puan Ortalamaları, Standart Sapmaları ve Düzeltilmiş Son Test Ortalamaları	62
Tablo 22 Grupların Problem Çözme Envanterinden (ÇPÇE) Aldıkları Düzeltilmiş Son Test Toplam Puanlarıyla İlgili ANCOVA Tablosu	62
Tablo 23 Grupların Motivasyon Ölçeği (CIS) Puanlarına İlişkin (Son Test) Bağımsız Örneklem t-Testi Sonuçları	63

Tablo 24 Arařtırmadan Elde Edilen Sonuların Arařtırılan Deęiřkenler Aısından İncelenmesi	70
Tablo 25 Arařtırma Sonularının Maliyet ve Uygulanabilirlik Aısından İncelenmesi	71



Şekiller Dizini

Şekil 1. Robotun temel bileşenlerinin bir arada olduğu WeDo2.0 örneği.	28
Şekil 2. Bağımlı ve bağımsız değişkenler.	36
Şekil 3. Scratch 2.0 genel yapısı ve arayüzü.	44
Şekil 4. Scratch programında kullanılan blok türleri.	45
Şekil 5. Picobord ve kontrolünü sağlayan kod blokları.	45
Şekil 6. LEGO education WeDo 2.0 temel set.	46
Şekil 7. LEGO education WeDo 2.0 temel set parçalarının özellikleri.	47
Şekil 8. Blok tabanlı programlama ve blok tabanlı robotik programlama ortamı. ...	54
Şekil 9. Kontrol ve deney gruplarına ait uygulama sürecinden bir görünüm.	54



Simgeler ve Kısaltmalar Dizini

DG: Deney Grubu

KG: Kontrol Grubu

MEB: Milli Eğitim Bakanlıđı

N: İstatistik veri sayısı

p: Anlamlılık deęeri

sd: Serbestlik derecesi

sh: Standart hata

ss: Standart sapma

t: Hesaplanan istatistik t deęeri

x: Aritmetik Ortalama

Bölüm 1

Giriş

Bu bölümde problem durumu, araştırmanın amacı ve önemi, araştırma problemi, sayıtlar ve sınırlılıklar ile tanımlar yer almaktadır.

Problem Durumu

Teknolojide meydana gelen hızlı değişim ve gelişmeler 20. yüzyılın sözel anlatıma dayalı eğitim yapısında köklü değişikliklere sebep olmuştur. Gelişen teknoloji ile birlikte bilgi ve iletişim teknolojileri alanında yeni öğrenme ortamları kullanılmaya başlanmış, eğitim sistemi yapısal değişime uğrayarak teknoloji yoluyla öğrenme ön plana çıkmıştır. Bilgi çağında bireyler, iş ve yaşamlarında başarıya ulaşmak için birtakım üst düzey becerileri ve öğrenme eğilimlerini kapsayan gereksinimlere ihtiyaç duymaya başlamışlardır (Ekici, Abide, Canbolat, & Öztürk, 2017). Bireyin günlük hayatta karşılaştığı problemleri tanımlaması, problemin çözümüne ilişkin bir plan yaparak en uygun çözüm yöntemini veya stratejisini uygulamaya koyması ve ileride karşılaşacağı benzer problemlerin çözümünde kullanmak üzere elde ettiği sonucu değerlendirebilmesi için problem çözme becerisi oldukça önemlidir. Ayrıca problemi parçalara ayırıştırıp yeniden tanımlamasını sağlayan analitik düşünme ve problemlere çözüm ararken farklı stratejiler kullanarak yaratıcı-yenilikçi yaklaşımlara göre çözüm üreten tasarım odaklı düşünme becerilerini geliştirmek için programlama eğitimi önemli bir gereksinim haline gelmiştir. Bu nedenle bireyler programlama öğrenmenin farklı, kolay yollarını aramakta ve elde ettikleri tecrübelerle kişisel gelişimlerine katkı sağlamaktadırlar. Farklı becerilerin gelişmesine katkıda bulunan programlama eğitimine sadece bilişim sektöründe yer almak isteyen bireyler değil, toplumun her kesiminden bireyler gereksinim duyarlar.

Programlama eğitimi, bilişim teknolojilerinin yönetimini sağlayan temel yapıtaşlarından olan yazılımların oluşturulması için gerekli bir eğitim alanıdır (Kert & Uğraş, 2009). Günümüzde bilişim sektöründe tanınmış birçok yazılımcının ortak özelliği küçük yaşlarda bilgisayar programlamayı öğrenmiş olmalarıdır. Örneğin Bill Gates program yazmaya 13 yaşında başlamış, daha sonraki yıllarda "Microsoft" yazılım şirketini kurmuştur. Günümüzde yaygın olarak kullanılan bir sosyal ağ olan Facebook'un kurucusu Mark Zuckerberg de kod yazmaya çok küçük yaşta başlayan

bir diğerk yazılımcıdır (Aytekin, Çakır, Yücel, & Kulaözü, 2018). Küçük yaşlardaki çocukların kodlama öğrenerek tasarım mantığını kavramaları yeni fikirler ortaya koyma, bulunan fikirleri uygulamaya geçirme, hatalar ile karşılaşılması durumunda hataları düzelterek çözüm üretme ve birlikte çalışma kabiliyetlerini arttıracaktır (Demirer & Sak, 2016). Üretken ve yaratıcı bireylerin yetişmesi sonucunda yazılım alanında ülke ihtiyaçları karşılanarak geleceği değıştirecek projeler ortaya konacaktır (Karabak & Güneş, 2013).

Çocuklar için programlama eğitimi zor ve karmaşık bir süreç olduğundan dolayı küçük yaşlardaki çocuklar programlaya yeterince ilgi duymamaktadırlar (Kert & Uğraş, 2009). Metin tabanlı programlama dillerinin söz dizimleri çocuklar için oldukça zordur. Programlama eğitiminde karşılaşılan zorluklardan dolayı görsel programlama ortamları tasarlanmış, böylece programlama öğrenmenin zor olan süreçleri kolaylaştırılmıştır (Resnick vd., 2009). KoduLab Game, Alice ve Scratch gibi blok tabanlı programlama yazılımları eğitim ve öğretimde yaygın olarak kullanılmaktadır. Blokları sürükleme sistemine dayalı bu yazılımlardan bazıları çevrimiçi olarak da hizmet vermektedir. Blockly, Code.org, Code Combat ve Codecademy çevrimiçi olarak kullanılabilen ortamlardır. Yaygın olarak kullanılan bu yazılımlardan biri olan Scratch, Massachusetts Teknoloji Enstitüsünde (MIT) oluşturulan bir proje grubu tarafından hazırlanmış olup, programlamaya yeni başlayan kullanıcılara kolaylık sağlamak amacıyla kod bloklarının görselleştirilerek geliştirildiği basit bir programlama dili editörüdür. Scratch yazılımı, programlamaya yeni başlayan 8 yaş ve üzerindeki öğrencilerin animasyon, sunu ve oyun gibi çeşitli yazılımlar geliştirebildikleri eğitim amaçlı olarak kullanılan blok tabanlı ücretsiz bir editördür (Kert & Uğraş, 2009). Scratch programında yapısal olarak 3 farklı blok tipi bulunmakla beraber her blok bir komuta karşılık gelir.

Eğitimde fen ve teknoloji derslerinde sık kullanılan robotlar da programlama eğitiminde kullanılmaktadır (Fidan & Yalçın, 2012). Robotlar öğrencilere eğlenceli, eğitsel ve işbirlikli etkinlikler halinde mühendislik ve teknolojinin temel kavramlarını öğretirken aynı zamanda inşa etme, tasarım ve programlama becerilerini geliştirmektedir (Fidan & Yalçın, 2012). Günümüzde programlama öğretiminde programlanabilir fiziksel robot kullanımı, sanal robot programlama ortamları ve robot programlama yazılımları kullanımı oldukça yaygındır (Numanoğlu & Keser 2017). Fiziksel programlanabilir robot kiti olarak; Lego Mindstorms Kitleri (NXT, Ev3), Lego

Wedo, Lego Boost, Parallax Robotics Kitleri, Fischertechnik Kitleri, Makeblock Kitleri, Dash ve Dot, Primo, Control iRobot® Create® 2, Intel RealSense Robotic Development Kit ve Robo Mind örnek gösterilebilir.

Robotlar hem metin hem de blok tabanlı programlama ortamları ile programlanabilir. C, Python ve Java metin tabanlı programlama ortamları iken Enchanting, Robo Pro, Modkit, miniBlog, Arduino (S4A), Open Roberta, Blockly ve mBlock blok tabanlı olarak kullanılabilen ortamlardır (Costelha & Neves, 2018; Kalelioğlu, Gülbahar, & Doğan, 2018). Scratch programı sahip olduğu eklentiler aracılığıyla eğitsel Lego robot kitleri veya elektronik devre bileşenleriyle etkileşime girerek bu kitler üzerinde kontrol sağlayabilir (Çatlak, Tekdal, & Baz, 2015). Öğrenciler, robot kitleri sensörlerini kullanarak Scratch programıyla etkileşim kurduklarında yazdıkları programın tepkilerini fiziksel ortamda anında görebilirler. Fiziksel ortamda gözlenen bu tepkiler programlama ile ilgili kavramların somutlaşmasını sağlar. Böylece öğrencilerin analiz, yaratıcılık, farklı düşünme, sistematik düşünme ve problem çözme gibi birtakım becerilerinin gelişmesi sağlanabilir (Ersoy, Madran, & Gülbahar, 2012). Lego robotları öğrencilerin mühendislik tasarım süreçleri ve programlama mantığının yanında matematiksel düşünme, işbirlikli çalışma, yaratıcılık ve problem çözme becerilerini geliştirmeye de katkıda bulunur (Fidan & Yalçın, 2012).

Programlama öğrenmede ve problem çözme becerisinin kazandırılmasında blok tabanlı kodlama ortamlarının mı yoksa robotik destekli blok tabanlı kodlama ortamlarının mı daha etkili bir öğrenme aracı olduğu sorusuna cevap verecek çalışmalar sınırlıdır. Robotların ilgi çekici özelliklere sahip olması öğrenciler tarafından sıkıcı olarak değerlendirilen kodlama sürecindeki olumsuz durumu ortadan kaldıracaktır.

Araştırmanın Amacı ve Önemi

Araştırmanın genel amacı, eğitsel robot kiti kullanarak blok tabanlı kodlama eğitimi alan ortaokul öğrencilerinin programlama erişimi, motivasyon ve problem çözme becerilerini ortaya koymaktır. Ülkemizde ortaokullarda kodlama eğitimi ilk olarak 2012 yılında Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programına dâhil edilmiş fakat müfredat içeriği esnek bir çerçevede sunulmuştur. 2018 yılında ise 5. ve 6. sınıflarda haftada iki saat zorunlu olarak okutulacak derste programlama

becerileri konu ağırlığı %50 oranına yükseltilecek bir eğitim dönemi sürececek şekilde planlanmıştır. Öğretim programı incelendiğinde programlama aracı olarak blok tabanlı yazılımlara ve çevrimiçi ortamlara yer verildiği görülmekte, ders kazanımlarının arasında “robot” kavramının yer almadığı görülmektedir. Yenilenen program çerçevesinde algoritma tasarımına ilişkin anlayış geliştirme, sözel ve görsel olarak ifade etme, problem çözmek için değişken atama, doğrusal yapılar, karar yapısı, döngü ve fonksiyon yapılarını kullanma, problemleri çözmek için uygun programlama yaklaşımını seçme ve uygulama konusunda beceriler kazandırılması amaçlanmıştır (Milli Eğitim Bakanlığı, 2018).

Romero ve Dupont (2016), eğitsel robot kitlerinin eğitimde kullanılmasıyla gelişen becerileri eleştirel düşünme, birlikte çalışma, yaratıcılık, problem çözme bilgi işlemsel düşünme becerileri olarak ele almıştır. Robotik eğitimi ile programlama eğitiminin ortak becerileri geliştirdiği söylenebilir. Bu nedenle programlanabilir robotların bu becerilere fazladan olabilecek etkisi merak konusudur. Robotik eğitimi özellikle ilkokul ve ortaokul alt sınıflarında soyut kavramların somutlaştırılması sayesinde öğrenmeyi kolaylaştırmaktadır. Somut dönemden yeni çıkmış olan çocuklar için programlama sonuçlarını otantik dünya üzerinde görebilme, dokunabilme ve tasarımında değişiklik yapabilme öğrenime katkıları açısından önemlidir. Programlama uygulaması süresince robotlar ilgi uyandırıcı özellikleri ile motivasyona olumlu katkı sağlayabileceği gibi teknik problem çözme süreçleri motivasyonel açıdan olumsuz etkiye sebep olabilir.

Blok tabanlı kodlama yazılımı Scratch eğitimiyle ilgili yapılan çalışmalar incelendiğinde, dünya genelinde ilkokul ve ortaokul seviyesinde yeterince çalışma yapılmasına rağmen ülkemizde ilkokul ve ortaokul düzeyinde yeterli çalışma yapılmadığı görülmüştür. Ülkemizde yaygın olarak kullanılan blok tabanlı programlama yazılımı Scratch kullanılarak verilen programlama eğitimiyle aynı yazılıma komple başlangıç robot kitlerinin entegre edilmesiyle verilen programlama eğitiminin çeşitli değişkenler açısından karşılaştırıldığı çalışmaya rastlanmamıştır. Bu nedenle Türkiye’deki öğrencilerin programlama becerilerini arttırmak için kodlama ortamının ve kodlanan aracın özelliklerinin ele alınması önemli görülmektedir.

Öğrenciler robotlar sayesinde otantik dünya ile soyut programlar arasındaki bağlantıyı kurabilmektedirler. Bu çalışma, bilişim teknolojileri sınıflarında yapılacak

uygulamalarda robot kitlerinin kodlama yazılımıyla birlikte kullanılmasıyla ortaya çıkacak olumlu veya olumsuz durumların belirlenmesi açısından bilişim teknolojileri öğretmenlerine yol gösterici olması bakımından önemlidir. Ayrıca bu çalışma gelecek nesiller için daha etkili öğrenme ortamlarının oluşturulmasına yönelik robot teknolojileri kullanılarak verilecek kodlama eğitiminin öğretim programlarına entegre edilme gereksiniminin belirlenmesi açısından önemlidir. Bu çalışmadan elde edilen bulgular gelecek yıllarda daha verimli bir kodlama öğretimi için gerekli olan adımların atılmasını sağlayacak sonuçları ortaya çıkarması açısından önemlidir. Ayrıca bu çalışmada elde edilen bulguların diğer araştırmalardan elde edilen bulgularla karşılaştırılması alanyazına katkı sağlayacaktır. Bu noktadan hareketle bu çalışmada kodlama eğitiminde robot kullanılarak öğrencilerin erişim düzeyleri, problem çözme beceri düzeyleri ve motivasyon seviyeleri araştırılmış, grupların programlama erişimi, problem çözme becerisi ve motivasyon düzeyleri arasındaki farklılığa bakılmıştır.

Araştırma Problemi

Blok tabanlı kodlama eğitimi alan ortaokul öğrencilerinin, erişim, problem çözme becerisi ve motivasyon düzeylerinin artırılmasında eğitsel robot kitlerinin etkisi nedir?

Alt problemler. Araştırma kapsamında aşağıdaki alt problemlere yanıt aranmıştır. Blok tabanlı kodlama eğitimi alan ve blok tabanlı robotik kodlama eğitimi alan öğrencilerin;

1. Deney ve kontrol gruplarının erişim testinden aldıkları ön test ve son test puanları arasında anlamlı bir fark var mıdır?
2. Deney ve kontrol gruplarının erişim testinden aldıkları son test puanları arasında anlamlı bir fark var mıdır?
3. Deney ve kontrol gruplarının erişim puanları arasında anlamlı bir fark var mıdır?
4. Deney ve kontrol gruplarının problem çözme envanterinden aldıkları ön test ve son test puanları arasında anlamlı bir fark var mıdır?
5. Deney ve kontrol gruplarının problem çözme envanterinden aldıkları son test puanları arasında anlamlı bir fark var mıdır?

6. Deney ve kontrol gruplarının problem çözme envanterinden aldıkları puanlar arasında anlamlı bir fark var mıdır?
7. Deney ve kontrol gruplarının motivasyon ölçeğinden aldıkları son test puanları arasında anlamlı bir fark var mıdır?

Sayıltılar

- Veri toplama amacıyla kullanılan ölçek ve testlerin araştırılan değişkenleri ölçtüğü varsayılmıştır.
- Çalışmaya katılan deney ve kontrol grubu öğrencilerinin denk gruplar olup olmadıklarının belirlenmesinde, öğrencilerin yaş ortalamaları, önceki eğitim öğretim dönemine ait matematik akademik başarı puanları, erişim ve problem çözme becerileri ön test puanlarının belirleyici olduğu varsayılmıştır.
- Çalışmaya katılan deney ve kontrol grubu öğrencilerinin deneysel uygulama öncesi ve sonrasında kendilerine uygulanmış olan ölçme araçlarına samimi cevaplar verdikleri varsayılmıştır.
- Çalışmaya katılan deney ve kontrol grubu öğrencilerinin daha önce hiçbir programlama eğitimi almadığı varsayılmıştır.
- Çalışmaya katılan deney ve kontrol grubu öğrencileri arasında herhangi bir etkileşim olmadığı varsayılmıştır.
- Araştırmacının çalışmaya katılan deney ve kontrol grubu öğrencilerine deneysel uygulama süresince tarafsız davrandığı varsayılmıştır.
- Araştırmacının çalışmaya katılan deney ve kontrol grubu öğrencilerine yaptığı uygulamaların eşdeğer olduğu varsayılmıştır.
- Erişim testinin ve öğretim programının geçerlik ve güvenilirliği için uzman kanılarına başvurulmasının yeterli olduğu varsayılmıştır.

Sınırlılıklar

- Araştırmanın çalışma grubu 2017-2018 eğitim-öğretim yılında, Bolu ilindeki bir ortaokulun 5. ve 6. sınıflarında eğitim gören 87 öğrenci ile sınırlıdır.

- Araştırmanın uygulama süreci 12 haftalık süreyle sınırlıdır.
- Bilişim teknolojileri laboratuvarında bilgisayar ve eğitsel robot seti sayısının sınırlı olması, öğrencilerin gruplar halinde çalışma yapmasını zorunlu kılmıştır.
- Eğitim programının geçerlemesi bir uzman görüşü ile sınırlıdır.
- Erişi testinin geçerlik ve güvenirlik çalışmalarının müfredat değişikliği nedeniyle yapılamaması ile sınırlıdır.

Tanımlar

Robotik Kodlama: Robotların kodlama dilleri kullanılarak oluşturulan yazılımlarla kontrol edilmesidir.

Erişi: Öğrencilerin eğitim programı öncesindeki davranışları ile eğitim programı sonrasındaki davranışları arasındaki tutarlı farka denir (Ertürk, 1984). Araştırmada grupların ön test-son test programlama başarı puanları arasındaki fark erişiş puanı olarak alınmıştır.

Motivasyon: Bireylerin kendi arzu ve istekleri doğrultusunda belirli bir amacı gerçekleştirme çabalarıdır (Koçel, 2005).

Problem Çözme: Bireylerin önceki yaşantılardan elde ettiği kuralları basit biçimde uygulanmasından farklı olarak alternatif çözüm yolları üreterek sorunların üstesinden gelebilmesidir (Korkut, 2002).

Bölüm 2

Araştırmanın Kuramsal Temeli ve İlgili Araştırmalar

Bu bölümde çalışmayla ilgili kuramsal bilgiler ve ilgili araştırmalar yer almaktadır. Bu kapsamda öncelikle programlama ve programlama eğitimi, programlama öğretiminde yaşanan zorluklar, programlama öğretiminde gerekli olan beceriler, problem ve problem çözme becerileri, öğrenme ortamları ve motivasyon ile eğitimde robotik kavramları ile ilgili bilgilere yer verilmektedir. Son olarak ilgili araştırmalar bölümünde ise eğitimde robotik kullanım süreçlerine ilişkin çalışmalar yer almaktadır.

Programlama ve Programlama Eğitimi

Programlama. Gelişen dünyada bireylerin gelecekte aktif ve üretken olmaları yaratıcı düşünme, eleştirel düşünme, algoritmik düşünme, yansıtıcı düşünme ve problem çözme becerilerinin kazandırılmasına bağlıdır. Öğrencilerin bu becerileri kazanmasında programlama öğretiminin katkısı büyüktür (Benzer & Erümit, 2017). Programlama kavramı ile ilgili çok çeşitli tanımlar bulunmaktadır. Programlama genel anlamda bilgisayara bir şeyin nasıl yapılacağını bildirmektir (Pine, 2009). Başka bir tanıma göre programlama; problem çözme, yaratıcılık ve yanal düşünmeyi içeren soyut bir beceridir (Lecky-Thompson, 2007). Kesici ve Kocabaş (2007) programlamayı gerçek hayattaki işlerin yapımı ve karşılaşılan problemlerin çözümü için gerekli komutlar dizisinin, bilgisayarın anlayabileceği makine diline çevrilmesi ve daha sonra derlenip bireylerin kullanabileceği şekilde uygulama haline getirilmesi olarak tanımlamıştır. Arabacıoğlu, Bülbül ve Filiz (2007) ise programlama dili kullanarak oluşturulan kod satırlarının bir araya getirilmesi sonucunda problemlerin çözülmesi olarak tanımlamıştır. Ersoy vd.'e (2011) göre programlama, özel kelime ve sembollerin belirli bir amacı gerçekleştirmek üzere bir araya getirilerek oluşturdukları komutlar bütünü olan programlama dilleri ile bilgisayarın belirli işleri yapabilmesini sağlayan programları oluşturma sürecidir. Belirtilen tanımlardan yola çıkarak programlama; mevcut bir problemin çözümü için kullanılan kodların, belirli kavramları kullanarak, belirli kurallara göre yazımı sonucunda uygulama oluşturma süreci olarak tanımlanabilir. Dile özgü komutların belirli kurallar çerçevesinde yazılmasına kodlama ya da programlama, programlama sonucu elde edilen son ürüne ise program ya da uygulama adı verilir. Bilgisayarlar

verilen görevleri kendi başlarına yerine getiremediklerinden dolayı yapacakları işlemleri bildiren bilgisayar programına ihtiyaç duyarlar. Bilgisayar programı, bir bilgisayar tarafından anlaşılabilen bir dizi yönergedir. Bu yönergeler aynı zamanda kodlama olarak da bilinen bilgisayar programlaması ile gerçekleştirilir. Bilgisayar programcıları bilgisayarlara her türlü görevi yaptırmak için kod yazarlar (Holland & Minnick, 2015).

Bilgisayar programlama öğretiminde öncelikle algoritma ve akış şeması mantığının etkin bir şekilde öğrenilmesi gerekmektedir. Bir program oluşturulmadan önce programın algoritmasını ve akış şemasını bilmek gerekir. Algoritmalar, bir problemi çözmek amacıyla sırayla gerçekleşecek olan adımlardan oluşurlar. Bu adımlar çizgiler, dörtgen, daire vb. geometrik şekillerden oluşan akış şeması kullanılarak görsel hale getirilebilir (Aslanyürek, 2007). Diğer bir ifadeyle algoritma programlama dillerine yol gösteren yöntemlerin adım adım sıralanması, akış şeması ise algoritmanın şekillerle gösterilmesidir. Günlük hayatta karşılaşılan birçok problemin üstesinden gelirken genellikle bilinçsiz olarak kullanılan algoritma, kullanılan programlama diline göre programın öncelikle hangi adımlarının mantıklı bir sırada gerçekleşeceğini öngörmek amacıyla programlama eğitiminde bilinçli olarak kullanıldığında kolaylık sağlayacaktır. Bu nedenle algoritma eğitiminin programlama öğretiminden önce verilmesi gerekir. Algoritma yazılırken mümkün olduğunca kısa komutlar kullanılmalıdır.

Programlama eğitiminin geçmişi. Günümüze kadar her birinin kendine özgü deyim ve yazım kurallarının bulunduğu çok sayıda programlama dili geliştirilmiştir. Programlama dillerinin gelişimi elektronik bilgisayarların ortaya çıkmasından daha eskiye dayanmaktadır. Elektronik veri işleme kapasitesine sahip ilk bilgisayar olan ENIAC 1940'lı yıllarda kullanılırken programcılık alanında daha eskiye dayanan çalışmalar mevcuttur.

1837 yılında İngiliz matematikçi Charles Babbage, "Analitik Makine" adını verdiği programlanabilir bir mekanik bilgisayar tasarlamıştır. Charles Babbage'ın tasarladığı analitik makineyi kullanan asistanı Ada Lovelace bir algoritma hazırlamış ve Bernoulli sayılarını hesaplamıştır. Ada Lovelace'in kullandığı algoritma dünyanın ilk bilgisayar programı olarak kabul edilmiştir (Fuegi & Francis, 2003). 1937 yılında İngiliz matematikçi Alan Turing, "Turing Makinası" adını verdiği mantıksal sistem içerisinde hesaplanabilir her işlemi yapan bir algoritma geliştirmiştir (Karaçay,

2006). Günümüzde kullanılan programlama mantığının temeli Turing makinasının işlemleri yapış biçimine dayanmaktadır (Ekiz, Vatansever, Zengin, & Demir, 2000). 1940'lı yıllarda düşük hız ve bellek kapasiteli elektrikle çalışan bilgisayarların geliştirilmesi programcıları elle ayarlanmış assembly dili kullanarak program yazmaya zorlamıştır. Bu yıllar modern programlama dillerinin ortaya çıkmaya başladığı dönemdir. İlk yüksek seviyeli derlenebilir programlama dili FORTRAN, IBM adına 1954'de John Backus ve ekibi tarafından geliştirilmiştir. FORTRAN'ın programcılık kavramına getirmiş olduğu en önemli yeniliklerden biri taşınabilirliğinin yüksek olmasıdır (Ersoy & Ersoy, 2018). Günümüzde hala kullanılmakta olan diller arasında McCarthy tarafından 1958'de geliştirilmiş LISP ile Grace Hopper ve Bob Bemer tarafından 1959'da geliştirilen COBOL bulunmaktadır. 1960'lı ve 1970'li yıllar "goto" kullanmadan programlama yapmak anlamına gelen "yapılandırılmış programlama" dönemine geçişin gerçekleştiği dönemdir.

1960'lı yılların sonlarında, araştırmacılar öğrencilerin programlamayı nasıl öğrendiğini araştırmak amacıyla veri toplamaya başlamıştır. Başta Papert olmak üzere bu araştırmacılar programlama ortamlarının bir takım becerileri geliştirmede bir araç olarak kullanılabileceği yönünde çeşitli iddialarda bulunmuşlardır (Guzdial & du Boulay, 2019). Programlama ortamlarının eğitim amaçlı kullanımı 1960'lı yılların sonlarında LOGO programlama dilinin ortaya çıkmasıyla başlamıştır (Calao, Moreno-Leon, Correa, & Robles, 2015). LOGO, LISP programlama dilinden türetilmiş olup programlama eğitimiyle matematiksel problem çözme becerilerine katkı sağlamak amacı ile geliştirilmiştir (McNerney, 2004).

1980'li yıllar nesne yönelimli programlamanın ortaya çıktığı dönemdir. İnternetin 1990'ların ortalarında yaygınlaşması programlama dillerinin gelişimini önemli derecede etkilemiştir. Özellikle, Java ve JavaScript programlama dilleri bu dönemde yaygınlaşmıştır. Bilgisayar programlamanın geçmişi incelendiğinde yazılım geliştiren bilim insanlarının özellikle matematik alanında çalışmalar yaptıkları görülmektedir. Alanyazın incelendiğinde programlama ve matematik arasında bir ilişki olduğu görülmektedir (Byrne & Lyons, 2001; Lewis & Shah, 2012; Ventura, 2005).

Programlama eğitiminde grup çalışması. Bilgisayar programlama etkinlikleri geleneksel anlamda bireysel çalışma olarak görülse de birtakım programlama görevlerinde grup çalışmalarının yapıldığı görülmektedir. Son yıllarda

hızla popüler hale gelen “aşırı programlama” (XP) ile işbirliğine dayalı programlamanın önemi artmıştır. Hızlı yazılım geliştirmeye dayalı bir yöntem olan “aşırı programlama” yönteminde iki programcının bir bilgisayarı kullanarak birlikte çalıştıkları ikili kodlama yöntemi kullanılmaktadır. Bu yöntemde programcılardan biri aktif olarak kod yazarken, diğer programcı yazılacak kodu gözden geçirerek mantık ve sözdizimi hatalarını bulup, taktiksel ve stratejik eksiklikleri tamamlama görevini yerine getirir (McDowell, Werner, Bullock, & Fernald, 2002). İşbirliğine dayalı programlama etkinliklerinin olumlu katkılarını ortaya çıkaran çalışmalar bulunmaktadır. Yapılan çalışmalarda işbirliğine dayalı programlamanın bireysel programlamaya göre daha hızlı ve hatasız kodlamaya olanak sağladığı belirlenmiştir (Lui & Chan, 2003; Nosek, 1998; Williams & Kessler, 2000).

Yapılandırmacı öğrenme yaklaşımı ile önem kazanan işbirlikli öğrenme, öğrencilerin öğrenme sürecinde daha fazla sorumluluk aldıkları ve daha aktif olmalarını sağlayan öğrenme yaklaşımıdır (Yaşar, 1998). Programlama dilleri ile yazılan kapsamlı programlar, bir proje grubu oluşturularak etkili bir şekilde yürütülebilmektedir. Benzer şekilde programlama eğitimi verilirken de işbirlikli proje grupları oluşturularak öğrenciler öğrenmekte oldukları programlama dilini kullanarak çeşitli programlar ortaya çıkarırlar (Kert & Uğraş, 2009).

Programlama Öğreniminde Gerekli Bilgiler

Program yazma, üst düzey bilişsel beceri gerektiren süreçlerden olduğundan oldukça zordur (Gomes, Mendes, & Marcelino, 2015; Fesakis & Serafeim, 2009; Kert & Uğraş, 2009). Programlama öğrenimi için öğrenciler farklı becerilere gereksinim duyar. Sözdizimsel (Syntactic) Bilgi, Kavramsal (Conceptual) Bilgi ve Stratejik (Strategic) bilgi olmak üzere birbiri ile ilişkili üç farklı programlama bilgisine gereksinim vardır (Bayman & Mayer, 1988). Bu bilgi ve beceriler şu şekilde açıklanabilir:

1. Sözdizimsel Bilgi (Syntactic Knowledge). Herhangi bir programlama diline ait yazım kuralları bilgisidir. Programlama dillerinin sözdizimleri farklılık gösterebileceği gibi benzerlik de gösterebilir. Programlama dillerinin sözdizim kuralları oldukça katıdır.

2. Kavramsal Bilgi (Conceptual Knowledge). Programlama dilinde kullanılan kavramların ve bu kavramların görevlerini kapsayan bilgisidir. Kavramsal

bilgi programlama dilleri arasında oldukça benzerlik gösterdiğinden dolayı herhangi bir programlama dilinde elde edilen bilgiler bir diğer programlama diline transfer edilebilmektedir.

3. Stratejik Bilgi (Strategic Knowledge). Programlama sürecinde sözdizimsel bilginin ve kavramsal bilginin birlikte kullanılarak problemin çözüme kavuşturulması için gerekli olan algoritma geliştirebilme bilgisidir.

McGill ve Volet (1997) programlama bilgilerinden sözdizimsel ve kavramsal bilgiyi Bildirimsel-Sözdizimsel, Bildirimsel-Kavramsal, İşlemsel-Sözdizimsel ve İşlemsel-Kavramsal olmak üzere dört grupta ele almışlardır (Tablo 1).

Tablo 1

McGill ve Volet'in (1997) Programlama Dilleri Öğretim Tablosu

	Bildirimsel Bilgi (Declarative Knowledge)	İşlemsel Bilgi (Procedural Knowledge)
Sözdizimsel Bilgi (Syntactic Knowledge)	Programlama dilinin yazım kuralları bilgisidir. Örnek: Pascal programlama dilinde her komutun satır sonuna noktalı virgül işaretinin konulması gerektiğini bilmek. Örnek: BASIC'te bir prosedür ile fonksiyon arasındaki yazım ayrımını fark etmek.	Programlama sürecinde yazım kurallarına uyabilme bilgisi. Örnek: BASIC programından bir metin dosyasını açıp okuyabilme.
Kavramsal Bilgi (Conceptual Knowledge)	Program çalışırken gerçekleşen olayların ne anlama geldiğini kavrayabilme bilgisidir. Örnek: Bir sözde kod parçasının ne yaptığını açıklayabilme.	Programlamada karşılaşılan problemlere çözüm üretebilme bilgisidir. Örnek: Bazı verilerin ortalamasını hesaplamak için bir kod parçası tasarlama.
Stratejik Bilgi (Strategic Knowledge)	Yeni bir problem durumunda gerekli programı tasarlama, kodlama ve test etme becerisidir.	

