



ONDOKUZ MAYIS ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ

ANABİLİM DALI

**PROGRAMLAMA ÖĞRETİMİNDE ROBOTİK VE SCRATCH
UYGULAMALARININ ÖĞRENCİLERİN BİLGİ İŞLEMSEL
DÜŞÜNME BECERİLERİ VE AKADEMİK BAŞARILARINA
ETKİSİ**

Elif ŞİMŞEK

Danışman

Dr. Öğr. Üyesi Polat ŞENDURUR

YÜKSEK LİSANS TEZİ

Haziran, 2018

TELİF HAKKI

Bu tezin tüm hakları saklıdır. Kaynak göstermek koşuluyla tezin teslim tarihinden itibaren(.....) ay sonra tezdin fotokopi çekilebilir.

YAZARIN

Adı : Elif

Soyadı : ŞİMŞEK

Bölümü : Bilgisayar ve Öğretim Teknolojileri

İmza :

Teslim Tarihi :

TEZİN

Türkçe Adı : Programlama Öğretiminde Robotik ve Scratch Uygulamalarının Öğrencilerin Bilgi İşlemsel Düşünme Becerileri Ve Akademik Başarısına Etkisi

İngilizce Adı : The Effect Of Robotics And Scratch Applications On Computational Thinking Skills And Academic Achievement In Programming Instruction

ETİK İLKELERE UYGUNLUK BEYANI

Tez yazma sürecinde bilimsel ve etik ilkelere uyduğumu, yararlandığım tüm kaynakları kaynak gösterme ilkelerine uygun olarak kaynakçada belirttiğimi ve bu bölümler dışındaki tüm ifadelerin şahsıma ait olduğunu beyan ederim.

Yazar Adı Soyadı:

İmza:

KABUL VE ONAY

Elif ŞİMŞEK tarafından hazırlanan “Programlama Öğretiminde Robotik ve Scratch Uygulamalarının Öğrencilerin Bilgi İşlemsel Düşünme Becerileri ve Akademik Başarısına Etkisi” adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Ondokuz Mayıs Üniversitesi **Bilgisayar ve Öğretim Teknolojileri Eğitimi** Anabilim Dalı’nda Yüksek Lisans tezi olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Polat ŞENDURUR

Bilgisayar ve Öğretim Teknolojileri Eğitimi, Ondokuz Mayıs Üniversitesi

Başkan: Doç. Dr. İbrahim ÇETİN

Bilgisayar ve Öğretim Teknolojileri Eğitimi, Abant İzzet Baysal Üniversitesi

Üye: Dr. Öğr. Üyesi Hatice Gökçe BİLGİÇ DOĞAN

Bilgisayar ve Öğretim Teknolojileri Eğitimi, Ondokuz Mayıs Üniversitesi

Bu tezin **Bilgisayar ve Öğretim Teknolojileri Eğitimi** Anabilim Dalı’nda Yüksek Lisans tezi olması için şartları yerine getirdiğini onaylıyorum.

Tarihi: __/__/____

Prof. Dr. Ali ERASLAN

Eğitim Bilimleri Enstitüsü Müdürü

(İmza ve Mühür)

Çok sevgili aileme



TEŞEKKÜRLER

Bu araştırma pek çok bilişim teknolojileri öğretmeninin takip ettiği ve kullandığı iki yöntemi, görsel programlamayı ve robot programlamayı karşılaştırarak bu iki yöntemin bilgi işlemsel düşünme ve akademik başarı üzerindeki etkisini incelemeyi amaçlamıştır. Araştırma konusunun belirlenmesinde ve yazım sürecinin her aşamasında engin bilgi ve deneyimini, değerli zamanını esirgemeyerek çalışmayla yakından ilgilenen, yol gösteren danışman hocam Polat ŞENDURUR'a sonsuz teşekkürlerimi sunarım.

Çalışma boyunca bana destek olan yüksek lisans mesai arkadaşım Cennet TERZİ'ye, hayatımın her evresinde yanımda olan çok değerli aileme teşekkürü bir borç bilirim.

PROGRAMLAMA ÖĞRETİMİNDE ROBOTİK VE SCRATCH UYGULAMALARININ ÖĞRENCİLERİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİLERİ VE AKADEMİK BAŞARISINA ETKİSİ

Yüksek Lisans Tezi

Elif ŞİMŞEK

ONDOKUZ MAYIS ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ

Haziran, 2018

ÖZ

Bu araştırmanın amacı programlama öğretimi sürecinde öğrencilerin bilgi işlemsel düşünme becerilerine ve akademik başarılarına görsel programlama ve robotik programlama etkinliklerinin etkisini ortaya çıkarmaktır. Araştırmaya iki gruba ayrılan toplam altmış öğrenci katılmıştır. Araştırma öncesinde öğrencilerin temel bilgisayar bilgisi ölçülmüştür. Daha sonra öğrencilere bir ay boyunca görsel programlama ve robotik eğitimi verilmiştir. Ardından Scratch ve mBlock ortamlarına uygun birbirine eşdeğer programlama başarı testleri uygulanmıştır. Öğrencilerin bilgi işlemsel düşünme pratikleri öğrencilerle yapılan görüşme ile ölçülmüştür. Daha sonra gruplar yer değiştirilerek 1. gruba mBlock, 2. gruba Scratch eğitimi verilmiştir. Eğitim sonrası ölçüm tekrarlanmıştır.

Araştırmada nicel araştırma yöntemlerinden yarı deneysel araştırma yönteminin sınıfta kontrol grup tasarımı kullanılmıştır. Veri toplama aracı olarak akademik başarı testleri ve bilgi işlemsel düşünme pratiklerini ölçen görüşme formu kullanılmıştır.

Araştırma sonuçları incelendiğinde hem akademik başarıda hem de bilgi işlemsel düşünme pratiklerinde iki grubun da eşdeğer puanlar aldığı gözlemlenmiştir. Sonuçlar programlama temel eğitimi için iki yöntemin de kullanılabilir olduğu şeklinde yorumlanmıştır. Çalışma alan yazına katkı sağlayarak, bilişim teknolojileri öğretmenleri ve STEM alanında çalışan diğer branştaki öğretmenlere, öğrencilerine programlama eğitimi verirken nasıl bir yol takip etmeleri hakkında rehberlik sunabilecek sonuçlar ortaya koymuştur.

Anahtar Kelimeler : Bilgi işlemsel düşünme, akademik başarı, scratch, mblock, görsel programlama, robotik, programlama öğretimi

Sayfa Sayısı : 93

Danışman : Dr. Öğr. Üyesi Polat ŞENDURUR



THE EFFECT OF ROBOTICS AND SCRATCH APPLICATIONS ON COMPUTATIONAL THINKING SKILLS AND ACADEMIC ACHIEVEMENT IN PROGRAMMING INSTRUCTION

MS Thesis

Elif ŞİMŞEK

ONDOKUZ MAYIS UNIVERSITY

GRADUATE SCHOOL OF EDUCATIONAL SCIENCES

April, 2018

ABSTRACT

The aim of this research is to reveal the effect of visual programming and robotic programming activities on the students' computational thinking skills and academic achievement during the programming instruction process. A total of 60 students, who were divided in two groups, participated in the research. The background computer knowledge of the students was measured before the research. Then students were trained in visual and robotic programming for a month. Next, achievement tests, which are equivalent in Scratch and mBlock environments, were applied. The students' practices of computational thinking was measured by interviewing. After that, the groups were changed and given mBlock training for Scratch group and Scratch training for mBlock group. Then the measurement was repeated.

In the research, the posttest control group design of the semi-experimental research method from the quantitative research methods was used. Academic achievement tests and interview forms that measure computational thinking practices were used as data collection tools.

At the end of the research, it was concluded that there was no difference between groups in academic achievement and in computational thinking practices. Interpretation of results revealed that two of the applications can be used for basic programming education. This research is considered as being able to make contribution to the field and literature. In addition results have the potential to offer

guidance to information and communication technology teachers and teachers working on the field of STEM by providing programming training.

Key Words : Computational thinking, academic achievement, scratch, mblock, visual programming, robotics, programming instruction

Number of Pages : 93

Advisor : Dr. Polat ŞENDURUR



İÇİNDEKİLER

TELİF HAKKI.....	II
ETİK İLKELERE UYGUNLUK BEYANI.....	III
KABUL VE ONAY	IV
TEŞEKKÜRLER	VI
ÖZ.....	VII
ABSTRACT	IX
TABLolar LİSTESİ.....	XIII
BİRİNCİ BÖLÜM.....	1
I. GİRİŞ.....	1
1.1 Araştırma Konusu ve Problemi	3
1.2 Araştırmanın Amacı.....	5
1.3 Araştırmanın Problem Cümlesi.....	5
1.4 Araştırmanın Alt Problemleri.....	5
1.5 Sayıtlar	5
1.6 Sınırlılıklar	6
1.7 Araştırmanın Önemi	6
İKİNCİ BÖLÜM	8
II. KURAMSAL ÇERÇEVE.....	8
2.1 Bilgi İşlemsel Düşünme	8
2.2 Programlama ve Programlama Öğretimi	12
2.3 Scratch Öğretimi	16
2.4 Robotik ve Eğitsel Robotik	21
2.5 mBlock Öğretimi	22
ÜÇÜNCÜ BÖLÜM	30
III. YÖNTEM	30
3.1 Araştırmanın Deseni	30
3.2 Evren ve Örneklem	30
3.3 Veri Toplama Araçları.....	33
3.4 Süreç	39
3.5 Analiz.....	47
3.6 Geçerlilik ve Güvenirlik.....	48
DÖRDÜNCÜ BÖLÜM	49

IV. BULGULAR.....	49
4.1 Akademik Başarı İle İlgili Sonuçlar	49
4.2 Bilgi İşlemsel Düşünme Pratikleri İle İlgili Sonuçlar.....	56
BEŞİNCİ BÖLÜM	61
V. SONUÇ, TARTIŞMA VE ÖNERİLER	61
5.1 Özet.....	Hata! Yer işareti tanımlanmamış.
5.2 Sonuç ve Tartışma	Hata! Yer işareti tanımlanmamış.
5.2.1 Akademik Başarının Karşılaştırılması	Hata! Yer işareti tanımlanmamış.
5.2.2 Bilgi İşlemsel Düşünme Pratiklerinin Karşılaştırılması	Hata! Yer işareti tanımlanmamış.
5.3 Öneriler	71
KAYNAKÇA	73
EKLER.....	80

TABLolar LİSTESİ

Tablo 1: Öğrenci Tablosu.....	33
Tablo 2: Etkinlik ve Kazanım Tablosu	34
Tablo 3: Deneme ve Yanılma Puanlama Ölçeği	36
Tablo 4: Test Etme ve Hata Ayıklama Puanlama Ölçeği	37
Tablo 5: Yeniden Kullanma ve Karıştırma Puanlama Ölçeği.....	38
Tablo 6: Özetleme ve Modüler Hale Getirme Puanlama Ölçeği	39
Tablo 7: Temel Bilgisayar Becerisi Testi Tanımlayıcı Verileri.....	50
Tablo 8: 1. Sınava Ait Tanımlayıcı Veriler.....	50
Tablo 9: 2. Sınava Ait Tanımlayıcı Veriler.....	51
Tablo 10: Gruplar arası etki	53
Tablo 11: Gruplar arası etki	55
Tablo 12: Bilgi İşlemsel Düşünme Pratikleri Tanımlayıcı İstatistikleri	56
Tablo 13: Box's Test	58
Tablo 14: Çoklu değişken testi.....	59
Tablo 15: Levene Testi 2.....	60

ŞEKİLLER LİSTESİ

Şekil 1: Blok Paleti	17
Şekil 2: Kodlama Alanı	18
Şekil 3: Proje Ekranı	18
Şekil 4: Scratch Ekranı.....	19
Şekil 5: Blok Paleti	23
Şekil 6: Kodlama Alanı.....	24
Şekil 7: Proje Ekranı	25
Şekil 8: mBot Monte Etme (https://forum.makeblock.com/t/mbot-line-sensor-not-working-out-of-box/7678 kaynağından alınmıştır)	26
Şekil 9: mBot Paket İçeriği(https://forum.makeblock.com/t/mbot-line-sensor-not-working-out-of-box/7678 kaynağından alınmıştır)	27
Şekil 10: mBot Bağlantılar ve Sensörler (https://forum.makeblock.com/t/mbot-line-sensor-not-working-out-of-box/7678 kaynağından alınmıştır)	28
Şekil 11: Gruplara Sorulan Sorular	35
Şekil 12: Öğrenciler Scratch ile Programlama Yaparken	42
Şekil 13:Scratch Penguen Dansı Etkinliği	43
Şekil 14:Robotik Penguen Dansı Etkinliği	43
Şekil 15:mBot Yangın Var Etkinliği.....	44
Şekil 16:mBot Park Et Etkinliği.....	44
Şekil 17:Öğrenciler mBot Programlarken.....	45
Şekil 18: Scratch Programlama.....	45
Şekil 19:Bağımlı ve kovaryans değişkenleri arası scatter plot grafiği.....	52
Şekil 20:Bağımlı ve kovaryans değişkenleri arası scatter plot grafiği.....	54

SİMGELER VE KISALTMALAR

MEB	Milli Eğitim Bakanlığı
YÖK	Yüksek Öğretim Kurumu
TBBT	Temel Bilgisayar Becerisi Testi
1. Grup	Önce Scratch Sonra mBlock Eğitimi Alanlar
2. Grup	Önce Mblock Sonra Scratch Eğitimi Alanlar

BİRİNCİ BÖLÜM

I. GİRİŞ

Bilişim teknolojileri öğretmeni olarak görsel programlama yazılım ortamı olan Scratch'ı kullandığım 3 yıllık süreçte Scratch programlama dili öğrencilerin programlama kavramlarıyla zorlanmadan tanışmalarına olanak sağlamıştır. Scratch'ın hata mesajı vermemesi, sürükle bırak gibi kolay bir yöntemi içermesi, anında görsel çıktılar vermesi, yapılan çalışmalarını paylaşma imkanı sunması, sadece programlamaya değil tasarıma da yer vermesi gibi avantajlarının programlama eğitimine giriş sürecinde öğrencilere oldukça fazla katkı sağlamıştır. Ayrıca öğrenciler metin tabanlı programlamada yaşadıkları zorlukların çoğunu bu görsel programlama aracında yaşamamış, programlamaya karşı olumsuz bir tutum geliştirmemiş, programlamanın çok zor olduğunu düşünmemiştir.

Görsel programlama araçlarının yaygınlaşmasının ardından robotik de her ne kadar ilk kez 1969 yılında Seymour Papert ve Logo programıyla kodlanan Kaplumbağa robotuyla programlama öğretiminde kullanılmaya başlanmış olsa da son yıllarda bilgi işlemsel düşünme becerisi kavramının popülerleşmesiyle yeniden gündeme gelmiştir. Pek çok araştırmaya konu olmuş, programlama öğretiminde tercih edilmeye başlanmıştır (Barr ve Stephenson, 2011; Feurzeg, 2006). Robotik, programlama öğretimini desteklediği ve disiplinlerarası etkinliklerde de kullanılabilirdiği için pek çok ulusal ve uluslararası proje ve yarışmanın da içeriğini oluşturmuştur. Uluslararası olan projelere TERECOP, ER4STEM, Centrobot; yarışmalara ise RobotChallenge, First Lego League, RoboCup Junior, Robo League, World Robot Olympiad WRO Hellas örnek olarak verilebilir. Ulusal olan projelere Kodlasam, KodluYoz, KodlaManisa, KodlaRize, Turkcell Geleceği Yazanlar, Kod Adı 2023 yarışmalara ise RobotAfyon, Robiyaman, ODTÜ Robot Günleri, Yıldız Savaşları örnek olarak verilebilir. 2007 ve 2017 yılları arasındaki eğitimde robotik kullanımı ile ilgili yayınlanan 45 makalenin incelendiği çalışmada STEM, fen ve teknoloji, robot uygulamaları, bilgisayar bilimleri, kamplar, yabancı dil gibi

branşlarla ilgili çalışıldığı sonucuna varılmıştır (Yolcu ve Demirer, 2010). Ayrıca incelenen makalelerde robotik kullanımının etkili olduğu görülmüştür.

Scratch programlama dili, bilişim sınıfına sahip bir okul için herhangi bir erişilebilirlik problemi oluşturmamaktadır. Hem internet arayüzü kullanılarak hem de masaüstü yazılımı aracılığıyla kullanılabilir. İnternet üzerinden de öğrenciler scratch.mit.edu adresinden kayıt olarak çalışmalarını yapabilmektedir. Fakat robotik Scratch'a göre oldukça maliyetli olmaktadır. Çünkü robotik eğitimi için robot setlerinden biri tercih edilmelidir. Fiyatı düşük olan bir robot seçildiğinde ise dayanıklılık, pil ömrü, yetersiz sensör sayısı ve benzeri durumlara ek olarak arayüzünün kullanılabilirlik problemleri ile karşılaşmaktadır. Fiyatı yüksek olan robotlar sağlam olmakta aynı zamanda yeterli sensör sayısına ve anlaşılır/kullanımı kolay bir arayüze sahip olabilmektedir. Ancak bu robotlar pek çok devlet okulunun ya az sayıda sahip olabileceği ya da hiç sahip olamayacağı robotlardır. Az sayıda olan robotlar da öğrenci mevcudunun fazla olduğu öğrenme ortamlarında oldukça yetersiz kalmaktadır.

Scratch gibi görsel programlama araçları ve bunun yanında programlamayı bir sonraki boyuta geçiren robot programlama ya da robotik etkinlikler hem benzer özellikler taşımakta hem de belli alanlarda birbirlerinden farklılaşmaktadır. Robotiğin fiziksel çevre ile etkileşime geçmesi ya da görsel programlama ortamlarının program çıktılarını hızlı ve görsel olarak sunabilmesi bu iki ortam arasındaki farklılığa örnek olarak verilebilir. Ortam farklılıkları programlama becerisinin gelişimi ve bilgi işlemsel düşünme becerilerinin kazanılması süreçlerini etkileme potansiyeline sahiptir. Bu noktadan yola çıkarak, Scratch gibi bir görsel programlama aracının ve benzer bir ortamda robot programlama süreçlerinin hem programlama becerisi hem de bilgi işlemsel düşünmenin bir alt boyutu olan bilgi işlemsel düşünme pratikleri üzerine ne şekilde etki ettiklerini ortaya çıkarmak bu araştırmanın temel amacını oluşturmaktadır.

1.1 Araştırma Konusu ve Problemi

21. yüzyılda değişen yaşam şartlarıyla beraber eğitimin hedefleri ve toplumun bireylerden beklentileri de değişmiştir. Bu değişimlerin bireylerde gerektirdiği beceriler değişken olmakla beraber problem çözme, kritik düşünme, iletişim, esneklik ve adapte olabilme, bilgi ve teknoloji okuryazarlığı, küresel yetkinlikler ve finansal okuryazarlık olarak belirlenmiştir (The Partnership for 21st Century Skill, 2009).

Wing (2006) bilgi işlemsel düşünme becerisinin 21. yüzyıl becerilerinden biri olduğunu iddia etmektedir. Bundy (2007) ise bilgi işlemsel düşünmenin küçümsenmeyecek entellektüel bir devrim olduğunu, araştırmacılara yeni türde sorular sorma ve yeni türde cevapları kabul etme imkânı sunduğunu söylemiştir. 21. yüzyılı anlamak için öncelikle bilgi işlemsel düşünmeyi anlamalısınız ifadesi ise bilgi işlemsel düşünme becerisinin içinde bulunduğumuz yüzyıl için önemini vurgulamaktadır.

Bilgi işlemsel düşünmeyi Cuny, Snyder ve Wing (2010) bir bilgi işlemcisi (bilgisayar, makine, insan) tarafından etkili bir biçimde gerçekleştirilecek şekilde problemlerin formüle edildiği düşünme süreci olarak tanımlamıştır. 21. yüzyıl becerilerinden biri olarak kabul edilen kodlama becerisi mantıksal akıl yürütme becerisinin bir parçası olarak görülmektedir (Dg Connect, 2014). Bilgi işlemsel düşünme becerisi iletişim, sosyal yeterlilik, takım halinde çalışma gibi beceriler dahil edilerek öğretilir. Kodlama soyut kavramları somutlaştırabilir, yaratıcılık, takımla çalışma ve problem çözme gibi becerilerle birleştirilebilir. Bunlar da kodlamayı bilgi işlemsel düşünme becerisinin kazanılmasında etkili bir araç yapabilmektedir (Education and training 2020 [ET 2020], 2016).

Pfeiffer (1962) bilgisayarların bizler için büyük yardımcıları olduğunu ve sadece etrafımızda bilgisayarların olmasının bile düşünce biçimimizi değiştirmemiz için yeterli olduğunu söylemiştir. Bilgi işlemsel düşünme becerisi Pfeiffer'in sözünü ettiği değişen düşünme biçimidir. Smith, Cypher ve Tesler (2000) yaptıkları araştırmada C++ ve Java gibi geleneksel programlama dillerinin bilgisayarın düşünme biçimine çok yakın bir temsil gücüne sahip olduğunu göstermiştir. Fakat görsel programlama dilleri ise alana özgü oldukları için geleneksel dillere göre güçlü

olmamalarına rağmen insan diline daha yakındır (Lye ve Koh, 2014). K-12 öğrencilerine bilgi işlemsel düşünmenin kavramlar, pratikler ve bakış açılarından oluşan üç boyutunu kazandırabilmek için görsel programlama araçlarını tercih etmek geleneksel programlama dillerini tercih etmekten daha iyi olacaktır. Çünkü görsel programlama dilleri noktalı virgül ve kıvrımlı parantez kullanımı gibi gereksiz söz dizimini azaltmaktadır. Bunlar da bilişsel yükün azalmasına ve öğrencilerin program yazmanın mekaniği yerine programlamanın mantığı ve yapılarına odaklanmasını sağlamaktadır (Kelleher ve Pausch, 2005, s. 131). Ayrıca görsel programlama araçları öğrencilerin bilgi işlemsel düşünme becerilerinin bilgi işlemsel düşünme pratikleri boyutunu, çıktılarını animasyonlu nesnelere şeklinde görülebilir kıldığı için sergilemektedir (Lye ve Koh, 2014). Bu tür görselleştirmeler ise bilgi işlemsel düşünme pratiklerinin test etme ve hata ayıklama gibi alt boyutlarını kolaylaştırarak öğrencilerin problem çözme alt boyutunu daha kolay elde etmelerini sağlamaktadır.

Son zamanlarda kodlama becerisinin bireylere kazandırılmasında MIT tarafından geliştirilen Scratch oldukça popüler bir görsel programlama aracıdır (Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010) . MIT tarafından geliştirilen, Scratch ile hemen hemen aynı arayüze sahip mBlock görsel programlama aracı da robotik uygulamaları için sunduğu programlama ortamıyla dikkat çekmektedir.

İki programlama ortamı da kodların belirli bir mantık sırasında sürüklenip bırakılmasıyla kullanıcılara kolay, pratik ve eğlenceli bir programlama deneyimi sunmaktadır. Bu da araştırmada bu iki programlama ortamının tercih edilme sebeplerinden biridir. mBlock ve Scratch benzer arayüzlere sahip olmasına rağmen araştırmada Scratch robotsuz, mBlock ise sadece robot etkinliklerinde kullanılmıştır.

Programlama eğitiminde kodlarını robotlar aracılığıyla somutlaştıran, yazmış oldukları kodları robotun hareketlerinde görebilen öğrencilerin, Scratch ile oluşturmuş oldukları kodların çıktılarını sadece ekranda gören öğrencilere göre bilgi işlemsel düşünme pratikleri ve akademik başarıları arasında anlamlı bir fark olup olmadığı çalışmanın yanıt aradığı soruların temelini oluşturmaktadır.

Literatürde Scratch ve robotik eğitimi ile ilgili birçok çalışma yer almaktadır fakat bu ikisi arasındaki bilgi işlemsel düşünme becerilerini ve akademik başarıyı karşılaştıran ve inceleyen çalışmalar oldukça azdır. Dolayısıyla bu çalışmanın alan yazına katkı

sağlayabileceği, bilişim teknolojileri öğretmenleri ve STEM alanında çalışan diğer branştaki öğretmenlere, öğrencilere programlama eğitimi verirken nasıl bir yol takip etmeleri hakkında rehber olabileceği düşünülmektedir.

1.2 Araştırmanın Amacı

Araştırmanın amacı programlama öğretiminde kodlarını Scratch ekranı üzerinden test eden ve çalıştıran öğrenciler ile mBlock'ta oluşturulmuş kodlarını robotlarının (mbot) hareketleriyle test edip çalıştıran öğrenciler arasındaki bilgi işlemsel düşünme pratikleri ve akademik başarı değişkenlerini karşılaştırmaktır. Bu karşılaştırmanın sonuçlarından yola çıkarak programlama öğretiminde izlenmesi gereken yollara dair veriye dayalı kanıtlar ortaya konması hedeflenmiştir.

1.3 Araştırmanın Problem Cümlesi

Milli Eğitim Bakanlığına bağlı ortaokulda öğrenim görmekte olan 10-11 yaş aralığındaki (5. Sınıf) öğrencilere verilen programlama öğretiminde Scratch kullanarak kodlarını Scratch ekranı üzerinden test eden ve çalıştıran öğrenciler ile mBlock ile kodlarını robotlarının (mbot) hareketleriyle test edip çalıştıran öğrenciler arasındaki bilgi işlemsel düşünme pratikleri ve akademik başarıları arasında anlamlı bir fark var mıdır?

1.4 Araştırmanın Alt Problemleri

1. Scratch görsel programlama dili ile verilen programlama eğitimi ile robot destekli programlama eğitimini alan öğrencilerin programlamaya yönelik akademik başarıları arasında bir fark var mıdır?
2. Scratch görsel programlama ortamı kullanılarak verilen programlama eğitimi ile robot destekli programlama eğitimini alan öğrencilerin bilgi işlemsel düşünme pratikleri arasında bir fark var mıdır?

1.5 Sayıtlar

Bu araştırmada aşağıdaki sayıtlardan hareket edilmiştir.

1. Araştırmada, öğrencilerin araştırma sırasında kullanılan ölçme araçlarına samimi ve doğru cevaplar verdikleri varsayılmıştır.

2. Scratch ve mBlock ile programlama öğrenen öğrencileri değerlendirmek için hazırlanmış olan aynı öğrenme alanlarını kapsayan etkinliklerin birbirine eşdeğer olduğu varsayılmıştır. Öğrencilerle yaptıkları projelerle ilgili görüşmeler yapılacaktır.
3. 1. ve 2. grubun yer aldığı sınıflarda bulunan akademik başarı notları değerlendirilmeyen ve bilgi işlemsel pratikleri ile ilgili görüşmeler yapılmayan öğrenciler öğrenme ortamında etkisiz varsayılmıştır.

1.6 Sınırlılıklar

1. Araştırma, 2016-2017 öğretim yılında Samsun ili Tekkeköy ilçesinde yer alan Tekkeköy İmam Hatip Ortaokulu 5. sınıf öğrencileri ile sınırlıdır.
2. Değişkenler Scratch ve mBlock uygulamalarıyla sınırlıdır.
3. Araştırma, değişkenleri ölçmek için geliştirilen ölçeklerle toplanan bilgilerle sınırlandırılmıştır.
4. Araştırmada eğitimler dört farklı sınıfta verilmiştir. Görüşmelerin yapılması ve akademik başarı testlerinin değerlendirilmesi ise verilerin toplanması ve değerlendirilmesi sürecini kolaylaştırmak için bu sınıflardan rastgele olarak seçilen onbeşer öğrenci ile sınırlandırılmıştır.

1.7 Araştırmanın Önemi

Programlamanın öneminin arttığı günümüzde programlama öğretimi de oldukça önemli hale gelmiştir. Üniversite ve liselerde verilen programlama dersi ilk ve ortaokullarda verilmeye başlanmış, hatta eğitim okul öncesine kadar inmiştir. Bu durum küçük yaş gruplarında uygulanabilir ve etkili bir programlama öğretimi için daha çok araştırma ve çalışma yapılmasını doğurmuştur. Programlamanın daha küçük yaş gruplarında öğretimi ile ilgili pek çok araştırma yapılmıştır ve yapılmaktadır. Küçük yaş gruplarında programlama öğretimi nasıl olmalı, hangi yöntemler kullanılmalı, hangi araçlar tercih edilmeli, programlama öğretimi ile öğrencilere hangi beceriler ne kadar etkili kazandırılıyor, programlama öğretimi disiplinlerarası öğretimi destekler mi, hangi disiplinlerin verilmesinde daha etkili olur şeklinde uzayıp giden sorulara cevap aranmaktadır.

Pfeiffer (1962) bilgisayarların bizler için büyük yardımcıları olduğunu ve sadece etrafımızda bilgisayarların olmasının bile düşünce biçimimizi değiştirmemiz için yeterli olduğunu söylemiştir. Yıllar sonra Papert (1996) ise bilgi işlemsel düşünme becerisi kavramını dile getirmiştir. Bilgi işlemsel düşünme becerisi Cuny, Snyder ve Wing (2010) tarafından bir bilgi işlemcisi (bilgisayar, makine, insan) tarafından etkili bir biçimde gerçekleştirilecek şekilde problemlerin formüle edildiği düşünme süreci olarak tanımlanmıştır. Wing (2006, s. 35) in “Sadece bilgisayar bilimciler için değil herkes için temel bir beceri” adlı makalesiyle bilgi işlemsel düşüncenin önemini vurgulanmış ve yaygınlaşarak, popülerleşmiştir.

Bilgi işlemsel düşünme becerisinin müfredatta yerini alması için 3 yaklaşım mevcuttur (Weinberg, 2013). Bunlar bilgisayarsız öğrenme, oyun ve robot programlama, başlangıç öğrenme ortamlarıdır. Bilgisayarsız öğrenme çeşitli kart oyunları ve fiziksel etkinlikleri, oyun ve robot programlama oyun tasarlama-programlama ve robot programlama etkinliklerini, başlangıç öğrenme ortamları ise scratch ve code.org gibi görsel programlama araçlarıyla oluşturulan etkinlikleri içermektedir. Araştırmada bu yaklaşımlardan robot programlama ve başlangıç öğrenme ortamları yaklaşımları tercih edilmiştir. Araştırmada sadece bilgi işlemsel düşünme pratikleri değil akademik başarıda değerlendirilmiştir.

Bu araştırma programlama öğretiminde takip edilen yöntemlerden görsel programlama ve robot ile programlama öğretimlerinden hangisinin bilgi işlemsel düşünme pratikleri ve programlamaya yönelik akademik başarının artırılmasında daha etkili olduğu sorusuna cevap aramak için yapılmıştır. Bu araştırmanın sonuçlarından yola çıkarak programlama öğretiminde izlenmesi gereken yollara dair veriye dayalı kanıtlar ortaya konması hedeflenmiştir.

Mannila ve diğerleri (2014) öğretmenlerin bilgi işlemsel düşünmenin öğretimi ile ilgili ortak bakış açılarının olmadığını ve bilgi işlemsel düşünmenin öğretim programlarına nasıl eklenebileceğiyle ilgili araştırmalara ihtiyaç olduğunu belirtmiştir. Son zamanlarda ise bilgi işlemsel düşünmeyle ilgili yapılan çalışmalarda bilgi işlemsel düşünmenin öğretim programına ve STEM alanına entegre edilmesi ile ilgili araştırma sayısında artış görülmüştür (Özçınar, 2017). Yapılan araştırmanın sonuçları kullanılarak programlama öğretiminin bilgi işlemsel düşünme becerisi odaklı, daha verimli ve etkili olabileceği düşünülmektedir. Araştırma bilişim

teknolojileri öğretmenlerine kodlama öğretiminde yol gösterici, yeni araştırma konularına rehberlik edebilecek nitelikleri taşımaktadır.

İKİNCİ BÖLÜM

II. KURAMSAL ÇERÇEVE

Bu bölümde bilgi işlemsel düşünmeden, programlamadan ve programlama öğretiminden, görsel programlamadan, görsel programlama aracı olan Scratch'tan ve de hem görsel programlama aracı olan hem de robot programlamaya imkan sunan mBlock'tan bahsedilmiştir.