Programlama Öğretiminde Yaşanan Zorluklar

Karmaşık ve soyut yapıları nedeniyle programlama zor bir konu olarak kabul edilmektedir. Programlama öğrenimine yönelik yapılan araştırmalar sonucu, programlamaya yeni başlayanların programlama öğrenme sürecinde birçok sorunla karşılaştığı ortaya çıkmıştır (Dunican, 2002; Lister vd., 2004; Robins, Rountree, & Rountree, 2003; Proulx, 2000). Ismail, Ngah ve Umar (2010) programlama öğreniminde karşılaşılan problemlerle ilgili birçok çalışmayı gözden geçirmiş ve elde edilen bulguları dört maddede toplamıştır:

- Etkin olmayan öğretim yöntem ve stratejilerini kullanmak
- Problemleri analiz etmede yetersiz kalmak
- Etkin olmayan problem çözme teknikleri kullanmak
- Programlama sözdizimini ve yapılarını kullanmada yetersiz kalmak

Programlama öğreniminin zorluklarının sebebi olarak, kullanılan ders materyalleri, öğretim yöntemleri ve öğretim programından kaynaklanan zorluklar (Gray, Boyle, & Smith, 1998; Huet, Tavares, Weir, Ferguson, & Wilson, 2003; Meisalo, Suhonen, Torvinen, & Sutinen, 2002; Tavares, Brzezinski, Huet, Cabral, & Neri, 2001), programlama kavramlarının sözdiziminin kurallarını doğru şekilde kullanamama (Kölling & Rosenberg, 2001; Piteira & Costa, 2013; Sheard & Hagan, 1998), değişken ve veri türü gibi gerçek hayatta karşılığı olmayan birçok soyut kavramın yer alması (Esteves & Mendes, 2004; Jenkins, 2002; Lahtinen, Ala-Mutka, & Jarvinen, 2005; Ozoran, Çağıltay, & Topallı, 2012; Proulx, 2000), öğrencilerin programlama öğrenmeye karşı olumsuz tutum göstermeleri (Anastasiadou & Karakos, 2011; Farkas & Murthy, 2005; Hongwarittorn & Krairit, 2010; Korkmaz & Altun, 2013; Özyurt & Özyurt, 2015), programlama öğreniminde motivasyon eksikliği (Gomes & Mendes, 2007; Tella, 2007) gibi durumlar gösterilebilir. Programlama öğrenimine yeni başlayan öğrencilerin programlama performanslarını arttırmak için farklı öğretim yöntemleri kullanılmasına rağmen, birçok çalışmada öğrenci başarısı yükseltilememiştir. (Carbone & Sheard, 2002; Kay vd., 2000; Stein, 1999; Williams & Kessler, 2000).

Dunican'a (2002) göre programlama eğitimi alan öğrencilerden bazıları sadece bir zorlukla karşılaşırken, bazı öğrenciler ise aynı anda birkaç zorlukla karşılaşabilir. Bu durum öğrenciler üzerinde daha çok olumsuz etki gösterdiğinden

öğrencinin programlama konusundaki ilgi ve istekleri hızla azalmaktadır (Sheard & Hagan, 1998). Mannila, Peltomäki ve Salakoski'ye (2006) göre, acemi programcılar programlamayı öğrenirken yalnızca problem çözme durumu ile karşı karşıya kalmaz; aynı zamanda programlama dilinin sözdizimi ve anlam bilgisi gibi farklı bilgilerine gereksinim duyarlar. Başka bir ifadeyle acemi programcılar, programlama problemini bir sözde kod ile ifade edebilecek problem çözme becerisine sahip olmasına rağmen sözde kodu sözdizimsel olarak doğru şekilde bilgisayar programına dönüştürmede sorun yaşayabilirler (Dunican, 2002; Rist, 1996).

Stamouli, Doyle ve Huggard'a (2004) göre başarısız bir programlama öğrenim süreci bazı öğrencilerin dersi bırakarak programlama öğrenmekten vazgeçmesine, bazılarının ise derse devam etmesine rağmen ileriki yaşamlarında programlamadan uzak durup farklı bir meslek seçmelerine neden olacaktır. Bu nedenle programlamaya yeni başlayan öğrencilerin programla eğitiminde karşılaşılabilecek zorlukları en aza indirmek için bir takım düzenlemeler yapmak gerekmektedir. Herbert (2011) ise programlamadaki sözdizimi bilgisine daha az gereksinim duyan, görselleştirilmiş, anında geri bildirim sağlayan programlama araçlarının kullanımının programlama öğreniminde daha etkili olduğunu belirtmiştir.

Gomes ve Mendes (2007) yaptıkları bir çalışma ile programlama öğretiminde karşılaşılan zorluklar için öğrencilere ve araştırmacılara öğretim yöntemleri, çalışma yöntemleri, öğrencinin ilgi ve tutumları, programlamanın doğası ve psikolojik etkiler hakkında birtakım tespitlerde bulunmuştur:

- Öğretimin bireyselleştirilememesi,
- Öğretmenin uyguladığı yöntem ve stratejilerin her bir öğrencinin öğrenme stiline uygun olmaması,
- Dinamik kavramların anlatımında statik materyaller kullanılması,
- Öğrencilerin bireysel altyapılarının öğretmenler tarafından dikkate alınmaması,
- Öğretmenlerin bir programlama dili kullanarak problem çözmeyi teşvik etmek yerine, programlama dili ve sözdizimsel ayrıntılarını öğretmeye daha fazla odaklanmaları,
- Öğrencilerin yanlış ders çalışma yöntemleri kullanmaları,
- Öğrencilerin problemi nasıl çözeceklerini bilmemeleri,

- Öğrencide mantıksal ve matematiksel düşünme becerisinin yeterli düzeyde olmaması,
- Öğrencilerin programlamada özel uzmanlık alanlarının olmaması,
- Programlamanın soyut kavramlar içermesi,
- Programlama dillerinin çok karmaşık sözdizimlerine sahip olması
- Öğrencilerin motivasyonlarının düşük olması
- Öğrencilerin hayatlarının zor bir döneminde programlama öğrenmek zorunda kalmaları

programlama öğreniminde karşılaşılan güçlüklerin temelini oluşturmaktadır.

Mannila, Peltomäki ve Salakoski (2006), programlama öğrenme sürecinde öncelikli olarak “basit” programlama dilinin kullanılmasının ileride kullanılacak “daha karmaşık” bir dile etkisini görebilmek amacıyla iki bölümden oluşan bir çalışma yapmıştır. Çalışmada öncelikle programlamaya yeni başlayıp yarısı basit (Python) diğer yarısı karmaşık (Java) programlama dili kursuna katılmış yaşları 16 ile 19 arasında değişen öğrenciler tarafından yazılmış 60 farklı programdaki hataları incelemişler, zorluğun kullanılan dile bağlı olup olmadığını görmek amacıyla karşılaştırmalar yapmışlardır. Çalışmanın devamında ise basit dilden daha karmaşık dile geçiş sürecini incelemek amacıyla basit dil eğitimi alan sekiz öğrencinin karmaşık dil öğrenme süreçleri takip edilmiş, böylece basit ve karmaşık programlama dillerinin hem ilk programlama dili olarak hem de karmaşık dile geçiş olan ileriki aşamada kullanımları karşılaştırılmıştır. Elde edilen bulgularda “basit” olarak tanımlanan programlama diliyle hazırlanan yazılımlarda daha az yazım hatası ve mantık hatası ortaya çıktığı görülmüştür. Araştırmanın nitel bölümünde ise öncelikle “basit” bir programlama dili öğrenmenin “karmaşık” dile geçiş sürecinde faydalı olduğu sonucu ortaya çıkmış, dillerle aradaki tek farklılığın sözdiziminden kaynaklandığı görülmüştür. Ayrıca karmaşık programlama dilinde daha fazla sözdizimsel ayrıntı bulunmasına rağmen bu durumun karmaşık programlama dilinin öğreniminde sorun çıkarmadığı bulunmuştur.

Derus ve Ali (2012), bir üniversitede elektrik mühendisliği bölümünde eğitim gören öğrencilerin temel programlama öğreniminde karşılaştıkları zorlukları araştırmak ve incelemek için bir çalışma yapmıştır. Temel Programlama dersine devam eden 105 öğrenciyle görüşmeler yaparak programlamanın farklı konularını anlama düzeylerini ve programlama öğrenmelerinde karşılaştıkları zorlukları tespit

etmişlerdir. Araştırma bulgularına göre çok boyutlu diziler ve döngü ifadeleri en zor anlaşılan konular olmuştur. Ayrıca araştırmacılar, öğrencilerin en çok programlama sürecinde programlama yapısının temel kavramlarını anlamada, belirli bir görevi tamamlamak için bir program tasarlamada ve programlama dilinin sözdizimini öğrenmede zorlandıklarını belirtmişlerdir.

Tan, Ting ve Ling (2009), öğrencilerin programlama öğrenmesini zorlaştıran faktörleri ve oluşturulacak öğrenme sürecinde hangi öğretim yönteminin uygulanabileceğini ortaya koymak üzere Malezya Multimedia Üniversitesi'nde eğitimine devam eden ve "Bilgisayar Programlama 1" dersini alan 182 lisans öğrencisine anket uygulamışlardır. Araştırmadan elde edilen bulgular sonucunda öğrencilerin programlama dersinde ileri düzey programlama konularını anlamada zorlandıkları, örneklerden yararlanarak ve alıştırmayı yaparak programlamayı öğrenmeyi tercih ettikleri ve ders anlatım yoluyla öğrenmenin ilgi seviyelerini azalttığı belirlenmiştir. Ayrıca alternatif bir çözüm yolu olarak bilgisayar programlama öğrenmek için, oyun temelli öğrenme gerekliliğine dair bulgu elde edilmiştir. Bu nedenle, araştırmacılar programlama konularına yönelik oyun tabanlı bir öğrenme yöntemi önermiştir.

Kinnunen ve Malmi (2008), programlama dersinde en fazla zorlanılan konuları tespit etmek amacıyla üniversitede eğitim gören 459'u dersi geçen ve 119'u dersi bırakan toplam 578 öğrenci ile bir çalışma yapmışlardır. Çalışmanın sonuçlarına göre öğrenciler en fazla hata bulma ve kendi kodunu tasarlama konularında zorluk yaşadıklarını belirtmişlerdir.

Programlama Öğretiminde Gerekli Olan Beceriler

Programlama yapabilme yeteneğini doğrudan etkileyen beceriler problem çözme ve analitik düşünme becerileridir (Akçay & Çoklar, 2016; Linn & Clancy, 1992; Maheshwari, 1997; Şahiner & Kert, 2016). Dalton ve Goodrum (1991) bilgisayar programlamanın problem çözme stratejileri ile birlikte kullanıldığında problem çözme becerilerinin öğretilmesine de katkı sağlayabileceğini öne sürmüştür. Allison, Orton ve Powell (2002), öğrenciler tarafından öğrenilmesi zor olan programlama becerilerinin öğretilmesinin de zor olduğunu belirtmiştir.

Problem ve Problem Çözme Becerileri

Problem. Alanyazında problem kavramı araştırıldığında birbirinden farklı pek çok tanım olduğu görülmektedir. Latince bir sözcük olan “problem” kelimesinin Türkçe karşılığı “sorun” anlamına gelmektedir. Sorun; engelli ve sıkıntılı bir durumun çözümlenmesi, öğrenilmesi anlamlarına gelmektedir. Eğitim alanında ise “problem” kavramı daha yaygın olarak kullanılmaktadır (Kalaycı, 2001). Türk Dil Kurumu’na (2018) ait güncel Türkçe sözlükte problem, “Teoremler veya kurallar yardımıyla çözülmesi istenen soru, mesele” olarak açıklanmaktadır. Karasar’a (2012) göre problem, bireyde fiziksel ya da zihinsel yönden tedirginlik uyandıran, birden fazla çözüm yolu bulunan kararsızlık durumu olarak tanımlamıştır. Cüceloğlu’na (2006) göre problem, bireyin gerçekleştirmek istediği hedefine ulaşmasını güçleştiren engellerin ortaya çıkardığı durum iken Dewey’e (1933) göre problem, bireye karşı koyan, zihninde karmaşıklık yaratan her şey olarak tanımlanır (Gelbal, 1991; Güçlü, 2003; Baykul, 2009). Bingham (2004) problemi, bireylerin hedeflerine ulaşmak için bir araya getirerek elde ettiği mevcut güçlerinin karşısına çıkan engeller olarak tanımlarken, Morgan (1999) ise bireyin hedefine ulaşırken karşılaştığı engeller sonucu oluşan bir çatışma durumu olarak tanımlamıştır. Sonuç olarak problem için, “birey veya grup için üstesinden gelinmesi gereken durum” olduğu söylenebilir.

Problem çözme. Problem çözme kavramı, günümüze kadar birçok araştırmacı tarafından tanımlanmaya çalışılmıştır. Anderson’a (1980) göre problem çözme, bilişsel işlemlere sırasıyla bir hedef doğrultusunda yön verilmesidir. Korkut’a (2002) göre bilişsel becerilerden farklı olarak duyuşsal ve davranışsal becerileri de kapsayan problem çözme, bireyin önceki yaşantılardan elde ettiği kuralları basit biçimde uygulanmasından farklı olarak alternatif çözüm yolları üreterek sorunların üstesinden gelebilmesi olarak tanımlanır. Problem çözme becerisi, bireyin çözüm üretmek amacıyla gerekli olan kuralları bir araya getirerek bir problemin çözümünde kullanabilme düzeyidir (Bilen, 2006). D’Zurilla ve Goldfried (1971) problem çözmeyi, problemle başa çıkabilmek için etkili tepki seçeneklerini oluşturup bunların içinden en etkili olduğu düşünüleni seçebilmeyi içeren hem bilişsel hem de davranışsal bir süreç olarak ele almışlardır. Bu tanımlara göre problem çözme, istenilen bir duruma ulaşma yolunda karşılaşılan engelleri ortadan kaldırmak için mevcut bilgilerin kullanılma durumudur.

Problemler farklı özelliklerde olabilmektedir. Problem çözümlerine dayalı deneysel düşünmeyi savunan John Dewey, problem çözmeyi düşünmeye dayalı bir davranış olarak tanımlamış, bilimsel yöntemin basamaklarını beş adımdan oluşan bir süreç olarak belirlemiştir. Bunlar;

- Güçlük yaratan bir durumla karşı karşıya kalma,
- Bu durumda problemi keşfedip tanımlama,
- Olası çözüm yolları belirleme ve denenceler kurma,
- Denenceleri sınaama, olası sonuçları değerlendirme,
- Uygulama sonuçlarına göre denenceleri askıya alma, değiştirme ve düzeltme (Sönmez, 2011).

Bingham (2004), problem çözmeyi bir amaca ulaşırken karşılaşılan güçlükleri yenme süreci olarak ele almaktadır. Bu tanıma göre problemlerin üç temel özelliği vardır. Bu özellikler;

- Bireyin belirlediği bir amaç
- Amaca ulaşırken bireyin karşısına çıkan bir engel
- Bireyi amacına ulaşmaya yönelten içsel bir gerginlik

Bireyin amaca ulaşırken engellerle daha önceden karşılaşmamış olması, amaca ulaşmaya iten içsel bir gerginlik yaratır. Bu süreçte birey gerginlikten kurtulmanın yollarını aramaktadır.

Öğülmüş'e (2001) göre bir durumda problemden bahsedilebilmesi için;

- Mevcut durumla olması gereken durum arasında bir farkın bulunması,
- Bireyin bu farkı fark etmesi ya da algılaması,
- Algılanan farkın bireyde gerginliğe yol açması,
- Bireyin gerginliği ortadan kaldırmak amacıyla girişimlerde bulunması,
- Bireyin gerginliği ortadan kaldırmaya yönelik girişimlerinin engellenmesi gereklidir.

Kneeland'e (2001) göre problem çözme sürecinin basamakları şu şekildedir:

- Problemin farkına varma,

- Gerekli bilgileri toplayarak problemin temeline inme,
- Çözüm yollarını araştırma ve bulma,
- En uygun çözüm yolunu tespit etme,
- Problemi çözme.

Polya (1957) ise problem çözme sürecini dört basamakta incelemiştir. Bu basamaklar şu şekildedir:

Problemin anlaşılması. Problemin çözülmesi için ilk olarak problemi anlamak gerekir. Bu aşamada öğrenciden problemi tekrar etmesi istenir. Problemden bilinmeyenlerin neler olduğu, verilerin ne olduğu ve nelerin istenildiği, koşulun ne olduğu, problemde kimlerin ne şekilde etkilendiği ortaya konulmalıdır. Bu aşamada öğrenciler problemi başkasının anlayacağı şekilde kendi cümleleriyle akıcı bir şekilde yeniden ifade edebilmeli, problemi özetleyebilmeli, problemin bilinmeyenlerini, verilerini ve koşullarını gösterebilmeli, problemde verilen bilgileri şekil veya diyagramla görselleştirebilmelidir.

Çözümle ilgili stratejinin seçilmesi. Problemi anladıktan sonra problemi çözmek için bir plan yapmak gereklidir. Problemlerin türüne göre farklı çözüm yolları olduğundan doğru çözüm planını tercih ederek uygulamak gerekir. Bu aşamada öğrencilerden problemin çözümü için uygun stratejileri belirlemeleri istenir. Öğrencilere daha önce benzer bir problemle karşılaşmış olmalarının düşünmeleri sağlanarak aynı çözümün kullanılıp kullanılmayacağı tespit edilir. Bu yöntem işe yaramadıysa problem dönüştürülmeli ve çeşitlendirilmelidir. Problem çözülmiyorsa benzer bir problem çözülmeye çalışılmalıdır. Benzer problemleri çözerken asıl problemde uzaklaşıldığı durumlarda verilerin tamamen kullanma durumu ve koşullardan tam olarak yararlanma durumu sorgulanmalıdır.

Seçilen stratejinin uygulanması. Çözüm planının uygulanması aşamasında öncelikle yapılan işlemler incelenmelidir. Planın belirlenen strateji doğrultusunda uygulanıp uygulanmadığı gözden geçirilmeli, çözüm basamakları kontrol edilmeli ve çözüm basamaklarının doğruluğu ispat edilmelidir.

Geriye bakış, çözümün değerlendirilmesi. Son olarak, sonucun doğru olup olmadığının kontrol edilmesi gerekmektedir. Öğrenci planını uygulamış, çözümü yazmış ve her basamağı kontrol etmiş olmasına rağmen, çözüm yolu en etkin yol

olmayabilir, hatalar bulunabilir. Bu nedenle çözüm yolu ve sonuç kontrol edildikten sonra, elde edilen bilgiler yeniden şekillendirilerek başka problemlerin çözümünde kullanılmalıdır.

Problem çözme ile programlamayı ilişkilendiren bazı araştırmalar programlama öğretiminin öğrencilerde problem çözme becerilerini geliştirmeye yönelik olumlu katkı sağladığını ortaya koymuştur. Casey (1997) öğrencilere problem çözme becerilerinin öğretimi için programlama eğitiminin uygun olabileceği varsayımı ile deneysel bir çalışma yapmıştır. Araştırma sonucunda öğrencilerin programlama eğitiminde göstermiş olduğu başarı ile problem çözme becerilerindeki artışın doğru orantılı olduğu sonucuna ulaşılmıştır. Alkan (2018) programlama eğitiminin özel yetenekli öğrencilerin problem çözme becerileri üzerindeki etkisini araştırmak için ortaokul 5. sınıfta öğrenim gören 35 öğrenci ile deneysel bir çalışma gerçekleştirmiştir. Bu çalışma sonucuna göre, programlama eğitiminin öğrencilerin problem çözme becerilerine olumlu yönde katkı sağladığı belirlenmiştir. Bu çalışmanın sonuçlarıyla benzerlik gösteren başka çalışmalar da bulunmaktadır (Begosso & Silva, 2013; Brown vd., 2008; Coşar, 2013; Çetin, 2012; Hooshyar, Ahmad, Shamshirband, Yousefi, & Horng, 2015; Kim, Chung, & Yu, 2013; Taylor, Harlow, & Forret, 2010).

Lai ve Yang (2012), görselleştirilmiş programlama eğitiminin öğrencilerin problem çözme becerileri ve mantıksal akıl yürütme becerileri üzerindeki etkisini araştırmak amacıyla 6. sınıfta öğrenim gören 130 öğrenci ile deneysel bir çalışma gerçekleştirmiştir. Çalışma sonucuna göre, görselleştirilmiş programlama eğitiminin öğrencilerin problem çözme becerilerine olumlu yönde katkı sağladığı belirlenmesine rağmen, mantıksal akıl yürütme becerileri üzerinde önemli bir katkı belirlenmemiştir. Calder (2010) ise yaptığı çalışmada, görsel programlama araçlarının problem çözme becerilerini geliştirdiğini vurgulamıştır.

Kalelioğlu ve Gülbahar (2014) blok tabanlı programlama yazılımının ilköğretim 5. sınıf öğrencilerinin problem çözme becerilerine olan etkisini incelemiştir. Çalışmanın sonucunda, blok tabanlı programlama ortamının öğrencilerin problem çözme becerilerini geliştirmesine rağmen anlamlı olarak katkı yapmadığı belirlenmiştir. Kukul ve Gökçearsan'ın (2014) yaptığı çalışmanın sonuçları da bu çalışmayla benzerlik göstermektedir.

Öğrenme Ortamları ve Motivasyon

Motivasyon kavramı araştırıldığında birbirinden farklı pek çok tanımın olduğu görülmektedir. Kökeni latince bir sözcük olan “motus” kelimesinin karşılığı olan “hareket” kelimesine dayanmaktadır. Motivasyon kelimesinin Türk Dil Kurumu’na (2018) ait güncel Türkçe sözlükteki karşılığı “isteklendirme, güdüleme” olarak açıklanmaktadır. Motivasyon ilk olarak Woodworth (1918) tarafından “bir organizmayı çeşitli şekillerde harekete geçiren enerji birikimi” olarak tanımlanmıştır. Keller (1983) ise davranışın yönü ve büyüklüğü olarak tanımlamış, gösterilen çabanın motivasyon belirtisi olduğunu ifade etmiştir. Sternberg ve Williams’a (2002) göre motivasyon, bireylerin hedeflerine ulaşırken yardım aldığı içsel bir enerji veya zihinsel bir güç iken Koçel (2005) motivasyonu bireylerin kendi arzu ve istekleri doğrultusunda belirli bir amacı gerçekleştirme çabaları olarak tanımlamıştır.

Motivasyon kuramları içsel ve dışsal olarak ayrılabilir. İçsel motivasyon bireyin bir işi bir doyum elde etmek için kendi isteğiyle başarmak istemesi iken dışsal motivasyon dışsal etkilerin özendirici bir takım ödüller kullanılarak bireyi harekete geçirmesidir (Ercan, 2004; Ryan & Deci, 2000). İçsel motivasyonda birey kendi isteği ile harekete geçerken, dışsal motivasyonda eylem kendisi ile ilişkili olmayıp bireyi ödül, rica ve ceza gibi dışarıdan gelen uyaranların etkisiyle harekete geçirmektedir (Akbaba, 2006).

Motivasyonun öğrencilerin akademik başarıları üzerindeki etkisini inceleyen çeşitli çalışmalar yapılmış, bu çalışmalar sonucunda motivasyon düzeylerinin öğrenci başarısı üzerinde önemli etkisinin olduğu ortaya çıkmıştır (Bruinsma, 2003; Eymur & Geban, 2011; McKenzie & Schweitzer, 2001; Sankaran & Bui, 2001). Öğretim sürecinde öğrenci motivasyonunun akademik başarı ile ilişkisini etkileyen faktörlerin nelerden kaynaklandığını ortaya çıkarmak amacıyla farklı modeller geliştirilmiştir. Bu modellerden birisi Keller tarafından geliştirilmiş olan ARCS Motivasyon Modelidir (Gürol & Demirli, 2006). ARCS Motivasyon Modelinin amacı öğretimi ilgi çekici hale getirmektir (Keller, 1987a). ARCS Modelinde her birinin üç alt ögesi olan dört ana kategori bulunmaktadır. Bu kategoriler dikkat (attention), uygunluk (relevance), güven (confidence) ve tatmin (satisfaction)’dir (Keller, 1987a; Keller, 1987b). ARCS adını bu ana kategorilerin İngilizce baş harflerinden almıştır. Jenkins’e (2001) göre öğrenciler motive olamazlarsa programlama öğrenmede

sıkıntı yaşarlar. Stamouli vd.'e (2004) göre, eleştirel düşünme ve matematiksel yetenek gibi çoklu yeteneklerin yanında ders motivasyonu programlama başarısının artmasında etkilidir.

Alanyazında yapılan çalışmalarda motivasyon ile akademik başarı arasında pozitif yönlü anlamlı ilişkiler olduğu bulunmuş, motivasyonu yüksek olan öğrencilerin daha başarılı olduğu tespit edilmiştir (Credé & Phillips, 2011; Yazıcı & Altun, 2013; Yılmaz & Çavaş, 2007). Wang, Huang ve Hwang (2014), görsel programlama ortamlarının öğrencilerin motivasyonlarını olumlu olarak etkilediğini tespit etmişlerdir. Diğer araştırmalarda da benzer sonuçların bulunduğu görülmektedir (Erol, 2015; Genç & Karakuş, 2011; Jiau, Chen, & Ssu, 2009; Ozoran, Çağıltay, & Topallı, 2012; Ruf, Muhling, & Hubwieser, 2014).

Blok Tabanlı Görsel Programlama

Programlama öğreniminde karşılaşılan zorlukları ortadan kaldırmak amacıyla günümüzde görsel programlama ortamlarının kullanımı oldukça yaygınlaşmıştır (Malan & Leitner, 2007). Bu ortamlar sözdizimi bilgisi gerektirmeyen sürükle bırak yöntemine dayalı arayüzleri sayesinde herhangi bir kod yazımına ihtiyaç duymadan öğrencilerin kolaylıkla projelerini oluşturabilmelerine imkân sağlamaktadır. Bu ortamlarda kullanılan kodların bloklar halinde olması, soyut kavramlar olan “değişken, döngü, dizi” gibi temel programlama yapılarını öğrenenler açısından daha anlaşılır hale getirmektedir.

Baz (2018), kodlama eğitiminde dünya çapında kullanılan öğretici yazılımların özelliklerini ve birbirleri ile gösterdikleri benzerlik ya da farklılıkları tespit etmek amacıyla yaptığı çalışmada elde ettiği bulguları özetlemiştir. Tablo 2’de 7-11 yaş seviyesine uygun olan yazılımların karşılaştırılması ele alınmıştır.

Tablo 2

Kodlama Eğitiminde Kullanılan Öğretici Yazılımlar

Kodlama Ortamı	Web Adresi	Desteklediği İşletim Sistemi	Ücret Durumu	Dil Desteği	Mobil Uyum	Örnek Projeler	Sosyal Medya/Blog	Yardım Desteği
Scratch	http://www.scratchjr.org/	Mac, Linux, Windows	Ücretsiz	✓	✓	✓	✓	✓
code.org	https://code.org/	Mac, Linux, Windows	Ücretsiz	✓	✓	✓	✓	✓
Bitsbox	https://bitsbox.com/	Mac, Linux, Windows, iOS, Android	Ücretli	✓	✓	✓	✓	✓
Code Monkey	https://www.playcodemonkey.com/	Mac, Linux, Windows	Ücretli	✓	✓	✓	✓	✓
Code Combat	https://codecombat.com/	Mac, Linux, Windows	Ücretsiz	✓	✓	x	✓	✓
Lightbot	https://lightbot.com/	Mac, Linux, Windows, iOS, Android	Ücretli	✓	✓	✓	✓	✓
Grok Learning	https://groklearning.com/	Mac, Linux, Windows	Ücretsiz	x	✓	✓	✓	✓
Kidsruby	http://kidsruby.com/	Mac, Linux, Windows	Ücretsiz	x	✓	✓	✓	✓
Bomberbot	http://landing.bomberbot.com/	Mac, Linux, Windows	Ücretli	✓	✓	✓	✓	✓
Touch Develop	https://www.touchdevelop.com/	Mac, Linux, Windows	Ücretsiz	x	✓	x	✓	✓
Tech Rocket	https://www.techrocket.com	Mac, Linux, Windows	Ücretli	..	✓	✓	✓	✓
RoboMind	https://www.robomindacademy.com	Mac, Linux, Windows, iOS, Android	Ücretli	✓	✓	✓	✓	✓
Mad Learn	http://crescerance.com	Mac, Linux, Windows, iOS, Android	Ücretsiz	x	✓	✓	✓	✓
Green Foot	https://www.greenfoot.org	Mac, Linux, Windows	Ücretsiz	x	✓	✓	✓	✓
Thimble by Mozilla	https://thimble.mozilla.org	Mac, Linux, Windows	Ücretsiz	✓	✓	✓	✓	✓
App Inventor	http://appinventor.mit.edu	Mac, Linux, Windows	Ücretsiz	x	✓	✓	✓	✓
AllcanCode	https://www.allcancode.com	Mac, Linux, Windows	Ücretli	✓	✓	✓	✓	✓

Blok Tabanlı Programlama Öğretiminin Katkıları ile İlgili Alanyazın

Bu bölümde blok tabanlı görsel programlama (Scratch) öğretiminin katkılarıyla ilgili ilköğretim, ortaöğretim ve üniversite düzeylerinde yapılan bilimsel çalışmalar incelenmiştir.

Gülmez (2009), programlama öğretiminde görselleştirme araçlarını kullanmanın öğrenci başarısına ve motivasyonuna olan etkisini araştırmış ayrıca ilköğretim seviyesindeki öğrencilerin programlama başarılarının yordanmasına katkı sağlaması amacıyla algoritma geliştirme başarıları ile ilişkisi olan dersleri belirlemeye çalışmıştır. Deneme ve ilişkisel tarama modelinin kullanıldığı çalışma iki deney grubu ile yürütülmüş, deney gruplarından biri akış şeması modelli yazılımı kullanırken, diğer deney grubu ise algoritmayı hikayeleştiren yazılımı kullanarak eğitim almıştır. Çalışmadan elde edilen bulgulara göre çalışma grupları arasında motivasyon açısından anlamlı bir fark bulunmadığı belirlenmiş, öğrencilerin algoritma geliştirme başarıları ile Türkçe, Matematik, İngilizce ve Bilişim Teknolojileri derslerindeki akademik başarıları arasında anlamlı bir ilişki tespit edilmiştir.

Nam, Kim ve Lee (2010), Scratch kullanarak gerçekleştirdiği yapı iskele destekli eğitim ile programlama öğrenmenin problem çözme becerileri üzerindeki etkilerini belirlemek amacıyla ilköğretim 6. sınıfta eğitim gören 60 öğrenci ile deneysel bir çalışma yapmıştır. Çalışma sonucunda elde edilen bulgulara göre yapı iskele destekli eğitim alan öğrencilerin problem çözme becerilerinin anlamlı bir şekilde arttığı belirlenmiştir.

Lai ve Lai (2012), fen öğreniminde Scratch kullanarak programlama yapmanın etkililiğini belirlemek amacıyla 5. sınıf öğrencileri ile tek gruplu yarı deneysel bir çalışma yapmıştır. Çalışmadan elde edilen bulgulara göre öğrenciler mantıksal düşünme ve problem çözmede daha iyi performans göstermiştir. Ayrıca araştırmada kullanılan anketten gelen geri bildirimlere göre öğrencilerin %71'inden fazlası bilgisayar öğreniminde Scratch kullanmayı tercih ettiğini, %58'inden fazlası fen öğrenimlerinde Scratch ile tasarım yapmayı tercih ettiğini ve % 54'ünden fazlası ise gelecekte diğer bilim alanlarında benzer bilgisayar programlama projeleri yapmak istediklerini belirtmişlerdir.

Wilson, Hainey ve Connolly (2012) oyun tabanlı bir yapı kullanarak programlama yeteneğini değerlendirilmek amacıyla İskoçya'da bir ilköğretim

okulunda eğitimine devam eden 60 öğrenci ile bir çalışma yapmıştır. Çalışmadan elde edilen bulgulara göre öğrencilerin programlama beceri seviyelerinde ilerleme kaydedilmiştir. Shin ve Park (2014), Scratch ile öğrencilerin problem çözme becerilerini ölçmek amacıyla 6. sınıfta eğitim gören 46 öğrenci ile bir çalışma yapmıştır. Çalışmadan elde edilen bulgulara göre Scratch yazılımının öğrencilerin problem çözme becerilerini arttırmada katkı sağladığı ortaya çıkmıştır.

Ouahbi, Kaddari, Darhmaoui, Elachqar ve Lahmine (2015), öğrencilerin programlama temellerini öğrenmeleri amacıyla Scratch oyun ortamını kullanarak lisede öğrenim gören 69 öğrenciyle deneysel bir çalışma yapmışlardır. Bir grup öğrenci Scratch ortamını kullanarak basit oyunlar oluştururken diğer grup Pascal programlama diline dayanan geleneksel öğrenme yöntemi ile eğitim almıştır. Öğrencilerin programlama seviyelerini, oyun alışkanlıklarını, gelecek yıllara yönelik programlama motivasyonlarını ve ilgilerini belirlemek amacıyla çalışma öncesinde bir anket uygulanmıştır. Çalışma sonunda elde edilen bulgulara göre Scratch programlama öğrenme ortamını kullanan öğrencilerin daha iyi motive oldukları ve programlamaya devam etme isteklerinin daha fazla olduğu ortaya çıkmıştır.

Erol (2015), Scratch ile programlama öğretiminin öğrencilerin motivasyon ve programlama başarısına etkisini incelemek amacıyla üniversite ikinci sınıfta öğrenim gören 52 öğrenci ile deneysel bir çalışma yapmıştır. Çalışma sonucu elde edilen bulgularda katılımcıların programlama başarı puanları ön testte her iki grupta da benzer iken son testlerde deney grubu lehine anlamlı bir farklılık ortaya çıkmıştır. Araştırmanın nitel verilerinden elde edilen sonuçlara göre deney grubunda yer alan katılımcılar Scratch ile oyun tasarımı etkinliklerini eğlenceli ve kolay bulmuş, Scratch programının motivasyonu arttırmada etkili olduğunu belirtmişlerdir. Kontrol grubunda yer alan katılımcılar ise akış diyagramları ile problem çözme sürecinin zor ve sıkıcı olduğunu, bu durumun motivasyonlarını düşürdüğünü belirtmişlerdir.

Koorsse, Cilliers ve Calitz (2015), Bilişim teknolojileri programlama zorluklarını etkileyen faktörleri bir bilişim teknolojileri uzmanı bakış açısıyla ortaya çıkarmak için yardımcı programlama araçlarını (PAT) değerlendirmiştir. Bu araştırmanın bulgularına göre, bilim insanları programlama yapılarını kavramada PAT kullanımının yararlı olduğunu belirtmiş, fakat PAT kullanan bilişim teknolojileri uzmanlarının programlama yapılarını kavramasına ve programlamaya yönelik motivasyonlarının artmasına yönelik kesin bir kanıt bulunamamıştır.