2.1 Bilgi İşlemsel Düşünme

Wing (2006) bilgi işlemsel düşünmenin 21. yüzyılın temel becerilerden biri olacağını iddia etmiş ve bilgi işlemsel düşünmeyi yaygınlaştırmıştır fakat Papert (1980) 26 yıl öncesine dayanan bilgi işlemsel düşünmenin ilk ve orta öğretimde öğretimi fikrini Wing'den çok daha önceleri ortaya atmıştır.

Wing (2006) bilgi işlemsel düşünmeyi bilgisayar bilimlerini temel alan kavramları kullanarak problem çözmenin, sistem tasarlamının ve insan davranışlarını anlamının bir yolu olarak tanımlamıştır. Cuny, Snyder ve Wing (2010) ise bir bilgi işlemcisi (bilgisayar, makine, insan) tarafından etkili bir biçimde gerçekleştirilecek şekilde problemlerin formüle edildiği düşünme süreci olarak tanımlamıştır. Ayrıca bu beceriyi bir bilgisayar, insan ya da hem insan hem de bilgisayar tarafından üretilen bir çözümü kabul ederek problemin formüle edilmesi sürecindeki zihinsel etkinlik olarak da tarif etmişlerdir. Özden (2015) ise bilgisayarları, karşılaşılan problemlerin çözümünde üretim aracı olarak kullanabilmek için gereken bilgi, beceri ve tutumlara sahip olmayı bilgi işlemsel düşünme becerisi olarak tanımlamıştır. Wing (2006) bilgisayarların sıkıcı olduğunu, insanın bilgisayardan çok daha akıllı ve yaratıcı

olduğunu, bilgisayarları etkileyici hale insanların getirdiğini, insanların hayalgücüyle sınırlı olan fonksiyonel sistemler inşa edebileceğini söylemiştir.

Bundy (2007) bilgi işlemsel düşünmenin hem fen hem de sosyal bilimlerde araştırmaları etkilediğini ve büyük miktarlardaki verilerin kolayca işlenerek yeni soruların sorulmasını ve yeni cevaplara kolayca ulaşılmasını sağladığını belirtmiştir. Curzon (2015) ise bilgi işlemsel düşünmeyi insanlar için problem çözme becerisi olarak tanımlamıştır. ISTE (International Society for Technology in Education) (2015)'e göre ise bilgi işlemsel düşünme problem çözme yaklaşımlarından biridir ve düşünce ile teknoloji ilişkisini güçlendirmektedir.

2016'da ikincisi düzenlenen akran öğrenme etkinliğinde (ET 2020, 2016) bilgi işlemsel düşünme problemlerin formüle edilmesi ve çözümünde bilgisayar bilimleri kavramlarını kullanabilme becerisi olarak tanımlanırken, bir bilgisayar bilimcisi gibi düşünme olarak da özetlenmiştir. Eğitim alanında hızla yükselen bir trend olan bilgi işlemsel düşünme becerisinin mantıklı düşünme, problem çözme ve dijital yetkinlik kazandırma potansiyeline sahip olduğu vurgulanmıştır.

Barr, Harrison ve Conery'e (2011: ss 21) göre bilgi işlemsel düşünme aşağıdaki adımları içeren bir problem çözme sürecidir:

- Bilgisayar veya diğer araçların kullanılarak problemin formülleştirilmesi
- Verilerin mantıklı bir şekilde düzenlenerek analiz edilmesi
- Modelleme ve simülasyon gibi soyutlama ile verilerin sunulması
- Algoritmik düşünme ile çözümlerin otomatikleştirilmesi
- En etkili, yeterli aşamalar seçilerek ve kaynaklar kullanılarak tanımlama, analiz etme ve uygun çözümleri uygulama
- Problem çözme sürecinin çeşitli problemlere genellenmesi ve dönüştürülmesi

Yukarıdaki adımlar bilgi işlemsel düşünme becerisinin pek çok beceriyi kapsadığını göstermektedir. ISTE (2015) bilgi işlemsel düşünme becerisinin eleştirel düşünme, yaratıcı düşünme, problem çözme, algoritmik düşünme, iletişim becerileri ve

işbirlikli öğrenmeyi kapsadığını ifade etmekte ve bu becerilerle tanımlanabileceğini vurgulamaktadır.

Bilgi işlemsel düşünme becerisi iletişim, sosyal yeterlilik, takım halinde çalışma gibi beceriler dahil edilerek öğretilir. Kodlama soyut kavramları somutlaştırabilir, yaratıcılık, takımla çalışma ve problem çözme gibi becerilerle birleştirilebilir. Bunlar da kodlamayı bilgi işlemsel düşünme becerisinin kazanılmasında etkili bir araç yapabilir (ET 2020, 2016.).

2016 yılında ISTE öğrenciler için standartlar belirlemiştir. Bu standartlardan biri de bilgi işlemsel düşünür (Computational Thinker) olmalıdır. Her öğrencinin sahip olması gereken 21. Yüzyıl becerilerinden biri olarak karşımıza çıkan bilgi işlemsel düşünmenin kazandırılmasında kullanılan yöntemlerden biri olan programlama birçok ülkenin ilk ve ortaöğretim programında zorunlu olarak yer alırken bazı ülkelerin öğretim etkinliklerinin bir parçası olarak desteklenmiştir (Grover ve Pea, 2013; Mannila ve diğerleri, 2014). 2014 yılında İngiltere’de programlama ilkokuldan itibaren zorunlu ders olmuştur (UK Digital Skills Taskforce, 2014). Anayasasında ücretsiz internetin bir vatandaşlık hakkı olduğu ibaresi bulunan tek ülke Estonya’da ProgeTiiger adı verilen öğretim programı ile 7 yaşından itibaren programlama öğretilmektedir (Demirkan, 2016; Robertson, 2012). 2016’da ise Finlandiya’da kodlama ilkokul öğrencileri için zorunlu ders olmuştur (Hiltunen, 2016). İsrail’de 1995 yılından günümüze kadar liselerde bilgisayar bilimleri müfredatı güncellenerek uygulanmaktadır. Ayrıca 2011 yılında uygulamaya başladıkları STEP (Science and Technology Excellence Program) adlı program ile ortaokullarda da bilgisayar bilimleri müfredatına yer vermeye başlanmıştır (Gal-Ezer ve Stephenson, 2014). Japonya’da ise 2020 yılında ilkokulda, 2021’de ortaokulda, 2022’de ise lisede zorunlu ders olması gündemdedir (Bethune, 2016).

Ülkemizde ise bir dönem Bilişim Teknolojileri dersinin etkililiği seçmeli ders olması, sadece bir ders saati olması ve dersin daha çok okuma olarak işlenmesi, yazma boyutunun gerektiği kadar etkili olmaması yüzünden uzun süre eleştirilmiştir (Özdiç ve Altun, 2014). 2013-2014 eğitim öğretim yılına başlarken dersin içeriği ve niteliği bu eksiklik göz önünde bulundurularak değiştirilmiştir. Bilişim Teknolojileri olan dersin adı Bilişim Teknolojileri ve Yazılım olarak değiştirilmiştir. Ders 5 ve 6.

sınıflarda zorunlu olmuş ayrıca ders saati de birden ikiye çıkarılmıştır. Talim ve Terbiye Kurulu tarafından yayınlanan çerçeve programda “Problem Çözme, Programlama ve Özgün Ürün Geliştirme” öğrenme alanı eklenmiştir (Karabak ve Güneş, 2013).

Hem programlama eğitiminin 21. Yüzyıl becerilerinden biri olma öngörüsü hem de görsel programlama dillerinin ortaya çıkması dünyada programlama eğitimine olan ilgi ve eğilimi arttırmıştır. CS For ALL (Herkes İçin Bilgisayar Bilimleri) Amerika genelinde anasınıfından liseye kadar tüm öğrencilerin bilgisayar bilimleri okuryazarlığını arttırmak için başlatılan bir harekettir (Mejia, 2017). “Computer Science Education Week” kapsamında düzenlenen “Kodlama Saati (Hour of Code)” etkinliğinde eski Amerika Başkanı Barack OBAMA, Bill Gates ve Mark Zuckerberg gibi dünyaca ünlü yazılım devlerinin katılımıyla hazırlanan videolar tüm dünyada ses getirmiştir. Barack OBAMA videoların birinde “Bu beceriyi öğrenmek sadece sizin geleceğiniz için değil ülkemizin geleceği için de önemlidir” demiştir (Code.org, 2013 Aralık 8). Kodlama yapan ilk başkan ünvanını almıştır. Küresel bir hareket olan “Kodlama Saati (Hour of Code)” 180’den fazla ülkede 10 milyonlarca öğrenciye ulaşmıştır. Kodlama Saati CODE.org tarafından organize edilen Microsoft, Apple, Amazon gibi büyük şirketlerin yer aldığı bir organizasyondur. Bilgisayar bilimlerini yaygınlaştırmayı, kadınların ve azınlık ırkların bilgisayar bilimlerine ilgisini arttırmayı amaçlamaktadır.

Ülkemizde bu gelişmelere ayak uydurmak için 2017 yılında Bilişim Teknolojileri ve Yazılım dersi öğretim programında değişikliğe gidilmiştir. Öğretim programının genel amaçlarının 5. maddesi “Problem çözme ve bilgi işlemsel düşünme becerileri edinmelerini ve geliştirmelerini amaçlamak.” olmuştur. Bu amaçla programlama öğretimi ile ilgili etkinliklere verilen süre arttırılmıştır. Ayrıca “Bilgi teknolojileri alanında her öğrenciye teknolojiden yararlanma ve bilgi-işlemsel düşünme becerisi kazanabilme fırsatı sağlanmalıdır.” öğretim programının uygulanmasında dikkat edilecek hususlardan biri olarak belirtilmiştir. Bu da ülkemizde takip edilen öğretim programında 21. Yüzyıl becerilerinden biri olan bilgi işlemsel düşünmenin önemsendiğinin ve temel becerilerden biri olarak görüldüğünün göstergesi olabilir.

Bilgi işlemsel düşünme becerisini konu alan pek çok araştırma mevcuttur. Bilgi işlemsel düşünme becerisinin k12 düzeyinde tanımının kapsamıyla (Allan ve diğerleri, 2010; Barr ve Stephenson, 2011; National Academies of Science, 2010) ve değerlendirilmesiyle ilgili çok az fikir birliği vardır (Brennan ve Resnick, 2012). Barr ve Stephenson (2011) bilgi işlemsel düşünmenin içinde bulunduğumuz yüzyıl için son derece önemli olduğunu sınıf ortamlarına dahil edilebilmesi için bir tanımlamanın gerekli olduğunu söylemektedir. Bilgi işlemsel düşünmenin sınıf ortamına taşınması için yeterli kaynak ve etkinliklerin olmadığını da vurgulamaktadır (Barr ve Stephenson, 2011). Brennan ve Resnick (2012) tasarım tabanlı öğrenme etkinlikleriyle genç öğrenenlerin bilgi işlemsel düşünme becerisi gelişimini değerlendirebilecek bilgi işlemsel kavramlar, bilgi işlemsel uygulamalar ve bilgi işlemsel bakış açıları olmak üzere üç aşamalı bir yöntem geliştirmişlerdir.

İlk ve ortaokulda bilim, teknoloji, mühendislik ve matematik alanlarını birleştiren sistematik ve çok disiplinli yaklaşıma STEM denilmektedir. STEM içerisinde bulunduğumuz çağın şartlarına uygun olarak eğitim sisteminin yapılandırılmasını temel almaktadır. Son zamanlarda oldukça yaygınlaşan ve artan bir ilgiye sahip olan STEM eğitiminin tamamlayıcı becerilerinden biri de bilgi işlemsel düşünmedir (ISTE ve CSTA, 2011).

Pek çok güncel araştırmaya konu olan bilgi işlemsel düşünme becerisinin öğrencilere kazandırılmasında en çok tercih edilen ve en etkili yöntemlerden biri programlama öğretimidir.

2.2 Programlama ve Programlama Öğretimi

Farklı amaçları gerçekleştirmek amacıyla elektronik aygıtların programlanmasında özel kelime ve semboller içeren komutlardan oluşan programlama dilleri kullanılır (Van-Roy ve Haridi, 2004). Komutların yazılmasıyla ortaya belli bir amaca hizmet eden programlar ve yazılımlar çıkar (Türk Dil Kurumu (TDK), 2017). Farklı teknolojiler için hazırlanmış pek çok programlama dili vardır ve her dilin kendine has yazım kuralları ve deyimleriyle hedefe uygun komutlar oluşturularak çalıştırılır (Van-Roy ve Haridi, 2004). Dile has kodların yazılması işlemine kodlama veya programlama, son olarak ortaya çıkan ürüne ise program veya uygulama denir. Programlama yazma, test etme ve hata ayıklama gibi süreçlerden oluşan ve üst düzey

bir bilişsel beceri gerektiren karmaşık bir problem çözme süreci olarak ifade edilebilir (Kert ve Uğraş, 2009; Fesakis ve Serafeim, 2009).

Son zamanlarda yapılan araştırmalar göstermiştir ki 21. yüzyıl becerilerinin kazandırılmasında programlama öğretiminin yeri son derece önemlidir. Akpınar ve Altun'un (2014) yapmış olduğu araştırma ilk ve ortaokul öğrencilerinde tasarım ve programlama araçları öğretiminin problem çözme, analitik düşünme, uzamsal düşünme ve işbirlikli çalışma gibi becerilerin gelişimini desteklediğini ortaya koymuştur.

Eğitim, ülkelerin gelişmişlik seviyelerinin değişiminde önemli bir yere sahiptir (Tolunay ve Akyol, 2006). Nitelikli ve alanında uzman insan gücü gelişmiş toplumlarda ihtiyaç duyulan önemli bir unsurdur. Bu insan gücünü yetiştirerek gelişmişlik seviyesini arttırmak için eğitim kaçınılmazdır. Günümüzde toplumların güç unsuru bilgidir ve içinde bulunduğumuz çağa adını vermiştir (Kalkınma Bakanlığı, 2015). Bilgi ve iletişim teknolojileri bilginin işlenmesi, depolanması ve kullanılmasına imkan sunmaktadır. Bilgi ve iletişim teknolojisi araçları bilgiyi işleyecek, depolayacak ve kullanabilecek hale amaca uygun şekilde programlanarak getirilebilir. Bu da gelişmişlik seviyesini arttırmak için programlama eğitiminin gerekliliğini gösterebilir.

Programlama öğretiminde başarıyı etkileyen faktörler motivasyon, programlamaya karşı tutum, programlama dilinin anlaşılabilirliği (Jenkins, 2002) ve öğretim yöntem ve tasarımı (Mayer, 1976) olarak sayılabilir. Programlamaya yeni başlayanlar için programlama oldukça zor bir süreç olarak bilinmektedir (Aşkar ve Davenport, 2009; Kelleher ve Pausch, 2005; Korkmaz ve Altun, 2013;). Birçok araştırmada programlama kurslarına başlayanların kursları bırakma oranının oldukça yüksek olduğu (Ala-Mutka, 2004) ve kursiyerlerin bu kurslarda çok az ilerleme kaydettiği (Robins, Rountree ve Rountree, 2003) sonucuna varılmıştır. Programlamaya dair kalıplaşmış önyargılar programlamaya ilişkin olumsuz tutum geliştirilmesine neden olabilmektedir (Özyurt ve Özyurt, 2015). Bunlardan biri de programlamanın programlamaya yeni başlayanların çoğunun uzman programcı olamayacağı kadar zor olduğu düşüncesidir (Turkle, 2005).

İlköğretim düzeyindeki öğrencilerin programlama ile karşılaşması onların bilişsel gelişimine (Akpınar ve Altun, 2014; Atmatzidou ve Demetriadis, 2016; Clements ve

Gullo, 1984; Lewis, 2010; Papert, 1980) ve programlamaya dair algılarına (Kalelioğlu ve Gülbahar, 2014; Lewis, 2010) olumlu katkı sağlamaktadır.

Bilgi işlemsel düşünme becerisinin k-12 müfredatında yerini alması için 3 yaklaşım mevcuttur (Weinberg, 2013). Bunlar bilgisayarsız öğrenme, oyun ve robot programlama, başlangıç öğrenme ortamlarıdır. Küçük yaş grubundaki öğrencilere bilgi işlemsel düşünme becerisi kazandırmak için görsel programlama yazılımları (başlangıç öğrenme ortamları) tercih edilen araçlardan biridir (Armoni ve diğerleri, 2015; Brennan ve Resnick, 2012). Görsel programlama araçları deneyimi az olanlara bile kod yazım hatalarına düşmeden tasarım ve üretim imkânı ve kullanım kolaylığı sunan k-12 düzeyindekiler için programlama öğretiminde uygun bir ortamdır (Grover ve Pea, 2013).

Lai ve Yang'ın (2011) 6. Sınıf öğrencileriyle yapmış olduğu çalışmanın sonucunda görsel programlama araçlarının problem çözme becerisine olumlu etkileri olurken mantıksal düşünme becerisinde ise anlamlı bir fark yoktur. Yapılan başka çalışmalarda ise görsel programlama araçlarının problem çözme becerileri, matematiksel, mantıksal düşünme, yapılandırma ve yaratıcılık becerilerine olumlu etkileri olduğu görülmüştür (Calder, 2010; Wilson, Hainey ve Connolly, 2013).

Günümüzde bilgi ve iletişim teknolojisi çeşitli meslek alanlarında, sanatta, beşeri ve sosyal bilimlerde karşılaşılan pek çok probleme çözüm üreteceğinden, programlama becerisine sahip bireylere ihtiyaç vardır. Bu doğrultuda dünyada pek çok ülkenin müfredatı programlama eğitimini içermektedir (Settle ve Perkovic, 2010). Dünyada ve ülkemizde programlama öğretimi eğitimin ilk basamağından son basamağına kadar farklı öğretim programlarında verilmektedir. Ayrıca özel kurslar aracılığıyla da bireylere ihtiyaçlarına göre programlama dersleri verilmektedir.

Programlama öğretimine yardımcı yazılım ve araçlar olarak ya da görsel yazılım araçları olarak adlandırılan kod bloklarının sürükleyip bırakılması gibi kolay bir yöntemle kullanıma olanak sunan yazılımlar yaygınlaşmıştır. Elektronik devre elemanlarının programlayarak programlama eğitiminde kullanılmasını sağlayan, somut ürünlerin ortaya konulmasını sağlayan yazılımlar da mevcuttur. Bu yazılımlara aşağıdaki programlar örnek verilebilir.

Scratch: Ücretsiz bir görsel programlama uygulaması olan Scratch, Massachusetts Teknoloji Enstitüsü (MIT) Medya Laboratuvarında geliştirilen blok programlama

yapısına sahip, sürükle bırak etkileşimi ile kullanılan bir yazılımdır. Animasyon, oyun ve etkileşimli sunular oluşturma ve geliştirme olanağı sunan Scratch, özellikle küçük yaştaki öğrencilerin dikkatini çekecek programlama olaylarını içeren bir yapıya sahiptir.

AppInvertor: Android işletim sistemleri için mobil uygulamalar oluşturmaya ve geliştirmeye olanak sağlayan görsel bir programlama ortamı olan AppInvertor sürükle bırak etkileşimi ile kullanılan bir yazılımdır.

mBlock: mBlock, Scratch benzeri grafiksel programlama arayüzüdür. Program, günlük hayatta kullandığımız dile en yakın ve en anlaşılır şekilde kullanılan kod bloklarını barındırır. mBlock ile animasyon, oyun ve etkileşimli sunular yapabileceğimiz gibi mbot adlı robot kitini programlayabiliriz. Ayrıca arduino kart programlamaya da olanak tanımaktadır.

Alice: Programlamaya yeni başlayanların temel programlama kavramlarını daha kolay anlamalarını ve üç boyutlu sanal dünyalar oluşturmalarını sağlayan, öğrenilmesi kolay bir görsel programlama aracıdır. Alice, kullanıcının ilgi çekici nesnelere oluşturmasını ve bu nesnelere kod blokları sayesinde hareket vermelerini sağlamaktadır. Alice'in ana ekranında önceden programlanmış nesnelere ve basit metodları içeren kod blokları vardır. Öğrenciler bu blokları sürükleyerek bir hikâyeyi anlatabilir ya da karmaşık eylemler içeren bir sahne oluşturabilir.

S4A: Scratch tabanlı görsel bir programlama aracı olan S4A'nın içerisinde Arduino'yu kontrol etmeye imkân sunan programlama blokları bulunmaktadır. Bu program blokları sayesinde açık kaynak kodlu bir mikrodenetleyici kart olan Arduino programlanabilir. S4A Türkçe dil desteği de sunmaktadır. Yazılımın bir diğer artısı da yalnızca bilgisayardan veri gönderme değil Arduino'dan gelen veriyi de okuyabilmesidir.

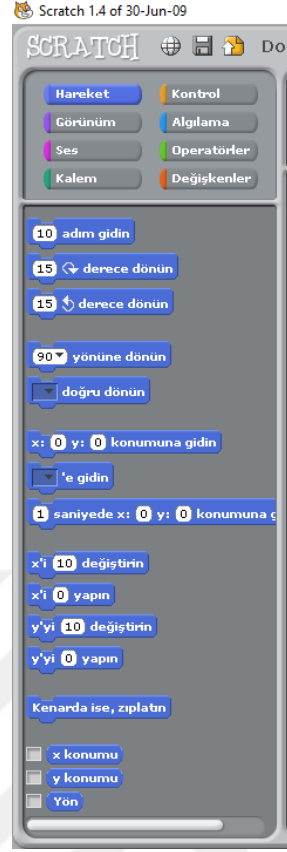
KoduGameLab: Microsoft tarafından geliştirilmiş Kodu Game Lab, basit görsel programlama dili aracılığıyla programlama becerisi olmayan çocuk veya gençlerin oyun geliştirmelerine, oynayabilmelerine ve oyunlarını arkadaşlarıyla paylaşabilmelerine imkan sunan bir yazılımdır.

2.3 Scratch Öğretimi

Ücretsiz bir görsel programlama uygulaması olan Scratch, MIT Medya Laboratuvarı'nda geliştirilmiştir. İntel Bilgisayar Kulüp Evleri gibi okul sonrası bilgisayar merkezlerindeki (8-16 yaş) öğrencilerin ihtiyaçları ve ilgileri Scratch'ın tasarımına temel oluşturmuştur (Resnick, Kafai ve Maeda, 2003). Scratch projesi 2003 yılında başlamıştır. Scratch yazılımı ve web sayfası ise 2007'de herkesin kullanımına sunulmuştur (Maloney ve diğerleri, 2010). Programlamayı daha görsel ve eğlenceli şekle getirmeyi amaçlamaktadır. Çoklu ortam temeli üzerine oturtulan Scratch ile animasyon, oyun, sunu, simülasyon gibi çeşitli yazılımlar geliştirilebilmektedir. Scratch, programlama deyim ve ifadelerinin sürüklenip bırakıldığı bir arayüze sahiptir ve bu şekilde projeler kolaylıkla oluşturulmaktadır. (Meerbaum-Salant, Armoni ve Ben-Ari, 2013). Ayrıca Scratch hazırlanan uygulamaları İnternette paylaşma olanağı da sunmaktadır.

Pek çok kullanıcı Scratch'ı kod bloklarını sürükleyip bırakarak, deneme yanılma yaparak ya da var olan projelerdeki kodlamaları keşfedip, inceleyerek öğrenmektedir. Scratch programlama ortamı kullanıcılara tek pencereci kullanıcı arayüzüyle kolay erişim, canlılık ve keşfedilebilirlik, kodların görsel çıktısını sunarak kolay kontrol etme, hata mesajsız kodlama, verileri somutlaştırma, komut bloklarını en aza indirmeye olanaklarını sunmaktadır (Maloney ve diğerleri, 2010). Scratch diğer programlama ortamlarına göre daha akılda kalıcı, daha anlamlı ve daha sosyal bir ortam sunmaktadır.

Scratch uygulama ekranı blok paleti, kodlama alanı ve proje ekranı olmak üzere üç ana kısımdan oluşmaktadır.



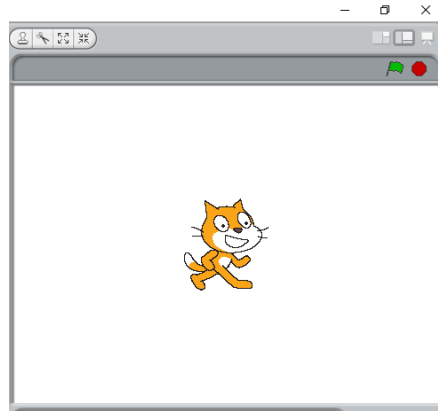
Şekil 1: Blok Paleti

Şekil 1’de Scratch ortamında yer alan Blok paleti yer almaktadır. Blok paletinde kodlar uygun bloklara yerleştirilerek gruplandırılmıştır. Scratch’ın araştırmamızda kullandığımız sürümü olan 1.4’de hareket, görünüm, ses, kalem, kontrol, algılama, operatörler ve değişkenler blokları yer almaktadır. Burada yer alan kodlar yazılmak istenen program doğrultusunda kodlama alanına sürüklenerek bırakılır.



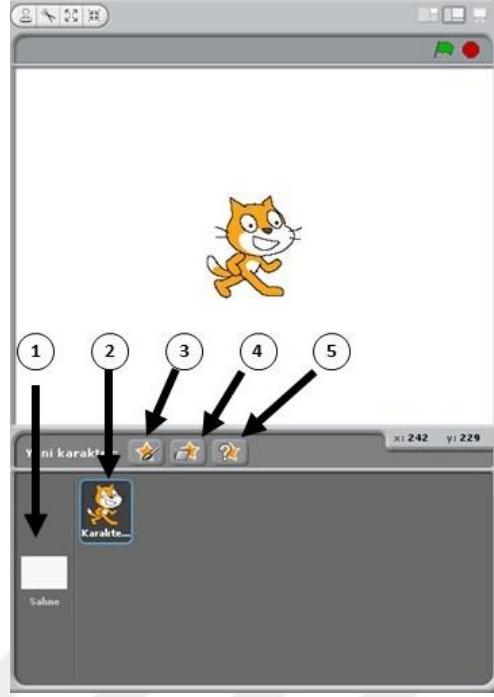
Şekil 2: Kodlama Alanı

Şekil 2’de Scratch’a ait kodlama alanı yer almaktadır. Bu alanda sürükle bırak yöntemiyle getirilen kodlar yer alır. Kodlar burada mantıksal bir şekilde sıralanıp birleştirilerek programlama işlemi gerçekleştirilir.



Şekil 3: Proje Ekranı

Şekil 3’de yer alan Proje ekranında ise mantıksal sıraya dizilerek hazırlanmış olan kodların oluşturduğu çıktı görüntülenmektedir.



Şekil 4: Scratch Ekranı

Scratch da proje ekranının altında yer alan Şekil 4’de 1 numara ile işaret edilen alana tıklandığında sahne seçili hale gelir. Çalışmanın sahnesi ile ilgili kodlamalar sahne seçili hale geldikten sonra kodlama alanına kod blokları sürüklenip bırakılarak yapılabilir. 2 numara ile gösterilen simge, çalışmada olan karaktere aittir. Çalışmaya karakter eklendikçe karakterler bu alanda gösterilecektir. Scratch da sadece çalışmada var olan karakterlere ve sahneye kodlama işlemi yapılır. Şekilde karakter seçili olduğu için kodlama alanına sürüklenip bırakılan kod bloklarıyla sadece seçili karakter kodlanabilir. 3 numara ile gösterilen simgeye tıklandığında karakter çizimi yapılabilir. 4 numaralı simgeye tıklandığında Scratch’ın kütüphanesinde var olan karakterlerden istenilen biri çalışmaya eklenebilir. 5 numaralı simgeye tıklandığında Scratch kütüphanesinden rastgele bir karakter çalışmaya eklenir.

Hairball tarafından desteklenmekte olan Dr. Scratch (Moreno, 2014) açık kaynaklı, ücretsiz bir web aracıdır. Scratch projelerini analiz etmek için geliştirilmiştir. Soyutlama, problem ayrıştırma, benzerlik, mantıksal düşünme, senkronizasyon, akış kontrolü, kullanıcı etkileşimi ve veri sunumu açısından Scratch projelerine otomatik olarak bilgi işlemsel düşünme skoru vermektedir. Yapılan araştırmada bu aracın bilgi işlemsel düşünme becerisini ölçmede kullanıcıya ait birden fazla projeyi incelemesi

gerektiđi, bilgi işlemsel düşünmeye ait hata ayıklama ve harmanlama gibi yeteneklerin ölçülmesi için yetersiz olduđu, öğrencilerin kodlama becerilerini geliştirmeleri, öğretmenlerin Scratch projelerini değerlendirerek öğrencileri yönlendirmeleri açısından kullanışlı bir araç olduđu sonucuna varılmıştır. Uygun kod bloklarıyla oluşturulmuş basit bir çalışmanın işlevselliđi göz ardı edilerek yüksek puan alması yazılımın eleştirilen ve geliştirilmesi istenen yönlerinden biri olmuştur. Ayrıca yaratıcılık ve özgünlük gibi becerilerin değerlendirilemeyeceđi vurgulanmıştır. Öğretmenlere sadece programın verdiđi puanlamaya göre öğrencileri değerlendirmemeleri tavsiye edilmektedir (Moreno-León ve Robles, 2015).

6. sınıflar için bilgisayar okur yazarlığı kursu kapsamında verilen Scratch eğitimi sonrasında öğretmenler Scratch'ın programlama öğretimini daha eğlenceli, daha motive edici, daha kolay anlaşılabilir hale getirdiđini, yaratıcılıđa imkân sunduđunu ve disiplinlerarası öğrenmeyi sağlayabileceđi sonuçlarına ulaşmışlardır (Flanagan, 2015). Ayrıca Kobsiripat (2014) tarafından ilköğretim öğrencilerine yönelik yapılan çalışmada da Scratch'ın yaratıcı düşünme becerisini olumlu olarak etkilediđi sonucuna varılmıştır.

Scratch ile ilgili 32 adet makalenin incelendiđi araştırmada Scratch ile ilgili öne çıkan konular yaratıcı düşünme, mantıksal düşünme, problem çözme, algoritma ve programlama öğretimi, simülasyon geliştirme, oyun programlama, diđer dillerle karşılaştırma, diđer derslerde kullanım, programlamaya ilişkin öğrenci görüşleri ve scratch programının tanıtılması olarak bulunmuştur (Çatlak, Tekdal ve Baz, 2015). Ayrıca araştırmada Scratch'ın programlama öğretiminde etkili bir dil olduđu, programlamayı zevkli ve eğlenceli hale getirdiđi, programlamaya Scratch ile başlamanın derse olan ilgi ve motivasyonu olumlu yönde etkilediđi bulgularına ulaşılmıştır.

Scratch ile ilgili yapılan her çalışma olumlu olmamıştır. Logo ve Scratch uygulamaları kullanan 6. sınıf öğrencilerinin tutum ve öğrenme çıktılarının karşılaştırıldıđı bir çalışmada varsayılanın aksine LOGO'yu kullanan öğrencilerin özgüvenlerinin daha yüksek olduđu fakat her iki grubun da döngüleri yorumlamada birbirine yakın performans gösterdiđi ve iki grubun da programlamaya devam etme konusunda istekli olmadığı sonucuna ulaşılmıştır (Lewis, 2010).

21. yüzyılda sahip olunması gereken temel becerilerden bilgi işlemsel düşünme öğrenenlere son zamanlarda hızla yaygınlaşan eğitsel robotik ya da görsel programlama uygulamalarıyla kazandırılabilir. Görsel programlama ortamlarının öğrencileri programlama öğrenmede motive ettiği, programlama öğrenmeyi kolaylaştırdığı, yaratıcı düşünme becerisini geliştirdiğini, öğrencilerin programlama öğrenme konusundaki özgüvenini arttırdığı, uygun etkinlikler ve projelerle diğer branşların da öğrenimine katkı sağladığı yapılan araştırmaların sonuçlarındandır. Eğitsel robotik ise öğrenenlere gerçek fiziksel bir aygıtta kodların nasıl çalıştığını görme, elektronik kavramlarla tanışma, programlama öğrenmede motive edici olma, özgüveni artırma, programlamayı kolaylaştırma, diğer branşlarla ilişkilendirme olanakları sunmaktadır. Ancak, öğrenenlere sunulan bu olanaklar onların bilgi işlemsel düşünme becerisine ne kadar olumlu katkı sağladığı, bilgi işlemsel düşünme becerisinin kazandırılmasında görsel programlama ortamları yerine eğitsel robotiği tercih etmeli miyiz gibi soruların cevabına ulaşmak bu aşamada cevaplanması gereken sorulardandır.