Erol ve Kurt (2017) Scratch ile programlama eğitiminin öğrenci motivasyonu ve programlama başarısı üzerindeki etkisini incelemek amacıyla üniversitede öğrenim gören 52 öğrenci ile deneysel bir çalışma yapmışlardır. Çalışmadan elde edilen bulgulara göre hem deney grubu hem de kontrol grubunun başarısı artmış, bu artışın deney grubu lehine anlamlı olduğu belirlenmiştir. Ayrıca deney grubunun motivasyon puanları artarken kontrol grubunun puanları azalmıştır.

Vatansver (2018), Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmak amacıyla 226 öğrenci ile karma yöntem kullanarak bir çalışma yapmıştır. Araştırmadan elde edilen sonuçlara göre öğrencilerin problem çözme becerilerinin cinsiyete ve sınıf seviyesine göre farklılık göstermediği belirlenmiştir.

Erdem (2018) yaptığı deneysel çalışmada ortaokul 5. sınıfa devam eden 75 öğrenciyle, yüz yüze eğitim ve ters yüz sınıf modeli olan teknoloji destekli öğrenme ortamlarında Scratch programlamayı öğrenmelerinin ve programlama öğretiminin öğrencilerin bilgi işlemsel düşünme becerilerine etkisini araştırmıştır. Çalışmadan elde edilen bulgulara göre yüz yüze eğitim ve ters yüz sınıf modeli ile eğitim alan öğrencilerin Scratch programlama öğrenmelerinde ve bilgi işlemsel düşünme becerilerinde istatistiksel olarak anlamlı bir fark olmadığı görülmüştür. Nitel verilerden elde edilen sonuçlara göre ters yüz sınıftaki öğrenciler kendi kendine öğrenmenin daha iyi olduğunu belirtirken, yüz yüze eğitim sınıftaki öğrenciler öğretmen anlatımının daha iyi olduğunu belirtmişlerdir. Ayrıca ters yüz sınıftaki öğrenciler anlık yardım alamadıklarını belirtirken yüz yüze sınıftaki öğrenciler sınıf içi etkenlerden, dersin hızından ve öğretmen anlatımından dolayı konuyu kaçırdıklarını belirtmişlerdir. Ayrıca her iki grup öğrencileri keşfederek öğrenmenin daha iyi olabileceğini, oyun ve animasyon yapabilecek becerilerinin geliştiğini belirtmişlerdir.

Kereki (2008) öğrencilerin programlama deneyimlerini ve motivasyonlarını geliştirmek amacıyla iki farklı eğitim kurumunda bulunan öğrenciler ile Scratch - Visual Basic ve Scratch - Java ortamları kullanarak deneysel bir çalışma yapmıştır. Çalışmanın sonuçlarına göre ön test ve son test puanları arasında gruplar arasında istatistiksel olarak anlamlı bir fark bulunmaz iken Scratch ile eğitim alan deney grubu öğrencilerinin motivasyon puanlarının daha yüksek olduğu ortaya çıkmıştır.

Alanyazında yapılan çalışmalardan elde edilen sonuçlara göre Scratch yazılımının programlama öğretiminde kolaylıklar sağladığı, problem çözme, ilgi ve motivasyon üzerinde olumlu etki oluşturduğu, sadece programlama öğretiminde değil diğer derslerin öğretiminde de kullanılabileceği belirlenmiştir.

Eğitimde Robotik Kullanımı

Robot kelimesi köken olarak Çekçe'de "mecburi hizmet, angarya, köle emeği" anlamına gelen "robota" kelimesinden türetilmiştir. Alinyazında robot kavramıyla ilgili çeşitli tanımlar ortaya atılmıştır. Bunlar arasında en kapsamlı olan birkaçı şöyledir: Amerikan Robot Enstitüsü robotu "Robot, yeniden programlanabilen; malzemeleri, parçaları, özel cihazları çeşitli programlanmış hareketlerle yapılacak işe göre taşıyan veya işleyen çok fonksiyonlu bir manipülatördür." şeklinde tanımlamıştır.

Robotların çalışma ve kullanımını ifade eden robotik kavramı ise ilk olarak Isaac Asimov tarafından kullanılmıştır. Robotik; elektronik, mekanik veya sibernetik yapılardan oluşan, kendi kendine çalışabilen ya da uzaktan kumanda ile kontrol edilebilen, sensörleri, kontrol kartı, eyleyicileri ve fiziksel yapıları ile nesnelere tutmak, kavramak, hareket ettirmek, taşımak, üretim yapmak gibi görevleri yerine getirebilen yapay sistemlerdir (Çamoğlu, 2015). Şabanoviç ve Yannier'e (2003) göre robotiğin eğitim alanında kullanılması teknolojik bilgiye sahip olma, araştırma ve bir araya getirme becerileri ile birlikte çalışma becerilerini geliştirmektedir.

Robotun Temel Bileşenleri.

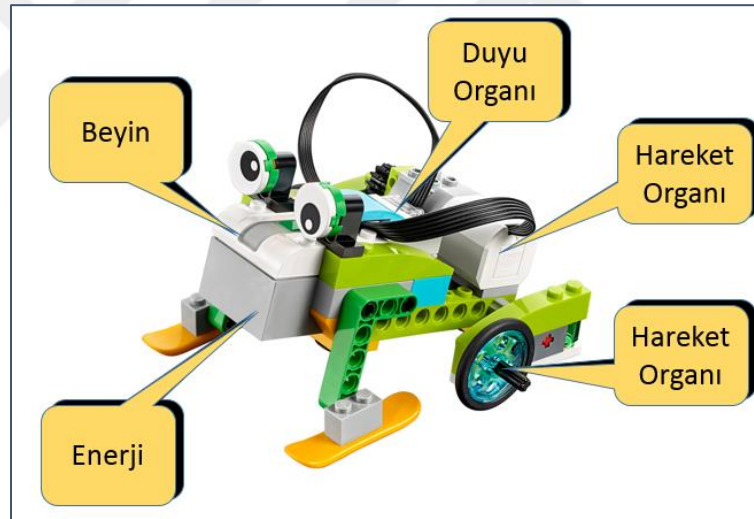
Robot; kontrol kartı, motor ve algılama sistemleri (sensörler) birleşiminden oluşur.

Kontrol Kartı. Bilgisayar programının yürütüldüğü "beyin" olarak da bilinir. Genellikle, robotun hareketli parçalarının izleyeceği komutları verir.

Motor ve Mekanik Parçalar. Motor, güç uygulandığında dönmeye neden olan bir çıkış cihazıdır (Gaudiello & Zibetti, 2016). Mekanik parçalar ise robotun hareket etmesini, dönmesini, tutmasını ve kaldırılmasını sağlayan pistonlar, kışkaçlar, tekerlekler ve dişlilerden oluşan parçalardır.

Sensörler. Bir robota girdi sağlayan ve çevresini hissetmesini sağlayan parçalardır (Gaudiello & Zibetti, 2016). Çevredeki fiziksel büyüklükleri tespit eden

sensörler programlanabilir robota önemli işlevler katan robot ile bilgisayar sistemi arasındaki farkı ortaya koyan en temel parçalardan biridir (Komis & Misirli, 2016). Sensörler robotların buldukları ortamla bağlantısını sağlarlar (Özfiyat, 2009). Böylece bir robot çevresindeki nesnelere boyutlarını, nesnelere arasındaki mesafeyi ve buldukları ortamın diğer özelliklerini sensörleri yardımıyla öğrenir. Robot uygulamalarında en sık olarak ses sensörü, dokunmatik sensör, eğim sensörü, ışık sensörü, mesafe sensörü, ısı sensörü ve ultrasonik sensör kullanılmaktadır (Robotiksistem, 2019). Thymio gibi bazı robotik sistemlerde, sensörler kullanıcılardan veya çevreden bilgiler alarak robotla iletişimi sağlarken LEGO® WeDo™ gibi diğer robotik sistemlerde, sensörler robotun eylem yeteneğini genişletmektedir (Komis & Misirli, 2016). Robotların temel bileşenleri olan kontrol kartını beyin olarak, sensörlerini duyu organı olarak, motoru ve mekanik parçalarını ise hareket organı olarak düşündüğümüzde robotlar insanlarla benzerlik gösterir (Şekil 1).



Şekil 1. Robotun temel bileşenlerinin bir arada olduğu WeDo2.0 örneği.

Robotik Setler. Robotik; mekanik, elektronik ve programlama becerileri gerektiren oldukça zor ve maliyetli bir süreç olması sebebiyle hazır robotik setler geliştirilmiştir. Hilal, Wagdy ve Khamis (2007) yaptığı çalışmada robotik setleri beş kategoriye ayırmıştır: Bu kategoriler;

1. Gövde İnşa Kitleri: LEGO Teknik, FischerTechnik ve Meccano gibi bir robotun mekanik bir iskeletinin oluşturulmasını sağlayan kitlerdir.

2. Elektronik Kitler: Beyin görevindeki mikroişlemci, elektromekanik bileşenler ve sensörlerden oluşan kitlerdir.

3. Programlama Kitleri: Microsoft Robotic Studio veya Gazebo gibi robotların programlamasını ve taklit edilmesini sağlayan simülasyon ortamlarıdır.

4. Programlanabilir Robotlar: Robotun gövde veya elektronik tasarımında esneklik sağlamamasına rağmen kullanıcı tarafından yeniden programlanabilen robot kitlerdir.

5. Komple Başlangıç Kitleri: Gövde tasarımı, elektronik bileşenleri, mekanik ve yazılım işlevselliği açısından esneklik sağlayan tüm robotik kitleri içerir. Çok yönlü başlangıç kitleri ve çok yönlü olmayan başlangıç kitleri olmak üzere ikiye ayrılırlar. Çok yönlü başlangıç kitleri, farklı görünümlere sahip robotlar oluşturmak için kullanılacak genel parçalardan oluşur. Çok yönlü olmayan başlangıç kitleri ise endüstriyel robotlar, mobil robotlar, biyolojiden ilham alınarak oluşturulan robotlar, sosyal robotlar ve yarışma odaklı robotlardır.

Costelha ve Neves (2018) özellikle STEM eğitiminde robot kullanımının yaygınlaşmasını sağlamak amacıyla eğitimde kullanılan eğitsel robot setlerinin özelliklerini incelemiş, araştırma sonucu elde edilen bulguları tablo halinde özetlemiştir. Bu tablodan elde edilen bulgulara göre programlanabilir eğitsel robotların yaş grubu, metin tabanlı programlama dili, görsel programlama dili ve bağlantı türü gibi özellikleri Tablo 3'te özetlenmiştir.

Tablo 3

Programlanabilir Eğitsel Robotlar

Eğitsel Robot	Yaş Grubu	Metin Tabanlı Programlama Dili	Görsel Programlama Dili	Kablosuz Bağlantı türü
Anprino (Arthur)	N.A	Arduino tabanlı	Arduino	Bluetooth
Bee-Bot	3+	Hayır	Hayır	Yok
Blue-Bot	5-7	Hayır	Üretici Özel	Bluetooth
Bot'n Roll ONE A	N.A	Arduino tabanlı	Open Roberta	Yok
BQ Printbot Evolution	8+	Arduino tabanlı	Üretici Özel, Arduino	Bluetooth
BrickPi	10+	Python, Java, NodeJS	Scratch	Wifi, Bluetooth
Cubelets	4+	C	Blockly	Bluetooth

Cubetto	3+	Arduino tabanlı	Fiziksel, Arduino	Bluetooth
Cue	11+	Javascript	Üretici Özel, Blockly	Bluetooth
Dash	6+	Hayır	Üretici Özel, Blockly	Bluetooth
Edison Robot	4-16	Python	Üretici Özel	Yok
Finch	6+	Python, Java,C	Üretici Özel, Scratch, Snap	Yok
GoPiGo3	9+	Python, NodeJS	Üretici Özel, Scratch	Wifi, Bluetooth
KIBO 18 Kit	4+	Hayır	Fiziksel programlama	Yok
Lego Mindstorms Ev3	10+	EV3Basic, C, Java	Üretici Özel, Scratch, LabView	Wifi, Bluetooth
LEGO WeDo 2.0	7+	Java	Üretici Özel, Scratch	Bluetooth
Makeblock mBot	8+	Python	Scratch	IR
Marty	8+	Python, NodeJS, C++, C#	Scratch	Wifi
Ozobot Bit 2.0	6-10	Hayır	Üretici Özel, Fiziksel	Yok
Ozobot Evo	8-14	Hayır	Üretici Özel, Fiziksel	Bluetooth
Phiro Pro	9+	Hayır	Üretici Özel, Scratch, Snap, Pocket Code	Bluetooth
Pro-Bot	3+	LabView	Üretici Özel	Yok
RoboticsTxt Disc. Set	10+	Python	Üretici Özel	Wifi, Bluetooth
Sphero SPRK+	8+	Javascript	Üretici Özel	Bluetooth
Stem Robotics Erp	8+	Hayır	Üretici Özel	Bluetooth
Thymio II Wireless	6+	Evet	Üretici Özel, Blockly, Scratch	Üreticiye özgü
VEX IQ Super Kit	8+	easyC, Python	Üretici Özel, Scratch, Blockly	Üreticiye özgü
Zowi	8+	Arduino tabanlı	Üretici Özel, Arduino	Bluetooth

Programlama ve robotik. Son yıllarda robotların eğitimde kullanılması sonucu eğitim teknolojisi alanında yeni ortamlar ve araçlar ortaya çıkmıştır. Bu yeni ilgi alanı eğitsel robotik olarak adlandırılmıştır. Robotik eğitiminin geliştirdiği beceriler eleştirel düşünme, birlikte çalışma, yaratıcılık, problem çözme ve bilgi işlemsel düşünme becerileri şeklindedir (Romero & Dupont, 2016).

Robotik teknolojileri özellikle erken çocukluk dönemi eğitiminde kullanıldığında dişliler, kollar, eklemler, motorlar ve sensörler ile programlama

döngüler ve değişkenler gibi soyut ve mekanik kavramları somut ve eğlenceli hale getirir. Bu nedenle öğrenciler tarafından programlanan robotlar verilen komutların etkisini anında gösterdiğinden dolayı soyut kavramları somut hale getirmede kullanılabilir araçlardır (Bers, 2007; Üçgül, 2017). Ayrıca eğitsel robotlar eğlenceli etkinlikler sunarak öğrenme ortamında öğrencilerin ilgisini ve merakını uyandırarak uygulamalı eğitimde kullanılabilir benzersiz bir öğrenme aracıdır (Eguchi, 2010).

Blok Tabanlı Robotik Programlama Öğretiminin Katkıları ile İlgili Alanyazın

Alanyazında eğitimde robot kullanımının öğrenmeye katkılarıyla ilgili ilköğretim, ortaöğretim ve üniversite düzeylerinde yapılan birçok araştırma bulunmaktadır. Şenol ve Büyük (2015) fen ve teknoloji dersinde robotik destekli yapılan deneylerin öğrencilerin bilimsel süreç becerileri ile derse yönelik motivasyonlarına etkilerini ortaya koymak üzere 7. sınıfa devam eden 40 öğrenci ile karma yöntem kullanarak bir çalışma yürütmüştür. Deney grubunda etkinlikler robot teknolojisi kullanılarak gerçekleştirilirken kontrol grubunda aynı etkinlikler müfredattaki haliyle uygulanmıştır. Çalışmadan elde edilen bulgulara göre robotik destekli fen deneylerinin gerçekleştirildiği deney grubu öğrencilerinin bilimsel süreç becerileri ile fen ve teknoloji dersine yönelik motivasyonları kontrol grubu öğrencilerine göre anlamlı düzeyde farklılık göstermiştir.

Kurebayashi, Kamada ve Kanemune (2009), Japonya'nın Shizuoka eyaletindeki bir ortaokulun 2. sınıfında programlama eğitimi alan 135 öğrenci ile otonom robotlar kullanarak bir çalışma yapmıştır. Çalışmadan elde edilen bulgulara göre programlama öğrenme öğrenciler için "zor ve eğlenceli" bir süreç olmasına rağmen robotla programlama öğrenmenin programlama sürecinde zorluk çeken öğrenciler için yararlı olduğu sonucu ortaya çıkmıştır.

McGill (2012) programlama eğitiminde robot kullanmanın öğrencileri programlama öğrenmeye motive edip etmediğini tespit etmek amacıyla bir çalışma yapmıştır. Çalışmadan elde edilen bulgulara göre robotlar, öğrencilerin programlamayı öğrenmeye yönelik tutumlarını olumlu yönde etkilemesine rağmen kullanılan ölçeğin alt boyutu olan ilgi, güven ve memnuniyet üzerinde anlamlı farklılık ortaya çıkmadığı belirtilmiştir. Sonuçlar ayrıca öğrencilerin genel motivasyon düzeylerinin etkilenmediğini göstermiştir.

Liu, Newsom, Schunn ve Shoop (2013) programlama eğitiminde fiziksel robotlarla etkileşim kuran öğrenciler ile sanal robotlarla etkileşime giren öğrenciler arasında öğrenme farklılıklarını araştırmak için lisede öğrenim gören 30 öğrenci ile deneysel yöntem kullanarak bir çalışma gerçekleştirmiştir. Bu çalışma sonucuna göre, hem fiziksel robot kullanan öğrenciler hem de sanal robot kullanan öğrenciler programlama öğrenimi açısından aynı başarıyı göstermesine rağmen öğretmen ve öğrencilerin programlamaya daha iyi odaklanabilmelerinden dolayı sanal sınıfta verilen eğitimin daha kısa sürede tamamlandığı ve verimli bir öğrenme gerçekleştiği belirtilmiştir.

Liang, Fleming, Man ve Tillo (2013) LEGO MindStorms'un öğrenme sürecini destekleyip desteklemediğine dair öğrenci görüş ve izlenimlerini araştırmak amacıyla Çin'de bulunan bir üniversitede öğrenim gören büyük çoğunluğu daha önce programlama eğitimi almayan 480 öğrenci ile karma yöntem kullanarak 7 hafta süren bir çalışma gerçekleştirmiştir. Bu çalışma sonucuna göre, öğrenciler robotlar ile programlama öğrenmenin ilgi çekici olduğunu ve grup çalışmasının eğlenceli olduğunu belirtmişlerdir.

Lykke, Coto, Mora, Vandel ve Jantzen (2014) öğrencilerin programlama beceri düzeylerini ve motivasyonunu belirlemek amacıyla LEGO Mindstorms robot kiti kullanarak probleme dayalı öğrenme (PBL) ve probleme dayalı öğrenme + LEGO Mindstorms (PBL + LM) kombinasyonu ile eğitim alan iki farklı deney grubu ve geleneksel eğitimin yapıldığı kontrol grubu ile çalışma yapmıştır. Çalışma sonuçlarına göre öğretim tasarımlarının öğrencilerin fizyolojik ve duygusal durumlarını etkilediği belirtilmiştir. Ayrıca öğrenciler PBL tasarımının işbirliğini, etkileşimi ve duygusal mutluluğu teşvik ettiğini belirtirken, PBL + LM tasarımının robot kullanımı nedeniyle ilgi çekici ve motive edici olmasına rağmen robottan kaynaklanan kısıtlamalar nedeniyle sinir bozucu olduğunu, geleneksel öğrenmenin ise kuramsal kavrama açısından güven verici olmasına rağmen etkileşim açısından stres ve gerginlik yarattığını belirtmişlerdir.

Álvarez ve Larrañaga (2015) Bask Country Üniversitesi'nde yaklaşık 100 öğrenciyle yaptıkları çalışmada Lego robotlarının programlama eğitimindeki etkisini araştırmak için bir çalışma yapmıştır. Bu çalışma sonucuna göre öğrencilerin motivasyon ve öğrenme sürecine ilişkin algılarında anlamlı bir artış belirlenmiştir. Robot kullanımı nedeniyle öğrencilerin ilgisinin ve motivasyonunun arttığı, ayrıca

robotların öğrencilerin dersi daha iyi kavramalarına yardımcı oldukları belirtilmiştir.

Korkmaz (2016) tarafından yapılan bir çalışmada Lego MindStorms EV3 robotları bilgisayar mühendisliği bölümünde C++ programlama dersinde öntest sontest kontrol gruplu deneysel bir çalışmada kullanılmıştır. Bu çalışmanın sonuçlarına göre Lego MindStorms EV3 robotları kullanılan deney grubundaki öğrencilerin akademik başarıları kontrol grubu öğrencilerine göre anlamlı bir şekilde daha iyi bulunmuştur. Çalışmada ayrıca Lego MindStorms EV3 robotlarının öğrencilerin bilgisayar programlamaya yönelik tutumlarına da olumlu bir etki yaptığı belirtilmiştir.

Çankaya, Durak ve Yünkül (2017), programlama eğitimi alan öğrencilerin başarılarını ve görüşlerini araştırmak için 6. ve 7. sınıfta öğrenim gören 9 öğrenci ile karma araştırma yöntemi ile bir çalışma gerçekleştirmiştir. Bu çalışma sonucuna göre, öğrencilerin yaratıcı problem çözme becerileri ile performans puanları arasında pozitif yönde, anlamlı ve orta düzeyde bir ilişki olduğu tespit edilmiş, öğrencilerin robotlarla programlama eğitimine yönelik olarak olumlu tutuma sahip oldukları, yapılan eğitimin motivasyonu artırdığı, eğlenceli olduğu ve programlama öğrenmelerine olumlu yönde katkı sağladığı, robotlarla yapılan programlama eğitiminin klasik programlama eğitimine tercih edilebileceği belirtilmiştir.

Numanoğlu ve Keser (2017) mBot eğitsel robot kiti platformunun programlama öğretiminde kullanılabilirliğini belirlemeyi, mBlock blok tabanlı robot programlama ortamının özelliklerini, kullanımını, programlama yapısını ve programlama öğretimine yönelik uygulama geliştirmeyi belirlemek amacıyla bir çalışma yapmıştır. Çalışma sonucuna göre elde edilen bulgular mBlock programlama ortamı ve mBot robot kiti kullanılarak yapılan programlama eğitimi ile programlamanın temel kavramlarını içeren uygulamaların kolayca oluşturulup kullanılabileceğini göstermiştir. Ayrıca öğrencilerin soyut kavramları kolayca somutlaştırabileceği ve yazdığı programın etkisini anında görerek problem çözme ve bilgi-işlemsel düşünme becerilerini daha kolay ve hızlı bir şekilde geliştirebileceği belirtilmiştir.

Şanal ve Erdem (2017) kodlama ve robotik üzerine çalışma yapma durumlarının öğrencilerin problem çözme süreçleri farklılıklarına etkisini belirlemek amacıyla bir çalışma yapmıştır. Araştırmada öğrencilere çözmeleri için teknik ve sosyal olmak üzere iki problem verilmiş, öğrencilerden problemleri çözmeleri

istenmiştir. Kodlama yapan öğrencilerin problem çözme süreçlerinin kodlama yapmayan öğrencilerin problem çözme süreçlerinden daha verimli olduğu belirtilmiştir.

Durak, Yılmaz ve Yılmaz (2018) robot etkinlikleri ile gerçekleştirilen programlama öğretim sürecinde ortaokul öğrencilerinin görüşlerini belirlemek amacıyla 7. ve 8. sınıfta öğrenim görmekte olan 55 öğrenci ile 10 hafta süren nitel bir çalışma yapmıştır. Çalışma sonucuna göre öğrenciler robotik programlama etkinliklerinin bilgisayar kullanım becerilerini geliştiren zor fakat eğlenceli bir süreç olduğunu ve Scratch kullanımının programlama kavramlarının öğrenilmesini desteklediğini belirtmişlerdir.

Şimşek (2018) görsel programlama ve robotik programlama etkinliklerinin öğrencilerin bilgi işlemsel düşünme becerilerine ve akademik başarılarına etkisini bulmak amacıyla ortaokulda eğitim gören 60 öğrenci ile deneysel bir çalışma yapmıştır. Araştırmadan elde bulgular incelendiğinde hem akademik başarıda hem de bilgi işlemsel düşünme pratiklerinde iki grubun puanları arasında farklılık bulunmamıştır.

Alanyazında yapılan çalışmalardan elde edilen sonuçlara göre robotik etkinliklerinin programlama öğretiminde kolaylıklar sağladığı, problem çözme, ilgi ve motivasyon üzerinde olumlu etki oluşturduğu, robotların sadece programlama öğretiminde değil diğer derslerin öğretiminde de kullanılabileceği belirlenmiştir.

Bölüm 3

Yöntem

Çalışmada araştırma yöntemi olarak yarı-deneysel desenlerden kontrol gruplu ön test son test deney modeli kullanılmıştır (Karasar, 2012). Ön test-son test kontrol gruplu yarı deneysel desen ile araştırmanın öncesinde ve sonrasında elde edilen veriler karşılaştırılmış ve ilgili değişkenler arasındaki ilişki test edilmiştir.

Yapılan araştırmada deney ve kontrol gruplarına deneysel işlemler başlamadan önce ve deneysel işlemler bittikten sonra “Programlama Erişi Testi” ve “Çocuklar İçin Problem Çözme Envanteri” ölçeği uygulanmıştır. Deneysel işlemler bittikten sonra ayrıca “Motivasyon Ölçeği (CIS)” uygulanmıştır. Araştırmanın yapısına uygun olarak oluşturulan ön test-son test kontrol gruplu deney modelinin tasarımı Tablo 4’de sunulmuştur.

Tablo 4

Ön Test-Son Test Kontrol Gruplu Deney Modeli Tasarım Tablosu

Gruplar	Ön test	Uygulama	Son test
DG	ET ₁	U ₁	ET ₂
	PÇE ₁		PÇE ₂
			MÖ
KG	ET ₁	U ₂	ET ₂
	PÇE ₁		PÇE ₂
			MÖ

DG: Deney Grubu

ET₁: Erişi Testi (Ön test)

KG: Kontrol Grubu

ET₂: Erişi Testi (Son test)

U₁: Scratch+LegoWedo2.0

PÇE₁: Problem Çözme Envanteri (Ön test)

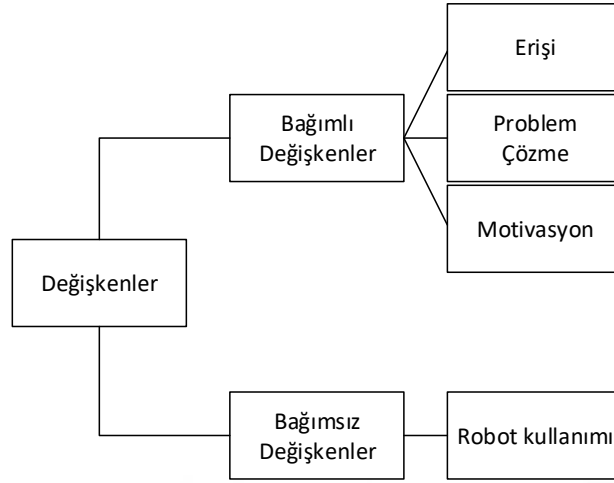
U₂: Scratch

PÇE₂: Problem Çözme Envanteri (Son test)

MÖ: Motivasyon Ölçeği (Son test)

Bu araştırmanın bağımsız değişkeni kodlama eğitiminde robot kullanımı bağımlı değişkenleri ise erişim, problem çözme becerisi ve motivasyondur. Çalışmada bağımlı değişkenlere ait nicel veriler, gerekli izinleri alındıktan sonra uygulanan

ölçekler ve araştırmacı tarafından geliştirilen programlama erişimi testi ile toplanmıştır. Değişkenlerin şekilsel gösterimi Şekil 2’de sunulmuştur.



Şekil 2. Bağımlı ve bağımsız değişkenler.

Çalışma Grupları

Araştırma Bolu ili Merkez ilçesinde bulunan bir devlet okulunun 5. ve 6. sınıflarında eğitimlerine devam eden 87 öğrenci ile yapılmıştır. Çalışma gruplarına okulun 5. ve 6. sınıflarının tüm şubeleri dahil edilmiştir. Bu sınıflar için 4'er şube bulunan okulda her sınıf seviyesinden rastgele olarak seçilen 2 şube (toplam 4 şube) programlama eğitiminde blok tabanlı görsel programlama yazılımı ve robot kiti kullanan deney grubunu oluştururken, diğer 2 şube (toplam 4 şube) sadece blok tabanlı görsel programlama yazılımı kullanan kontrol grubu olmuştur. Çalışma grubuna dahil edilen şubelerdeki öğrencilere ait cinsiyet (Tablo 5), yaş dağılımları (Tablo 6), matematik becerileri (Tablo 7), erişim ön test puanları (Tablo 9) ve problem çözme ön test puanları (Tablo 10) ilgili tablolarda sunulmuştur.

Tablo 5

Deney ve Kontrol Gruplarının Cinsiyete Göre Dağılımı

Gruplar	Kız		Oğlan		Toplam N
	N	%	N	%	
Blok Tabanlı (KG)	24	55,8	19	44,2	43
Blok Tabanlı Robotik (DG)	20	45,5	24	54,5	44
Toplam	44	50,6	43	49,4	87

Tablo 5 incelendiğinde, blok tabanlı ve blok tabanlı robotik öğretimlerinin yapıldığı sınıflardaki öğrencilerin sayısının cinsiyet açısından birbirlerine benzer oldukları görülmektedir.

Tablo 6

Grupların Yaş Ortalamaları

Gruplar	Şube	N	\bar{x}
Blok Tabanlı (KG)	5A, 5C, 6B, 6C	43	10,8
Blok Tabanlı Robotik (DG)	5B, 5D, 6A, 6D	44	10,7
Toplam		87	10,75

Tablo 6'da, blok tabanlı (KG) programlama eğitimi alan öğrencilerin yaş ortalamasının 10,8, blok tabanlı robotik (KG) programlama eğitimi alan öğrencilerin yaş ortalamasının ise 10,7 olduğu görülmektedir. Tablo 6'da görüldüğü üzere blok tabanlı ve blok tabanlı robotik öğretimlerinin yapıldığı gruplar yaş ortalamaları bakımından birbirlerine benzerdir.

Tablo 7

Grupların Matematik Akademik Başarı Ortalamaları

Gruplar	Şube	N	\bar{x}
Blok Tabanlı (KG)	5A, 5C, 6B, 6C	43	70,29
Blok Tabanlı Robotik (DG)	5B, 5D, 6A, 6D	44	74,27
Toplam		87	72,56

Programlama eğitiminde matematiksel yetenek önemlidir. Öğrencilerin matematik akademik başarıları puanları, e-okul sisteminden alınarak hesaplanan 1. döneme ait yazılı sınav ortalamalarından elde edilmiştir. Tablo 7'de, blok tabanlı (KG) programlama eğitimi alan öğrencilerin 1. dönem matematik akademik başarı ortalamasının 70,29, blok tabanlı robotik (KG) programlama eğitimi alan öğrencilerin 1. dönem matematik akademik başarı ortalamasının ise 74,27 olduğu görülmektedir. Deney ve kontrol gruplarında yer alan öğrencilerin 1. dönem matematik akademik başarı ortalamaları bağımsız örneklem t-testi ile karşılaştırılmıştır. Sonuçlar Tablo 8'de sunulmuştur.

Tablo 8

Grupların Matematik Akademik Başarı Ortalamalarının Karşılaştırılması

Gruplar	N	\bar{x}	ss	sd	t	p
Blok Tabanlı (KG)	43	70,29	19,37			
Blok Tabanlı Robotik (DG)	44	74,27	18,54	85	-1,10	,275

Bağımsız örneklem t-testi, kontrol grubu ile deney grubu arasındaki matematiksel beceri farklılığını ortaya koymak için kullanılmıştır. Analiz sonucunda, blok tabanlı ($\bar{x}=70,29$, $ss=19,37$) ve blok tabanlı robotik ($\bar{x}=74,27$, $ss=18,54$) grupları arasında istatistiksel olarak anlamlı bir fark bulunmamıştır ($t(85)=-1,10$, $p>0,05$). Bu sonuçlara göre, programlama eğitimi öncesinde deney ve kontrol gruplarının matematiksel yetenek açısından denk olduğu görülmüştür.

Deney ve kontrol gruplarında yer alan öğrencilerin erişiş ön testinden almış oldukları puanlar bağımsız örneklem t-testi ile karşılaştırılmıştır. Sonuçlar Tablo 9’da sunulmuştur.

Tablo 9

Grupların Erişiş Ön Test Puanlarının Karşılaştırılması

Gruplar	N	\bar{x}	ss	sd	t	p
Erişiş Ön test Puanları						
Blok Tabanlı (KG)	43	77,27	26,92			
Blok Tabanlı Robotik (DG)	44	79,86	20,88	85	-0,50	,616

Bağımsız örneklem t-testi, kontrol grubu ile deney grubunun programlama ön bilgileri arasındaki farklılığı ortaya koymak için kullanılmıştır. Analiz sonucunda, blok tabanlı ($\bar{x}=77,27$, $ss=26,92$) ve blok tabanlı robotik ($\bar{x}=79,86$, $ss=20,88$) grupları arasında istatistiksel olarak anlamlı bir fark bulunmamıştır ($t(85)=-0,50$, $p>0,05$). Bu sonuçlara göre deneysel uygulama öncesinde deney ve kontrol gruplarının programlama ön bilgisi açısından denk olduğu görülmüştür.

Deney ve kontrol gruplarında yer alan öğrencilerin “Çocuklar İçin Problem Çözme Envanteri” ön testinden almış oldukları puanlar bağımsız örneklem t-testi ile karşılaştırılmıştır. Sonuçlar Tablo 10’da sunulmuştur.