2.4 Robotik ve Eğitsel Robotik

Robotik öğrenenlere mekanik, motorlar, sensörler, programlama ve dijital alanlar ile ilgili öğrenme imkânı sunan bir daldır. 21. Yüzyılda Robotiğin artan popülaritesi ile eğitsel robotik kavramı ortaya çıkmıştır (Bers ve Marina, 2010). Eğitsel robotik kitleri ile robotik lise, ortaokul ve ilkokullarda yaygınlaşmaya başlamıştır (Rogers, Wendell ve Foster, 2010).

Bilişim teknolojileri öğrencilere gerçek dünya yapıları aracılığıyla anlamlı projeler geliştirme imkânı sunmaktadır (Papert, 1993). 1969 yılında Seymour Papert genç öğrenenler için Logo programını ve Logo programıyla kodlanan Kaplumbağa robotunu tasarlayarak bu alandaki ilk örnekleri vermiştir.

Eğitsel robotik uygulamaya dönük, eğlenceli etkinliklerle öğrencilerin ilgi ve meraklarını tetikleyen benzersiz eğitim ortamları sunmaktadır (Eguchi, 2010). Ayrıca öğrenenin farklı alanlarda da öğrenmesini sağlarken 21. yy. da sahip olunması gereken temel becerilerin kazanılmasında da etkilidir (Eguchi, 2010; Benitti, 2012). Eğitsel robotiğin amaçları öğrenmede motivasyonu arttırmak, genç öğrenenleri

etkileyerek bilim ve teknoloji alanındaki öğrenme isteklerini arttırmak ve problem çözme kapasitelerini arttırmaktır (Resnick ve Silverman, 2005).

Programlama becerilerini geliştirmek için Eğitsel robotikle ilgili olarak farklı yaklaşımlar mevcuttur (Eguchi, 2010: ss 4008-4013). Bunlar:

- Temaya Dayalı Öğretim Programı Yaklaşımı: Özel bir konuyu içeren müfredat alanlarıdır.
- Proje Tabanlı Yaklaşım: Öğrencilerin gerçek dünya problemlerini çözmek için gruplar halinde çalışması yaklaşımıdır.
- Hedef Odaklı Yaklaşım: Öğrenciler çoğunlukla okul dışı Robot turnuvalarında yarışmaktadırlar. Buna uluslararası düzeyde FIRST Lego Lig (<http://www.firstlegoleague.org>) ve RoboCupJunior (<http://www.robocupjunior.org>), ulusal düzeyde ise Robot Afyon örnek olarak verilebilir.

Dünyada ve ülkemizde robotik eğitimini yaygınlaştırmak için pek çok proje yapılmakta, atölyeler oluşturulmakta, kurslar, konferanslar verilmektedir. Ülkemizde eğitsel robotiğin kullanımı kodlama öğretiminde daha yaygın olarak kullanılmaktadır. Kodlama ile ilgili ulusal projelerde robotik eğitimi de verilmektedir. Ulusal projelere Samsun'da KODLASAM, Rize'de KODLARİZE, Manisa'da KODLAMANIŞA, Trabzon'da KODLAYAP, Gençlik ve Spor Bakanlığı tarafından yürütülen KOD ADI 2023 örnek olarak verilebilir.

2.5 mBlock Öğretimi

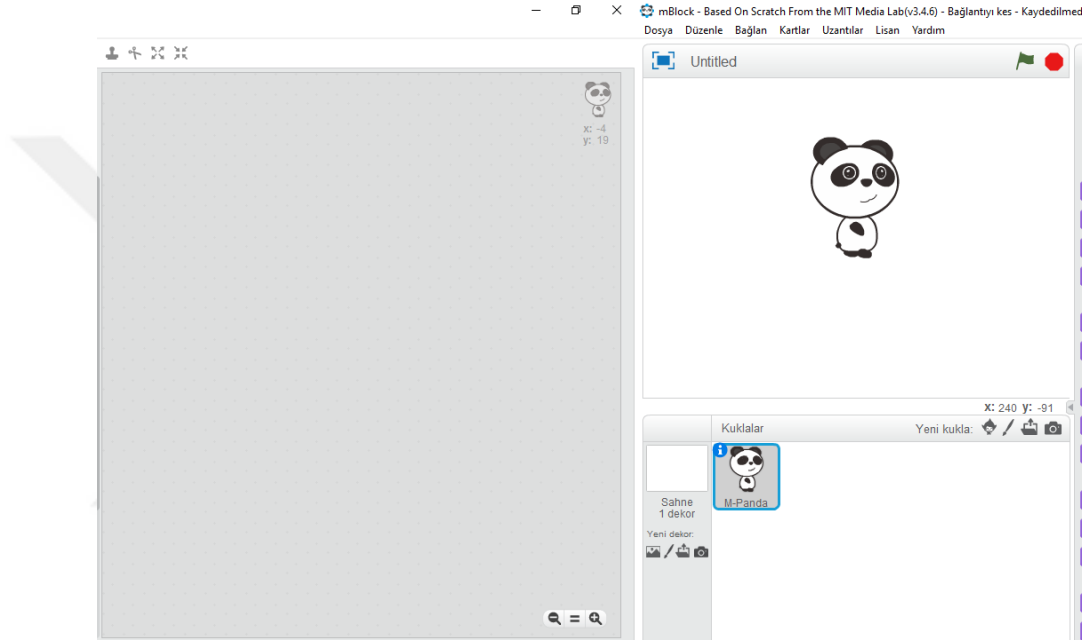
Makeblock markası alüminyum parçalardan çok farklı mekanik aksamlar kuran ve evrensel geliştirme kartları ile bu sistemlerin kontrolüne imkân sağlayan, eğitim odaklı ürünler geliştiren bir markadır. Geliştirme kartlarında Arduino platformunun temel alınması ile beraber yazılım ortamında da Scratch temelli mBlock grafiksel programlama arayüzü kullanılmıştır. Bu sayede çalışma yapan kişiler evrensel donanım ve yazılımlara aşina olup, proje sonunda geliştireceği daha farklı ve özgün projelerde markadan bağımsız ve özgür olmaktadır.

mBlock, Scratch benzeri grafiksel programlama arayüzüdür. Program, günlük hayatta kullandığımız dile en yakın ve en anlaşılır şekilde kullanılan kod bloklarını barındırır. Grafiksel programlama dilleri temelde normal kod yazma gibi olsa da görsellik ön planda olduğu ve sürükle-bırak şeklinde bir yöntemle yazıldığı için öğrenmesi oldukça kolaydır. mBlock uygulama ekranı Proje ekranı, blok paleti ve kodlama alanı olmak üzere üç ana kısımdan oluşmaktadır.



Şekil 5: Blok Paleti

Şekil 5’de yer alan blok paletinde belirli başlıklar altında toplanmış kod blokları yer almaktadır. mBlock da Scratch’den farklı olarak robotlar adlı blok yer almaktadır. Burada yer alan kod blokları yazmak istenilen program doğrultusunda kodlama alanına sürüklenerek bırakılır.



Şekil 6: Kodlama Alanı

Şekil 6’da yer alan kodlama alanına sürüklenip bırakılan kod blokları programın istenilen şekilde çalışması için mantıksal bir sıralamaya sahip olmalıdır.



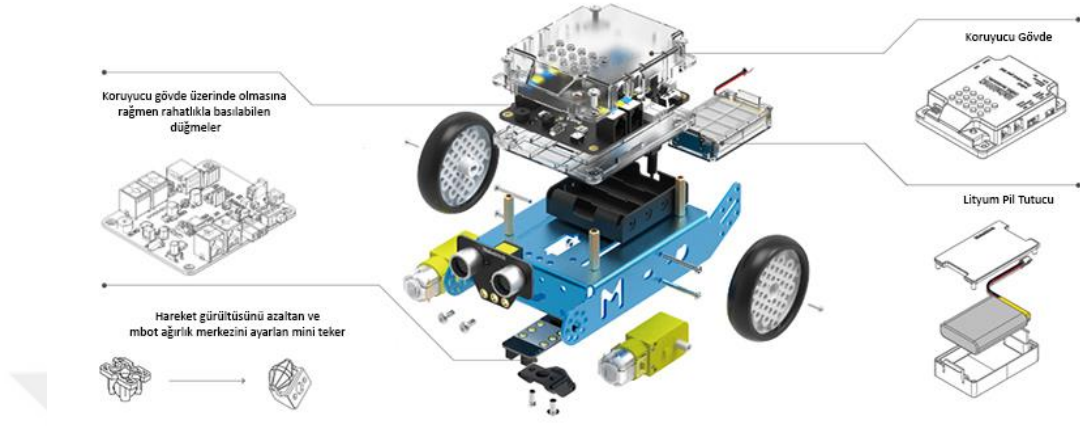
Şekil 7: Proje Ekranı

Şekil 7’de yer alan proje ekranında kodlama alanında mantıksal olarak sıralanmış kodların ekran çıktısı ve projede kullanılan karakterler (kuklalar) görüntülenir.

mBlock uygulama ekranı Scratch ile hemen hemen aynı arayüze sahiptir. Scratch’den farklı olarak Robotlar adlı içerisinde robot programlamayı sağlayacak kod blokları bulunan sekme mevcuttur (Şekil 6). mBlock ile mbotumuzun iletişim kurmasını sağlamak için bazı yöntemler mevcuttur. Bunlar:

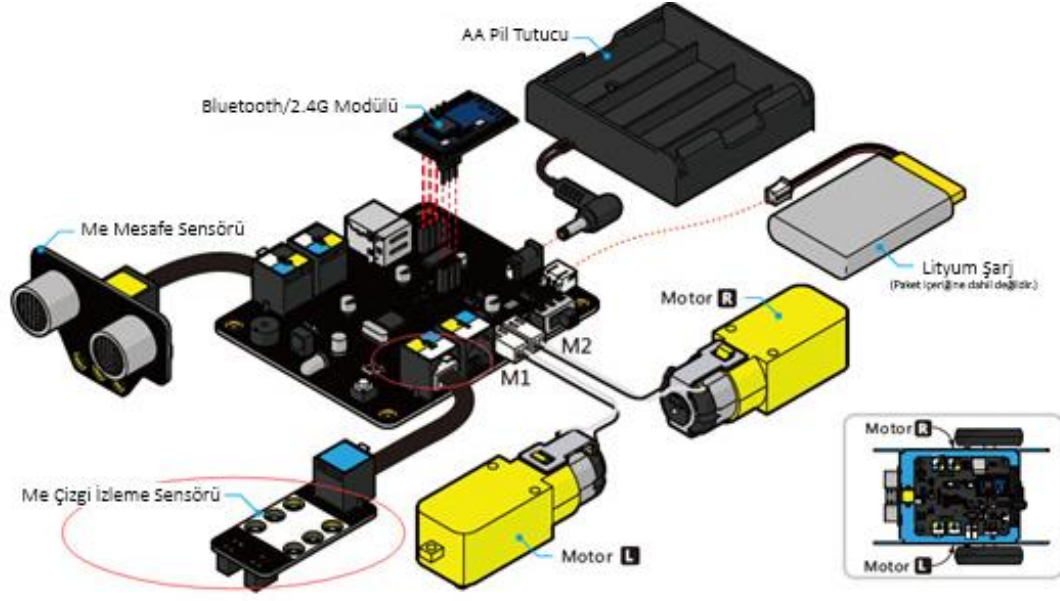
- Ghz Kablosuz Bağlantı: Bu yöntemle mbotla kablosuz olarak anlık iletişim kurulmaktadır. Yapılan kodlamalar anında mbotta gözlenebilir. Masaüstü ve dizüstü bilgisayarlar için idealdir. Tablet ve akıllı telefonlara kurulan uygulamayla uyumlu değildir (20 Kasım 2017). Uygulama ileri bir tarihte 2.4 Ghz ile bağlantı için uyumlu hale gelebilir.
- Bluetooth ile Kablosuz Bağlantı: Bu yöntemle mbotla kablosuz olarak anlık iletişim kurulmaktadır. Yapılan kodlamalar anında mbotta gözlenebilir. Tablet ve akıllı telefonlara uyumlu olan mBlock uygulaması ile kullanmak için daha idealdir. 2.4 Ghz bağlantıya göre bağlantı hızı yavaştır.

- Kablo ile Bağlantı: Utf kablo aracılığıyla bağlandığında yapılan kodlamalar robotun kullanmış olduğu kart olan Arduino Uno'ya yüklendikten sonra mbotta gözlemlenebilir.



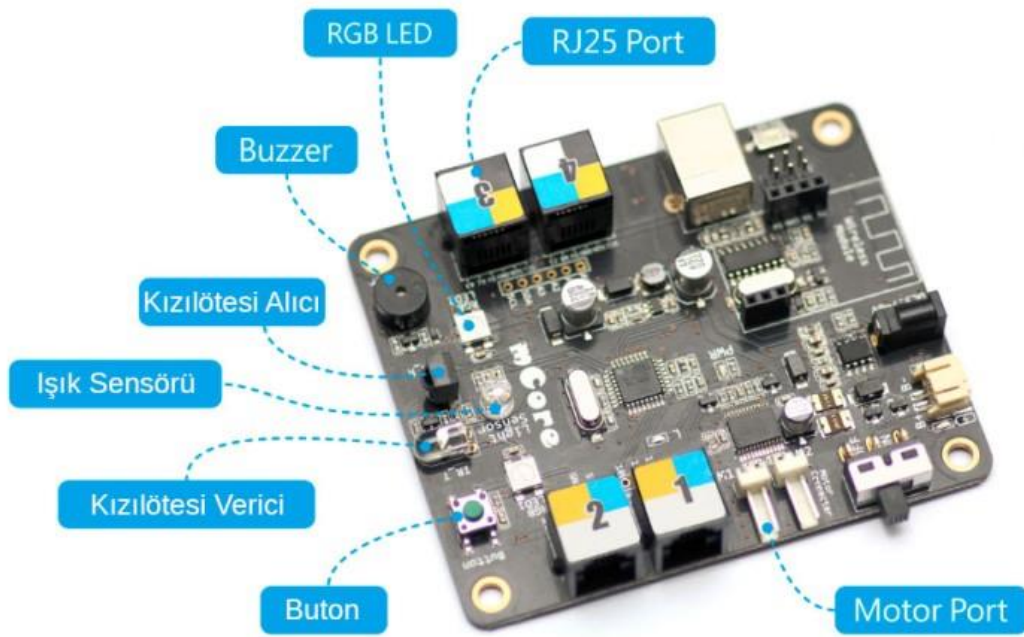
Şekil 8: mBot Monte Etme (<https://forum.makeblock.com/t/mbot-line-sensor-not-working-out-of-box/7678> kaynağından alınmıştır)

Mbot satın alındığında demonte halde gelmektedir. Şekil 8'de mbotun nasıl monte edileceği gösterilmiştir. Ayrıca mbotun özelliklerinden bazıları vurgulanmıştır.



Şekil 9: mBot Paket İçeriği (<https://forum.makeblock.com/t/mbot-line-sensor-not-working-out-of-box/7678> kaynağından alınmıştır)

Şekil 9'da mBot'a ait sensörler, motorlar, modüller ve bataryalar yerleştirileceği yerler ile birlikte gösterilmiştir. mBot alındığında pakette mesafe sensörü, çizgi izleme sensörü, 2 adet DC motor, isteğe göre ya bluetooth ya da 2.4 G bağlantı modülü, AA pil tutucu (lityum pil için koruyucu paket içeriğine dahil fakat lityum pil dahil değildir.) ve mcore (mBot çekirdeği yani programlanabilir kart) bulunmaktadır. Ayrıca mcore üzerinde sensörleri bağlamak için RJ25 portları, RGB kodlar kullanılarak oluşturulan renklerin çıktısını veren RGB Led, buzzer, uzaktan kumanda ile yönetim için kızılötesi alıcısı, mesaj ve sayı gönderebilmek için kızılötesi vericisi, ışık sensörü, düğme (buton), sağ ve sol tekerlekleri kontrol eden motorlar için motor girişleri bulunmaktadır. Paket içerisinden demonte halindeki mBotu birleştirmek için vida ve somun ayrıca robotu uzaktan kontrol etmek için de uzaktan kumanda çıkmaktadır.



Şekil 10: mBot Bağlantılar ve Sensörler (<https://forum.makeblock.com/t/mbot-line-sensor-not-working-out-of-box/7678> kaynağından alınmıştır)

Şekil 10'da gösterilen mCore mbotun programlanabilmesini sağlayan karttır. mCore üzerinde sensörlerin bağlanacağı 4 adet RJ25, DC motorların bağlanacağı motor portları, ışık sensörü, buzzer, uzaktan kumandayla kullanabilmek için kızılötesi alıcı ve verici, RGB led bulunmaktadır.

Programlama öğretiminde mbotun kullanılabilirliğini inceleyen bir araştırmanın sonuçlarına göre mBlock yazılımı ve mbot kullanılarak programlamanın temel kavramları olan döngüler, koşul yapıları, fonksiyonlar-prosedürler, değişkenler, listeler ve dizileri içeren uygulamalar kolayca oluşturulup kullanılabilir. Ayrıca programlama öğretiminde robot kullanarak soyut kavramlar kolayca somutlaştırılabilir ve yazılan programın etkisini anında gören öğrencilerin problem çözme ve bilgi-işlemsel düşünme becerileri daha kolay ve hızlı bir şekilde gelişebilir sonucuna varılmıştır (Numanoğlu ve Keser, 2017).

Robotik öğretimi gerçekleştiren 27 tane öğreticiden toplanan verilerin yorumlandığı araştırmada öğrencilere göre robotik eğitiminin eğlenceli ve verimli olduğu, sürecin öğrencinin hayalgücünü geliştirdiği, oyun ve eğlence ortamı sunduğu, öğrencilerin kendi ürünlerini oluşturma imkânı sunduğu sonucuna varılmıştır. Ayrıca öğretmenler sonuca ulaşma isteği, robotların hareket etmesi, pekiştirici ve kısa molaların

meşguliyet ve motivasyon üzerinde olumlu etkileri olduğunu vurgularken kullanılacak senaryonun kısa, öğrenci özelliklerine uygun, ilgi çekici ve gerçek yaşamla ilişkili olması gerektiğini belirtmişlerdi (Küçük ve Şişman, 2017).

Beşinci sınıflara yapılan okul sonrası robotik programında öğrencilerin STEM, problem çözme ve takım çalışmasına yönelik tutum ve beceri gelişimleri belirlenmeye çalışılmıştır. Çalışma sonrasında öğrencilerin STEM'e olan ilgisinin arttığı, takımla çalışma ve problem çözme becerilerinin arttığı ve takım çalışması ve problem çözmeye karşı olumlu tutum geliştirildiği sonucuna varılmıştır (Ebelt, 2012). Dijital oyunlar ve robotiğin STEM'e olan tutumun değişiminin incelendiği çalışmada her ikisinin de STEM'e karşı olumlu tutum geliştirdiği fakat dijital oyunların robotiğe göre olumlu tutum geliştirmede daha etkili olduğu sonucu elde edilmiştir (Ilias ve Evripidis, 2016).

ÜÇÜNCÜ BÖLÜM

III. YÖNTEM

Bu bölümde araştırmanın deseni, çalışma evreni, örnekleme, veri toplama araçları, veri toplama süreci ve verilerin nasıl analiz edileceğine ilişkin bilgiler verilmiştir.

3.1 Araştırmanın Deseni

Bu araştırmada 10-12 yaş grubu öğrencilerinin programlama öğretiminde robotik ve Scratch uygulamalarının öğrencilerin bilgi işlemsel düşünme pratiklerine ve akademik başarılarına etkisinin karşılaştırılması ve incelenmesi amaçlanmıştır. Araştırma nicel araştırma yöntemlerinden yarı deneysel araştırma yöntemi kullanılarak gerçekleştirilmiştir. Yarı deneysel araştırma yöntemlerinden son test kontrol grup tasarımı kullanılmıştır.

Araştırma 30'ar kişilik 4 farklı sınıftaki öğrencileri incelediğinden öğrenciler rastgele gruplara atanmamıştır. Bu nedenle gerçek deneysel tasarıma yer verilememiştir. Her sınıftan rastgele 15'er öğrenci seçilerek ilk iki sınıftan seçilen 30 kişi birinci grup, kalan iki sınıftan seçilen 30 öğrenci ise ikinci grup olarak sınıflandırılmıştır. Sınıflardaki tüm öğrenciler eğitimleri almalarına rağmen sadece seçilen 60 öğrencinin verileri değerlendirilmiştir. Birinci grup programlama öğretimine Scratch ile başlamış, ikinci grup ise mBlock ve robot programlama ile başlamıştır.

Eğitim sonunda öğrencilerin bilgi işlemsel düşünme pratikleri ve programlamaya yönelik akademik başarıları ölçülmüştür. Daha sonra gruplar yer değiştirilerek aynı eğitimler farklı gruplara verilmiştir. 1. gruba mBlock ile robot, 2. gruba Scratch etkinlikleri yapılmış ve akademik başarıları tekrar ölçülmüştür.

3.2 Evren ve Örneklem

Araştırmanın evrenini 5 ve 6. sınıf öğrencileri yani 10-12 yaş grubu öğrenciler oluşturmuştur. Çalışma evrenine bağlı olarak Samsun'un Tekkeköy ilçesinde öğrenim gören 5. sınıf öğrencileri seçilmiştir. Öğrenciler 10-11 yaş aralığındadır,

araştırma yapılmadan önce bir buçuk dönem (yaklaşık 8 ay) bilişim teknolojileri dersi almışlardır. Bu süre boyunca öğrencilere aşağıdaki kazanımlar verilmiştir.

- Bilişim Teknolojileri sınıf kurallarını bilir.
- Bilgi ve iletişim teknolojisi araçlarını listeler.
- Bilişim teknolojilerin günlük hayatımızdaki kullanım amaçlarını ve hayatımızdaki önemini açıklar.
- Belirli bir amaç için kullanılması gereken bilişim teknolojisine karar verir.
- Farklı teknolojilerin olumlu ve olumsuz yönlerini değerlendirir.
- Verilen bağlamda bilginin uygunluğunu değerlendirir.
- Bilgisayarı açıp kapatabilir. Fare ve klavyeyi doğru kullanabilir.
- Kelime işlemci programını tanır. Kelime işlemci programlarında uygulanan temel dosyalama işlemlerini (belge açma-kapatma-kaydetme) yapar.
- Kelime işlemci programlarında uygulanan metin işlemlerini (yazı yazma, işlemleri geri alma, taşıma, kopyalama, arama) yapar.
- Kelime işlemci programlarında sayfaların yapılarını düzenler ve çıktılar alır.
- Kelime işlemci programlarında biçimlendirme işlemlerini yapar. Kelime işlemci programları içinde tablolama işlemlerini yapar.
- Dosyalama işlemlerini yapar. Hücrelerde seçme, veri girme, düzenleme, kopyalama ve taşıma işlemlerini yapar.
- Sayfa yapısını ayarlar, çıktı alır. Biçimlendirme işlemlerini (hücre ve yazı özelliklerini değiştirme) yapar.
- Basit formüller oluşturur. Formülleri kopyalar, taşır.
- Sunum programında dosyalama işlemlerini yapar. Metin işlemlerini yapar.
- Sunularda düzenlemeler yapar.
- Sunulardaki nesnelere ve yazılara animasyon verir.
- Yazılım kavramını anlar.
- Yazılıma örnek verir.

- Donanım kavramını anlar.
- Donanım birimlerine örnekler verir.
- Hafıza birimlerini kavrar.
- Masaüstü ve mobil işletim sistemlerini tanır.
- İnternet tarayıcılarını kavrar.
- Arama motorlarını kullanır. İndirme işlemleri yapar.
- Mail hesabı açar. Mail alır ve gönderir.
- Gizlilik ve güvenlik boyutlarını açıklar.
- Zararlı yazılımları tanır. Zararlarından korunmak için önlemler alır.
- Bilişim suçlarının neler olduğunu bilir.
- Elektronik ortamdaki verilerin sınıflanması ve saklanması kullanılan yaklaşımları değerlendirir.

Öğrencilerin Programlamayla ilgili bilgi ve deneyimleri yoktur. Programlama ile ilk kez Scratch ve mBlock kullanarak araştırma sürecinde tanışmışlardır. Robotlarla da ilk kez bu araştırmayla tanışmışlardır.

Öğrenciler dört farklı sınıftan kura yöntemiyle seçilmişlerdir. Öğrencilerin yarısı kız, yarısı erkek olacak şekilde kura çekilmiştir. Dört sınıftan 15'er öğrenci olmak üzere toplam 60 öğrenci belirlenmiştir.

Tablo 1'de çalışmada yer alan öğrenci özellikleri ve sayıları gösterilmiştir. Grup 1 ve grup 2 de 15 kız 15 erkek öğrenci olmak üzere 30'ar öğrenci, toplam 60 öğrenci bulunmaktadır.

Tablo 1: Öğrenci Tablosu

Grup Adı	Cinsiyet	Öğrenci Sayısı	Toplam Öğrenci
Grup 1	Erkek	15	30
	Kız	15	
Grup 2	Erkek	15	30
	Kız	15	
Toplam Öğrenci			60

3.3 Veri Toplama Araçları

Çalışmada veri toplama araçları öğrencilerin ders içerisindeki etkinlikler doğrultusunda ortaya çıkardıkları ürünler ile ilgili görüşme formu (Tablo 3, 4, 5 ve 6) ve alan uzmanı tarafından hazırlanmış olan testler TBBT (Ek 2), 1. sınav (Ek 3) ve 2. sınavdır (Ek 4).

Bilgi işlemsel düşünme becerisini ölçmek için ise Harvard Üniversitesi tarafından geliştirilmiş olan ölçek kullanılmıştır. Çalışmada öğrenciler dört ana uygulama alanına göre değerlendirilmiştir. Her uygulama alanına ait 4 alt başlık mevcuttur. Bu başlıklar doğrultusunda öğrencilerle yapılan görüşme düşük (1), normal (2), yüksek (3) olmak üzere puanlandırılmıştır (Assessing Development Of Computational Practices,2019).

TBBT, 1. ve 2. sınav alanda 10 yıllık tecrübesi olan bilişim teknolojileri öğretmeni tarafından hazırlanmıştır. Testler iki farklı bilişim teknoloji öğretmeni tarafından incelenerek gerekli düzenleme ve değişiklikler yapılmıştır.

Öğrencilerin akademik başarılarının karşılaştırılmasında referans almak için alan uzmanı tarafından hazırlanmış TBBT öğrencilere uygulanmıştır. Öğrenciler 2 gruba ayrılmıştır. İlk gruba Scratch, ikinci gruba ise robotik etkinlikleri ile bilgi işlemsel düşünme becerisinin kazandırılması hedeflenmiştir. Her iki gruba dört haftalık eğitim verilmiştir.

Dört haftalık eğitimde yapılan etkinlikler ve hedeflenen kazanımlar Tablo 2’de gösterildiği gibidir:

Tablo 2: Etkinlik ve Kazanım Tablosu

Hafta	Etkinlik Adı	Grup	Etkinlik / Kazanımlar
1	Arayüzü Tanıma	Scratch	Scratch arayüzü tanıtılır. Arayüzü tanımları için fırsat verilir.
		Robotik	mBlock arayüzü tanıtılır. Arayüzü tanımları için fırsat verilir.
2	Dans Et	Scratch	Scratch da seçilen bir karaktere hareket, görünüm, kontrol ve ses bloklarında bulunan kodlar kullanılarak Penguen Dansı yaptırılır. Döngülerle çalışabilir.
		Robotik	mBlock da kontrol ve robot blokları kullanılarak robota penguen dansı yaptırılır. Döngülerle çalışabilir. Motorları kontrol edebilir.
3	Yangın Var	Scratch	Seçmiş olduğu karakteri belli bir alanın dışına çıkarmadan sürekli o alanın içinde hareket ettirecek kodları oluşturması beklenmektedir. Kontrol yapılarını kullanılabilir.
		Robotik	Duvarlarla çevrilmiş bir alanın içerisinde robotu alanın dışına çıkarmadan sürekli olarak hareket ettirecek kodları oluşturması beklenmektedir. Kontrol yapılarını kullanılabilir. Sensörlerle çalışabilir (mesafe sensörü).
4	Park Et	Scratch	Seçmiş olduğu karakteri kullanıcının yön tuşlarını kullanarak uygun alana getirmesini sağlayacak kodları oluşturması beklenmektedir. Kullanıcı kontrollü çalışmalar oluşturabilir.
		Robotik	Robotun yön tuşlarıyla kontrol edilebilir hale getirilmesini sağlayacak kodları oluşturabilir. Kullanıcı kontrollü çalışmalar oluşturabilir.

Öğrencilerin akademik başarısını ölçmek için kullanılan ölçeklerde Scratch ve mBlock gruplarına sorulan sorular hemen hemen benzer olup genelde sorulardaki şekillerin arayüzü (robot grubuna mBlock, görsel programlama grubuna Scratch arayüzü) değişmiştir (Şekil 11). Gruplara verilen ilk eğitimden sonra 1. sınav yapılmıştır. Öğrenci başarısını arttırmak için önce Scratch mı yoksa robotik mi öğretmeli sorusuna cevap bulabilmek için aynı öğrenci gruplarına Scratch öğrenenlere robotik, robotik öğrenenlere ise Scratch eğitimi verilmiştir. Ardından 2. sınav yapılmıştır. Eğitim sonundaki sınav sonuçları karşılaştırılmıştır.



Şekil 11: Gruplara Sorulan Sorular

Öğrencilerin bilgi işlemsel düşünme pratiklerini ölçmek için Harvard Üniversitesi tarafından geliştirilmiş olan ölçek kullanılmıştır. Çalışmada öğrenciler dört ana uygulama alanına göre değerlendirilmiştir. Her uygulama alanına ait 4 alt başlık mevcuttur. Bu başlıklar doğrultusunda öğrencilerle yapılan görüşme düşük (1), normal (2), yüksek (3) olmak üzere puanlandırılmıştır (Assessing Development Of Computational Practices,2019).

Brennan ve Resnick (2012) ürün tabanlı görüşmeler, tasarım senaryoları ve öğrenci dosyaları olmak üzere üç farklı yaklaşımı kullanarak bilgi işlemsel düşünme becerisinin daha güvenilir bir şekilde ölçülebileceğini vurgulamıştır. Bu çalışmada ürün tabanlı görüşmeler kullanılarak öğrencilerin bilgi işlemsel düşünme pratikleri ölçülmeye çalışılmıştır. Buna göre bilgi işlemsel düşünme becerisi üç boyut altında ölçülmektedir. Bunlar bilgi işlemsel düşünme kavramları, bilgi işlemsel düşünme pratikleri, bilgi işlemsel bakış açılarıdır. Bilgi işlemsel pratikler deneme-yineleme, test etme-hata ayıklama, yeniden kullanma-karıştırma, özetleme ve modüler hale getirme olmak üzere dört ana bölüm altında nitelendirilmiştir.

Aşağıdaki tablolar öğrencilerin bilgi işlemsel düşünme pratikleri (Deneme-Yineleme, Test Etme-Hata Ayıklama, Yeniden Kullanma-Karıştırma, Özetleme ve Modüler Hale Getirme) ölçülerek öğrenci gelişiminin değerlendirilmesi için kullanılmıştır. İlk sütunda öğrenci (tasarım günlüğü veya röportajın bir parçası gibi) için bir soru yer almaktadır. İkinci, üçüncü ve dördüncü sütunlar ise becerinin ne düzeyde olduğunu göstermektedir.

Tablo 3: Deneme ve Yanılma Puanlama Ölçeği

Deneme ve Yineleme	Düşük	Normal	Yüksek
Projeyi nasıl oluşturacağını tanımla	Öğrenci projeyi oluşturmayla ilgili temel bilgileri ama özel bir projeyi yapacağıyla ilgili detay vermez.	Öğrenci özel bir projeyi oluşturmayla ilgili genel bilgileri verir.	Öğrenci özel bir projenin farklı elemanlarıyla ve nasıl geliştirilebileceğiyle ilgili bilgileri verir.
Projede üzerinde çalışırken dendiğini farklı şeyleri tanımla	Öğrenci dendiğiyle ilgili özel bir örnek veremez.	Öğrenci projesinde dendiğiyle ilgili genel bir örnek verir.	Öğrenci projesinde dendiği farklı şeylerle ilgili özel örnekler verir.
Yapmış olduğu düzenlemeleri tanımla ve neden yapmış olduğunu anlat	Öğrenci değişiklik yapmadığını söyler ya da sadece değişiklik yaptığını söyler ama örnek veremez.	Öğrenci yapmış olduğu bir özel değişikliği tanımlar.	Öğrenci yapmış olduğu özel değişiklikleri ve neden yapmış olduğunu tanımlar.
Yeni birşeyler dendiğini zamanı tanımla	Öğrenci birşeyler dendiğine dair örnekler sunamaz.	Öğrenci yeni birşey dendiğine dair genel bir örnek verir.	Öğrenci projesinde dendiği özel yeni şeyleri tanımlar.