Tablo 10

Grupların Problem Çözme Becerileri Ön Test Puanları

Gruplar		N	\bar{x}	ss	sd	t	p
ÇPÇE Ön Test Puanları	Blok Tabanlı (KG)	43	89,12	11,34	85	0,82	,412
	Blok Tabanlı Robotik (DG)	44	87,00	12,58			

Bağımsız örneklem t-testi, kontrol grubu ile deney grubunun problem çözme becerileri arasındaki farklılığı ortaya koymak için kullanılmıştır. Analiz sonucunda, blok tabanlı ($\bar{x}=89,12$, $ss=11,34$) ve blok tabanlı robotik ($\bar{x}=87,00$, $ss=12,58$) grupları arasında istatistiksel olarak anlamlı bir fark bulunmamıştır ($t(85)=0,82$, $p>0,05$). Buna göre deneysel uygulama öncesinde deney ve kontrol gruplarının problem çözme becerisi açısından denk olduğu görülmüştür. Bulunan bu değerler grupların yansız bir biçimde oluşturulduğunu göstermektedir.

Veri Toplama Araçları

Araştırma kapsamında veri toplamak amacıyla ölçme araçları olarak, “Kişisel bilgi formu”, “Programlama erişim testi”, “Çocuklar için problem çözme envanteri” ve “Motivasyon ölçeği” kullanılmıştır. Aşağıda veri toplama araçlarının uygulanmasıyla ilgili bilgiler verilmiştir.

Kişisel bilgi formu. Araştırmacı tarafından hazırlanan kişisel bilgi formu; çalışma grubuna ait bilgilerin alınması için katılımcıların yaşını, cinsiyetini ve bilgisayar sahipliğini belirlemeye yönelik sorulardan oluşmaktadır.

Programlama erişim testi. Öğrencilerin uygulama öncesinde ve sonunda; algoritma geliştirme, blok tabanlı kodlama programı öğrenimi ve programlama yapıları ile ilgili başarılarını değerlendirmek amacı ile araştırmacı tarafından geliştirilmiştir. Algoritma geliştirme, blok tabanlı programlama öğrenimi ve programlama becerilerinin değerlendirilmesine yönelik olarak hazırlanan test için öncelikle İnternet kaynakları taranmıştır. Sorular hazırlanmadan önce MEB Eğitim Bilişim Ağı (EBA) ve Gülbahar Güven (2018) tarafından hazırlanan MEB Bilişim Teknolojileri 5. sınıf öğretmen rehber kitabı ve öğrenci materyal kitabındaki uygulama, örnek ve tanımlar incelenmiş, sorular ders müfredatına uygun olarak hazırlanmıştır. Güvenirliği arttırmak adına soru sayısı olabildiğince çok tutulmuş,

daha sonra ise uzman görüşüne sunulmuştur. Uygun olmayan soruların elenmesi amacıyla programlama dersi veren bir akademisyenin görüşü alınmış; dile, bilişim teknolojileri ve yazılım dersi programına ve öğrenci seviyesine uygunluğuna yönelik olarak alınan dönütler sonrasında uygun olmayan soruların elenmesi işlemi yapılmıştır. Teste konu alanı uzmanlarının görüşleri ve önerileri doğrultusunda son şekli verilmiştir.

Eriş testi iki bölüm halinde oluşturulmuş (Programlama Eriş Testi I, Programlama Eriş Testi II) ve öğrencilerin bu testlerden aldıkları puanlar toplanarak toplam başarı puanları elde edilmiştir. Birinci test tek bölüm olarak hazırlanmış olup algoritma ve strateji geliştirmeye yönelik olarak hazırlanmıştır. Algoritma geliştirme soruları ders müfredatındaki temel algoritma geliştirme konularına paralel olarak hazırlanmış, 13 adet ve her biri 5 puan olmak üzere toplam 65 puanlık kısa cevap, doğru-yanlış, boşluk doldurma, eşleştirme ve klasik biçimde sorulardan oluşmaktadır. İkinci test ise beş bölüm olarak hazırlanmış olup blok tabanlı programlama bilgisini ve programlama yapılarını ölçmeye yönelik olarak hazırlanmıştır. Blok tabanlı programlama öğreniminde kullanılan Scratch programı arayüzünün kullanımını, kullanılan blokların işlevlerinin ne derece öğrenildiğini ve Scratch programındaki programlama yapılarını ölçmek amacıyla hazırlanmıştır.

- A bölümü, her biri 2 puan olmak üzere 10 adet sorudan oluşan toplam 20 puanlık çoktan seçmeli,
- B bölümü, her biri 2 puan olmak üzere 4 adet sorudan oluşan toplam 8 puanlık iki yanıtı birebir eşleştirme,
- C bölümü, her biri 1 puan olmak üzere toplam 8 puanlık birebir eşleştirme,
- D bölümü, her biri 4 puan olmak üzere 5 adet sorudan oluşan toplam 20 puanlık 4 yanıtı bire bir olmayan eşleştirme,
- E bölümü ise her biri 4 puan olmak üzere 14 adet sorudan oluşan toplam 56 puanlık çoktan seçmeli olup testin tamamından alınabilecek en yüksek puan 112'dir.

Çocuklar için problem çözme envanteri (ÇPÇE). Araştırma kapsamında öğrencilerin problem çözme becerilerini ölçmek amacıyla Serin, Bulut Serin ve Saygılı (2010) tarafından geliştirilmiş olan “İlköğretim Düzeyindeki Çocuklar İçin Problem Çözme Envanteri (ÇPÇE)” kullanılmıştır.

Öğrencilerin problem çözme konusunda kendilerini algılama düzeylerini ölçebilmek amacıyla hazırlanan envanter onu geliştiren araştırmacılar tarafından 4., 5., 6., 7. ve 8. sınıf düzeylerinde 568 ortaokul öğrencisine uygulanmıştır. Envanterin hangi alt faktörlerden oluştuğunu belirlemek amacıyla araştırmacılar tarafından yapılan faktör analizi sonucunda ölçme envanterinin “Problem çözme becerisine güven” (12 madde), “Öz denetim” (7 madde) ve “Kaçınma” (5 madde) olmak üzere üç faktör ve toplam 24 maddeden oluştuğu tespit edilmiştir. Ölçüm aracını geliştiren araştırmacılar tarafından her bir faktöre ait Cronbach α iç tutarlık katsayıları sırasıyla 0,85, 0,78 ve 0,66 olarak, toplamda ise 0,80 olarak hesaplanmıştır. Bu haliyle envanterin güvenilir bir ölçüm aracı olduğu düşünülmektedir. Ölçüm aracının test-tekrar test güvenilirlik puanları ise sırasıyla 0,84, 0,79 ve 0,70 olarak, envanterin tamamının güvenilirlik katsayısı ise 0,85 olarak hesaplanmıştır. Envanterin kapsam geçerliği uzman görüşleri alınarak, yapı geçerliği ise açıklayıcı faktör analizi ile belirlenmiştir. Bu üç faktörlü yapının doğruluğunun test edilmesi amacı ile doğrulayıcı faktör analizi uygulanmıştır.

Tablo 11

Ölçeğin Alt Faktörleri ve Cronbach Alfa Katsayıları

Faktör Adı	Madde Sayısı	Maddeler	Cronbach Alfa
Problem Çözme Becerisine Güven	12	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23	0,85
Özdenetim	7	2, 4, 6, 8, 10, 12, 14	0,78
Kaçınma	5	16, 18, 20, 22, 24	0,66

5’li Likert türündeki ölçekte bulunan maddeler için “Her Zaman”, “Sık Sık”, “Arada Sırada”, “Ender Olarak” ve “Hiçbir Zaman” dereceleri kullanılmıştır. “Problem Çözme Becerisine Güven” faktörü altındaki 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 numaralı maddeler değerlendirilirken öğrencilerin aldıkları puanlar 1’den 5’e yaklaştıkça artmıştır. Ters puanlanan “Öz Denetim” faktöründeki 2, 4, 6, 8, 10, 12,

14 numaralı maddeler ve “Kaçınma” faktörü altındaki 16, 18, 20, 22, 24 numaralı maddeler değerlendirilirken puanlar ise 1’den 5’e yaklaştıkça azalmıştır. Ters puanlanan maddeler recode yapılarak tekrar düzenlenmiştir. Ölçekten alınabilecek en düşük puan 24 iken en yüksek puan ise 120’dir. Öğrencilerin ölçekten aldıkları puana göre kendilerini problem çözme becerileri konusunda yeterli görme durumları ortaya çıkacaktır (Serin vd., 2010).

Motivasyon ölçeği (CIS-Derse karşı tutum ölçeği). Araştırma kapsamında gerçekleştirilen çalışmanın öğrenciler üzerindeki motivasyonel etkilerini belirlemek amacıyla CIS - Derse Karşı Tutum Ölçeği kullanılmıştır. Ölçek öğrencilerin derse karşı tutumlarını ölçmek amacıyla Keller ve Subhiyah (1987) tarafından geliştirilmiştir (Keller, 2006). Motivasyon Ölçeği (CIS-Derse Karşı Tutum Ölçeği) “Dikkat (Attention)” (8 madde), “Uygunluk (Relevance)” (9 madde), “Güven (Confidence)” (8 madde) ve “Doyum (Satisfaction)” (9 madde) olmak üzere 4 faktör ve toplamda 34 maddeden oluşmaktadır.

Ölçeğin Türkçeye çevirisi ilk olarak Varank (2003) tarafından yapılmıştır. Ölçeğin İngilizceden Türkçeye çevirisi Amerika’da Türk Dili ve Edebiyatı bölümünde okuyan, her iki dilde de ileri düzeyde dil yetkinliğine sahip iki öğrenciye kontrol ettirilmiş ve daha sonra gerekli düzeltmeler yapılarak tamamlamıştır. Ölçek daha sonra 8. sınıfta eğitime devam eden 195 öğrenciye uygulanarak güvenilirlik analizleri yapılmış; Alpha güvenilirlik katsayısı ölçeğin tamamı için 0,83, dikkat alt faktörü için 0,55, uygunluk alt faktörü için 0,59, güven alt faktörü için 0,67 ve doyum alt faktörü için 0,59 olarak bulunmuştur. Elde edilen değerler neticesinde ölçeğin ortaokul öğrencileri için güvenilir bir ölçek olduğu ortaya çıkmıştır.

Acar (2009) tarafından gerçekleştirilen çevirisinde ise CIS ölçeğinin Keller’in (2006) çalışmasındaki en son sürümü kullanılmıştır. Ölçeğin Türkçeye çevrilmesi amacıyla ilk olarak ölçekteki ifadeler 3 farklı tercüme bürosunda Türkçeye, daha sonra da farklı 3 tercüme bürosunda Türkçeden İngilizceye çevrilmiş, ölçekte bulunan 34 maddenin dil bakımından anlamlı olup olmadığını görmek için konu uzmanlarının görüşü alınmıştır. Uzman görüşleri sonrasında, ölçeğin anlaşılabilirliğini test etmek için pilot uygulama yapılmış ve öğrencilerin görüşleri alınmıştır. Daha sonra ölçekteki ifadelerin kendi aralarında tutarlık gösterip göstermediğini test etmek için pilot uygulama çalışması dâhilinde güvenilirlik analizi yapılmıştır. Alpha güvenilirlik katsayısı ölçeğin tamamı için 0,93, “dikkat” alt faktörü için 0,73, “uygunluk”

alt faktörü için 0,85, “güven” alt faktörü için 0,71 ve “doyum” alt faktörü için 0,77 olarak bulunmuştur. Elde edilen değerler neticesinde çalışmada kullanılan motivasyon ölçeğinin güvenilir bir ölçek olduğu ortaya çıkmıştır.

Tablo 12

Ölçeğin Alt Faktörleri ve Cronbach Alfa Katsayıları

Faktör Adı	Madde Sayısı	Maddeler	Cronbach Alfa
Dikkat	8	1, 4, 10, 15, 21, 24, 26, 29	0,73
Uygunluk	9	2, 5, 8, 13, 20, 22, 23, 25, 28	0,85
Güven	8	3, 6, 9, 11, 17, 27, 30, 34	0,71
Doyum	9	7,12, 14, 16, 18, 19, 31, 32, 33	0,77

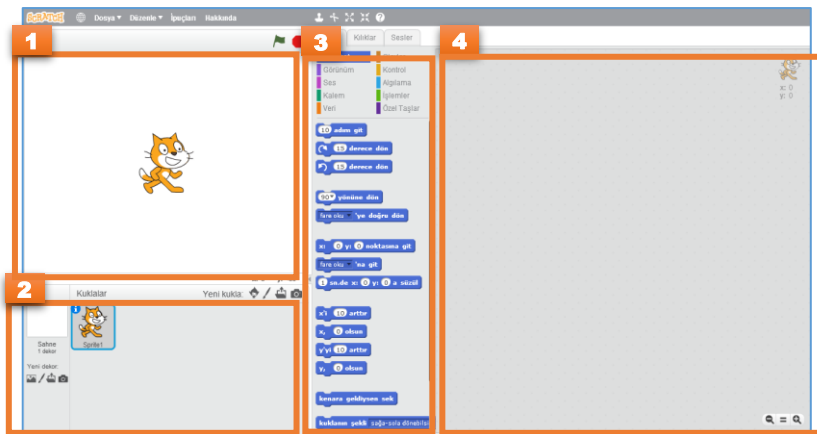
5’li Likert türündeki ölçekte bulunan maddeler için “Doğru Değil (1)”, “Biraz doğru (2)”, “Orta Derece Doğru (3)”, “Oldukça Doğru (4)” ve “Çok Doğru (5)” dereceleri kullanılmıştır. “Dikkat” faktörü altındaki 4 ve 26 numaralı maddeler, “Uygunluk” faktörü altındaki 8 ve 25 numaralı maddeler, “Güven” faktörü altındaki 6, 11 ve 17 numaralı maddeler, “Doyum” faktörü altındaki 7 ve 31 numaralı maddeler olumsuz ifadelerden oluştuğu için ters olarak puanlanmıştır. Bu maddeler değerlendirilirken öğrencilerin aldıkları puanlar 1’den 5’e yaklaştıkça azalmış, olumlu ifadelerden oluşan diğer maddeler değerlendirilirken öğrencilerin aldıkları puanlar 1’den 5’e yaklaştıkça artmıştır. Ters puanlanan maddeler recode yapılarak tekrar düzenlenmiştir. Ölçekten alınabilecek en düşük puan 34 iken en yüksek puan 170’dir.

Blok tabanlı programlama ortamı (Scratch). Massachusetts Teknoloji Enstitüsü (MIT) Medya Laboratuvarı tarafından geliştirilen ücretsiz bir yazılım olan Scratch, çoklu dil desteği, çevrimdışı kullanılmasının yanında çevrimiçi kullanım özelliğinin olması ve oluşturulan projelerin dünya genelinde diğer kullanıcılarla paylaşılabilmesi sayesinde yaygın olarak kullanılmaktadır. Scratch, programlamaya yeni başlayanlar için programlama kavramını eğlenceli bir yolla sunan bir kodlama yazılımıdır. Scratch, basit sürükle ve bırak yöntemi kullanılarak, kodlamanın temellerini öğrettiği gibi aynı zamanda gerçek bilgisayar programları da oluşturmaya yarar (Holland & Minnick, 2016). Türkçe de dâhil olmak üzere 40’in üzerinde dil desteğiyle Scratch, kullanıcılarına ücretsiz olarak sunulmaktadır. İlk olarak 2007

yılında geliştirilen Scratch, 2012 yılında web ortamında Scratch 2.0 sürümü ile hem çevrimiçi hem de çevrimdışı kullanım özelliğine sahip olmuştur. Scratch'ın İnternet ortamına girmesi, dünya çapında daha fazla büyümesine ve daha fazla kişiye ulaşmasına olanak sağlamıştır. Öğrencilerin kendi yaptığı etkileşimli hikayelerini, oyunlarını, animasyonlarını programladıkları ve oluşturdukları çalışmalarını paylaşması; yaratıcı düşünme, sebep sonuç ilişkisi kurma ve takım halinde çalışma gibi temel yaşam becerilerini kazanmaları açısından öğrenciler için büyük bir fırsat olmuştur (Scratch, 2019). Kordaki (2012) Scratch programının özelliklerini şu şekilde belirtmiştir:

- Özgürce etkinlik yapma
- Projeler üzerinde deneme yapma
- Projeler üzerinde değişiklik yapma
- Doğru olan kodun eksik bir kısmı üzerinde çalışma
- Kodun yanlış kısmı üzerinde çalışma
- Karakutu çalışmaları yapma
- İşbirlikli öğrenme sağlama

Scratch yazılımının arayüzü 4 bölümden oluşmaktadır (Şekil 3). Şekil 3'de 1 numaralı alan kuklanın (karakterin) hareketlerini sergilediği sahneyi, 2 numaralı alan projede yer alan kuklaların özelliklerinin ve sahne arka dekorunun düzenlendiği bölümü, 3 numaralı alan kod bloklarının renklere göre sınıflandırıldığı bölümü ve 4 numaralı alan seçilen kuklanın davranışlarını belirleyen kod yığınlarının yerleştirildiği bölümü belirtmektedir.



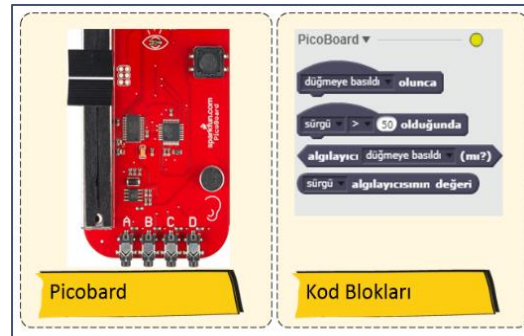
Şekil 3. Scratch 2.0 genel yapısı ve arayüzü.

Scratch yazılımında, doğrusal yapı blokları, karar yapı blokları, çoklu karar yapı blokları ve döngü blokları ile kuklaların davranışları kontrol edilebildiği gibi bu kod blokları kullanılarak çizim ve matematiksel işlemler yapılabilir. Scratch programında yapısal olarak bloklar; komut blokları, fonksiyon blokları, tetikleme (şapkalı) blokları ve kontrol blokları olmak üzere temel olarak 4 çeşittir (Şekil 4). Yığın blokları olarak adlandırılan komut ve kontrol blokları girinti ve çıkıntıları sayesinde birbirleriyle birleşerek yığın oluşturabilmektedir. Şapka blokları olarak da adlandırılan tetikleme blokları bir olay gerçekleştiğinde kendinden sonra gelen yığın bloklarını çalıştırır. Herhangi bir girinti ya da çıkıntıya sahip olmayan fonksiyon blokları ise sayısal, metinsel ya da mantıksal veri döndürerek diğer bloklara veri girişi sağlarlar (Marji, 2014). Scratch programında kullanılan blok türleri Şekil 4'te sunulmuştur.

Blok Tipi	Örnek Blok
Komut Blokları	
Tetikleme Blokları	
Fonksiyon Blokları	
Kontrol Blokları	

Şekil 4. Scratch programında kullanılan blok türleri.

Yazılım ve donanım arasında etkileşim kurarak otantik dünyayı algılayabilme özelliği olan Scratch ile oluşturulan projelerde bilgisayar dışındaki bazı elektronik sistemler de algılanabilir ve bunlara tepki verilebilir. Eklenti desteği olan Scratch için farklı geliştiriciler tarafından oluşturulmuş eklentiler ile elektronik cihazlar ve robot kitleri gibi harici donanımlara hükmedilmektedir. Kullanılan robotik kitin özelliklerine göre kod blokları değişkenlik göstermektedir. Şekil 5'te Scratch yazılımında kullanılan elektronik kit ve bu kitin farklı blokları görülmektedir.



Şekil 5. Picobard ve kontrolünü sağlayan kod blokları.

LEGO® Education WeDo 2.0 robot kiti. İlkokullar ve ortaokulların alt sınıflarında öğrencilerin yaş ve gelişim düzeylerine uygun olarak hazırlanmış olup öğrencilerin yaratıcılıklarını kullanarak ilgi çekici projeler oluşturmalarına olanak sağlayan bir lego setidir (Lego Education, 2019). Öğrencilerin yaratıcılıklarını ve hayal güçlerini kullanarak kendi robotlarını oluşturabilecekleri bu setle STEM becerilerinin yanında programlanabilir olması nedeniyle kodlama becerileri de geliştirebilmektedir. Scaradozzi, Sorbi, Pedale, Valzano ve Vergine'e (2015) göre Lego Wedo 2.0 küçük yaştaki öğrencilere kolay ve eğlenceli bir şekilde temel mühendislik kavramlarını öğretmeyi sağlar.

Massachusetts Institute of Technology (MIT) araştırmacıları 11 yaşından büyük çocuklar için son derece başarılı olan LEGO Mindstorms robotik kitinden sonra, daha küçük yaş grubundaki çocukların robot geliştirme ve eğitim sürecinde öğrencilerin yaşadıkları olumsuz durumları ortadan kaldırmak amacıyla LEGO® Education WeDo setini 2009 yılında tasarlayarak piyasaya sürmüşlerdir. Böylece daha küçük yaş grubundaki öğrenciler de LEGO robotiğini tanımışlardır. Lego WeDo 2.0 yazılımı tabletlerde ve kişisel bilgisayarlarda çalışmaktadır. Genel olarak WeDo 2.0 seti aşağıda belirtilen parçalardan oluşmaktadır (Şekil 6, Şekil 7):

1. Bir adet WeDo 2.0 akıllı tuğla
2. Bir adet motor
3. Bir adet hareket sensörü
4. Bir adet eğim sensörü
5. Robotu inşa etmekte kullanılan 280 parçadan oluşan dişliler, tekerlekler



Şekil 6. LEGO education WeDo 2.0 temel set.

WeDo 2.0 Akıllı Tuğla

WeDo 2.0 Temel setinin kontrol kartı niteliğinde olan Akıllı Tuğla, WeDo sensörlerinin ve motorlarının çalışmasını sağlar. Bluetooth® Düşük Enerji (BTLE) teknolojisini kullanarak bilgisayarlara bağlanan akıllı tuğla üzerinde 1 adet RGBLed, 1 adet açma kapama düğmesi ile sensör ve motor bağlantıları için iki adet port bulunmaktadır. Enerji kaynağı olarak ise iki adet AA pil kullanır (Lego Education, 2019).



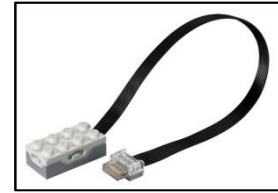
Hareket Sensörü

WeDo 2.0 hareket sensörü, sensör ile nesne arasındaki uzaklığı ölçer. Hareket sensörü nesnenin şekline ve yansıtıcı özelliklerine bağlı olarak, yaklaşık 15 cm uzaklıkta bulunan engelleri algılayabilir. Nesne hareket ettiğinde, mesafe uzaklığını karşısında bulunan diğer nesneye göre değiştirir. Hareket sensörü hareketi ölçebildiği gibi aynı zamanda mesafeyi de ölçebilir. Bu nedenle mesafe sensörü olarak adlandırılır (Lego Education, 2019).



Eğim Sensörü

Eğim sensörü robotun yatay ya da dikey denge durumuna göre veri elde eden algılayıcısıdır. WeDo 2.0 eğim sensörü yedi farklı şekilde denge durumunu algılayabilir. Bunlar, bu tarafa eğim, diğer tarafa eğim, yukarı eğim, aşağı eğim, eğilme yok, herhangi bir tarafa eğim ve sallayın şeklindedir (Lego Education, 2019).



WeDo 2.0 Motor

Kullanımı oldukça kolay olan bu orta boylu, orta kuvvetli motor robotun hareketini sağlar. WeDo 2.0 akıllı tuğla ile birlikte kullanılır.



Şekil 7. LEGO education WeDo 2.0 temel set parçalarının özellikleri.

Sensörleri veya motoru kullanmak için LEGO WeDo 2.0 akıllı tuğlanın üzerinde bulunan portlarından herhangi birine bağlamak gereklidir. Akıllı tuğla ise bilgisayara Bluetooth® Düşük Enerji (BTLE) kullanarak bağlanmalıdır. WeDo yazılımını kullanarak, bir bilgisayara üç farklı akıllı tuğla bağlanabilir. Böylece daha fazla motor veya sensor bir arada kullanılabilir (Lego Education, 2019). WeDo 2.0 seti ile oluşturularak tasarlanan robotlar Scratch yazılımı kullanılarak kodlanabilir.

Uygulama Süreci

Araştırmanın uygulama süreci 2017-2018 Eğitim-Öğretim yılı 2. döneminde, Bilişim Teknolojileri ve Yazılım dersinde gerçekleştirilmiştir. Bilişim Teknolojileri ve Yazılım dersi problem çözme becerilerinin ve programlama öğretiminin de yer aldığı 5. ve 6. sınıflarda haftada 2 ders saati olarak okutulan bir derstir. Haftalık 2 ders olmak üzere ders anlatımları her iki grup için 12 hafta boyunca devam etmiştir. Ön test verileri uygulama öncesinde müfredat kapsamında yapılmış olup son test verileri ise 12 haftalık uygulama sürecinden sonra dönem sonunda yapılmıştır. Uygulamada her sınıf seviyesinden iki şubeye blok tabanlı görsel programlama öğretimi (Scratch), aynı sınıf seviyesinden diğer iki şubeye ise blok tabanlı robotik programlama öğretimi (Scratch + WeDo 2.0) gerçekleştirilmiştir. Araştırmada birbirine kolay bağlanabilen parçalardan oluşan, birçok nesneyi modelleyebilecek yeterli sayıda yapı ve hareket bileşenine sahip blok tabanlı (Lego) robot kiti kullanılmıştır. Öğrenciler daha önceden herhangi bir robotik eğitimi almamalarından dolayı ortaokul alt sınıflarının seviyesine uygun olan WeDo 2.0 tercih edilmiştir. WeDo 2.0 robotik kiti elektronik devre bileşenlerinden uzak yapısıyla robotik deneyim ve elektronik bilgisi gerektirmediğinden başlangıç eğitiminde tercih edilmekte ve “Scratch” gibi görsel programlama yazılımlarıyla kolaylıkla programlanabilmektedir. Tasarım süreci zaman alan bir süreçtir. Bu nedenle yalnızca robotların programlamadaki etkisini görmek ve uygulamaları eşdeğer sürede gerçekleştirmek için programlanabilir parçalar araştırmacı tarafından daha önceden tasarlanıp inşa edilerek öğrencilere sunulmuştur.

Scratch programı diğer blok tabanlı kodlama yazılımlarına göre daha fazla özelliği olması nedeniyle tercih edilmiştir. Bu özellikler; ücretsiz olması, dil desteği olması, programlamayı kolaylaştıran görsel programlama ortamı sunması, öğrencilerin ilgi ve dikkatini çekebilecek özelliklere sahip olması, çevrimiçi ve

çevrimdışı kullanılabilmesi, dünya çapında yapılan başka projeleri inceleme ve düzenleme özelliği ve Lego robotlarıyla uyumlu olmasıdır. Uygulama aşamasında eğitim sürecinde hazırlanan içeriklere uygun bir ilerleyiş takip edilmiştir.

Uygulama sürecinde gerçekleştirilen ders anlatımları ve toplamda 2 hafta süren veri toplama aşaması araştırmacı tarafından gerçekleştirilmiştir. Bu kapsamda dersin planlaması, eğitim içeriklerinin oluşturulması, derslerin işlenmesi ve katılımcıların araştırma sürecinde takibi ve çalışma gruplarının oluşturulması kontrol edilmiştir. Uygulama süreci ve ders kazanımları tüm ayrıntıları ile Tablo 13'te sunulmuştur (Milli Eğitim Bakanlığı, 2018).

Tablo 13

Uygulama Süreci ve Ders Kazanımları

UYGULAMA SÜRECİ			
HAFTALAR	DENEY GRUBU	KONTROL GRUBU	SÜRE
1. Hafta	ÖN TESTLER	Veri Toplama Araçları (Ön Testler)	2 DERS
		Kişisel Bilgi Formu Problem Çözme Envanteri Erişi Testi	10 dak. 15 dak. 1 saat
2. Hafta	PROBLEM ÇÖZME KAVRAMLARI VE YAKLAŞIMLARI	Algoritma ve Strateji Geliştirme Kazanımları	2 DERS
		5.5.1.6. Problemi çözmek için gerekli değişken, sabit ve işlemleri açıklar. 5.5.1.7. Problem çözümünde kullanılacak operatörlere örnek verir. 6.5.1.1. Verileri toplayarak türlerine göre sınıflandırır. 6.5.1.2. Sabitleri ve değişkenleri problem çözümünde kullanır.	
3. Hafta	PROBLEM ÇÖZME KAVRAMLARI VE YAKLAŞIMLARI	Algoritma ve Strateji Geliştirme Kazanımları	2 DERS
		5.5.1.8. Problem çözümünde ifade ve eşitliklere örnek verir. 5.5.1.9. Problem çözümünde işlem önceliğine örnek verir. 6.5.1.3. Bir problemi alt problemlere böler. 6.5.1.4. Temel fonksiyonları problem çözme sürecinde kullanır.	
4. Hafta	PROBLEM ÇÖZME KAVRAMLARI VE YAKLAŞIMLARI	Algoritma ve Strateji Geliştirme Kazanımları	2 DERS
		5.5.1.12. Algoritma kavramını açıklar. 5.5.1.13. Bir problemin çözümü için algoritma geliştirir. 6.5.1.5. Problemin çözümü için bir algoritma geliştirir. 6.5.1.6. Bir algoritmanın çözümünü test eder.	
5. Hafta	PROBLEM ÇÖZME KAVRAMLARI VE YAKLAŞIMLARI	Algoritma ve Strateji Geliştirme Kazanımları	2 DERS
		5.5.1.14. Akış şeması bileşenlerini ve işlevlerini açıklar. 5.5.1.15. Bir algoritma için akış şeması çizer. 6.5.1.7. Farklı algoritmaları inceleyerek en hızlı ve doğru çözümü seçer. 6.5.1.8. Hatalı bir algoritmayı doğru çalışacak biçimde düzenler.	

6. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.2. Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanıtır.		
	6.5.2.1. Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanıtır.		
	6.5.2.2. Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.		
7. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.4. Doğrusal mantık yapısını açıklar.		
	5.5.2.5. Doğrusal mantık yapısını kullanan algoritmalar geliştirir.		
	6.5.2.5. Doğrusal mantık yapısını içeren programlar oluşturur.		
	6.5.2.6. Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.		
8. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.5. Doğrusal mantık yapısını kullanan algoritmalar geliştirir.		
	6.5.2.5. Doğrusal mantık yapısını içeren programlar oluşturur.		
	6.5.2.6. Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.		
9. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.7. Karar yapılarını içeren algoritmalar geliştirir.		
	6.5.2.7. Karar yapısını içeren programlar oluşturur.		
	6.5.2.8. Karar yapısını içeren programları test ederek hatalarını ayıklar.		
10. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.7. Karar yapıları içeren algoritmalar geliştirir.		
	6.5.2.9. Çoklu karar yapıları içeren programlar oluşturur.		
	6.5.2.10. Çoklu karar yapısını içeren programları test ederek hatalarını ayıklar.		
11. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.7. Karar yapıları içeren algoritmalar geliştirir.		
	5.5.2.9. Döngü yapısı içeren algoritmalar oluşturur.		
	6.5.2.7. Karar yapısını içeren programlar oluşturur.		
	6.5.2.11. Döngü yapısını içeren programlar oluşturur.		
12. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.7. Karar yapıları içeren algoritmalar geliştirir.		
	5.5.2.9. Döngü yapısı içeren algoritmalar oluşturur.		
	6.5.2.7. Karar yapısını içeren programlar oluşturur.		
	6.5.2.11. Döngü yapısını içeren programlar oluşturur.		
13. Hafta	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch+Lego WeDo 2.0)	2 DERS
	5.5.2.9. Döngü yapısı içeren algoritmalar oluşturur.		
	6.5.2.7. Karar yapısını içeren programlar oluşturur.		
	6.5.2.11. Döngü yapısını içeren programlar oluşturur.		
14. Hafta	Veri Toplama Araçları (Son Testler)		2 DERS
	Problem Çözme Envanteri		15 dak.
	Motivasyon ölçeği		15 dak.
	Erişi testi		1 saat

Uygulanan Eğitim İçerikleri

Eğitim sürecinde anlatılacak konular ders müfredatına göre gerçekleştirilmiş, ders anlatımlarını içeren bölümlerde gerekli olan bilgi aktarımı çalışmaları için 5. sınıf öğretmen kılavuzu kitabındaki uygulama, örnek ve tanımlardan yararlanılmıştır (Gülbahar Güven, 2018). Uygulama örneklerinin belirlenmesinde ise programlama dili öğretimi ile ilgili ders kitapları ve İnternet kaynakları incelenmiştir. Eğitim programındaki blok tabanlı kodlama etkinlikleri Milli Eğitim müfredatına uygun olarak alanı bilişim teknolojileri olan öğretmen tarafından hazırlanmış ve geçerliği için uzman görüşü alınmıştır. Uygulanan eğitim içerikleri ile ilgili bilgiler Tablo 14'te sunulmuştur.

Tablo 14

Uygulama Sürecinde Uygulanan Eğitim İçerikleri

Deney ve Kontrol Grupları Eğitim İçerikleri	
Hafta 1 (2 Ders)	A. Bilgi - Veri ve veri türleri kavramları hakkında bilgi verilir. B. Bilgi - Sabit ve değişken kavramı hakkında bilgi verilir. C. Çalışma - Veri türü kavramına yönelik çalışma yapılır. D. Çalışma - Sabit ve değişken kavramına yönelik çalışma yapılır. E. Değerlendirme yapılır.
Hafta 2 (2 Ders)	A. Bilgi - Operatör ve İşlem Önceliği kavramları hakkında bilgi verilir. B. Çalışma - Operatör kavramına yönelik çalışmalar yapılır. C. Değerlendirme yapılır.
Hafta 3 (2 Ders)	A. Bilgi - Algoritma kavramı hakkında bilgi verilir. B. Çalışma - Algoritma kavramına yönelik çalışma yapılır. C. Bilgi - Akış şemaları kavramı hakkında bilgi verilir. D. Çalışma - Akış şemaları kavramına yönelik çalışma yapılır. E. Değerlendirme yapılır.
Hafta 4 (2 Ders)	A. Çalışma - Algoritma kavramına yönelik çalışma yapılır. B. Çalışma - Akış şemaları kavramına yönelik çalışma yapılır. C. Değerlendirme yapılır.
Hafta 5 (2 Ders)	A. Bilgi: Blok temelli programlama hakkında bilgi verilir. B. Bilgi: Scratch Programının arayüzü tanıtılır. C. Çalışma: Scratch bloklarına yönelik çalışma yapılır. D. Uygulama: Scratch "Kukla Dansı" uygulaması yaptırılır. (Hareket, görünüm, kontrol ve ses gruplarında bulunan kodlar kullanılır.)
Hafta 6 (2 Ders)	A. Bilgi: Scratch bloklarını tanıtılır. B. Uygulama: Scratch "Hareketi Severim" uygulaması yaptırılır. (Kontrol, olaylar ve hareket gruplarında bulunan kodlar kullanılır.)

	Kontrol Grubu Eğitim İçerikleri	Deney Grubu Eğitim İçerikleri
	Blok Tabanlı Programlama (Scratch)	Blok Tabanlı Robotik Programlama (Scratch + LegoWeDo 2.0)
Hafta 7 (2 Ders)	<p>A. Bilgi: Fiziksel Dünya ile Etkileşim hakkında bilgi verilir.</p> <p>B. Bilgi: Robotları hakkında bilgi verilir.</p> <p>C. Bilgi: Scratch doğrusal yapılar hakkında bilgi verilir.</p> <p>D. Uygulama: "Maymun ve Muz" uygulaması adım adım yapılır. Scratch yazılımı kullanılır.</p> <p>E. Değerlendirme yapılır.</p>	<p>A. Bilgi: Fiziksel Dünya ile Etkileşim hakkında bilgi verilir.</p> <p>B. Bilgi: Robotları hakkında bilgi verilir.</p> <p>C. Bilgi: Scratch doğrusal yapılar hakkında bilgi verilir.</p> <p>D. Uygulama: "Maymun ve Muz" uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 1 adet Lego WeDo SmartHub kullanılır. 1 adet Buton kullanılır. 1 adet Led kullanılır.</p> <p>E. Değerlendirme yapılır.</p>
Hafta 8 (2 Ders)	<p>A. Bilgi: Scratch değişken oluşturma hakkında bilgi verilir.</p> <p>B. Bilgi: Scratch karar yapıları hakkında bilgi verilir.</p> <p>C. Uygulama: "Elma Toplama" uygulaması adım adım yapılır. Scratch yazılımı kullanılır.</p> <p>D. Değerlendirme yapılır.</p>	<p>A. Bilgi: Scratch değişken oluşturma hakkında bilgi verilir.</p> <p>B. Bilgi: Scratch karar yapıları hakkında bilgi verilir.</p> <p>C. Uygulama: "Elma Toplama" uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 1 adet Lego WeDo SmartHub kullanılır. 1 adet Eğitim Sensörü kullanılır.</p> <p>D. Değerlendirme yapılır.</p>
Hafta 9 (2 Ders)	<p>A. Bilgi: Özel Taşlar (Fonksiyon) hakkında bilgi verilir.</p> <p>B. Bilgi: Scratch çoklu karar yapıları hakkında bilgi verilir.</p> <p>C. Uygulama: "Yarasa Vurma" uygulaması adım adım yapılır. Scratch yazılımı kullanılır.</p> <p>D. Değerlendirme yapılır.</p>	<p>A. Bilgi: Özel Taşlar (Fonksiyon) hakkında bilgi verilir.</p> <p>B. Bilgi: Scratch çoklu karar yapıları hakkında bilgi verilir.</p> <p>C. Uygulama: "Yarasa Vurma" uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 1 adet Lego WeDo SmartHub kullanılır. 1 adet Hareket (Mesafe) sensörü kullanılır. 1 adet Buton kullanılır.</p> <p>D. Değerlendirme yapılır.</p>

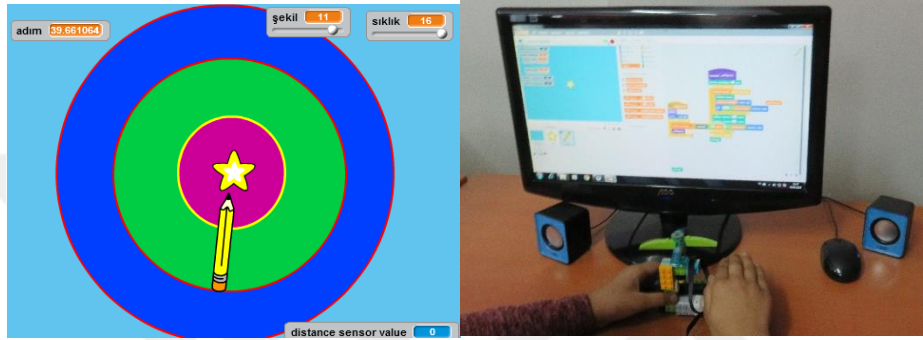
Hafta 10 (2 Ders)	<p>A. Bilgi: Diziler (Listeler) Oluşturma hakkında bilgi verilir.</p> <p>B. Bilgi: Scratch'de Döngü Yapısı hakkında bilgi verilir.</p> <p>C. Uygulama: Toplama Oyunu uygulaması adım adım yapılır. Scratch yazılımı kullanılır.</p> <p>D. Değerlendirme yapılır.</p>	<p>A. Bilgi: Diziler (Listeler) Oluşturma hakkında bilgi verilir.</p> <p>B. Bilgi: Scratch'de Döngü Yapısı hakkında bilgi verilir.</p> <p>C. Uygulama: Toplama Oyunu uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 1 adet Lego WeDo SmartHub kullanılır. 1 adet Hareket (Mesafe) sensörü kullanılır. 1 adet Motor kullanılır. Mekanik parçaları kullanılır.</p> <p>D. Değerlendirme yapılır.</p>
Hafta 11 (2 Ders)	<p>A. Uygulama: Kedi ve Fare uygulaması adım adım yapılır. Scratch yazılımı kullanılır.</p>	<p>A. Uygulama: Kedi ve Fare uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 2 adet Lego WeDo SmartHub kullanılır. 1 adet Hareket (Mesafe) sensörü kullanılır. 1 adet Motor kullanılır. Mekanik parçalar kullanılır.</p>
Hafta 12 (2 Ders)	<p>A. Uygulama: Şekil Çizme uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 1 adet Lego WeDo SmartHub kullanılır. 1 adet Hareket (Mesafe) sensörü kullanılır. 1 adet Motor kullanılır. Mekanik parçalar kullanılır.</p>	<p>A. Uygulama: Şekil Çizme uygulaması adım adım yapılır. Scratch yazılımı kullanılır. 1 adet Lego WeDo SmartHub kullanılır. 1 adet Hareket (Mesafe) sensörü kullanılır. 1 adet Motor kullanılır. Mekanik parçalar kullanılır.</p>

DeneySEL Çerçeve

Kişisel bilgi formunun, erişim testinin, problem çözme envanterinin ön test olarak Deney ve Kontrol gruplarına uygulanması ile uygulama süreci başlamıştır. Her iki gruba da ilk 4 hafta boyunca problem çözme kavramı ve yaklaşımları diğer bir ifadeyle programlamanın temel kavramları öğretilmiştir. Öğrencilerin otantik bir problemi ortaya koyup o probleme ait işlem adımlarını sistematik bir şekilde oluşturabilmelerini sağlamak amacıyla değişken, operatör, algoritma ve akış şemaları gibi konuların öğretimi yapılarak programlama yazılımı kullanmadan önce programlamanın temelleri hakkında bilgi verilmiş ve böylece derse alışmaları sağlanmıştır.

4 haftalık temel programlama eğitiminin ardından, 2 hafta boyunca hem deney hem de kontrol gruplarının kullanacağı blok tabanlı programlama yazılımının

arayüzünün ve özelliklerinin öğretimi gerçekleştirilmiştir. Daha sonraki süreçte ise kontrol grubundaki öğrenciler blok tabanlı görsel programlama ortamında eğitim alırken, deney grubundaki öğrenciler ise blok tabanlı görsel programlama ortamında eğitsel robot kitleri kullanarak blok tabanlı robotik eğitimi almışlardır (Şekil 8). Bu süreç 6 hafta boyunca devam etmiştir. Bu süreç içerisinde dersler anlatım yöntemiyle ve uygulamalı olarak bilgisayar laboratuvarında gerçekleştirilmiştir. Anlaşılmayan noktalarda gerekli tekrarlar ve düzeltmeler yapılmış, dönütler verilmiştir.



Şekil 8. Blok tabanlı programlama ve blok tabanlı robotik programlama ortamı.

Deney ve kontrol grupları şubeleri için dersler aynı bilgisayar laboratuvarında yapılmış olup, eğitim öncesinde bu laboratuvar yazılım ve donanım açısından eğitime uygun hale getirilmiştir. Kullanılan robot setleri sınırlı sayıda olduğu için öğrenciler için 4'erli çalışma grupları oluşturulmuştur. Şekil 9'da grupların uygulama sürecinden görseller sunulmuştur.



Şekil 9. Kontrol ve deney gruplarına ait uygulama sürecinden bir görünüm.

Toplamda 12 haftalık eğitimin ardından erişim testleri, problem çözme envanteri ve motivasyon ölçeği son test olarak her iki gruba da uygulanarak süreç sonlandırılmıştır.

Verilerin Analizi

Ön test ve son test olarak toplanan nicel veriler kontrol edilmiş daha sonra hesap tablosu programına aktarılmıştır. Hesap tablosu programında elde edilen nicel veriler tekrar gözden geçirilmiş ve istatistiksel analizlerinin yapılması için istatistik paket programına aktarılmıştır. Araştırma sonucunda elde edilen nicel veriler SPSS (Statistical Package for Social Sciences) paket programı kullanılarak nicel yaklaşıma göre analiz edilmiştir. Uygulanan test ve ölçeklerden elde edilen verilerin çözümlenmesi için frekans, aritmetik ortalama, standart sapma ve yüzde dağılımı; deney ve kontrol gruplarında yapılan testler ve ölçek puanları arasındaki farka bakmak için ise kovaryans analizi (ANCOVA), eşleştirilmiş örneklem t-testi ve bağımsız örneklem t-testi yapılmıştır.

Büyüköztürk'e (2004, s.105) göre kovaryans analizi (ANCOVA) bir araştırmada etkisi test edilen bir faktörün ya da faktörlerin dışında, bağımlı değişken ile ilişkisi bulunan başka bir değişkenin ya da değişkenlerin istatistiksel olarak kontrol edilmesini sağlayarak deneysel işlemin gerçek etkisinin ortaya çıkmasını sağlar. Araştırma öncesinde her iki gruba erişim testi ile problem çözme becerilerini saptamak için problem çözme envanteri uygulanmış; araştırmanın bitiminde aynı test ve envanter deney ve kontrol grubuna son test olarak tekrar uygulanmış böylece son test puanları arasındaki farka bakarak uygulamanın etkisi, her gruptaki ön test son test puanları arasındaki farka bakarak da araştırma öncesi ile sonrası arasındaki fark görülebilmektedir. Ayrıca her iki gruba son test olarak motivasyon ölçeği uygulanmış, böylece iki grup arasındaki son test motivasyon puanları ortalamaları arasındaki fark görülebilmektedir. Tüm istatistiksel analizlerde ,05 anlamlılık düzeyi temel alınmıştır.

Çalışmanın iç geçerliği

Her iki grupta bulunan öğrenciler de benzer yaş ve sınıf seviyesindeki öğrencilerden seçilmiştir. Çalışma sonunda testleri tam olarak dolduramayan ve doldurmak istemeyen, eğitime tam katılmayan öğrenciler belirlenerek araştırmanın kapsamı dışında tutulmuştur. Ön test ile son test sorularının aynı olmasından dolayı, öğrencilerin soruları hatırlamasını zorlaştırmak için ikinci ölçümler 13 haftalık bir zaman dilimi sonrasında gerçekleştirilmiştir. Ayrıca her iki gruba da verilen süreler eşit tutulmuştur. Veri toplayıcının farklılaşmasından kaynaklı tehditleri

engelleyebilmek için çalışma kapsamındaki tüm süreç alanı “Bilişim Teknolojileri” olan araştırmacı tarafından yönetilmiştir. Her iki gruba 12 hafta süresince ders anlatımı ve tüm verilerin toplanması yine aynı şekilde araştırmacı tarafından gerçekleştirilmiştir. Mekâna bağlı olarak ortaya çıkabilecek iç geçerlik tehditlerini engellemek için her iki gruba da uygulama aynı ortamda yapılmış ve veriler aynı ortamda toplanmıştır.

Çalışmanın dış geçerliği

Çalışma Bolu ili Merkez ilçesinde bulunan bir devlet ortaokulunun 5. ve 6. sınıflarına devam eden 8 farklı şubedeki 87 öğrenci ile sınırlıdır. Bu durum elde edilen sonuçların daha büyük gruplara genellenebilirliği bakımından sorun oluşturabilir. Elde edilen sonuçlar yalnızca benzer özelliklere sahip gruplar için genellenebilir.

Bölüm 4

Bulgular ve Yorumlar

Bu bölümde araştırmanın alt problemlerine ilişkin bulgulara yer verilmiştir. Blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin yapıldığı öğrencilerin erişileri, problem çözme becerileri ve motivasyon durumları sırasıyla incelenmiştir. Verilerin istatistiksel olarak çözümlenmesi sonucunda elde edilen bulgular tablolar halinde sunulmuştur.

Programlama Erişi Puanlarına İlişkin Bulgular

Birinci alt probleme ait bulgular. Araştırmanın birinci alt problemi “Deney ve kontrol gruplarının eriş testi sonucunda aldıkları ön test ve son test puanları arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Blok tabanlı robotik programlama eğitimi alan öğrencilerin eriş ön test puanları ile son test puanları arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını bulmak için, programlama eriş ön test ve son test puanları üzerinde eşleştirilmiş örneklem t-testi (paired samples t-test) analizi yapılmış ve etki büyüklüğü hesaplanmıştır. Elde edilen bilgiler Tablo 15’te sunulmuştur.

Tablo 15

Deney ve Kontrol Gruplarının Erişi Puanlarına İlişkin (Ön Test - Son Test) Eşleştirilmiş Örneklem t-Testi Sonuçları

Gruplar	Testler	N	\bar{x}	ss	sd	t	p
Blok Tabanlı Robotik (DG)	Erişi Ön test puanları	44	79,86	20,88	43	-14,82	,000
	Erişi Son test puanları	44	121,35	26,48			
Blok Tabanlı (KG)	Erişi Ön test puanları	43	77,27	26,92	42	-13,19	,000
	Erişi Son test puanları	43	106,44	27,22			

Tablo 15’te görüldüğü üzere deney grubu olan blok tabanlı robotik programlama grubunun eriş testi son test puan ortalaması ($\bar{x}=121,35$, $ss=26,48$) ön test puan ortalamasına göre ($\bar{x}=79,86$, $ss=20,88$) daha yüksek olup, bu fark istatistiksel olarak anlamlıdır ($t(43)=-14,82$, $p<0,05$). Kontrol grubu olan blok tabanlı programlama grubunun eriş testi son test puan ortalaması ($\bar{x}=106,44$, $ss=27,22$) ön test puan ortalamasına göre ($\bar{x}=77,27$, $ss=26,92$) daha yüksek olup bu fark istatistiksel olarak anlamlıdır ($t(42)=-3,19$, $p<0,05$).

İkinci alt probleme ait bulgular. Araştırmanın ikinci alt problemi “Deney ve kontrol gruplarının erişim testinden aldıkları son test puanları arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Deney ve kontrol gruplarında yer alan öğrencilerin erişim son testinden almış oldukları puanlar bağımsız örneklem t-testi ile karşılaştırılmıştır. Elde edilen veriler Tablo 16’da sunulmuştur.

Tablo 16

Grupların Erişim Puanlarına İlişkin (Son Test) Bağımsız Örneklem t-Testi Sonuçları

	Gruplar	N	\bar{x}	ss	sd	t	p
Erişim Son test Puanları	Blok Tabanlı (KG)	43	106,44	27,22	85	-2,59	,011
	Blok Tabanlı Robotik (DG)	44	121,35	26,48			

Bağımsız örneklem t-testi, blok tabanlı programlama eğitimi almış öğrenciler ile robot kullanarak blok tabanlı programlama eğitimi almış öğrencilerin programlama becerisi son test puanları arasındaki ilişkiyi ortaya koymak için kullanılmıştır. Tablo 16’da görüldüğü üzere yapılan analiz sonucunda, blok tabanlı ($\bar{x}=106,44$, $ss=27,22$) ve blok tabanlı robotik ($\bar{x}=121,35$, $ss=26,48$) grupları arasında istatistiksel olarak anlamlı bir fark bulunmuştur ($t(85)=-2,59$, $p<0,05$). Bu sonuçlara göre, blok tabanlı robotik programlama eğitimi alan öğrencilerin programlama başarısı blok tabanlı programlama eğitimi alan öğrencilere göre daha yüksektir.

Üçüncü alt probleme ait bulgular. Araştırmanın üçüncü alt problemi olan “Deney ve kontrol gruplarının erişim puanları arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Bu soruyu yanıtlamak için blok tabanlı programlama öğrenen grupla, blok tabanlı robotik programlamayı öğrenen gruptaki öğrencilerin ön test ve son test puanları hesaplanmıştır. Bu puanlar dikkate alınarak her iki grubun aritmetik ortalaması, standart sapması ve düzeltilmiş son test ortalaması bulunmuştur. Elde edilen değerler Tablo 17’de sunulmuştur.

Tablo 17

Grupların Ön Test Son Test Erişim Puan Ortalamaları, Standart Sapmaları ve Düzeltilmiş Son Test Ortalamaları

Gruplar	N	Ön Test Erişim Testi		Son Test Erişim Testi		Düzeltilmiş Son Test Puanları	
		\bar{x}	ss	\bar{x}	ss	\bar{x}	sh
Blok Tabanlı (KG)	43	77,27	26,92	106,44	27,22	107,60	2,52
Blok Tabanlı Robotik (DG)	44	79,86	20,88	121,35	26,48	120,22	2,50

Tablo 17’de görüldüğü üzere kontrol grubunun ön test erişim testindeki ortalaması 77,27, son test erişim testindeki ortalaması ise 106,44 iken deney grubunun ön test erişim testi ortalaması 79,86, son test erişim testi ortalaması ise 121,35’dir. Buna göre ön test erişim puanı kontrol edildiğinde grupların son test erişim testi ortalamalarında değişimler olduğu görülmektedir. Son test erişim testi düzeltilmiş ortalama puanları kontrol grubundaki öğrenciler için 107,60’a yükselirken; deney grubundaki öğrenciler için ise 120,22’ye düşmüştür. Deney ve kontrol grubundaki öğrencilerin son test erişim testi düzeltilmiş ortalama puanları arasındaki fark 12,62’dir. Bu farkın istatistiksel olarak anlamlı olup olmadığını anlamak için ANCOVA yapılmıştır. Analiz sonuçları Tablo 18’de sunulmuştur.

Tablo 18

Grupların Erişim Testinden Aldıkları Düzeltilmiş Son Test Puanlarıyla İlgili ANCOVA Tablosu

Varyans Kaynağı	Kareler		Kareler Ortalaması	F	p
	Toplamı	ss			
Düzeltilmiş Model	43131,365	2	21565,682	78,851	,000
Ön test	38296,535	1	38296,535	140,025	,000
Grup	3452,936	1	3452,936	12,625	,001
Hata	22973,860	84	273,498		
Düzeltilmiş Toplam	66105,224	87			

Tablo 18’deki bulgular, ortaokul öğrencilerinin Bilişim Teknolojileri ve Yazılım dersi, programlama öğretiminde blok tabanlı kodlama ortamı ile blok tabanlı robotik

kodlama ortamı erişim puanları arasında, ön test puanları kontrol altına alındığında istatistiksel olarak anlamlı bir fark olduğunu göstermektedir ($F(1,84)=12,625$, $p<0,05$). Diğer bir ifadeyle grupların programlama erişim son test puanları ön testten bağımsız olarak kullanılan öğrenme ortamına göre değişmektedir.

Problem Çözme Becerisi Puanlarına İlişkin Bulgular

Dördüncü alt probleme ait bulgular. Araştırmanın dördüncü alt problemi “Deney ve kontrol gruplarının problem çözme envanterinden aldıkları ön test ve son test puanları arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Blok tabanlı robotik programlama eğitimi ve blok tabanlı programlama eğitimi alan öğrencilerin erişim ön test puanları ile son test puanları arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını bulmak için programlama erişim ön test ve son test puanları üzerinde eşleştirilmiş örneklem t-testi (paired samples t-test) analizi yapılmıştır. Elde edilen bilgiler Tablo 19’da sunulmuştur.

Tablo 19

Grupların Çocuklar İçin Problem Çözme Envanterine-ÇPÇE İlişkin (Ön Test - Son Test) Eşleştirilmiş Örneklem t-Testi Sonuçları

Gruplar	Testler	N	\bar{x}	ss	sd	t	p
Blok Tabanlı Robotik (DG)	ÇPÇE Ön test puanları	44	87	12,58	43	-2,83	,007
	ÇPÇE Son test puanları	44	91,98	12,68			
Blok Tabanlı (KG)	ÇPÇE Ön test puanları	43	89,12	11,34	42	1,52	,136
	ÇPÇE Son test puanları	43	86,56	13,81			

Tablo 19’da görüldüğü üzere deney grubu olan blok tabanlı robotik programlama öğrencilerinin ÇPÇE son test puan ortalaması ($\bar{x}=91,98$, $ss=12,68$) ön test puan ortalamasına göre ($\bar{x}=87$, $ss=12,58$) daha yüksek olup, bu fark istatistiksel olarak anlamlıdır ($t(43)=-2,83$, $p<0,05$). Kontrol grubu olan blok tabanlı programlama öğrencilerinin ÇPÇE son test puan ortalaması ($\bar{x}=86,56$, $ss=13,81$) ön test puan ortalamasına göre ($\bar{x}=89,12$, $ss=11,34$) daha düşük olup bu fark istatistiksel olarak anlamlı değildir ($t(42)=1,52$, $p>0,05$).

Beşinci alt probleme ait bulgular. Araştırmanın beşinci alt problemi “Deney ve kontrol gruplarının problem çözme envanterinden aldıkları son test puanları arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Deney ve kontrol

gruplarında yer alan öğrencilerin “Çocuklar İçin Problem Çözme Envanteri” boyutlarının toplamının son testinden almış oldukları puanlar bağımsız örneklem t-testi ile karşılaştırılmıştır. Elde edilen veriler Tablo 20’de sunulmuştur.

Tablo 20

Grupların Problem Çözme Envanterine-ÇPÇE İlişkin (Son Test) Bağımsız Örneklem t-Testi Sonuçları

	Gruplar	N	\bar{x}	ss	sd	t	p
ÇPÇE Son Test Puanları	Blok Tabanlı (KG)	43	86,56	13,81			
	Blok Tabanlı Robotik (DG)	44	91,98	12,68	85	-1,91	,060

Bağımsız örneklem t-testi, blok tabanlı programlama eğitimi almış öğrenciler ile robot kullanarak blok tabanlı programlama eğitimi almış öğrencilerin problem çözme envanteri son test puanları arasındaki ilişkiyi ortaya koymak için kullanılmıştır. Tablo 20’de görüldüğü üzere yapılan analiz sonucunda blok tabanlı ($\bar{x}=86,56$, $ss=13,81$) ve blok tabanlı robotik ($\bar{x}=91,98$, $ss=12,68$) grupları arasında istatistiksel olarak anlamlı bir fark bulunmamıştır ($t(85)=-1,91$, $p>0,05$). Bu sonuçlara göre, blok tabanlı robotik programlama eğitimi alan öğrencilerin problem çözme beceri puanları, blok tabanlı programlama eğitimi alan öğrencilere göre daha yüksek çıkmasına rağmen bu fark istatistiksel olarak anlamlı değildir.

Altıncı alt probleme ait bulgular. Araştırmanın altıncı alt problemi olan “Deney ve kontrol gruplarının problem çözme envanterinden aldıkları puanlar arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Bu soruyu yanıtlamak için blok tabanlı programlama öğrenen grupla blok tabanlı programlamayı robotla birlikte öğrenen gruptaki öğrencilerin ön test ve son test puanları hesaplanmıştır. Bu puanlar dikkate alınarak her iki grubun aritmetik ortalaması, standart sapması ve düzeltilmiş son test ortalaması bulunmuştur. Elde edilen değerler Tablo 21’de sunulmuştur.

Tablo 21

Grupların Problem Çözme Envanteri (ÇPÇE) Ön Test - Son Test Puan Ortalamaları, Standart Sapmaları ve Düzeltilmiş Son Test Ortalamaları

Gruplar	N	ÇPÇE Envanteri Ön Testi		ÇPÇE Envanteri Son Testi		Düzeltilmiş Son Test Puanları	
		\bar{x}	ss	\bar{x}	ss	\bar{x}	sh
Blok Tabanlı (KG)	43	89,11	11,34	86,56	13,81	85,85	1,63
Blok Tabanlı Robotik (DG)	44	87,00	12,58	91,98	12,68	92,67	1,61

Tablo 21’de görüldüğü üzere kontrol grubunun problem çözme envanteri ön test ortalaması 89,11 son test ortalaması ise 86,56 iken deney grubunun problem çözme envanteri ön test ortalaması 87,00 son test ortalaması ise 91,98’dir. Buna göre ön test başarı puanı kontrol edildiğinde grupların son test başarı testi ortalamalarında değişimler olduğu görülmektedir. Son test başarı testi düzeltilmiş ortalama puanları kontrol grubundaki öğrenciler için 85,85’e düşerken deney grubundaki öğrenciler için ise 92,67’ye yükselmiştir. Deney ve kontrol grubundaki öğrencilerin son test başarı testi düzeltilmiş ortalama puanları arasındaki fark 6,82’dir. Bu farkın istatistiksel olarak anlamlı olup olmadığını anlamak için ANCOVA yapılmıştır. Elde edilen bilgiler Tablo 22’de sunulmuştur.

Tablo 22

Grupların Problem Çözme Envanterinden (ÇPÇE) Aldıkları Düzeltilmiş Son Test Toplam Puanlarıyla İlgili ANCOVA Tablosu

Varyans Kaynağı	Kareler		Kareler Ortalaması	F	p
	Toplamı	sd			
Düzeltilmiş Model	5998,93	2	2999,465	26,341	,000
Ön test	5360,28	1	5360,28	47,073	,000
Grup	1003,99	1	1003,99	8,817	,004
Hata	9565,30	84	113,873		
Düzeltilmiş Toplam	1556423,00	86			

Tablo 22’deki bulgular, Bilişim Teknolojileri ve Yazılım dersi, programlama öğretiminde blok tabanlı kodlama ortamında eğitim gören ortaokul öğrencilerinin

problem çözüme becerileri envanterinden aldıkları son test puanları ile blok tabanlı robotik kodlama ortamında eğitim gören ortaokul öğrencilerinin problem çözüme becerileri envanterinden aldıkları son test puanları arasında, ön test puanları kontrol altına alındığında istatistiksel olarak anlamlı bir fark olduğunu göstermektedir ($F(1,84)=8,817$, $p<0,05$). Diğer bir ifadeyle grupların problem çözüme envanteri toplam puanları ön testten bağımsız olarak kullanılan öğrenme ortamına göre değişmektedir.

Motivasyon Puanlarına İlişkin Bulgular

Yedinci alt probleme ait bulgular. Araştırmanın yedinci alt problemi “Deney ve kontrol gruplarının motivasyon ölçeğinden aldıkları son test puanları arasında anlamlı bir fark var mıdır?” şeklinde ifade edilmiştir. Bu soruya cevap bulmak için motivasyon ölçeği (CIS) son test puanları üzerinde bağımsız örneklem t-testi (independent samples t-test) analizi yapılmıştır. Elde edilen bulgular Tablo 23’de sunulmuştur.

Tablo 23

Grupların Motivasyon Ölçeği (CIS) Puanlarına İlişkin (Son Test) Bağımsız Örneklem t-Testi Sonuçları

	Gruplar	N	\bar{x}	ss	sd	t	p
Motivasyon (CIS)	Blok Tabanlı (KG)	43	118,49	16,64			
Son Test Puanları	Blok Tabanlı Robotik (DG)	44	124,45	20,25	85	-1,50	,137

Tablo 23’de görüldüğü üzere, bağımsız örneklem t-testi blok tabanlı programlama eğitimi almış öğrenciler ile robot kullanarak blok tabanlı programlama eğitimi almış öğrencilerin motivasyon (CIS) puanları arasındaki ilişkiyi ortaya koymak için kullanılmıştır. Blok tabanlı ($\bar{x}=118,49$, $ss=16,64$) ve blok tabanlı robotik ($\bar{x}=124,45$, $ss=20,25$) grupları arasında istatistiksel olarak anlamlı bir fark bulunmamıştır ($t(85)=-1,50$, $p>0,05$). Bu sonuca göre, blok tabanlı robotik programlama eğitimi alan öğrencilerin motivasyon puanları blok tabanlı programlama eğitimi alan öğrencilere göre daha yüksek olmasına rağmen bu fark istatistiksel olarak anlamlı değildir.

Bölüm 5

Sonuç, Tartışma ve Öneriler

Bu bölümde yapılan araştırmadan elde edilen bulgulardan hareketle ulaşılan sonuçların tartışılmasına ve önerilere yer verilmektedir.

Sonuç ve Tartışma

Günümüzde, robotlar pek çok alanda çok farklı görevler için kullanıldığı gibi eğitim alanında da kullanılmaktadır. Eğitim alanında fen ve teknoloji, matematik ve yabancı dil gibi derslerin öğretiminde, özel eğitime ihtiyaç duyan çocukların eğitiminde eğitim materyali olarak kullanılmaktadır. Robot geliştirme ve programlama süreçlerini kapsayan robotik alanı ise bir diğer kullanım alanıdır. Robotik alanında robotlar ilk olarak zaman gerektiren bir süreç olan inşa ve tasarım amacıyla kullanılmaya başlanmış, daha sonra programlanabilir hazır robot kitleri üretilmiştir. Buna bağlı olarak robotik programlama eğitiminde elektronik devre elemanlarının programlanması sonucu programlama ürününü fiziksel dünyada görmemizi sağlayan yazılımlar yaygın olarak kullanılmaktadır (Costelha & Neves, 2018).

Güncel araştırmalar, dünyanın birçok ülkesinde küçük yaştaki çocuklara yönelik programlama eğitimi ve robotik uygulamalarla gerçekleştirilen çalışmaların son yıllarda arttığını göstermektedir (Jung & Won, 2018; Kanbul & Uzunboylu, 2017; Major, Kyriacou, & Brereton, 2012). Programlama öğretimi kolaylaştırmak amacıyla görsel yazılım araçları olarak adlandırılan kod bloklarının sürüklenip bırakılması mantığına dayalı yazılımların kullanımı son yıllarda yaygınlaşmıştır. Yapılan çalışmalar özellikle Alice, code.org ve Scratch gibi blok tabanlı programlama dilleri ve Arduino ile öğrencilerin kendi robotlarını tasarlamaları, inşa etmeleri ve onları programlayarak harekete geçirmeleri için birbirine kolayca bağlanabilen parçalardan oluşan Lego tabanlı robotik ortamların yaygın olarak kullanıldığını göstermektedir (Costelha & Neves, 2018; Kalelioğlu, Gülbahar, & Doğan, 2018).

Problem çözme becerisi bireylerin sahip olması gereken en temel becerilerinden biridir (EARGED, 2011). Bu nedenle pek çok ülkenin müfredatı bu becerileri geliştirici nitelikte düzenlenmiştir. Programlama öğretimi ve robotik eğitiminin problem çözme becerilerini geliştirmede yaygın olarak kullanıldığı

görülmektedir (Çatlak vd., 2015; Yolcu & Demirer, 2017). Ayrıca robotların ilgi çekici olmaları nedeniyle programlama sürecinde öğrencilerin motivasyonlarına olumlu katkı sağladığı görülmektedir (Álvarez & Larrañaga, 2015; Çankaya vd., 2017; Ersoy vd., 2011).

Programlama öğretiminde kullanılan blok tabanlı görsel programlama yazılımları ve programlanabilir robotlarla ilgili birçok araştırma yapılmıştır. Bu görsel programa dillerinden Scratch yazılımıyla yapılan çalışmalar ağırlıklı olarak programlama derslerinde kullanım kolaylığı sebebiyle tercih edilirken Lego tabanlı robotik ortamlar ise inşa etme ve programlanabilir olmaları sebebiyle tercih edilmiştir. Programlama öğretiminin öğrencilerin programlama erişimine ve problem çözme becerilerine etkisi uygulama öncesi ve sonrası “Çocuklar İçin Problem Çözme Envanteri” ve alan uzmanı tarafından hazırlanan programlama testleri kullanılarak ölçülmüştür. Yapılan deneysel uygulamanın benzerlik göstermesinden dolayı öğrencilerin başarısını ölçmek için kullanılan ölçeklerde her iki gruba da aynı sorular sorulmuştur. Ayrıca uygulama sonrasında “Motivasyon Ölçeği (CIS)” kullanılarak robotların öğrenci motivasyonuna etkisi anlaşılmaya çalışılmıştır.