Tablo 3’de deneme ve yanılma puanlama ölçeği yer almıştır. Bu ölçekte öğrencilerden projesini nasıl oluşturacağı, projesi üzerinde çalışırken dendiği farklı şeyleri, yapmış olduğu düzenlemeleri ve nedenlerini, yeni bir şeyler dendiği zamanı

tanımlaması beklenmektedir. Yaptığı tanımlamalara göre öğrencilere düşük, normal ve yüksek olmak üzere 1 ile 3 arasında puan verilir. Düşük olanlara 1, normal olanlara 2, yüksek olanlara ise 3 puan verilmiştir.

Tablo 4: Test Etme ve Hata Ayıklama Puanlama Ölçeği

Test Etme ve Hata Ayıklama	Düşük	Normal	Yüksek
Projenin istediği gibi çalışmadığı anı tanımlama	Öğrenci projesinde problem içeren durumu tanımlayamaz.	Öğrenci projesinde neyin yanlış olduğunu tanımlar fakat ondan ne yapılması istendiğini tanımlayamaz.	Öğrenci ne olduğuna dair ve proje çalıştığında ondan ne yapması istendiğine dair özel bir örnek verir.
Bu hatayı nasıl tespit ettiğini tanımla	Öğrenci problemi tanımlayamaz.	Öğrenci yazıları (kodları) okuyup tanımlamasına rağmen kodlarda neyin probleme neden olduğuna dair özel bir örnek veremez.	Öğrenci yazıları (kodları) okuyup tanımlar ve kodlarda neyin probleme neden olduğuna dair özel bir örnek verir.
Problemi nasıl çözdüğünü tanımla	Öğrenci deney yapmadan ya da problemi çözmeden sorunun ne olduğunu tanımlayamaz.	Öğrenci genel bir örneği değiştirerek çalışıp çalışmadığını test eder.	Öğrenci özel bir örneği değiştirerek çalışıp çalışmadığını test eder.

Tablo 4’de test etme ve hata ayıklama puanlama ölçeği yer almıştır. Bu ölçekte öğrencilerden projesinin istediği gibi çalışmadığı anı, bu hatayı nasıl tespit ettiğini ve problemi nasıl çözdüğünü tanımlaması beklenmektedir. Yaptığı tanımlamalara göre öğrencilere düşük, normal ve yüksek olmak üzere 1 ile 3 arasında puan verilir. Düşük olanlara 1, normal olanlara 2, yüksek olanlara ise 3 puan verilmiştir.

Tablo 5: Yeniden Kullanma ve Karıştırma Puanlama Ölçeği

Yeniden Kullanma ve Karıştırma	Düşük	Normal	Yüksek
Başkalarının yazılarını (kodlarını) ve projelerini denerken onlardan nasıl ilham aldığını / faydalandığını tanımlar.	Öğrenci onların diğer projelerinden nasıl fikir ve ilham aldığını tanımlayamaz.	Öğrenci onlara ilham veren bir projeyi genel bir şekilde tanımlar.	Öğrenci onlara ilham veren projenin özel bir örneğini ve bunun onlara nasıl ilham verdiğini tanımlar.
Projende bir başka projenin bir kısmını kullandığı zamanı tanımla	Öğrenci diğer projelerden aldığı yazılara (kodlara), fikirlere veya kaynaklara nasıl uyum sağladığını tanımlayamaz.	Öğrenci diğer projelerden aldığı yazılara (kodlara), fikirlere veya kaynaklara nasıl uyum sağladığına dair genel bir tanımlama yapar.	Öğrenci diğer projelerden aldığı yazılara (kodlara), fikirlere veya kaynaklara uyum sağladığına ve bunun nasıl olduğuna dair özel örnekler verir.
Varolan bir projenin (Kendi projen ya da başkasının projesi) geliştirilmesi için değişikliği yapacağın zamanı tanımla	Öğrenci bir başka projedeki değişikliği tanımlayamaz.	Öğrenci bir başka projede yapılan değişikliklerin genel bir tanımını yapar.	Öğrenci bir başka projede yapılan değişikliklerin ve neden yapıldığının özel örneklerini verir.
Kimlerden ilham aldığını, destek aldığını tanımla.	Öğrenci kimseden destek almaz.	Öğrenci onu etkileyen öğrenci, kişi ya da projeleri tanımlar.	Öğrenci proje dökümanları, Scratch web sayfasındaki insanları ve onu etkileyen projeleri tanımlar.

Tablo 5’de yeniden kullanma ve karıştırma puanlama ölçeği yer almıştır. Bu ölçekte öğrencilerden başkalarının kodlarını ve projelerini denerken onlardan nasıl faydalandığını, projesinde bir başka projenin bir kısmını kullandığı zamanı, projenin geliştirilmesi için değişiklik yapacağı zamanı, kimlerden faydalandığını tanımlaması beklenmektedir. Yaptığı tanımlamalara göre öğrencilere düşük, normal ve yüksek

olmak üzere 1 ile 3 arasında puan verilmiştir. Düşük olanlara 1, normal olanlara 2, yüksek olanlara ise 3 puan verilmiştir.

Tablo 6: Özetleme ve Modüler Hale Getirme Puanlama Ölçeği

Özetleme ve Modüler Hale Getirme	Düşük	Normal	Yüksek
Projen için hangi karakterlere ihtiyacın olduğunu ve onlarla ne yapacağını tanımla.	Öğrenci karakterleri nasıl seçtiğini tanımlayamaz.	Öğrenci seçtiği karakterlerin tanımlamasını yapar.	Öğrenci projenin hedeflerine uygun karakterler hakkında nasıl karar verdiklerini özel olarak tanımlar.
Projende hangi yazılara (kodlara) ihtiyaç duyacağını nasıl karar verdiğini ve onların ne yapmaları gerektiğini tanımla.	Öğrenci yazıları (kodları) nasıl oluşturduklarını tanımlayamaz.	Öğrenci oluşturulan yazıların (kodların) genel tanımını yapar.	Öğrenci projenin amaçlarına uygun yazıları (kodları) nasıl oluşturduklarıyla ilgili özel bir tanım yapar.
Yazıları (kodları) nasıl organize ettiğini tanımlar.	Öğrenci yazıları (kodları) nasıl organize ettiğini tanımlayamaz.	Öğrenci yazıları (kodları) nasıl organize ettiğini genel olarak tanımlar.	Öğrenci yazıları (kodları) nasıl ve niçin organize ettiğini özel örnekler verir.

Tablo 6'da özetleme ve modüler hale getirme puanlama ölçeği yer almıştır. Bu ölçekte öğrencilerden projesi için hangi karakterlere ihtiyacı olduğunu ve onlarla ne yapacağını, projesinde hangi kodlara ihtiyaç duyacağına nasıl karar verdiğini ve onların ne yapmaları gerektiğini, kodları nasıl organize ettiğini tanımlaması beklenmektedir. Yaptığı tanımlamalara göre öğrencilere düşük, normal ve yüksek olmak üzere 1 ile 3 arasında puan verilmiştir. Düşük olanlara 1, normal olanlara 2, yüksek olanlara ise 3 puan verilmiştir.

3.4 Süreç

Araştırma konusunun belirlenme aşamasında araştırmacının bilişim teknolojileri öğretmeni olması önemli bir rol oynamıştır. Öğretmenlik deneyimlerinden yola

çıkılarak, bu araştırmanın bilişim teknolojileri öğretmenlerine ve bu alandaki eğitimlere pratik katkılar sağlayabileceği öngörülmüştür. Bilişim teknolojileri dersi öğretiminde programlama öğrenimine yeni başlayacak olan öğrenciler için programlamanın hangi programlama aracılığıyla öğretildiği ve programlama aracının akademik başarı üzerindeki etkisi araştırmacı tarafından merak edilmektedir. Çünkü iki yıldır araştırmacı tarafından Scratch ile programlamaya giriş yapılmıştır. Akademik olarak başarı kötü olmamasına rağmen başarının daha iyi olması beklenmektedir. Bu da araştırmacıyı programlama öğretimi için yeni yollar aramaya itmiştir.

Programlama öğretiminde kullanımı yaygınlaşan robot kitlerinin programlama öğretimini daha etkili hale getirdiğini gösteren pek çok araştırma mevcuttur. Bu da araştırmacının programlama öğretimi sürecinde Scratch (görsel programlama aracı) ile karşılaştırmak için araç olarak robotları kullanmaya karar vermesini sağlamıştır. Araştırmaya destek veren araştırmacının önerisiyle çağımızın her bireyde şart koştuğu temel becerilerden biri olarak görülen ve programlama öğretimiyle ilişkili olan bilgi işlemsel düşünme becerisi de araştırmaya dahil edilmiştir.

Araştırma konusuna karar verildikten sonra öğretim ortamı araştırma için uygun hale getirilmeye çalışılmıştır. Bilişim teknolojileri sınıfında 12 adet bilgisayar mevcuttur ve bilgisayarların hepsinde Scratch'ın 1.4 sürümü yüklüdür. Araştırmacıların yaptığı görüşmeler sonucunda maliyetinin düşük olması, paket içeriğinin programlama öğretimi için yeterli olması, programlama arayüzünün Türkçe olması, arayüzün Scratch ile hemen hemen aynı olması, kullanıcı dostu olması nedenlerle robot kiti tercihi mBottan yana olmuştur. Son olarak bilgisayarlara mBotu programlamayı sağlayan programlama ortamı olan mBlock kurulmuştur.

Etkinlikleri belirleme sürecinde etkinliğin Scratch ve mblock arayüzünde hemen hemen aynı kodlar kullanılarak yapılmasına dikkat edilmiştir. Bunun nedeni, etkinlikler arasında oluşacak olası farkların araştırma sonuçlarını minimum düzeyde etkilemesini sağlamaktır. Sünbül'e göre (2014: 20-34) öğretim ilkelerinden aşağıdaki kriterlere uygun olacak şekilde etkinlikler belirlenmiştir.

- Öğrenciye Görelik: Araştırmada 3 farklı etkinlik oluşturulmuştur. Oluşturulan etkinliklerde farklı özelliklerdeki öğrencilerin ilgi, ihtiyaç ve yetenekleri göz önünde bulundurulmuştur.

- Hedefe (Amaca) Uygunluk İlkesi: Her etkinlik programlama kavramlarını öğretmeye, programlama becerisi kazandırmaya yöneliktir.
- Bilinenden bilinmeye ilkesi: Öğrencilerin bildikleri konuların üzerine bilmedikleri konular eklenecek şekilde etkinlikler planlanmıştır.
- Nesnellik ve açıklık ilkesi: Öğrencilerin ikiden fazla duyu organını kullanmasını sağlayacak şekilde etkinlikler planlanmıştır.
- Ekonomiklik ilkesi: 4 haftalık süreçte öğrencilere hedeflenen kazanımları kazandırmak için süreç iyi bir şekilde planlandı.
- Hayatilik (Yaşama yakınlık) ilkesi: Etkinliklerin tasarımında öğrencilerin elde ettikleri kazanımları günlük hayatlarında da kullanabilmeleri özelliği göz önünde bulundurulmuştur.
- Aktüalite (güncellik) ilkesi: Etkinlikler gerçek hayatla ilişkili olacak şekilde tasarlandı.
- Aktivite (iş) ilkesi: Etkinlikler öğrencilerin aktif halde olmasını gerektirecek niteliktedir.
- Basitten karmaşığa (kolaydan zora) ilkesi: Etkinliklerin tasarımında ve sıralamasında basitten karmaşığa ilkesi göz önünde bulundurulmuştur.
- Bütünlük ilkesi: Etkinlikler öğrencilerin sadece akademik değil, sosyal, kültürel, ahlaki özellikleri geliştirecek niteliktedir.
- Sosyallik ilkesi: Etkinliklerde Scratch ile çalışanlar 2 veya 3, robotik ile çalışanlar ise 5 veya 6 kişilik gruplar halinde çalıştıkları için sosyal becerilerini geliştirme imkânı bulmuşlardır. (Şekil 12)



Şekil 12: Öğrenciler Scratch ile Programlama Yaparken

Etkinliklerin ilki penguen dansıdır. Bu etkinlikte 2. grup robotu sadece sağa sola, ileri geri müzikle uyumlu bir şekilde kodlarken (Şekil 14) 1. grup da aynı işlemi seçtikleri karaktere (Şekil 13) yaptırdı. Bu etkinlikteki amaç hareket bloklarını ve “..... saniye bekle” bloğunu kullanabilmek olarak belirlenmiştir.



Şekil 13:Scratch Penguen Dansı Etkinliği



Şekil 14:Robotik Penguen Dansı Etkinliği

İkinci etkinlik ise yangın var etkinliğidir. Bu etkinlikte 1. grup öğrencileri tasarladıkları alanın dışına çıkmadan alan içinde hareket eden karakter oluşturdu. 2. grup ise etrafı kutularla sarılmış dikdörtgen bir alanda kutulara çarpmadan dikdörtgen alan içinde hareket eden robotu programladılar (Şekil 15). Etkinliğin

amacı ise bir sonraki etkinlikte öğrenmiş oldukları hareket blokları ile “eğer...ise” kontrol bloğunu kullanabilmek olarak belirlenmiştir.



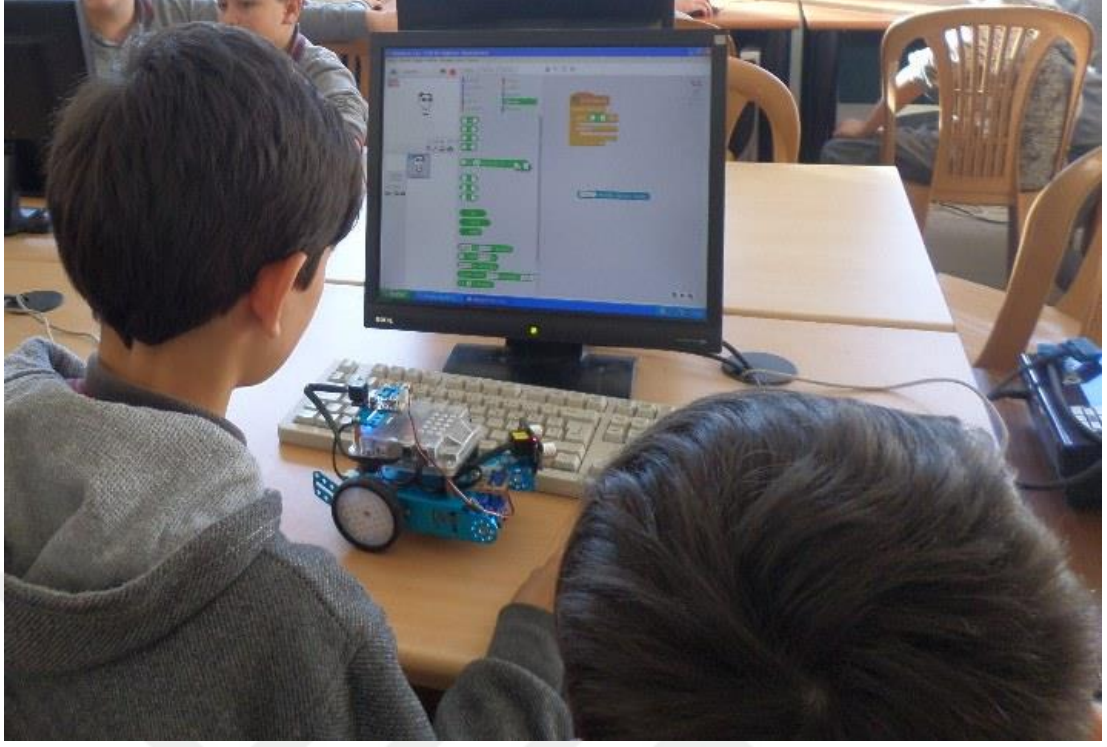
Şekil 15:mBot Yangın Var Etkinliği

Son etkinlik ise park et etkinliğidir. Bu etkinlikte 1. grup çizmiş oldukları park alanına, seçmiş oldukları araba karakterini yön tuşlarını kullanarak park etmeye çalıştılar. 2. grup ise hazırlanmış olan park alanına robotlarını yön tuşlarını kullanarak park ettiler (Şekil 16). Etkinliğin amacı ise klavye ile kullanıcı kontrolünü sağlayabilmek olarak belirlenmiştir.