Araştırmada yer alacak öğrenci grupları 5. ve 6. sınıfların 4 farklı şubesinden rastgele seçilmişlerdir. Grupların denk olup olmadığını tespit etmek amacıyla yapılan analizlerde grupların eğitim öncesinde cinsiyet dağılımı, yaş ortalaması, programlama ön bilgisi ve problem çözme becerileri açısından denk olduğu görülmüştür. Yapılan deneysel uygulama boyunca Scratch programıyla yapılan etkinlikler benzer olup tek farklılık kullanılan robotik kit materyali ve robotik kiti kontrol eden farklı kod bloklarıdır. Bu bloklar diğer Scratch blokları ile uyumlu olduklarından diğer kod blokları içinde rahatlıkla kullanılmıştır.

Deney ve kontrol gruplarının ön test puanlarına kıyasla son test erişim puanlarında istatistiksel olarak anlamlı bir farklılaşmanın olup olmadığını ortaya koymak amacıyla yapılan analizler sonucunda hem deney grubunun hem de kontrol grubunun uygulama sonrası elde edilen programlama erişim testleri puanlarında ön test puanlarına kıyasla istatistiksel olarak anlamlı bir fark olduğu görülmüştür. Alan yazına bakıldığında Gülmez (2009), görselleştirme araçları kullanmanın programlama öğretiminde pozitif etki oluşturduğunu; Kaucic ve Asic (2011) oyun programlama üzerine yaptığı çalışmada çok kısa bir sürede bile çocukların programlama öğrenmeleri konusunda ilerleme kaydettiklerini belirtmiştir.

Programlama başarısı yönünden bu sonuçla benzerlik göstermeyen çalışmalar da bulunmaktadır. Kereki'nin (2008) öğrencilerin programlama deneyimlerini ve motivasyonlarını geliştirmek amacıyla iki farklı eğitim kurumunda bulunan öğrenciler ile Scratch-Visual Basic ve Scratch-Java ortamları kullanarak yaptığı çalışma sonuçlarına göre hem ön test ve son testler arasında hem de gruplar arasında istatistiksel olarak anlamlı bir fark bulunmadığı görülmüştür. Ayrıca bu durumla ilgili olumsuz görüşler de bulunmaktadır. Örneğin, Koorsse, Ciliers ve Calitz (2015) programlama zorluğuna etki eden faktörleri bir bilişim teknolojileri uzmanı bakış açısıyla belirlemek amacıyla programlamaya yardımcı araçları değerlendirmişlerdir. Çalışmada bilişim teknolojileri uzmanları programlama yapılarını kavramada programlamaya yardımcı araçları kullanımının yararlı olduğunu belirtse de uzmanların görüşlerinden bu araçların programlama yapılarını kavramaya ve programlama motivasyonlarını arttırmaya yönelik kesin bir kanıt ortaya çıkmamıştır. Ayrıca programlama eğitiminde robot kullanımının başarıyı arttırdığına dair benzer sonuçların bulunduğu birçok çalışma bulunmaktadır (Álvarez & Larrañaga, 2015; Çankaya vd., 2017; Durak vd., 2018). Bu sonuçlardan yola çıkarak hem blok tabanlı programlama ortamının hem de eğitsel robotik kitlerinin programlama eğitiminde kullanılmasının programlama başarısını olumlu yönde etkilediği söylenebilir.

Her iki grubun da deneysel uygulama öncesi ve deneysel uygulama sonrası erişim testi puanlarının anlamlı şekilde farklılaşması uygulanan deneysel yöntemin etkisi konusunda kesin bir yargıya ulaşmamıza engel olmaktadır. Bu nedenle ön test puanları kontrol altına alınarak erişim analizi yapılmış, iki grup arasında erişim testi puanları arasında deney grubu lehine anlamlı bir fark ortaya çıkmıştır. Benzer olarak alanyazında Korkmaz (2016) robot tasarım faaliyetlerini içeren eğitim uygulamalarıyla yapılan etkinliklerin, blok tabanlı oyun etkinliklerine kıyasla öğrencilerin programlama akademik başarılarına daha fazla olumlu katkı sağladığını belirtmiştir. Bu sonuçlara göre uygulanan robotik kodlama ortamının 5. ve 6. sınıf öğrencilerinin ders başarıları üzerinde olumlu bir etkiye sahip olduğu sonucuna varılabilir. Her iki gruba da uygulanan erişim testinin araştırmacı tarafından eğitimde kullanılan aynı içeriklerden yararlanılarak geliştirilmesi ve her iki grubun aynı programlama ortamını kullanması erişim puanlarındaki farklılaşmanın kaynağının içerikle ve programlama ortamıyla ilişkili olmadığını, uygulanan robotik kodlama yaklaşımlarından kaynaklandığını göstermektedir. Gomes ve Mendes'e (2007) göre

programlama ortamının soyut kavramlar içermesi öğrenci başarısını olumsuz yönde etkiler. Robotlar soyut kavramları somutlaştırarak öğrenmeyi kolaylaştırmaktadır. Bu nedenle robotların programlamada kullanılması öğrenmeyi daha etkin hale getirmektedir. Somut dönemden yeni çıkmakta olan çocuklar için kodladıkları nesnelerin gerçek dünyada dokunabildikleri nesnelere dönüşmesi çok önemlidir. Robotlar, eğitimde öğretilmesi gereken mesafe, hız, zaman, eğitim ve sıcaklık gibi soyut kavramları anlamayı somut hale getirecek otantik dünya etkinliklerini planlama olanağı sağlamaktadır. Yapılan deneysel uygulamada her iki grup da Scratch programının benzer özelliklerini ve benzer kod yapılarını kullandığından erişim farkının Scratch programından kaynaklandığı düşünülemez. Farklı olarak blok tabanlı kodlama yapan öğrenciler Scratch kod bloklarıyla 2 boyutlu x (yatay) ve y (dikey) koordinatı bulunan ekranda hareketlerini sergileyen kuklayı programlarken robotik kodlama etkinliklerinde ise Scratch kod blokları 3 boyutlu fiziksel dünyada bulunan robotlara etki etmiştir. Ayrıca sensörlerine yapılan fiziksel müdahale hem robotun kendi üzerinde hem de robotun ilişkilendirilebileceği bir Scratch kuklası üzerinde etki göstermektedir. Dolayısıyla meydana gelen bu farklılığın sebebinin robotların somutlaştırma etkisinden kaynaklandığını söyleyebiliriz. Alanyazında yapılan benzer araştırmalarda bu durumla ilgili farklı bulgulara rastlanmıştır. Şimşek (2018) programlama öğretiminde robotik etkinliklerinin etkisini ortaya çıkarmak amacıyla iki farklı ortam kullanarak yaptığı araştırma sonucu robotik kodlama ve blok tabanlı kodlama ile yapılan programlama başarısının benzer olduğunu belirlemiştir. Öğrencinin hem blok tabanlı kodlama eğitimi hem de blok tabanlı robotik kodlama eğitimi alarak programlama başarısını arttıracak sonucuna ulaşmıştır. Ayrıca Scratch ortamında programlama sonucu elde edilen ürün iki boyutlu ekranda canlandırıldığından dolayı 3 boyutlu fiziksel dünyaya göre daha soyut kaldığını, öğrencilerin koordinat sistemini algılamada zorluk çektiğini, robotik etkinliklerde ise daha çok teknik problemlerin sürece olumsuz yansıdığını belirtmiştir.

Deney ve kontrol gruplarının problem çözme becerileri ön test puanlarına kıyasla son test puanlarında istatistiksel olarak anlamlı bir farklılaşmanın olup olmadığını ortaya koymak amacıyla yapılan analizler sonucunda deney grubunun problem çözme puanlarında ön test puanlarına kıyasla istatistiksel olarak anlamlı fark olduğu görülürken, kontrol grubunun problem çözme puanlarında ön test

puanlarına kıyasla istatistiksel olarak anlamlı bir fark olmadığı görülmüştür. Alanyazında yapılan çalışmaların sonuçlarına genel olarak baktığımızda hem blok tabanlı programlama ortamlarının hem de robotik etkinliklerin problem çözme becerilerine olumlu katkı yaptığı görülmektedir (Çankaya vd., 2017; Lai & Lai, 2012; Nam vd., 2010; Shin & Park 2014). Kalelioğlu ve Gülbahar'ın (2014) yaptığı çalışmanın sonuçları beklenmeyen bu durumla benzerlik göstermektedir. Kalelioğlu ve Gülbahar'ın (2014) yaptığı çalışmada da blok tabanlı programlama ortamının öğrencilerin problem çözme becerilerini geliştirmesine rağmen anlamlı fark oluşturmadığı belirlenmiştir.

Araştırmaya dahil edilen deney ve kontrol gruplarının problem çözme becerilerine ait son test puanları arasında anlamlı bir farklılaşma olup olmadığını sınamak amacıyla yapılan analizler sonucunda gruplar arasında anlamlı bir fark bulunmamıştır. Her iki grubun da deneysel uygulama öncesi ve deneysel uygulama sonrası problem çözme becerileri puanlarının anlamlı şekilde farklılaşmaması uygulanan deneysel yöntemin etkililiği konusunda kesin bir yargıya ulaşmamıza engel olmaktadır. Bu nedenle ön test puanları kontrol altına alınarak problem çözme becerileri analizi yapılmış, iki grup arasında problem çözme becerileri puanları arasında deney grubu lehine anlamlı bir fark ortaya çıkmıştır. Görsel blok tabanlı programlama eğitiminin robotik etkinliklerle birlikte yapılmasının robotlardan kaynaklanan ek problemlerin ortaya çıkmasına ve buna bağlı olarak öğrencilerin daha fazla sorumluluk almalarına sebep olduğu düşünülebilir.

Grup çalışması problem çözme becerilerine olumlu katkı yapar (Ge & Land, 2003). Gruplar halinde yapılan çalışmada her grubun çalışma yapabileceği dört bilgisayar varken blok tabanlı robotik programlama yapan gruba ayrıca bir robot seti verilmiştir. Robotla çalışan öğrenciler kendi aralarında daha fazla işbirliği yaptıklarından karşılaştıkları problemleri daha kolay çözmüşlerdir. Bu da çalışmanın sonuçlarını etkileyen durumlardan biri olabilir.

Çalışma gruplarının motivasyon son test puanlarında uygulama yapılan kodlama ortamlarına göre farklılaşma olup olmadığını tespit etmek için yapılan istatistiksel analizler sonucunda deney grubunun puanları daha yüksek çıkmasına rağmen bu farklılığın istatistiksel olarak anlamlı olmadığı görülmüştür. Alanyazında bu sonuçla benzerlik gösteren çalışmalar mevcuttur. McGill (2012) eğitsel robot kullanmanın öğrencileri programlama öğrenmeye motive edip etmediğini bulmak

amacıyla yaptığı çalışmada robotların programlama öğrenmeye yönelik tutumları olumlu yönde arttırdığını ancak ilgi, güven ve memnuniyet üzerinde etkisi olmadığını belirlemiştir. Sonuçlar ayrıca öğrencilerin genel motivasyon düzeylerinin değişmediğini göstermiştir. Alanyazında yapılan birçok çalışmada ise Scratch programının motivasyonu arttırdığına dair bulgular elde edilmiştir (Erol, 2015; Ouahbi vd., 2015).

Deney grubunda kullanılan robotik kit çok fazla elektronik ve donanımsal bilgi gerektirmediğinden, ayrıca kullanımı son derece kolay olduğundan öğrenme süreçlerine olumsuz bir durum yansımamıştır. Öğrenciler akıllı tuğla, motor, sensör, led ve buton gibi kavramlarla ilk defa karşılaşmışlardır. Bu nedenle öğrencilere daha önceden tasarlanmış robotlar verilmiş olup öğrenciler sadece programlama üzerine yoğunlaşmışlardır. Bu durum ek zaman kaybının önüne geçilmesine neden olmuştur. Buna rağmen süreçte en sık karşılaşılan problemler şu şekildedir:

- Bluetooth bağlantısında kopmalar,
- Sensörlerin istenilen şekilde ya da hiçbir şekilde tepki vermemesi,
- Tasarlanan robotu bir araya getiren parçaların birbirinden ayrılması,
- Motor hızını ve motor dönüş yönünü deneme yanılma yöntemi ile ayarlama ihtiyacı,
- Enerji kaynağı olan pillerin zayıflaması sonucu motor hızında yavaşlama.

Ortaya çıkan bu tür problemlerin robot kullanarak programlama öğrenen öğrencilerde motivasyon düşüşüne neden olduğu düşünülmektedir. Alanyazında benzer olarak Lykke vd. (2014) robotların ilgi çekici ve motive edici olmasına rağmen kısıtlamalar nedeniyle sinir bozucu etkiye neden olduğunu belirtmiştir.

Çalışma öncesinde öğrencilerin programlama ve robotik deneyimi yoktur. Blok tabanlı programlama yazılımları kullanarak programlama öğretimi son yıllarda yaygınlaşmıştır. Blok tabanlı programlama yazılımları kullanarak robotları programlama ise daha yeni bir yöntemdir. Yeni uygulamayla programlama öğrenmeye teşvik edilen öğrencilerin motivasyon düzeylerindeki artışın nedeni olarak teknolojiye yeni bir etki düşünülebilir. Yapılan çalışmada robotik etkinliklerle programlama öğrenen öğrencilerin motivasyon düzeylerinde anlamlı

olmamakla birlikte bir artış sağlanmıştır. Araştırmadan elde edilen sonuçlar Tablo 24 ve Tablo 25'de araştırılan değişkenler ve uygulanabilirlik açısından özetlenmiştir.

Tablo 24

Araştırmadan Elde Edilen Sonuçların Araştırılan Değişkenler Açısından İncelenmesi

Değişkenler		Blok Tabanlı Kodlama (KG)	Blok Tabanlı Robotik Kodlama (DG)
Programlama Erişi	Ön Test - Son Test t-Testi	Anlamli olarak artmiştir.	Anlamli olarak artmiştir.
	Son Test t-Testi	Deney grubu lehine anlamli farklılık ortaya çıkmıştır.	
	ANCOVA	Deney grubu lehine anlamli farklılık ortaya çıkmıştır.	
Problem Çözme Becerisi	Ön Test - Son Test t-Testi	Azalma olmakla birlikte anlamli değildir.	Anlamli olarak artmıştır.
	Son Test t-Testi	Anlamli farklılık ortaya çıkmamıştır.	
	ANCOVA	Deney grubu lehine anlamli farklılık ortaya çıkmıştır.	
Motivasyon	Son Test t-Testi	Deney grubunun puanları yüksek olmasına rağmen anlamli farklılık ortaya çıkmamıştır.	

Tablo 24'de görüldüğü üzere yapılan çalışma ölçülen değişkenler açısından incelendiğinde, deneysel uygulama sonunda her iki gruptaki öğrencilerin programlama erişimi puanlarında anlamli farklılık ortaya çıkmasına rağmen problem çözme becerilerinde deney grubundaki öğrencilerin problem çözme puanlarında anlamli farklılık ortaya çıkmış, kontrol grubundaki öğrencilerin problem çözme puanlarında anlamli farklılık bulunmamıştır. Araştırmaya katılan çalışma grupları arasında ise deneysel süreç sonunda, problem çözme becerileri ve programlama erişimi puanları arasında deney grubu lehine bir artış olmuştur. Her iki grubun motivasyon puanları karşılaştırıldığında deney grubunun puanı daha yüksek olmasına rağmen bunun anlamli olmadığı görülmüştür.

Tablo 25

Araştırma Sonuçlarının Maliyet ve Uygulanabilirlik Açısından İncelenmesi

	Blok Tabanlı Kodlama (KG)	Blok Tabanlı Robotik Kodlama (DG)
Maliyet	<ul style="list-style-type: none"> • Bilgisayar donanımı • Programlama yazılımı 	<ul style="list-style-type: none"> • Özel sınıf ortamı • Bilgisayar donanımı • Programlama yazılımı • Robot kiti
Uygulama Sorunları	<ul style="list-style-type: none"> • - 	<ul style="list-style-type: none"> • Robot kiti teknik sorunları • Uygulamanın zaman alması

Tablo 25’de görüldüğü üzere yapılan araştırma maliyet açısından incelendiğinde blok tabanlı kodlama etkinliklerinin başarıya ulaşması için sadece bilgisayar donanımına ihtiyaç varken, robotik etkinliklerinin başarıya ulaşması için çalışma yapılacak sınıf ortamı, bilgisayar donanımı ve kullanılacak robot kitine ihtiyaç olduğu görülmüştür. Yapılan çalışma uygulama sürecinde karşılaşılan sorunlar açısından incelendiğinde, blok tabanlı robotik programlama etkinliklerinde bir takım aksaklıkların meydana geldiği görülmüştür.

Tüm bu sonuçlar bağlamında gerçekleştirilen bu çalışma ile blok tabanlı robotik kodlama kullanımının uygulanabilirlik yönünden alanyazındaki açığı kapatmaya katkı sağlayacağı düşünülmektedir. Çalışma sonuçlarının özellikle ülke genelindeki politika yapıcılar, yasa koyucular, bireysel çalışan eğitimciler ve araştırmacılar için rehber niteliğinde olacağı, ayrıca geleceğe dönük araştırmalar için ışık tutacağı öngörülmektedir.

Öneriler

Bu bölümde robotik destekli kodlama eğitimi planlayan öğretmenler ve araştırmacılar için önerilere yer verilmektedir.

Programlamada öğretilen tüm kavramları robot kiti üzerinden göstermek oldukça zordur. Uygulama zorluğu ve sarf edilen çaba göz önüne alındığında robot kitleri programlama eğitiminde kısıtlı olarak kullanılabilir.

Fiziksel robot kitlerinde bulunan sensör görevini gören algılama blokları Scratch ortamında da bulunmaktadır. Etkili bir eğitim-öğretim sürecinde zamanın verimli kullanımı göz önüne alındığında Scratch programı robot kitine ihtiyaç olmadan programlama öğretiminde kullanılabilir. Etkili bir uygulama için robotik uygulamalar önceden test edilmeli ve robotların teknik kontrolleri yapılmalıdır.

Öğrenciler programlamadaki standart girdi - çıktı kavramlarını algılamada zorlanmışlardır. Etkili bir uygulama için bu kavramlar hakkında bilgi vermeden uygulamaya geçilmemelidir.

Etkili bir programlama öğrenme sürecinde öğrencilerin matematiksel becerileri önemlidir. Bu nedenle öncelikle matematik müfredat programı incelenerek programlama eğitimi içeriğinin matematik yeteneklerine uygun olarak seçilmesi verimi arttıracaktır.

Çalışma kapsamında geliştirilen erişim testinin içeriği algoritma stratejisi geliştirme, Scratch programı kullanımı ve temel programlama yapılarını içeren sorulardan oluşmaktadır. İleride yapılacak çalışmalarda sadece temel programlama yapılarını içeren sorulardan oluşan bir erişim testi geliştirilip gruplar arasında karşılaştırmalar yapılabilir.

Yapılan çalışma kapsamında robotların programlamaya etkisi problem çözme envanteri ve motivasyon ölçeğinin alt faktörlerinin toplam puanları üzerinden değerlendirilmiştir. Yeni yapılacak çalışmalarda alt faktörlerin bağımsız olarak değerlendirildiği bir araştırma gerçekleştirilebilir.

Yapılan çalışma robotik kitlerin öğrencilerin yaş seviyesi, metin tabanlı ve görsel tabanlı programlama desteği, bağlantı türü, sensör sayısı ve öğretime verdiği katkı göz önünde bulundurularak gerçekleştirilmiştir. Yeni yapılacak çalışmalarda farklı özelliklere sahip robot kitleri seçilerek farklı değişkenlerin değerlendirildiği araştırmalar gerçekleştirilebilir.

Yapılan çalışmada özel sınıf ortamı gerektiren maliyetleri yüksek robotik kitlerin kullanılması zorunlu olarak öğrencilerin gruplar halinde çalışmasına sebep olmuştur. Bu nedenle gelecek çalışmalarda sanal robotik ortamları kullanarak bireysel programlama öğrenen öğrenciler ile fiziksel robot ile grup çalışması yaparak programlama öğrenen öğrencilerin dahil olduğu bir çalışma gerçekleştirilebilir.

Yapılan çalışma fırsatlara erişimi daha az olan gruplar ile yapılmasına rağmen öğrencilerin bilgisayar sahipliği, İnternet erişimi, kendi çalışma odası olması, bilgisayar oyunu oynama alışkanlıkları ve cinsiyet faktörleri araştırma kapsamına dâhil edilmemiştir. Bütün bu faktörler ileriye dönük çalışmalarda süreç içerisinde araştırılması gereken önemli bir konu olarak görüldüğünden yapılacak yeni çalışmalara dahil edilebilir.

Yapılan çalışmada kullanılan robotik eğitim içeriği bir uzman tarafından incelenmiştir. Dolayısıyla uygulanan öğretim programının geçerlemesi sınınamamıştır. İleride yapılacak çalışmalarda aynı robotik eğitim içeriği üzerinden öğretim programının geçerlemesi yapılabilir. Böylece eğitim içeriğinin kazanımlara ve öğrenci sınıf seviyesine uygunluğun ölçülmesi mümkün olabilir.

Kaynaklar

- Acar, S. (2009). *Web destekli performans tabanlı öğrenmede ARCS motivasyon stratejilerinin öğrencilerin akademik başarılarına, motivasyonlarına ve tutumlarına etkisi* (Doktora Tezi). Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Akbaba, S. (2006). Eğitimde motivasyon. *Atatürk Üniversitesi Kazım Karabekir Eğitim Fakültesi Dergisi*, 13, 343-361.
- Akçay, A. ve Çoklar, A. N. (2016). Bilişsel becerilerin gelişimine yönelik bir öneri: Programlama eğitimi. A. İşman, H. F. Odabaşı ve B. Akkoyunlu (Eds.), *Eğitim Teknolojileri Okumaları 2016* 121-139. Ankara: The Turkish Online Journal of Educational Technology (TOJET).
- Alkan, A. (2018). Özel yetenekli öğrencilerin programlama dili öğretiminde kodu game lab yazılımının problem çözme becerileri düzeyine etkisi. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi*, 50, 480-493.
- Allison, I. K., Orton, P., & Powell, H. (2002). A virtual learning environment for introductory programming. In *Proceedings of the 3rd Conference of the LTSN-ICS.. Higher Education Academy Subject Centre for Information and Computer Sciences (HEA-ICS)*.
- Álvarez, A., & Larrañaga, M. (2016). Experiences incorporating lego mindstorms robots in the basic programming syllabus: lessons learned. *Journal of Intelligent & Robotic Systems*, 81(1), 117-129.
- Anastasiadou, S. D., & Karakos, A. S. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *The International Journal of Technology, Knowledge and Society*, 7(1), 37-51.
- Anderson, J. R. (1980). *Cognitive psychology and its implication*. San Fransisco: Freeman.
- Arabacıoğlu, C., Bülbül, H. & Filiz, A. (2007). *Bilgisayar programlama öğretiminde yeni bir yaklaşım*. Akademik Bilişim 2007 Konferansı, Dumlupınar Üniversitesi, Kütahya.

- Aslanyürek, M. (2007). Programlamaya Giriş ve Algoritmalar Ders Notları. [https://staff.emu.edu.tr/hasanoylum/Documents/courses/BTEP101/ALG\(Belge\).pdf](https://staff.emu.edu.tr/hasanoylum/Documents/courses/BTEP101/ALG(Belge).pdf) adresinden erişilmiştir.
- Aytekin, A., Çakır, F. S., Yücel, Y. B., & Kulaöz, İ. (2018). Geleceğe Yön Veren Kodlama Bilimi ve Kodlama Öğrenmede Kullanılabilecek Bazı Yöntemler. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi*, 5(5), 24-41.
- Baykul, Y. (2009). *İlköğretimde matematik öğretimi: 6.-8. sınıflar için*. Ankara: Pegem Akademi Yayıncılık.
- Bayman, P., & Mayer, R. (1988). Using conceptual models to teach Basic computer programming. *Journal of Educational Psychology*, 80(3), 291-298.
- Baz, F. Ç. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Current Research in Education*, 4(1), 36-47.
- Begosso, L. C., & Da Silva, P. R. (2013). Teaching computer programming: A practical review. In *2013 IEEE Frontiers in Education Conference (FIE)* (pp. 508-510). IEEE.
- Benzer, A. İ., & Erümit, A. K. (2017). The Analysis of the Graduate Theses Related to Programming Instruction. *Öğretim Teknolojileri & Öğretmen Eğitimi Dergisi*, 6(3), 99-110.
- Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York, NY: Teachers College Press.
- Bilen, M. (2006). *Plandan Uygulamaya Öğretim*. Ankara: Anı Yayıncılık.
- Bingham, A. (2004). *Çocuklarda problem çözme yeteneklerinin geliştirilmesi* (F. Oğuzkan, Çev.). İstanbul: Milli Eğitim Basımevi.
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer aided instruction as a vehicle for problem solving: Scratch boards in the middle years classroom. *ASEE Annual Conference and Exposition, Conference Proceedings*.
- Byrne, P., & Lyons, G. (2001). *The Effect of Student Attributes on Success in Programming*. Proceedings of ITICSE, pp. 49-52.

- Bruinsma, M. (2004). Motivation, Cognitive Processing and Achievement in Higher Education. *Learning & Instruction*, 14(6), 549-568.
- Büyüköztürk, Ş. (2004). *Veri analizi el kitabı*. Ankara: Pegem A Yayıncılık.
- Calao L. A., Moreno-León, J., Correa, H. E., Robles, G. (2015). Developing Mathematical thinking with Scratch. An experiment with 6th grade students. In: G. Conole, T. Klobučar, C. Rensing, J. Konert, E. Lavoué (Eds). *Design for teaching and learning in a networked world*, pp. 17-27. Lecture Notes in Computer Science, vol 9307. Springer, Cham.
- Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Carbone, A., & Sheard, J. (2002). A studio-based teaching and learning model in IT: what do first year students think? *SIGCSE Bull.*, 34(3), 213-217.
- Casey, P. J., (1997). Computer programming: A medium for teaching problem solving, *Computers in the Schools*, 13(1-2), pp. 41-51.
- Coşar, M. (2013). *Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri* (Doktora Tezi). Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Costelha, H., & Neves, C. (2018). Technical database on robotics-based educational platforms for K-12 students. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (pp. 167-172). IEEE.
- Credé, M. ve Phillips, L. A. (2011). A meta-analytic review of the Motivated Strategies for Learning Questionnaire. *Learning and Individual Differences*, 21(4), 337-346.
- Cüceloğlu, D. (2006). *İnsan davranışı*. İstanbul: Remzi Kitapevi.
- Çamoğlu, D. (2015). *Bilgisayar kontrollü robotik*. İstanbul: Dikeyksen Yayın Dağıtım.

- Çankaya, S., Durak, G., & Yünkül, E. (2017). Robotlarla programlama eğitimi: öğrencilerin deneyimlerinin ve görüşlerinin incelenmesi. *Turkish Online Journal of Qualitative Inquiry*, 8(4), 428-445.
- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch Yazılımı İle Programlama Öğretiminin Durumu: Bir Doküman İnceleme Çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3).
- Çetin, E. (2012). *Bilgisayar programlama eğitiminin çocukların problem çözme beceri üzerine etkisi* (Yüksek Lisans Tezi). Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- D'Zurilla T. J., & Goldfried M. R. (1971). Problem solving and behavior modification. *Journal of Abnormal Psychology*, 78(1), 107-126.
- Dalton, D. W., & Goodrum, D. A. (1991). The effects of computer programming on problem-solving skills and attitudes. *Journal of Educational Computing Research*, 7(4), 483-506.
- Demirer, V., & Sak, N (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Derus, S. R. M., & Ali, A. Z. M. (2012). Difficulties in learning programming: Views of students. *1st International Conference on Current Issues in Education*, 15-16 September 2012, Yogyakarta, Indonesia.
- Dewey, J. (1933). *How we think: A restatement of the relation of reflective thinking to the educative process*. Boston: DC Heath and Company
- Duncan, E. (2002). Making the analogy: Alternative delivery techniques for first year programming courses. In J. Kuljis, L. Baldwin ve R. Scoble (Eds). *Proceedings from the 14th workshop of the psychology of programming interest group* (pp. 89-99). Brunel University.
- Durak, H. Y., Yılmaz, F. G. K., & Yılmaz, R. (2018). Robot tasarımı etkinliklerinin programlama öğretiminde kullanılmasıyla ilgili ortaokul öğrencilerinin görüşlerinin incelenmesi. *Ege Eğitim Teknolojileri Dergisi*, 2(2), 32-43.
- EARGED (2011). *MEB 21. Yüzyıl Öğrenci Profili*. Ankara: MEB. https://www.meb.gov.tr/earged/earged/21.%20yy_og_pro.pdf adresinden erişilmiştir.

- Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. In D. Gibson & B. Dodge (Ed.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2010* (pp. 4006-4014). Chesapeake, VA: AACE.
- Ekici, G., Abide, Ö. F., Canbolat, Y., & Öztürk, A. (2017). 21. yüzyıl becerilerine ait veri kaynaklarının analizi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 6(1).
- Ekiz, H., Vatansever, F., Zengin, A., & Demir, Z. (2000). Hesaplamanın Tarihi ve Bilgisayarların Gelişimi. *Sakarya University Journal of Science*, 4(1-2), 73-82.
- Ercan, L. (2004). *Motivasyon. Sınıf Yönetiminde Yeni Yaklaşımlar* (Edit:L. Küçükahmet). Ankara: Nobel Yayınevi.
- Erdem, E. (2012). *Blok tabanlı ortamlarda programlama öğretimi sürecinde farklı öğretim stratejilerinin çeşitli değişkenler açısından incelenmesi* (Yüksek Lisans Tezi). Başkent Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Erol, O. (2015). *Scratch ile programlama öğretiminin bilişim teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisi* (Doktora Tezi). Anadolu Üniversitesi Eğitim Bilimleri Enstitüsü, Eskişehir.
- Erol, O., & Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Computers in Human Behavior*, 77, 11-19.
- Ersoy, E., & Ersoy, E. (2018). *Fortran Programlama Diline Giriş*. <http://dosyalar.ercanersoy.net/yayinlar/fortran-programlama-diline-giris.pdf> adresinden erişilmiştir.
- Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). *Programlama dilleri öğretimine bir model önerisi: robot programlama*. Akademik Bilişim 2011 Konferansı, Malatya: İnönü Üniversitesi.
- Ertürk, S. (1984). *Eğitimde Program Geliştirme*. Ankara: Yelkentepe Yayınları.
- Esteves, M., & Mendes, A. J. (2004, 20-23 Oct. 2004). *A simulation tool to help learning of object oriented programming basics*. Paper presented at the 34th Annual Frontiers in Education, 2004. FIE 2004.

- Eymur, G., & Geban, Ö. (2011). Kimya öğretmeni adaylarının motivasyon ve akademik başarıları arasındaki ilişkinin incelenmesi. *Eğitim ve Bilim*, 36(161).
- Farkas, D., & Murthy, N. (2005). Attitudes toward computers, the introductory course and recruiting new majors: Preliminary results. In *17th Workshop of the Psychology of Programming Interest Group, Sussex University* (pp. 268-277).
- Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. In *ACM SIGCSE Bulletin*, 41(3), 258-262.
- Fidan, U., & Yalçın, Y. (2012). Robot Eğitim Seti Lego Nxt. *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, 12(1), 1-8.
- Fuegi, J., & Francis, J. (2003). Lovelace & Babbage and the creation of the 1843'notes'. *IEEE Annals of the History of Computing*, 25(4), 16-26.
- Gaudiello, I., & Zibetti, E. (2016). *Learning robotics, with robotics, by robotics: Educational robotics*. Hoboken, NJ, USA: John Wiley & Sons.
- Ge, X. & Land, S. M., (2003). Scaffolding students' problem-solving processes in an illstructured task using question prompts and peer interactions. *Educational Technology Research and Development*, 51(1), 21-38.
- Gelbal, S. (1991). Problem çözme. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 6(6).
- Genç, Z. & Karakuş, S. (2011). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı*. 5th International Computer & Instructional 71 Technologies Symposium (ICITS), Elazığ, 22-24 Ekim 2011.
- Gomes, A., & Mendes, A. (2007, September). *Learning to program – difficulties and solutions*. Paper presented at International Conference on Engineering Education (ICEE), Coimbra, Portugal.
- Gomes, A.J., Mendes, A. J., & Marcelino, M. J. (2015). Computer Science Education Research: An Overview and Some Proposals. In *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming*, 1-29. IGI Global.

- Gray, J., Boyle, T., & Smith, C. (1998). A constructivist learning environment implemented in Java. *SIGCSE Bull.*, 30(3), 94-97.
- Guzdial & du Boulay (2019) The History of Computing Education Research. In S. A. Fincher & A. V. Robins (Eds.) *The Cambridge Handbook of Computing Education Research*. Cambridge, UK: Cambridge University Press.
- Güçlü, N. (2003). Lise Müdürlerinin Problem Çözme Becerileri. *Milli Eğitim Dergisi*, 160, 272-300.
- Gülbahar Güven, Y. (Ed.) (2018). *5. sınıf Bilişim Teknolojileri ve Yazılım Dersi Öğretmen Rehberi*. Ankara: Milli Eğitim Bakanlığı.
- Gülmez, I. (2009). *Programlama öğretiminde görselleştirme araçlarının kullanımının öğrenci başarı ve motivasyonuna etkisi* (Yüksek Lisans Tezi). Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Gürol, M. & Demirli, C. (2006, Nisan). *E-portfolio sürecinde öğrenci motivasyonu*. VI Uluslararası Eğitim Teknolojileri Konferansında sunulan sözlü bildiri, Gazimağusa.
- Herbert, C. W. (2011). *An introduction to programming using Alice 2.2 Second Edition*. USA: Course Technology, Cengage Learning.
- Hilal, A. R., Wagdy, K. M., & Khamis, A. M. (2007). A survey on commercial starter kits for building real robots. In *Proceedings of the International Conference on Electrical Engineering, CEE07, Coimbra, Portugal*.
- Holland, E., & Minnick, C. (2015). *JavaScript For Kids For Dummies*. Hoboken, New Jersey: John Wiley & Sons.
- Holland, E., & Minnick, C. (2016). *Adventures in Coding*. Indianapolis, Indiana: John Wiley & Sons.
- Hongwarittorn, N., & Krairit, D. (2010). Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming. In *The spring 8th International conference on Computing, Communication and Control Technologies*, pp. 6-9.

- Hooshyar, D., Ahmad, R. B., Shamsirband, S., Yousefi, M., & Horng, S. J. (2015). A flowchart-based programming environment for improving problem solving skills of Cs minors in computer programming. *The Asian International Journal of Life Sciences*, 24(2), 629-646.
- Huet, I., Tavares, J., Weir, G., Ferguson, J., & Wilson, J. (2003). *Co-operation in education: the teaching and learning of programming at the Universities of Aveiro and Strathclyde*. Paper presented at the *ICHED Conference*, Aveiro, Portugal.
- Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *TOJET: The Turkish Online Journal of Educational Technology*, 9(2).
- Jenkins, T. (2001). The motivation of students of programming. In *ACM SIGCSE Bulletin*, 33(3), pp. 53-56. ACM.
- Jenkins, T. (2002). *On the difficulty of learning to program*. Paper presented at 3rd annual Conference of LTSN-ICS, Loughbrorough University, Leicestershire, UK.
- Jiau, H. C., Chen, J. C., & Ssu, K. F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *IEEE Transactions on Education*, 52(4), 555-562.
- Jung, S., & Won, E. S. (2018). Systematic review of research trends in robotics education for young children. *Sustainability*, 10(4), 905.
- Kalaycı, N. (2001). *Sosyal bilgilerde problem çözme ve uygulamalar*. Ankara: Gazi Kitabevi.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners" perspective. *Informatics in Education-An International Journal*, 13(1), 33-50.
- Kalelioğlu, F., Gülbahar, Y., & Doğan, D. (2018). Teaching How to Think Like a Programmer: Emerging Insights. In *Teaching Computational Thinking in Primary Education* (pp. 18-35). IGI Global.

- Kanbul, S., & Uzunboylu, H. (2017). Importance of Coding Education and Robotic Applications for Achieving 21st-Century Skills in North Cyprus. *International Journal of Emerging Technologies in Learning*, 12(1).
- Karabak, D., & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 21(2-3), 163-169.
- Karaçay, T. (2006). *Olasılığın matematiksel temelleri ve yeni arayışlar*. Mantık, Matematik ve Felsefe IV. Ulusal Sempozyumu, İzmir, Türkiye.
- Karasar, N. (2012). *Bilimsel araştırma yöntemi*. Ankara: Nobel.
- Kaucic, B. & Asic, T. (2011). Improving introductory programming with Scratch? In *Proceeding of the 34th MIPRO International Conference*, pp. 1095–1100, Opatija, Croatia.
- Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses. *Computer Science Education*, 10(2), 109-128.
- Keller, J. M. & Subhiyah, R. (1987). *Manual for course interest survey (CIS)*. Tallahassee, FL, USA.
- Keller, J. M. (1983). Motivational Design of Instruction: In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status*, 384-434.
- Keller, J. M. (1987b). Strategies for stimulating the motivation to learn. *Performance and Instruction*, 26(8), 1-7.
- Keller, J. M. (2006). *Development of two measures of learner motivation :CIS and IMMS*. Unpublished manuscript. Florida State University.
- Keller, J.M. (1987a). Development and use of ARCS Model in instructional design. *Journal of Instructional Development*, 10(3), 1-10.
- Kereki, I. F. (2008). *Scratch: Applications in computer science 1*. 38th Annual Frontiers in Education Conference. Saratoga Springs, NY, October 22 - 25, 2008.

- Kert, S. B., & Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. *The First International Congress of Educational Research*, Çanakkale, Turkey.
- Kesici, T., & Kocabaş, Z. (2007). *Bilgisayar 2 Ders Kitabı*. Ankara: Semih Ofset.
- Kim, S., Chung, K., & Yu, H. (2013). Enhancing Digital Fluency through a Training Program for Creative Problem Solving Using Computer Programming. *The Journal of Creative Behavior*, 47(3), 171-199. doi:10.1002/jocb.30
- Kinnunen, P., & Malmi, L. (2008). *CS minors in a CS1 course*. Paper presented at the Proceedings of the Fourth International Workshop on Computing Education Research.
- Kneeland, S. (2001). *Problem Çözme* (N. Kalaycı, Çev.). Ankara: Gazi Kitabevi.
- Koçel, T. (2005). *İşletme Yöneticiliği*. İstanbul: Arıkan Yayınları.
- Kölling, M., & Rosenberg, J. (2001). Guidelines for teaching object orientation with Java. *SIGCSE Bull.*, 33(3), 33-36.
- Komis, V., & Misirli, A. (2016). The environments of educational robotics in Early Childhood Education: towards a didactical analysis. *Educational Journal of the University of Patras UNESCO Chair* 3(2), 238–246.
- Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, 82, 162-178.
- Kordaki, M. (2012). Diverse categories of programming learning activities could be performed within Scratch. *Procedia-Social and Behavioral Sciences*, 46, 1162-1166.
- Korkmaz, Ö. (2016). The Effect of Lego Mindstorms Ev3 Based Design Activities on Students' Attitudes towards Learning Computer Programming, Self-Efficacy Beliefs and Levels of Academic Achievement. *Online Submission*, 4(4), 994-1007.
- Korkmaz, Ö., & Altun, H. (2013). Engineering and ceit student's attitude towards learning computer programming. *The Journal of Academic Social Science Studies International Journal of Social Science*, 6(2), 1169-1185.

- Korkut, F. (2002). Lise öğrencilerinin problem çözme becerileri. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 22, 177-184.
- Kukul, V., & Gökçearslan, Ş. (2014). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. 8. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, (s. 58-63). Edirne.
- Kurebayashi, S., Kamada, T., & Kanemune, S. (2006). Learning computer programming with autonomous robots. In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives* (pp. 138-149). Springer, Berlin, Heidelberg.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *SIGCSE Bull.*, 37(3), 14-18.
- Lai, A. F., & Yang, S. M. (2011). The learning effect of visualized programming learning on 6th graders' problem solving and logical reasoning abilities. In *2011 International Conference on Electrical and Control Engineering* (pp. 6940-6944). IEEE.
- Lai, C., & Lai, M. (2012, 4-6 June 2012). *Using Computer Programming to Enhance Science Learning for 5th Graders in Taipei*. Paper presented at the 2012 International Symposium on Computer, Consumer and Control.
- Lecky-Thompson, G. W. (2007). *Just Enough C/C++ Programming*. Cengage Learning PTR.
- Lego Education (2019). LEGO® Education WeDo 2.0 Core Set. Retrieved from <https://education.lego.com/>
- Lewis, C. M., & Shah, N. (2012). *Building upon and enriching grade four mathematics standards with programming curriculum*. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA.
- Liang, H. N., Fleming, C., Man, K. L. & Tillo, T. (2013). A first introduction to programming for first-year students at a Chinese university using LEGO MindStorms. *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering*, 233-238. Retrieved from <https://doi.org/10.1109/TALE.2013.6654435>

- Linn M. C. & Clancy M. J. (1992). The case for case studies of programming problems. *Communications of the ACM*, 35(3), 121-132.
- Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., . . . Thomas, L. (2004). A multi-national study of reading and tracing skills in novice programmers. *SIGCSE Bull.*, 36(4), 119-150.
- Liu, A., Newsom, J., Schunn, C., & Shoop, R. (2013). Students learn programming faster through robotic simulation. *Tech Directions*, 72(8), 16.
- Lui, K. M., & Chan, K. C. (2003). *When does a pair outperform two individuals?* Paper presented at the International Conference on Extreme Programming and Agile Processes in Software Engineering.
- Lykke, M., Coto, M., Mora, S., Vandell, N., & Jantzen, C. (2014, April). Motivating programming students by problem based learning and LEGO robots. In *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 544-555). IEEE.
- Maheshwari, P. (1997, July). Teaching programming paradigms and languages for qualitative learning. In *Proceedings of the 2nd Australasian conference on Computer science education* (pp. 32-39). ACM.
- Major, L., Kyriacou, T., & Brereton, O. P. (2012). Systematic literature review: teaching novices programming using robots. *IET software*, 6(6), 502-513.
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM Sigcse Bulletin*, 39(1), 223-227.
- Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, 16(3), 211-227. Retrieved from <https://doi.org/10.1080/08993400600912384>
- Marji, M. (2014). *Learn to program with Scratch: A visual introduction to programming with games, art, science, and math*. San Francisco: No Starch Press.
- McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *SIGCSE Bull.*, 34(1), 38-42.

- McGill, M. M. (2012). Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education (TOCE)*, 12(1), 1-32.
- McGill, T. J., & Volet, S. E. (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education*, 29(3), 276-197.
- McKenzie, K., & Schweitzer, R. (2001). Who succeeds at university? Factors predicting academic performance in first year Australian university students. *Higher Education Research and Development*, 20(1), 21-33.
- McNerney, T. S. (2004). From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal and Ubiquitous Computing*, 8(5), 326-337.
- Meisalo, V., Suhonen, J., Torvinen, S., & Sutinen, E. (2002). Formative evaluation scheme for a web-based course design. *SIGCSE Bull.*, 34(3), 130-134.
- Miller, R. B., Kelly, G. N., & Kelly, J. T. (1988). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology*, 13(4), 348-357.
- Milli Eğitim Bakanlığı (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5. ve 6. sınıflar)*, Ankara: Milli Eğitim Bakanlığı. <http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=374> adresinden erişilmiştir.
- Morgan, C. T. (1999). *Psikolojiye giriş (13. baskı)* (H. Arıcı & S. Karakaş Çev.). Ankara: Meteksan Yayıncılık.
- Nam, D., Kim, Y., & Lee, T. (2010). The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill. In *ICCE2010* (pp. 723-727).
- Nosek, J. T. (1998). The case for collaborative programming. *Communications of the ACM*, 41(3), 105-108.
- Numanoğlu, M., & Keser, H. (2017). Programlama öğretiminde robot kullanımı-mbot örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*, 6(2), 497.

- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, 1479-1482.
- Ozoran, D., Çağıltay, N., & Topalli, D. (2012, November). *Using Scratch in introduction to programming course for engineering students*. Paper presented at 2nd International Engineering Education Conference (IEEC2012), Antalya.
- Öğülmüş (2001) *Kişilerarası Sorun Çözme Becerileri ve Eğitimi*. Ankara: Nobel
- Özfırat, M. (2009). Robotik Sistemler ve Madencilikte Kullanımının Araştırılması. *TÜBAV Bilim Dergisi*, 2(4), 412-425.
- Özyurt, Ö., & Özyurt, H. (2015). Bilgisayar Programcılığı Öğrencilerinin Programlamaya Karşı Tutum ve Programlama Öz Yeterliklerinin Belirlenmesine Yönelik Bir Çalışma. *Eğitimde Kuram ve Uygulama*, 11(1), 51-67.
- Pine, C. (2009). *Learn to Program*. Pragmatic Bookshelf.
- Piteira, M., & Costa, C. (2013). *Learning computer programming: study of difficulties in learning programming*. Paper presented at the Proceedings of the 2013 International Conference on Information Systems and Design of Communication, Lisboa, Portugal.
- Polya, G. (1957). *How to solve it. a new aspect of mathematical method*. Princeton, N.J.: Princeton University Press.
- Proulx, V. K. (2000). Programming patterns and design patterns in the introductory computer science course. *SIGCSE Bull.*, 32(1), 80-84.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silverman, B. & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.
- Rist, R. (1996). Teaching Eiffel as a first language. *Journal of Object-Oriented Programming*, 9(1), 30-41.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.

- Robotiksistem (2019). Sensör Çeşitleri. www.robotiksistem.com/sensor_nedir_sensor_cesitleri.html adresinden erişilmiştir.
- Romero, M. & Dupont, Y. (2016). *Educational Robotics: From Procedural learning to co-creative Project oriented challenges with Lego Wedo*. 8th Conference on Education and New Learning Technology, Barselona.
- Ruf, A., Muhling, A., & Hubwieser, P. (2014). *Scratch vs. Karel: Impact on learning outcomes and motivation*. Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education, Berlin, Germany, 5-7 November 2014.
- Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1), 54-67.
- Sankaran, S. R., & Bui, T. (2001). Impact of Learning Strategies and Motivation on Performance: A study in web-based instruction. *Journal of Instructional Psychology*, 28(3), 191-198.
- Scaradozzi, D., Sorbi, L., Pedale, A., Valzano, M., & Vergine, C. (2015). Teaching robotics at the primary school: an innovative approach. *Procedia-Social and Behavioral Sciences*, 174, 3838-3846.
- Scratch (2019). Scratch Hakkında. <https://scratch.mit.edu/about> adresinden erişilmiştir.
- Serin, O., Serin, N. B., & Saygılı, G. (2010). İlköğretim düzeyindeki çocuklar için problem çözme envanteri'nin (ÇPÇE) geliştirilmesi. *İlköğretim Online*, 9(2).
- Sheard, J., & Hagan, D. (1998). Experiences with teaching object-oriented concepts to introductory programming students using C++. *Technology of Object-Oriented Languages and Systems-TOOLS 24, IEEE Technology*, 310-319.
- Shin, S., & Park, P. (2014). A study on the effect affecting problem solving ability of primary students through the scratch programming. *Advanced Science and Technology Letters*, 59(1), 117-120.
- Sönmez, V. (2011). *Eğitim felsefesi*. Ankara: Pegem Yayıncılık.

- Spohrer, J. C., & Soloway, E. (1986). Novice mistakes: Are the folk wisdoms correct?. *Communications of the ACM*, 29(7), 624-632.
- Stamouli, I., Doyle, E., & Huggard, M. (2004, October). Establishing structured support for programming students. In *34th Annual Frontiers in Education, 2004. FIE 2004*. (pp. F2G-5). IEEE.
- Stein, L. A. (1999). Challenging the computational metaphor: Implications for how we think. *Cybernetics & Systems*, 30(6), 473-507.
- Sternberg, R.J., Williams, W.M. (2002). *Educational psychology*, USA: Allyn and Bacon.
- Şabanoviç, A., & Yannier, S. (2003). Robotlar: Sosyal etkileşimli makineler. *TÜBİTAK Bilim Teknik Dergisi*.
- Şahiner, A., & Kert, S. B. (2016). Komputasyonel düşünme kavramı ile ilgili 2006-2015 yılları arasındaki çalışmaların incelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 5(9).
- Şanal, S.Ö. & Erdem, M. (2017). Kodlama ve Robotik Çalışmalarını Problem Çözme Süreçlerine Etkisi: Sesli Düşünme Protokol Analizi. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu (24 - 26 Mayıs 2017)*. Malatya.
- Şenol, A. K., & Büyük, U. (2015). Robotik Destekli Fen Ve Teknoloji Laboratuvar Uygulamaları: Robolab. *Electronic Turkish Studies*, 10(3).
- Şimşek, E. (2018). *Programlama öğretiminde robotik ve scratch uygulamalarının öğrencilerin bilgi işlemsel düşünme becerileri ve akademik başarılarına etkisi (Yüksek Lisans Tezi)*. Ondokuz Mayıs Üniversitesi Eğitim Bilimleri Enstitüsü, Samsun.
- Tan, P. H., Ting, C. Y., & Ling, S.-W. (2009). Learning difficulties in programming courses: Undergraduates' perspective and perception. In IEEE (Ed.), *ICCTD 2009: 2009 International Conference on Computer Technology and Development, Volume 2* (pp. 42– 46).
- Tavares, J., Brzezinski, I., Huet, I., Cabral, A., & Neri, D. (2001). "Having coffee" with professors and students to talk about higher education pedagogy and academic success. Paper presented at the *24th International HERDSA Conference*, Newcastle, Australia.

- Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences*, 8, 561-570.
- Tella, A. (2007). The impact of motivation on student's academic achievement and learning outcomes in mathematics among secondary school students in Nigeria. *Eurasia Journal of Mathematics, Science & Technology Education*, 3(2), 149-156. Retrieved from <https://doi.org/10.12973/ejmste/75390>
- Türk Dil Kurumu. (2018). Türk Dil Kurumu Sözlükleri. <http://sozluk.gov.tr> adresinden erişilmiştir.
- Üçgül, M. (2017). Eğitsel robotlar ve bilgi işlemsel düşünme. Y. Gülbahar (Eds), *Bilgi İşlemsel Düşünmeden Programlamaya*. (ss.295-318) içinde. Ankara: Pegem.
- Varank, I. (2003). *The Effects of Computer Training on Turkish Teachers' Attitudes Toward Computers and The Effects of Computer-Supported Lessons on Turkish Students' Reported Motivation to Lessons* (Doctoral Thesis). Florida State University, USA.
- Vatanserver, Ö. (2018). *Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerisi üzerindeki etkisinin incelenmesi* (Yüksek Lisans Tezi). Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü, Bursa.
- Ventura, P. R. (2005). Identifying predictors of success for an objects-first CS1. *Computer Science Education*, 15(3), 223-243.
- Wang, H. Y., Huang, I. & Hwang, G. J. (2014). Effects of an Integrated Scratch and ProjectBased Learning Approach on the Learning Achievements of Gifted Students in Computer Courses. *2014 IIAI 3rd International Conference on Advanced Applied Informatics* içinde (ss. 382-387). 2014 IIAI 3rd International Conference on Advanced Applied Informatics.
- Williams, L., & Kessler, B. (2000). *The Effects of "Pair-Pressure" and "Pair-Learning" on Software Engineering Education*. Paper presented at the Proceedings of the 13th Conference on Software Engineering Education & Training.

- Wilson, A., Hainey, T., & Connolly, T. (2012). Evaluation of Computer Games Developed by Primary School Children to Gauge Understanding of Programming Concepts. In P. Felicia (Ed.), *Proceedings of the 6th European Conference on Games Based Learning* (pp. 549-558). Academic Conferences and Publishing Limited (ACPIL).
- Woodworth, R. S. (1918). *Dynamic psychology*. New York: Columbia University Press Yayınları.
- Yaşar, S. (1998) Yapısalcı Kuram ve Öğrenme-Öğretme Süreci. *Anadolu Üniversitesi Eğitim Fakültesi Dergisi*, 8(1-2), 68-75.
- Yazıcı, H. & Altun, F. (2013). The association between university students' internal and external motivation sources and their academic achievement. *International Journal of Social Science*, 6(6), 1241-1252.
- Yılmaz, H., & Çavaş, P. H. (2007). Reliability and validity study of the students' motivation toward science learning (smts) questionnaire. *Elementary Education Online*, 6(3), 430-440.
- Yolcu, V. & Demirer, V. (2017). A review on the studies about the use of robotic technologies in education. *SDU International Journal of Educational Studies*, 4(2), 127-139.

EK-A: Kişisel Bilgiler Formu

BİLGİ FORMU

Değerli Öğrencimiz;

Lütfen bu anketteki sorulara hiç kimseden çekinmeden samimi bir şekilde cevap veriniz. Çünkü anketlere vereceğiniz cevaplar sadece bilimsel amaçlar için kullanılacaktır. Uygun olan seçeneğin yanındaki "()" içine "X" işareti koymanız yeterlidir. Ankete vereceğiniz doğru cevaplar ve yardımlarınız için teşekkür ederiz.

Yaş:

Cinsiyet: () Kız () Oğlan

1. Kaç yıldır bilgisayar kullanıyorsunuz?
() 1-2 Yıl () 2-4 Yıl () 4-6 Yıl () 6-8 Yıl
2. Evinizde bilgisayar var mı?
() Evet () Hayır
3. Bilgisayarınız varsa, İnternet erişiminiz var mı? (Bilgisayarınız yoksa bu soruyu boş bırakın)
() Evet () Hayır
4. Evinizde çalışma odanız var mı?
() Evet () Hayır
5. Günlük bilgisayar (tablet, telefon vb.) oyunu oynama süreniz nedir?
() 0-1 saat () 1-3 saat () 3-5 saat () 5 saatten fazla
6. Geçen dönemki matematik notunuz (100 üzerinden) kaçtır?(.....)

Teşekkürler...

EK-B: Programlama Eriş Testi I

PROGRAMLAMA ERİŞİ TESTİ I- ALGORİTMA VE STRATEJİ GELİŞTİRME

1. Aşağıdaki ifadelerde bulunan boşluklara tanıma uygun kelimeyi yazınız.

(), günlük hayatımızda karşılaştığımız, çözüm aranması gereken ve çözümü için bilgi, mantık, deneyim ya da dikkat isteyen durumlardır.

(), bilgisayarların sonuca ulaşabilmek için algıladığı, işlediği, sonuç ürettiği veya daha sonra kullanmak üzere depoladığı her şeye denir.

(), belirli bir problemi çözmek veya bir amaca ulaşmak için tasarlanan yoldur.

() bir aracı, nesneyi ya da sayıyı işletmek/çalıştırmak anlamında kullanılır.

() bir sürecin adımlarını görsel ya da sembolik olarak gösterir.

2. Aşağıdaki ifadelerden doğru olanların başına “D”, yanlış olanların başına “Y” yazınız.

() Bilgisayarların işleyişinde bazı veriler değişkenler aracılığıyla depolanırken bazı veriler ise sabit olarak kalır.

() “VE” ve “VEYA” matematiksel operatörlere örnektir.

() Algoritma tasarımı yapmak bir problemi çözmek için plan yapmaktır.

() Veri türleri sadece değişken tanımlamak için kullanılabilir.

() Akış şemaları çeşitli şekillerden oluşur.

3. Aşağıdaki tabloda sunulan veriyi Sabit ya da Değişken olarak sınıflandırınız.

Veri	Sabit – Değişken
Her gün okula gelen öğrenci sayısı	
Günlük ders sayısı	
Bir ders saati süresi	
Tenefüste oynanan oyun sayısı	
Okuldan çıkış saati	

4. Aşağıdaki tabloda sunulan değişkenler için örnek veri yazıp veri türünü belirtiniz.

Değişken Adı	Veri	Veri Türü
Kitabın Adı		
Kitabın Yazarı		
Kitabın Türü		
Kitabın Sayfa Sayısı		
Kitabı Beğendin mi?		

5. Aşağıda sayılar arasındaki boşluklara +, -, *, /, > operatörlerinden uygun olanını yerleştiriniz.

10 2 = 5 12 6 4 6 = 10 4+5 9 5 ... 2 = 10

6. Aşağıda verilen tablodaki işlemlerin çıktılarını yazınız.

Giriş	İşlem	Çıkış
X = 8	$X * (Z - Y)$	
	$Z - X + Y$	
Y = 4	$Z + X / Y$	
Z = 12	$Z > Y$ VE $X < Y$	() Doğru () Yanlış
	$Z > X$ VEYA $Y > X$	() Doğru () Yanlış

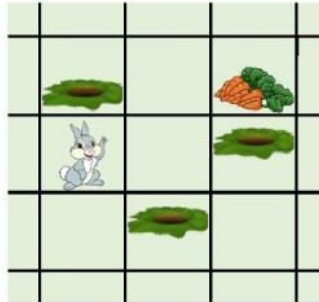
7. "Makarna yapmak" için gerekli olan adımlar aşağıda verilmiştir. Adımları doğru biçimde sıralamak için boşluklara gelmesi gereken uygun sayıyı yazınız.

- Adım 15 dk bekle
Adım Suyu kaynat
Adım Suyun içine at
Adım Paketi aç
Adım Süzgece dök

8. Yazı yazarken kurşun kaleminizin ucu kırıldığında problemi çözmek için gerekli olan adımları yazınız.

.....
.....
.....
.....
.....

9. Aşağıdaki resimde bulunan tavşanın deliklere düşmeden ve en kısa yoldan havuçlara ulaşması için "Sağa dön", "Sola dön", "İlerle" ifadelerini kullanarak gerekli algoritmayı yazınız.



10. Girilen sayının pozitif, negatif veya sıfır olduğunu göstermek için hazırlanan aşağıdaki algoritmanın doğru çalışabilmesi hangi adımın nasıl değiştirilmesi gerektiğini yazınız.

Adım __ : _____

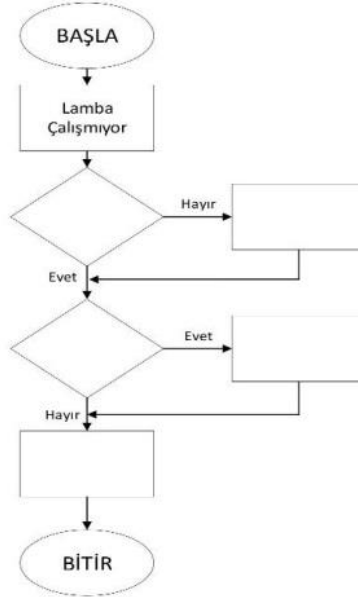
Adım 1 - BAŞLA
Adım 2 - "Sayıyı" YAZ
Adım 3 - Sayı Sıfırdan büyük ise "Sayı Pozitif" YAZ ve Adım 6'ya GİT
Adım 4 - Sayı Sıfırdan küçük ise "Sayı Negatif" YAZ ve Adım 6'ya GİT
Adım 5 - "Sayı Sıfıra Eşit" YAZ
Adım 6 - BİTİR

11. Suyun sıcaklık değeri; 0'dan küçük ise "Katı", 0-100 arasında ise "Sıvı" ve 100'den büyük ise "Gaz" yazdıran aşağıdaki algorithmda boşluğa gelmesi gereken ifadeyi yazınız.

Adım 1 - BAŞLA
Adım 2 - Suyun sıcaklığını GİR
Adım 3 - _____
Adım 4 - Sıcaklık 100'den büyükse "Gaz" YAZ ve Adım 6'ya GİT
Adım 5 - "Sıvı" YAZ
Adım 6 - BİTİR

12. Odanızdaki masa lambanız çalışmıyor. Aşağıda karışık olarak verilen yönergeleri doğru bir şekilde sıralayarak kutuların içine yazınız.

Lambanın fişi takılı mı?	Ampülü değiştir	Fişi tak	Yeni lamba al	Ampul patlak mı?
--------------------------	-----------------	----------	---------------	------------------



13. Farklı deęerler için bir karenin alanını hesaplayan ařađıdaki akıř řeması ekrana sırasıyla hangi deęerleri yazdırır?





















Ekran Çıktısı

EK-C: Programlama Eriş Testi II


PROGRAMLAMA ERİŞİ TESTİ II- SCRATCH

A. Aşağıdaki sorularda doğru olduğunu düşündüğünüz seçeneği işaretleyiniz.



1	Scratch programının kullanım amacı nedir? a) Ses, resim, müzik gibi araçları kullanarak oyun ve animasyon oluşturabiliriz. b) Sadece yazı yazmakta kullanabiliriz. c) Scratch bir bilgisayarın çalışması için gereken donanımdır. d) Scratch'ı İnternet'e girmek için kullanırız.	2	Scratch programında sahne ne anlama gelmektedir? a) Kullanılan blokların bulunduğu bölümdür. b) Projemizin canlandırıldığı yerdir. c) Blokları çekip bıraktığımız yerdir. d) Yazı yazabildiğimiz bölümdür.
3	 Yandaki düğmenin scratch programındaki görevi aşağıdakilerden hangisidir? a) Projeyi başlatır. b) Proje çalışmasını durdurur. c) Sahneyi tam ekran yapar. d) Scratch programını kapatır.	4	Scratch programında yatay ve dikey eksenler hangi harfle adlandırılır? a) z,t b) y,d c) x,y d) a,b
5	 Yanda gösterilen x ve y değerleri kuklanın (karakterin) hangi özelliğini ifade etmektedir? a) Yönünü b) Boyutunu c) Konumunu d) Dönüş biçimini	6	Kuklayı (karakteri) büyötmek için aşağıdaki düğmelerden hangisi kullanılır? a)  b)  c)  d) 
7	Kuklayı (karakteri) sahnede görünmez yapan kod bloğu aşağıdakilerden hangisidir? a)  b)  c)  d) 	8	Scratch programında, kod dizilerinin çalışmasını durduran blok aşağıdakilerden hangisidir? a)  b)  c)  d) 
9	Aşağıdakilerin hangisi Scratch hareket blokları içinde yer alır? a)  b)  c)  d) 	10	Aşağıdakilerden hangisi yanlıştır? a) Sahnede birden fazla kukla olabilir. b) Her kuklanın sadece 1 kılıfı vardır. c) Sahnedeki kuklalar bloklar yardımıyla hareketlendirilir. d) Scratch programıyla animasyonlar oluşturabiliriz.

B. Aşağıdaki soruları tabloya göre cevaplandırınız. Size göre doğru olan kutucukların numaralarını soruların yanına yazınız.

1 5 birim büyüt	2 tempoyu 20 arttır	3 x konumu	4 çalgıyı 1v yap
5 10 adım git	6 ses şiddeti	7 büyüklüğü %b 105 yap	8 fare oku 'na git

- Kaç numaralı kutucuktakiler ses blok grubunda yer alır?:
- Kaç numaralı kutucuktakiler kuklanın konumunu değiştirir?:
- Kaç numaralı kutucuktakiler  düğmesinin görevini yerine getirir?:
- Kaç numaralı kutucuktakiler diğer bloklara giriş değeri olarak kullanılır?:

C. Aşağıdaki tabloda verilen kodları uygun tanımlarla eşleştiriniz.

A.  tıklanınca	B.  olana kadar bekle	C. Puan '1' 1 arttır	D. fare oku 'a' mesafe
E. 15 derece dön	F. costume2 kılığına geç	G. iz bırak	H. bilgiim in uzunluğu
1. Karakterin istenilen derece kadar saat yönünde dönmesini sağlar.	2. Karakterin başka bir karakter ya da fare işaretçisiyle arasında olan mesafe bilgisini verir.	3. Girilen değerin uzunluk bilgisini verir.	4. Karakterin istenilen kostüme geçmesini sağlar.
5. Karakterin o anki konumunda ekrana kendi görüntüsünü yapıştırır.	6. Proje içerisinde bayrağa tıkladığında bu blok altında sıralanmış kod kümesini çalıştırır.	7. Koşul doğru olana kadar bekletir.	8. Var olan değişkenleri istenilen değer kadar artırır ya da azaltır.









A- 	B- 	C- 	D-
E- 	F- 	G- 	H-
I- 	J- 	K- 	L-
M- 	N- 	O- 	P-
R- 	S- 	T- 	U-

D. Aşağıda verilen görevleri gerçekleştirmek için gerekli kod bloklarını yukarıdaki tablodan seçerek kod bloğuna ait harfi kutucuklara yazınız.

No	Karakter	Görev	Blok
1	Kedi	Kedinin, / haber geldiğinde / sürekli olarak / 5 adım gitmesi / kenara geldiğinde geri dönmesi.	
2	Köpekbalığı	Köpekbalığının, / boşluk tuşuna basılınca / sürekli olarak / fare okuna dönerek / 5 adım hareket etmesi.	
3	Bas gitar	Basgitarın, / kukla tıkladığında / çalgıyı değiştirerek / müzik notasını çaldıktan sonra, / haber salması	
4	Fare	Farenin, / sürekli olarak / fare imleci konumuna gitmesi / kedi kuklasına değerse, / görünümü gizlemesi	
5	Top	Topun, / belirtilen renge değdiğinde, / "pop" sesi çalması, / "2 sn. süresince "oyun bitti" mesajı verip, / bütün komutları durdurması	



E. Aşağıdaki sorularda yandaki karaktere ait kod blokları verilmiştir. Bu kod bloklarına göre aşağıdaki soruları cevaplandırınız.

<p>1 Ekran çıktısı nedir?</p>  <p>a)A b)A c)B d)B B A B A A B A B B B A A</p>	<p>2 Girilen not=70 ise ekran çıktısı nedir?</p>  <p>a) Başarısız b) Başarısız, Başarılı c) Başarılı d) Başarılı, Başarısız</p>
<p>3 Ekran çıktısı nedir?</p>  <p>a) 2 b) 8 c) 15 d) 35</p>	<p>4 Ekran çıktısı hangi seçenek olamaz?</p>  <p>a) 7 b) 3 c) 6 d) 12</p>
<p>5 Ekran çıktısı nedir?</p>  <p>a)5 b)1 c)0 d)4 4 2 1 3 3 3 2 2 2 4 3 1 1 5 4 0</p>	<p>6 Ekran çıktısı sırasıyla 1, 2, 3, 4 ise girilen sayı kaçtır?</p>  <p>a) 4 b) 3 c) 5 d) 6</p>
<p>7 Ekran çıktısı nedir?</p>  <p>a) 1 b) 2 c) 5 d) 8</p>	<p>8 Ekran çıktısında "Merhaba" kaç defa tekrarlanır?</p>  <p>a) 2 b) 3 c) 5 d) 6</p>

9 Ekran çıktısı nedir?

a)3 b)3 c)2 d)2
 4 4 4 4
 5 6 6 5
 8 8 6 8

10 Ekran çıktısı nedir?

a) 95 b) 75
 c) 79 d) 57

11 Ekran çıktısı hangi seçenek olamaz?

a) 7 b) 2
 c) 3 d) 5

12 Bayrağa tıklanınca ekran çıktısı ne olur?

a) 55 b) 25
 c) 10 d) 0

13 "İsimler" listesine aşağıdaki blok kodları uygulanmıştır. Buna göre ekran çıktısı hangi seçenekte verilmiştir?

a) Elif b) Ahmet
 c) Ceren d) Gürkan

14 Yukarıdaki blok kodları uygulandıktan sonra kukla yandaki konuma gelirse ekran çıktısı ne olur?

a) 1. bölge b) 2. bölge
 c) 3. bölge d) 4. bölge

EK-Ç: Uygulama Basamakları Örnekleri (Kontrol Grubu)

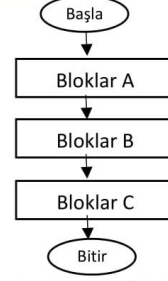
MUZ AVI

Programlama yapısı

Bu uygulama doğrusal yapıdadır. Komut, ilk bloktan başlar, ardından diğer tüm bloklar sırayla yürütülür.

Yandaki akış şeması diyagram doğrusal yapıyı gösterir. Program başladıktan sonra sırayla bloklar uygulanır ve sonunda biter. Doğrusal yapı, programlamanın temelini oluşturur.

Akış Şeması



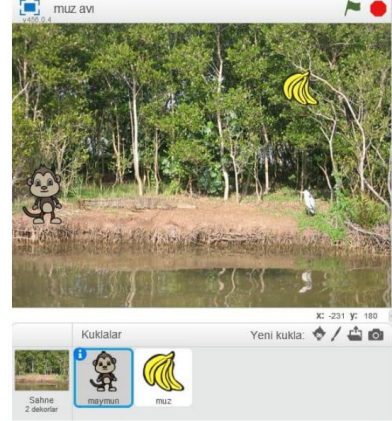
1

Bu etkinlikte birden fazla kukla kullanacağız. Etkinliğimizde, öncelikle sahnemizi sahne kütüphanesinden "lake (göl)" yapalım.



2

Kukla kütüphanesinden maymun ve muz seçerek sahnede uygun yerlere yerleştirelim. Bu etkinliğimizde maymun muza değdikten sonra muzun kaybolmasını sağlayacağız.



3

İlk olarak maymunun kodlarından başlayalım. Öncelikle boşluk tuşuna basıldıktan sonra maymunun, belirtilen x ve y koordinatlarına gidip kullanıcıdan bir tuşa (fare sol tuşu) basmasını beklemesini istiyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



4

Kullanıcı bir tuşa bastıktan sonra ses çıkması ve maymunun renk değiştirmesi için yanda işaretlenen kodları ekliyoruz.



MUZ AVI

5

Maymunun önce yatay sonra da dikey hareket etmesi için yanda işaretlenen kodları ekliyoruz.

```
65 notasını 0.5 vuruş çal
renk etkisi 20 olsun
2 sn.de x: 120 y: -25 a süzül
1 sn.de x: 120 y: 70 a süzül
```

6

Maymunun muza değdikten sonra bekleyip tekrar dikey hareket ettikten sonra kodların çalışmasını durdurmak için yanda işaretlenen kodları ekliyoruz.

```
2 sn.de x: 120 y: -25 a süzül
1 sn.de x: 120 y: 70 a süzül
0.5 saniye bekle
1 sn.de x: 120 y: -25 a süzül
hepsi durdur
```

7

Sıra geldi muzun kodlarına; Öncelikle muzun belirtilen x ve y koordinatlarında her zaman görünür olması gereklidir. Bunun için yandaki kodlarla başlıyoruz.

```
boşluk tuşu basınca
x: 120 y: 100 noktasına git
görün
```

8

Daha önce belirttiğimiz şekilde maymun muza değmekteydi. Dolayısıyla muzun maymunu algılayıp maymuna değene kadar beklemesi için yanda işaretlenen kodu ekliyoruz.

```
boşluk tuşu basınca
x: 120 y: 100 noktasına git
görün
maymun a değdi (mi?) olana kadar bekle
```

9

Maymun muza değdikten sonra ses çıkması ve muzun renk değişimi için yanda işaretlenen kodları ekliyoruz.

```
maymun a değdi (mi?) olana kadar bekle
renk etkisini 10 arttır
52 notasını 0.5 vuruş çal
```

10

Son olarak maymunun muza değdikten sonra bekleyip kaybolması için yanda işaretlenen kodu ekliyoruz.

```
renk etkisini 10 arttır
52 notasını 0.5 vuruş çal
0.5 saniye bekle
gizlen
```

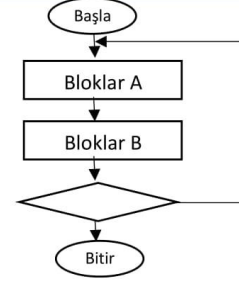

ELMA TOPLAMA

Programlama yapısı

Bu uygulama karar yapısındadır. Karar yapıları program yazma sırasında oluşturulan koşulun gerçekleşmesi ya da gerçekleşmemesi durumunu kontrol eder.

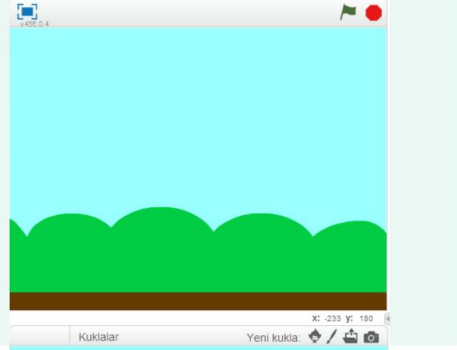
Yandaki akış şeması diyagramı karar yapısını gösterir. Programın akışı koşula göre değişiklik gösterir. Programın koşula göre karar alması ve o şekilde ilerlemesi gerekir.

Akış Şeması



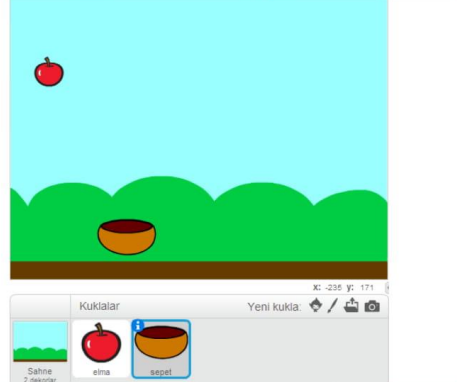
1

Bu etkinlikte birden fazla kukla kullanacağız. Etkinliğimizde, öncelikle sahnemizi sahne kütüphanesinden "blue sky (mavi gökyüzü)" yapalım.




2

Kukla kütüphanesinden sepet ve elma seçerek sahnede uygun yerlere yerleştirelim. Bu etkinliğimizde sepet ile elmaları yakalamaya çalışacağız.



3

İlk olarak sepetin kodlarından başlayalım.


Sepetimizin  düğmesine basıldıktan sonra klavye yön tuşlarıyla sağa ve sola hareket etmesini istiyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



4

ELMA TOPLAMA

Sıra geldi elmanın kodlarına;

Elmamızın  düşmesine basıldıktan sonra y konumunu azaltarak, dikey olarak düşmesini sağlayalım. Bunun için yandaki kodları ekleyelim.

5 Düşen elmanın kenara değdiğinde tekrar yukarı çıkması için yandaki kodları ekleyelim.

6 Elmanın her defasında farklı x konumundan düşmesi için x konumuna belirtilen değerler arasında rastgele sayı üretilmesini sağlamak için x konumunu yandaki gibi değiştirelim.

7 Düşen elmanın sepetin içine değdikten sonra tekrar yukarı çıkması için yandaki kodları ekleyelim.

8 Sıra geldi değişkenleri kullanmaya. Bunun için veri bloklarından "toplanan elma ve " düşen elma" değişkenleri oluşturuyoruz.

9 Her iki değişkenin de başlangıç değerini 0 olarak belirliyoruz. Başlangıç rengi etkisini 0 olarak belirliyoruz.

10 Alt kenara değdiğinde düşen elma sayısının artması ve elmanın renk değişimi için daha önceden oluşturduğumuz blokların arasına yanda işaretlenen kodu ekliyoruz.



ELMA TOPLAMA

11

Sepete deđdiđinde toplanan elma sayısının artması için daha önceden oluşturduğumuz blokların arasına yanda işaretlenen kodu ekliyoruz.



```
eđer rengine deđdi (mi?) ise
  renk etkisi 40 olsun
  toplanan elma 'i 1 arttır
  x: -200 ile 200 arasında bir sayı (1
```

12

Düşen elma sayısı=3 olduğunda “oyun bitti” mesajı vermesi, davul sesi çıkarması ve oyunun durması için yandaki kodları ekliyoruz.



```
eđer düşen elma = 3 ise
  12 davulunu 0.25 vuruş çal
  Oyun bitti de 2 saniye
  hepsi durdur
```

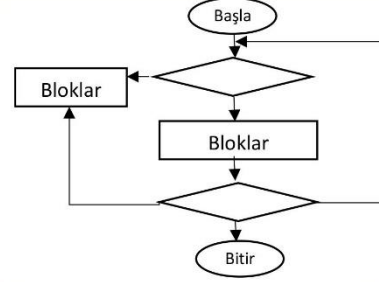
YARASA VURMA

Programlama yapısı

Bu uygulama çoklu karar yapısındadır. Karar yapıları program yazma sırasında oluşturulan koşulun gerçekleşmesi ya da gerçekleşmemesi durumunu kontrol eder.

Yandaki akış şeması diyagramı çoklu karar yapısını gösterir. Programın akışı koşula göre değişiklik gösterir. Programın koşula göre karar alması ve o şekilde ilerlemesi gerekir.

Akış Şeması



1

Bu etkinlikte birden fazla kukla kullanacağız. Etkinliğimizde, öncelikle sahnemizi sahne kütüphanesinden "castle2 (kale2)" yapalım.




2

Kukla kütüphanesinden büyücü, yarasaya ve yıldız seçerek sahnede uygun yerlere yerleştirelim. Bu etkinliğimizde büyücüden yıldız göndererek yarasayı vurmaya çalışacağız.




3

İlk olarak büyücünün kodlarından başlayalım.  düğmesine basıldıktan sonra boşluk tuşuna basıldığında büyücümüzün saat yönünde dönmesini, boşluk tuşuna basılmadığında ise saat yönü tersine dönmesini istiyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



4

Fare oku büyücüyeye yaklaştığında büyücünün yarasaya yıldız göndermesini istiyoruz. Bunun için  düğmesine basıldıktan sonra fare oku ve büyücü arasındaki mesafe 100 adımdan daha az kaldığında ateş haberini salmasını sağlıyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



YARASA VURMA

5



Yıldızın kodlarıyla devam ediyoruz. Ateş haberi geldiğinde yıldızın kendi kopyasını çıkartması için yandaki kod bloklarını ekliyoruz.

```
ateş haberi gelince
yıldız in ikizini yarat
```

6

Oluşturduğumuz kopyanın büyücüye gitmemesi ve büyücüyle aynı yöne dönüp 60 adım ileriden tekrar görünür olması için yandaki kod bloklarını ekliyoruz.

```
ikiz olarak başladığında
renk etkisi olsun
büyücü'na git
yönü değeri büyücü'ın yönüne dön
60 adım git
görün
```

7

Kopya yıldızın sahne kenarlarına kadar ilerlemesi için yandaki kod bloğunu ekliyoruz.

```
görün
kenar a değdi (mi?) olana kadar tekrarla
eğer yaras a değdi (mi?) ise
değilse
10 adım git
```

8

Kopya yıldızın kaybolmasını sağlamak için özel taş tanımlıyoruz.

```
tanımla kaybolma
renk etkisi olsun
10 saniye bekle
gizlen
bu ikizi sil
```

9

Tanımladığımız özel taşı kopya yıldızın kaybolması gereken yerlere yerleştiriyoruz.

```
görün
kenar a değdi (mi?) olana kadar tekrarla
eğer yaras a değdi (mi?) ise
kaybolma
değilse
10 adım git
```

10



Son olarak yarasanın kodlarını ekliyoruz. Yarasanın ekranda rastgele x konumunda görünüp büyücüye yönelmesi ve kenara değene kadar ilerlemesi için yandaki kod bloklarını ekliyoruz.

```
titizlenince
sürekli tekrarla
görün
x: 0 ile 100 arasında bir sayı (rast) y: 0 ile 100 arasında git
boyutu'ne değine dön
kenar a değdi (mi?) olana kadar tekrarla
10 adım git
```

11

Yarasa yıldız değere ise gizlenmesi, değmez ise büyücüye değdiğinde kod bloklarının sonlandırılması, büyücüye değmediğinde ise kenara değene kadar ilerlemesi için kod bloklarını yandaki gibi değiştiriyoruz.

```
kenar a değdi (mi?) olana kadar tekrarla
eğer yıldız a değdi (mi?) ise
gizlen
değilse
eğer büyücü a değdi (mi?) ise
hemen durdur
değilse
10 adım git
```

EK-D: Uygulama Basamakları Örnekleri (Deney Grubu)

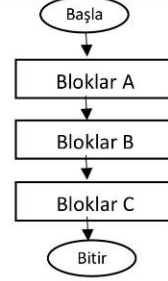
MUZ AVI

Programlama yapısı

Bu uygulama doğrusal yapıdadır. Komut, ilk bloktan başlar, ardından diğer tüm bloklar sırayla yürütülür.

Yandaki akış şeması diyagram doğrusal yapıyı gösterir. Program başladıktan sonra sırayla bloklar uygulanır ve sonunda biter. Doğrusal yapı, programlamanın temelini oluşturur.

Akış Şeması



1

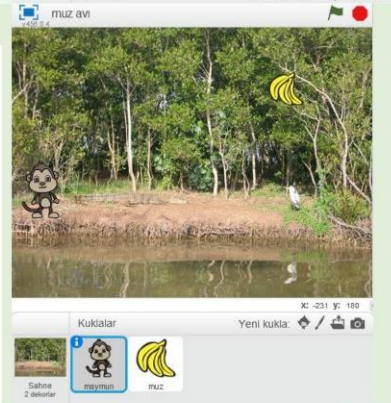
Bu etkinlikte Scratch ile otantik dünyanın etkileşimini sağlayacağız. Bunun Lego Wedo 2.0 Smart Hub (Beyin) kullanacağız. İlk etkinliğimizde smart Hub üzerinde bulunan buton, led ve ses kontrolünü sağlayacağız.

Bu etkinlikte birden fazla kukla kullanacağız. Etkinliğimizde, öncelikle sahne görüntümüzü sahne kütüphanesinden "lake (göl)" yapalım.



2

Kukla kütüphanesinden maymun ve muz seçerek sahnede uygun yerlere yerleştirelim. Bu etkinliğimizde maymun muza değdikten sonra muzun kaybolmasını sağlayacağız.



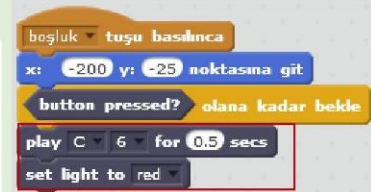
3

İlk olarak maymunun kodlarından başlayalım. Öncelikle boşluk tuşuna basıldıktan sonra maymunun, belirtilen x ve y koordinatlarına gittikten sonra kullanıcı Smart Hub üzerinde bulunan butona basana kadar beklemesini istiyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



4

Kullanıcı butona bastıktan sonra Smart Hub'tan ses çıkması ve led ışık değiştirmesi için yanda işaretlenen kodları ekliyoruz.



MUZ AVI

5

Maymunun önce yatay sonra da dikey hareket etmesi için yanda işaretlenen kodları ekliyoruz.

```
play C 6 for 0.5 secs
set light to red
2 sn.de x: 120 y: -25 a süzül
1 sn.de x: 120 y: 70 a süzül
```

6

Maymun muza değdikten sonra bekleyip tekrar dikey olarak geri hareket etmesi ve sonra kodların çalışmasının durması için yanda işaretlenen kodları ekliyoruz.

```
2 sn.de x: 120 y: -25 a süzül
1 sn.de x: 120 y: 70 a süzül
0.5 saniye bekle
1 sn.de x: 120 y: -25 a süzül
hepsi durdur
```

7

Sıra geldi muzun kodlarına; Öncelikle muzun belirtilen x ve y koordinatlarında her zaman görünür olması gereklidir. Bunun için yandaki kodlarla başlıyoruz.

```
boşluk tuşu basılınca
x: 120 y: 100 noktasma git
görün
```

8

Daha önce belirttiğimiz şekilde maymun muza değmekteydi. Dolayısıyla muzun maymunu algılayıp maymun değene kadar beklemesi için yanda işaretlenen kodu ekliyoruz.

```
boşluk tuşu basılınca
x: 120 y: 100 noktasma git
görün
maymun a değdi (mi?) olana kadar bekle
```

9

Maymun muza değdikten sonra Smart hub' dan ses çıkması ve led ışık renk değişimi için yanda işaretlenen kodları ekliyoruz.

```
maymun a değdi (mi?) olana kadar bekle
set light to green
play C 6 for 1 secs
```

10

Son olarak maymunun muza değdikten sonra bekleyip kaybolması için yanda işaretlenen kodu ekliyoruz.

```
set light to green
play C 6 for 1 secs
0.5 saniye bekle
gizlen
```

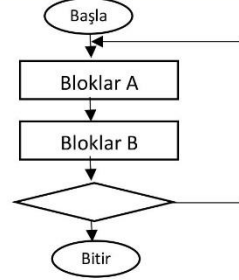
ELMA TOPLAMA

Programlama yapısı

Bu uygulama karar yapısındadır. Karar yapıları program yazma sırasında oluşturulan koşulun gerçekleşmesi ya da gerçekleşmemesi durumunu kontrol eder.

Yandaki akış şeması diyagramı karar yapısını gösterir. Programın akışı koşula göre değişiklik gösterir. Programın koşula göre karar alması ve o şekilde ilerlemesi gerekir.

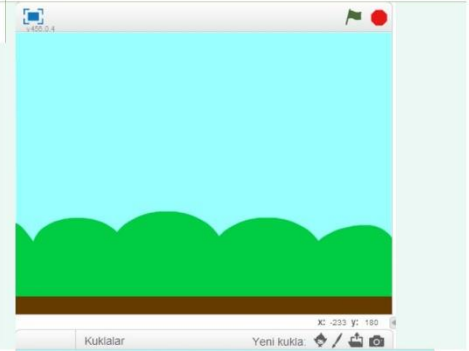
Akış Şeması



1

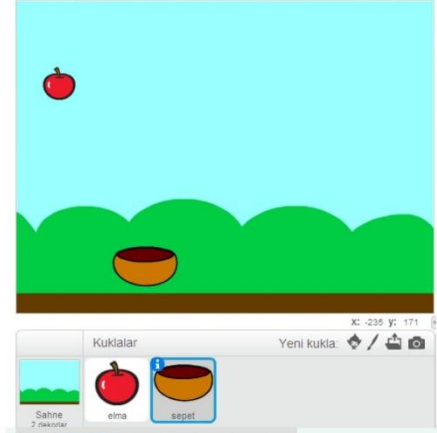
Bu etkinlikte birden fazla kukla kullanacağız. Bunun için Lego Wedo 2.0 Smart Hub (Beyin) ve eğim sensörü kullanacağız. Etkinliğimizde Smart Hub üzerinde bulunan led ve ses kontrolünü sağlayacağız.

Etkinliğimizde, öncelikle sahnemizi sahne kütüphanesinden “blue sky (mavi gökyüzü)” yapalım.




2

Kukla kütüphanesinden sepet ve elma seçerek sahnede uygun yerlere yerleştirelim. Bu etkinliğimizde sepet ile elmaları yakalamaya çalışacağız. Sepeti robotumuzda bulunan eğim sensörü ile kontrol edeceğiz.



3

İlk olarak sepetin kodlarından başlayalım.


Sepetimizin  düğmesine basıldıktan sonra eğim sensörüyle sağa ve sola hareket etmesini istiyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



ELMA TOPLAMA

4

Sıra geldi elmanın kodlarına;

Elmamızın  düğmesine basıldıktan sonra y konumunu azaltarak, dikey olarak düşmesini sağlayalım. Bunun için yandaki kodları ekleyelim.



```
tıklanınca
sürekli tekrarla
y'yi -5 arttır
```

5

Düşen elmanın kenara değdiğinde tekrar yukarı çıkması için yandaki kodları ekleyelim.



```
tıklanınca
sürekli tekrarla
y'yi -5 arttır
eğer kenar a değdi (mi?) ise
x: 0 y: 140 noktasına git
```

6

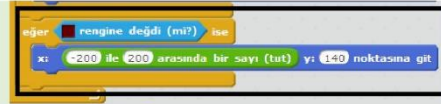
Elmanın her defasında farklı x konumundan düşmesi için x konumuna belirtilen değerler arasında rastgele sayı üretilmesini sağlamak için x konumunu yandaki gibi değiştirelim.



```
tıklanınca
sürekli tekrarla
eğer kenar a değdi (mi?) ise
x: -200 ile 200 arasında bir sayı (tut) y: 140 noktasına git
```

7

Düşen elmanın sepetin içine değdikten sonra tekrar yukarı çıkması için yandaki kodları ekleyelim.



```
eğer rengine değdi (mi?) ise
x: -200 ile 200 arasında bir sayı (tut) y: 140 noktasına git
```

8

Sıra geldi değişkenleri kullanmaya. Bunun için veri bloklarından "toplanan elma ve " düşen elma" değişkenleri oluşturuyoruz.



```
Veri | Özel Taşlar
Bir Değişken Oluştur
✓ düşen elma
✓ toplanan elma
düşen elma, 0 olsun
düşen elma, i +1 arttır
düşen elma, değişkenini göster
düşen elma, değişkenini gizle
```

9

Her iki değişkenin de başlangıç değerini 0 olarak belirliyoruz. Başlangıçta led ışığı kapatıyoruz.



```
tıklanınca
düşen elma, 0 olsun
toplanan elma, 0 olsun
set light to kapat
```


ELMA TOPLAMA

10

Elma düştüğünde kırmızı led ışık yanıp ses çıkması ve düşen elma sayısının artması için daha önceden oluşturduğumuz blokların arasına yanda işaretlenen kodu ekliyoruz.



```
eğer kenar a değdi (mi?) ise
set light to red
play C 6 for 1 secs
düşen elma 'i 1 arttır
x: -200 ile 200 arasında bir say
```

11

Elmanın sepete değdiğinde yeşil led ışık yanıp toplanan elma sayısının artması için daha önceden oluşturduğumuz blokların arasına yanda işaretlenen kodu ekliyoruz.



```
eğer rengine değdi (mi?) ise
set light to green
toplanan elma 'i 1 arttır
x: -200 ile 200 arasında bir say
```

12

Düşen elma sayısı=3 olduğunda "oyun bitti" mesajı vermesi, ses çıkarması ve oyunun durması için yandaki kodları ekliyoruz.



```
eğer düşen elma = 3 ise
play D 6 for 2 secs
Oyun bitti de 2 saniye
hepsi durdur
```

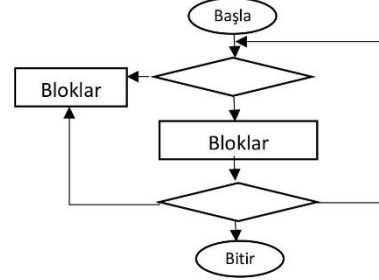
YARASA VURMA

Programlama yapısı

Bu uygulama çoklu karar yapısındadır. Karar yapıları program yazma sırasında oluşturulan koşulun gerçekleşmesi ya da gerçekleşmemesi durumunu kontrol eder.

Yandaki akış şeması diyagramı çoklu karar yapısını gösterir. Programın akışı koşula göre değişiklik gösterir. Programın koşula göre karar alması ve o şekilde ilerlemesi gerekir.

Akış Şeması



1

Bu etkinlikte birden fazla kukla kullanacağız. Bunun için Lego Wedo 2.0 Smart Hub (Beyin) ve mesafe sensörü kullanacağız. Etkinliğimizde Smart Hub üzerinde bulunan led ve buton kontrolünü sağlayacağız. Bu etkinlikte birden fazla kukla kullanacağız. Etkinliğimizde, öncelikle sahnemizi sahne kütüphanesinden "castle2 (kale2)" yapalım.




2

Kukla kütüphanesinden büyücü, yarasaya ve yıldız seçerek sahnede uygun yerlere yerleştirilim. Bu etkinliğimizde büyücüden yıldız göndererek yarasaya vurmaya çalışacağız.




3

İlk olarak büyücünün kodlarından başlayalım.  düğmesine basıldıktan sonra Smart Hub üzerindeki butona basıldığında büyücümüzün saat yönünde dönmesini, boşluk tuşuna basılmadığında ise saat yönü tersine dönmesini istiyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



4

Fare oku büyücüyeye yaklaştığında büyücünün yarasaya yıldız göndermesini istiyoruz. Bunun için  düğmesine basıldıktan sonra mesafe sensörü değeri 50'den daha az olduğunda ateş haberini salmasını sağlıyoruz. Bunun için yandaki kod bloklarını ekliyoruz.



YARASA VURMA

5



Yıldızın kodlarıyla devam ediyoruz. Ateş haberi geldiğinde yıldızın kendi kopyasını çıkartması için yandaki kod bloklarını ekliyoruz.

```
ateş haberi gelince
yıldız in ikizini yarat
```

6

Oluşturduğumuz kopyanın büyücüye gitmemesi ve büyücüyle aynı yöne dönüp 60 adım ileriden tekrar görünür olması için yandaki kod bloklarını ekliyoruz.

```
ikiz olarak başladığında
renk etkisi olsun
büyücü'na git
yönü değeri büyücü'ın yönüne dön
60 adım git
görün
```

7

Kopya yıldızın sahne kenarlarına kadar ilerlemesi için yandaki kod bloğunu ekliyoruz.

```
görün
kenar a değdi (mi?) olana kadar tekrarla
eğer yaras a değdi (mi?) ise
değilse
10 adım git
```

8

Kopya yıldızın kaybolmasını sağlamak için özel taş tanımlıyoruz.

```
tanımla kaybolma
renk etkisi olsun
10 saniye bekle
gizlen
bu ikizi sil
```

9

Tanımladığımız özel taşı kopya yıldızın kaybolması gereken yerlere yerleştiriyoruz.

```
görün
kenar a değdi (mi?) olana kadar tekrarla
eğer yaras a değdi (mi?) ise
kaybolma
değilse
10 adım git
```

10



Son olarak yarasanın kodlarını ekliyoruz. Yarasanın ekranda rastgele x konumunda görünüp büyücüye yönelmesi ve kenara değene kadar ilerlemesi için yandaki kod bloklarını ekliyoruz.

```
titizlenince
sürekli tekrarla
görün
x: 0 ile 100 arasında bir sayı (rast) y: 0 ile 100 arasında git
büyücü'ye değine dön
kenar a değdi (mi?) olana kadar tekrarla
10 adım git
```

11

Yarasa yıldız değere ise gizlenmesi, değmez ise büyücüye değdiğinde kod bloklarının sonlandırılması, büyücüye değmediğinde ise kenara değene kadar ilerlemesi için kod bloklarını yandaki gibi değiştiriyoruz.

```
kenar a değdi (mi?) olana kadar tekrarla
eğer yıldız a değdi (mi?) ise
gizlen
değilse
eğer büyücü a değdi (mi?) ise
hemen durdur
değilse
10 adım git
```


EK-E: Veri Toplama Araçları Kullanım İzinleri

İlköğretim düzeyindeki çocuklar için problem çözme envanteri Ölçek izni

3 ileti

Fatih Özer <fatihazer14tr@gmail.com>

22 Ekim 2017 01:46

Alıcı: nerguz.bulut@deu.edu.tr, nserin@eul.edu.tr, nerguzserin@gmail.com

Merhaba Nergüz Hocam,
Hacettepe Üniversitesi Bilgisayar ve Öğretim Teknolojileri bölümünde yüksek lisans öğrencisiyim. 2010 yılında Prof. Dr. Oğuz SERİN ve Doç. Dr. Gizem SAYGILI ile birlikte geliştirdiğiniz **İlköğretim düzeyindeki çocuklar için problem çözme envanteri'nin (ÇPÇE) geliştirilmesi** ölçeğini tez çalışmamda izniniz olursa kullanmak istiyorum. Bu konuda yardımcı olursanız çok memnun olurum.

Saygılarımla,,,

Fatih ÖZER
ANKARA

Nergüz Bulut Serin <nerguzserin@gmail.com>

23 Ekim 2017 18:05

Alıcı: Fatih Özer <fatihazer14tr@gmail.com>

Sayın Fatih Özer, " İlköğretim düzeyindeki çocuklar için problem çözme envanteri'nin (ÇPÇE) geliştirilmesi ölçeğini tez çalışmanızda kullanabilirsiniz. İyi çalışmalar dilerim.

Yrd.Doç.Dr.Sami ACAR <samiacar@gmail.com>

22 Ocak 2018 13:10

Alıcı: Fatih Özer <fatihazer14tr@gmail.com>

Sayın Fatih Özer,

J.M. Keller tarafından 2006 yılında son sürümü İngilizce olarak tarafıma e-posta ile gönderilen ve doktora tezimde Türkçe'ye uyarladığım CIS (Course Interest Survey) ölçeğini yüksek lisans tezinde kullanabilirsin. Ayrıca, çalışma sonuçlarını benimle de paylaşırsan memnun olurum.

Yüksek lisans tez çalışmada başarı dileklerle,

20 Ocak 2018 12:26 tarihinde Fatih Özer <fatihazer14tr@gmail.com> yazdı:

Merhaba Sami Hocam,
Hacettepe Üniversitesi Bilgisayar ve Öğretim Teknolojileri bölümünde yüksek lisans öğrencisiyim. Keller (2010) tarafından geliştirilen Türkçeye çevirerek uyarladığınız (CIS) Course Interest Survey Motivasyon ölçeğini tez çalışmamda izniniz olursa kullanmak istiyorum. Bu konuda yardımcı olursanız çok memnun olurum.

Saygılarımla...

EK-F: Etik Komisyonu Onay Bildirimi



T.C.
HACETTEPE ÜNİVERSİTESİ
Genel Sekreterlik

Sayı : 76000869/

433-1270

19 Mart 2018

EĞİTİM BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

İlgi: 23.02.2018 tarih ve 561 sayılı yazınız.

Enstitünüz Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı tezli yüksek lisans programı öğrencilerinden **Fatih ÖZER**'in **Prof. Dr. Hakan TÜZÜN** danışmanlığında yürüttüğü "**Kodlama Eğitiminde Robot Kullanılmasının Ortaokul Öğrencilerinin Erişi, Motivasyon ve Problem Çözme Becerilerine Etkisi**" başlıklı tez çalışması, Üniversitemiz Senatosu Etik Komisyonunun **06 Mart 2018** tarihinde yapmış olduğu toplantıda incelenmiş olup, etik açıdan uygun bulunmuştur.

Bilgilerinizi ve gereğini rica ederim.

Prof. Dr. Rahime M. NOHUTCU
Rektör a.
Rektör Yardımcısı

EK-G: Etik Beyanı

Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı bütün bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin bütününe kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

06 / 09 /2019


Fatih ÖZER

EK-Ğ: Yüksek Lisans Tez Çalışması Orijinallik Raporu

06/09/2019

HACETTEPE ÜNİVERSİTESİ

Eğitim Bilimleri Enstitüsü

Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı Başkanlığına,

Tez Başlığı: Kodlama Eğitiminde Robot Kullanımının Ortaokul Öğrencilerinin Erişi, Motivasyon ve Problem Çözme Becerilerine Etkisi

Yukarıda başlığı verilen tez çalışmamın tamamı (kapak sayfası, özetler, ana bölümler, kaynakça) aşağıdaki filtreler kullanılarak Turnitin adlı intihal programı aracılığı ile kontrol edilmiştir. Kontrol sonucunda aşağıdaki veriler elde edilmiştir:

Rapor Tarihi	Sayfa Sayısı	Karakter Sayısı	Savunma Tarihi	Benzerlik Oranı	Gönderim Numarası
06 / 09 / 2019	136	173,892	10 / 07 / 2019	%22	1168135982

Uygulanan filtreler:

1. Kaynaklar hariç
2. Alıntılar dâhil
3. 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve 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 eder, gereğini saygılarımla arz ederim.

Ad Soyadı: Fatih ÖZER

Öğrenci No.: N13227811

Ana Bilim Dalı: Bilgisayar ve Öğretim Teknolojileri Eğitimi

Programı: Bilgisayar ve Öğretim Teknolojileri Eğitimi

Statüsü: Y.Lisans Doktora Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Prof. Dr. Hakan TÜZÜN

EK-H: Thesis Originality Report

06/09/2019

HACETTEPE UNIVERSITY
Graduate School of Educational Sciences
To The Department of Computer Education and Instructional Technologies

Thesis Title: Effect of Using Robotics in Teaching Coding on Achievement, Motivation and Problem Solving Skills of Middle School Students

The whole thesis that includes the *title page, introduction, main chapters, conclusions and bibliography section* is checked by using Turnitin plagiarism detection software take into the consideration requested filtering options. According to the originality report obtained data are as below.

Time Submitted	Page Count	Character Count	Date of Thesis Defense	Similarity Index	Submission ID
06 / 09 / 2019	136	173,892	10 / 07 / 2019	%22	1168135982

Filtering options applied:

1. Bibliography excluded
2. Quotes included
3. Match size up to 5 words excluded

I declare that I have carefully read Hacettepe University Graduate School of Educational Sciences Guidelines for Obtaining and Using Thesis Originality Reports; that according to the maximum similarity index values specified in the Guidelines, my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge.

I respectfully submit this for approval.

Name Lastname: Fatih ÖZER
Student No.: N13227811
Department: Computer Education and Instructional Technologies
Program: Computer Education and Instructional Technologies
Status: Masters Ph.D. Integrated Ph.D.

ADVISOR APPROVAL

APPROVED
Prof. Dr. Hakan TÜZÜN

EK-I: Yayınlama ve Fikrî Mülkiyet Hakları Beyanı

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan "**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**" kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H.Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- o Enstitü/Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir. ⁽¹⁾
- o Enstitü/Fakülte yönetim kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren ... ay ertelenmiştir. ⁽²⁾
- o Tezimle ilgili gizlilik kararı verilmiştir. ⁽³⁾

06 / 09 /2019

Fatih ÖZER

"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"

- (1) Madde 6. 1. Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.
- (2) Madde 6. 2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç; imkânı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.
- (3) Madde 7. 1. Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir*. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.
Madde 7.2. Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir

* Tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu tarafından karar verilir.