Şekil 16:mBot Park Et Etkinliği

Şekil 17’de öğrenciler mBlock arayüzü aracılığıyla kablosuz olarak bağlandıkları mBot’u Park Et adlı etkinlik için programlamaya çalışıyorlar.



Şekil 17:Öğrenciler mBot Programlarken

Şekil 18’de öğrenci Scratch’ı kullanarak Park Et adlı etkinliği yapıyor.



Şekil 18: Scratch Programlama

Etkinlikleri belirledikten sonra karşılaştırılması yapılacak olan bilgi işlemsel düşünme becerisi ve akademik başarıyı ölçecek olan ölçeklerin belirlenmiştir. Bilgi işlemsel düşünme becerisini ölçmek için Harward Üniversitesi tarafından geliştirilmiş olan ölçek kullanılmıştır. Akademik başarıyı ölçmek için ise alan uzmanı tarafından hem Scratch hem de robotik gruplarında aynı kazanımları ölçecek ve aynı puana denk gelecek şekilde ölçekler hazırlanmıştır.

Öğrencilerin temel bilgisayar kullanım bilgisini ölçmek için alan uzmanı tarafından ön test hazırlanmıştır. Ön test Scratch veya robotik eğitimlerinden sonraki akademik başarıyı daha net görebilmek için hazırlanmıştır. Bu testte öğrencilerin bilgisayar donanımı (iç donanım, dış donanım, klavye özellikleri), yazılımı (işletim sistemleri, programlar, web yazılımları, web adresleri), internet güvenliği, bilgisayar ve sağlık gibi konulardaki akademik seviyesi ölçülmüştür.

Ön test sonrası öğrencilerle etkinlikler gerçekleştirilmeye başlanmıştır. Penguen dansı etkinliğinin hem robotik hem de 1. grup için oldukça keyifli geçtiği gözlemlenmiştir. Buna rağmen iki grupta da 2 ders saati (80 dakika) içerisinde görevini tamamlayamayanlar olmuştur. 1. grupta öğrencilerin karakter kostüm tasarımına ağırlık vermesi, 2. grubun ise robotun bağlanmasıyla ilgili sorunlar yaşaması etkinliğin tamamlanamama nedenleri olarak sıralanabilir.

Yangın var etkinliğinde ise hemen hemen her iki grupta görevi tamamlayabilmiştir. Tüm grupların görevi tamamlamalarının nedeni görevini tamamlayamayan öğrencilerin görevini kısa sürede tamamlayanlardan yardım alması olabilir. Park Et etkinliğinde de hemen hemen her iki grupta da görevi tamamlayamayan grup olmamıştır. Tüm grupların görevi tamamlamalarının nedeni görevini tamamlayamayan öğrencilerin görevini kısa sürede tamamlayanlardan yardım alması olabilir.

Etkinlikler sona erince son test yapılmıştır. Bilgi işlemsel düşünme becerileri ve akademik başarıları ölçülecek öğrenciler dört farklı sınıftan kura yöntemiyle seçilen 15'er öğrencinin oluşturduğu 60 kişilik gruptur. Sınıfların ikisini sadece kız öğrenciler, diğer ikisini de sadece erkek öğrenciler oluşturduğundan sonuçları analiz edecek grubun yarısını erkek, yarısını kız öğrenciler oluşturmuştur. Bilgi işlemsel düşünme becerisi ölçülecek öğrencilerle sırayla görüşme yapılmıştır. Görüşme esnasında öğrencilerin en çok beğendiği Scratch ya da Robotik etkinliği dosyası

açılarak çalışmalarını daha iyi hatırlayıp, yorumlamaları sağlanmıştır. Görüşmeler sadece öğrenci ve araştırmacının bulunduğu bilişim teknolojileri sınıfında yapılmıştır. Araştırmacı öğrenciye görüşme formunda yer alan soruları sormuş, öğrencinin sorulara verdiği cevaplar doğrultusunda öğrenciye puan vererek görüşme formunu doldurmuştur.

Çalışmanın ilk safhası bilgi işlemsel düşünme becerilerini ölçen ölçeğin uygulanmasından sonra sona erdi. İkinci safhanın uygulanması için 2 aylık yaz tatilinin geçmesi beklenmiştir. Tatil sonrası Scratch ve robotik gruplarının yerleri değiştirilerek etkinlikler gerçekleştirilmiştir. İlk aşamadan farklı olarak etkinlik sürecinde öğrenciler etkinlikleri daha kısa sürede tamamlamış ve etkinliklerini geliştirmeye vakit bulmuşlardır. Bunun nedeni olarak öğrencilerin programlama deneyimine sahip olmaları ve soyut düşünme becerisi kazanmaya başlamış olmaları gösterebilir. Etkinlikler sonrası Scratch ve robotik gruplarına uygun akademik başarı testleri uygulanmıştır.

3.5 Analiz

Gruplar arasındaki akademik başarılarının karşılaştırılması noktasında gruplar arası kovaryans analizi (ANCOVA) gerçekleştirilmiştir. Bu analizi gerçekleştirmek için en az iki gruptan oluşan bir bağımsız değişken, sonuçlar üzerindeki etkisinin kontrol edilmesi gereken bir değişken ve bir bağımlı değişken gerekmektedir. Analizdeki bağımlı değişken akademik başarı testinden alınan sonuçlar, bağımsız değişken ise grup bilgisini içeren değişken olarak belirlenmiştir. Bunun yanında öğrencilerin çalışmaya başladığı tarihe kadar aldıkları bilişim teknolojileri eğitime yönelik başarı düzeyleri de ölçülmüş ve analizde kontrol değişkeni olarak yer verilmiştir. Aynı değişkenler ile ANCOVA ikinci faz eğitimler sonunda tekrarlanmış ve öğrenci gruplarının aynı seviyede olduklarını anlamak için kullanılmıştır.

Bilgi-işlemsel düşünme pratikleri ile ilgili durumun irdelenmesi için ise çok değişkenli varyans analizi (MANOVA) çalıştırılmıştır. Bu analizde ise birden fazla ve birbiriyle de bağımlı olduğu düşünülen bağımlı değişken ve en az iki grup içeren bir bağımsız değişken yer almalıdır. Çalışmadaki bağımlı değişkenler içinde bilgi-işlemsel düşünme pratikleri ölçümünün alt boyutları yer almaktadır. Bağımsız değişken ise yine grup bilgilerini tutan değişken olarak belirlenmiştir.

3.6 Geçerlilik ve Güvenirlik

Bu bölümde çalışmaya etki edebilecek geçerlilik ve güvenilirlik unsurları kısaca açıklanmış ve alınan önlemler de tanıtılmaya çalışılmıştır. İç geçerlilik ele alınan konu olmuştur. Frankel, Wallen ve Hyun (2012)' e göre deneysel ve yarı deneysel çalışmalarda katılımcı özellikleri, katılımcı sayılarındaki değişiklikler, konum, katılımcıların tutumu ve uygulamadaki sorunlar iç geçerliliğe tehdit oluşturan unsurların başında gelmektedir. Çalışmamıza katılan öğrencilerin en önemli özelliği olarak görülen temel bilişim teknolojileri bilgi düzeyleri kontrol edilmiş ve analizlerin içine dahil edilmiştir. Ayrıca çalışmaya başlayan ve bitiren katılımcı sayısı da değişmediği için katılımcı kaybı tehdidi de gerçekleşmemiştir. Bunun yanında katılımcıların tutumunun çalışmaya etkisini en alt düzeye indirmek için iki uygulama benzer içerik ve ortamlarda gerçekleştirilmeye çalışılmıştır. Bu durum uygulamadan kaynaklı oluşacak tehditleri de en aza indirmiştir. Kullanılan tüm ölçekler de uzman görüşlerinden geçirilmiş ve gerekli düzenlemeler yapılmıştır. Güvenilirlik konusu için de yine ölçme araçlarının güvenilirliği üzerinde durulmuştur. Akademik başarı testi ve bilgi-işlemsel düşünme pratikleri ölçeği uzman görüşlerine tabi tutumu ve güvenilirlik açısından ölçme kabiliyetleri açısından incelenmiştir. İki aracın da güvenilirliği çalışma için uygun bulunmuştur.

DÖRDÜNCÜ BÖLÜM

IV. BULGULAR

Brennan ve Resnick (2012) ürün tabanlı görüşmeler, tasarım senaryoları ve öğrenci dosyaları olmak üzere üç farklı yaklaşımı kullanarak bilgi işlemsel düşünme becerisinin daha güvenilir bir şekilde ölçülebileceğini vurgulamıştır. Bu çalışmada ürün tabanlı görüşmeler kullanılarak öğrencilerin bilgi işlemsel düşünme pratikleri ölçülmeye çalışılmıştır. Buna göre bilgi işlemsel düşünme becerisi üç boyut altında ölçülmektedir. Bunlar bilgi işlemsel düşünme kavramları, bilgi işlemsel düşünme pratikleri, bilgi işlemsel bakış açılarıdır. Bilgi işlemsel düşünme kavramları sıra, döngüler, paralellik, olaylar, durumlar, operatörler ve veri olmak üzere yedi tanedir. Bilgi işlemsel pratikler ise, deneme-yineleme, test etme-hata ayıklama, yeniden kullanma-karıştırma, özetleme ve modüler hale getirme olmak üzere dört ana bölüm altında nitelendirilmiştir. Son olarak bilgi-işlemsel bakış açısı (derinlik) ise tanımlama, ilişkilendirme ve sorgulama olmak üzere üç unsuru içermektedir. Çalışmada Öğrencilerin bilgi işlemsel düşünme becerileri hakkında bilgi sahibi olmak amacı ile bu üç boyut içerisinde bilgi-işlemsel düşünme pratiklerine odaklanılmıştır. Aynı zamanda uygulama sonlarında programlamaya yönelik akademik başarı testleri de araştırmanın diğer temel boyutunu oluşturmuştur.

4.1 Akademik Başarı İle İlgili Sonuçlar

Tablo 7'e göre öğrencilerin bilgisayar becerisini ölçen sınavda 1. grubun not ortalaması 75.80 iken 2. grubun ise 68.83'dür.

Tablo 7: Temel Bilgisayar Becerisi Testi Tanımlayıcı Verileri

		Kontrol Türü	İstatistik	Standart Hata
TBBT	1. Grup	Ortalama	75.80	2.73
		Medyan	76.50	
		Std. Deviation	14.97	
		Minimum	48.00	
		Maksimum	100.00	
	2. Grup	Ortalama	68.83	3.12
		Medyan	69.00	
		Standart Sapma	17.12	
		Minimum	34.00	
		Maksimum	98.00	

Tablo 8'e göre 1. ve 2. gruba dört hafta süreyle verilen eğitimden sonra yapılan sınavda 1. grubun not ortalaması 81.96 iken 2. grubun not ortalaması 69.76'dir.

Tablo 8: 1. Sınava Ait Tanımlayıcı Veriler

		Kontrol Türü	İstatistik	Standart Hata
1. Sınav	1. Grup	Ortalama	81.96	1.95
		Medyan	83.00	
		Standart Sapma	10.73	
		Minimum	49.00	
		Maksimum	100.00	
	2. Grup	Ortalama	69.76	2.84
		Medyan	74.00	
		Standart Sapma	15.58	
		Minimum	36.00	
		Maksimum	94.00	

Tablo 9'a göre dört hafta süreyle 1. gruba robotik ve 2. gruba Scratch eğitimi verildikten sonra yapılan sınavda 1. grubun not ortalaması 81.76 iken 2. grubun not ortalaması 83.26'dır.

Tablo 9: 2. Sınava Ait Tanımlayıcı Veriler

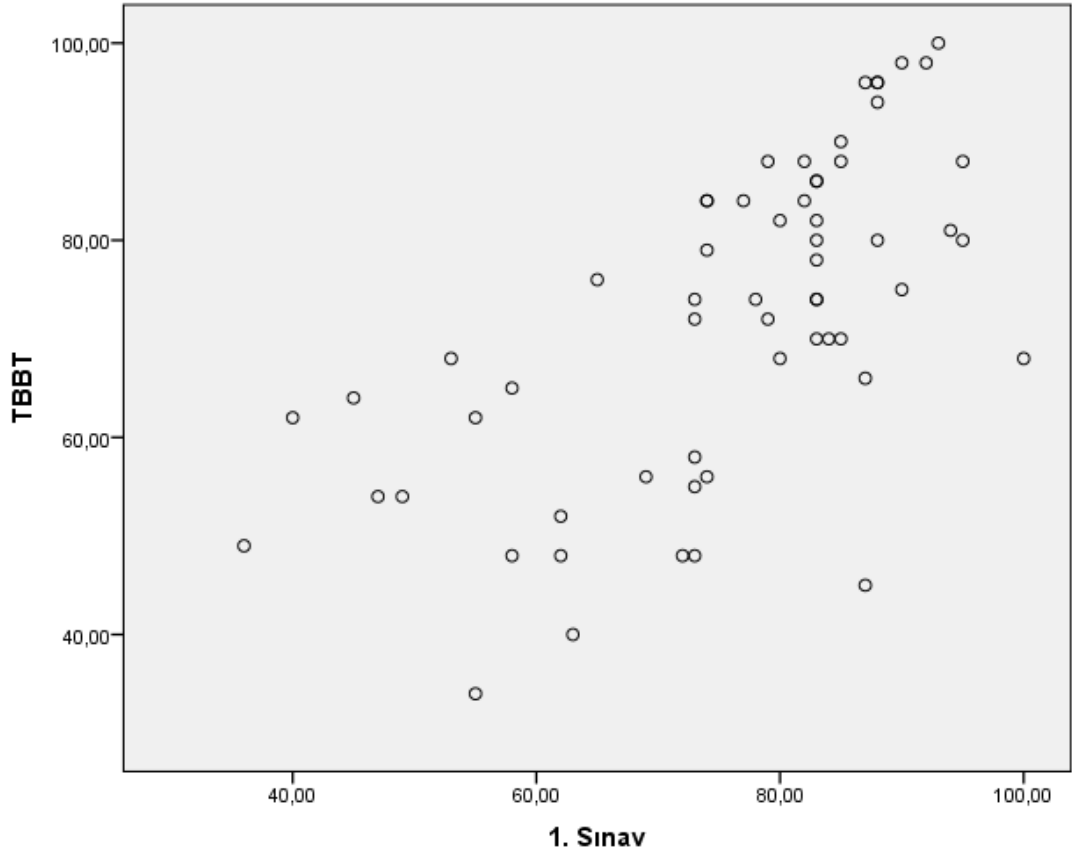
	Kontrol Türü	İstatistik	Standart Hata
1. Grup	Ortalama	81.76	2.21
	Medyan	82.50	
	Standart Sapma	12.13	
	Minimum	48.00	
	Maksimum	100.00	
	Ortalama	83.26	1.93
2. Sınav	Medyan	85.00	
	Standart Sapma	10.60	
	Minimum	54.00	
	Maksimum	100.00	

Grupların ilk Scratch ve Robotik programlama eğitimlerini tamamlamalarının ardından iki grup arasındaki programlamaya yönelik akademik başarıların farklılaşp farklılaşmadığını anlamak için ANCOVA (Analysis of covariance) gerçekleştirilmiştir. Bu analizde öğrencilerin temel bilgisayar becerileri kovaryans değişkeni olarak belirlenmiştir. Bu noktadan yola çıkılarak öncelikle analiz için gerekli olan varsayımlar test edilmiştir.

İlk varsayım olan bağımlı ve kovaryans değişkenlerinin sürekli değişken olmasıdır. Çalışmadaki iki değişken de sürekli değişken olarak çalışmada yer almıştır. İkinci varsayım ise bağımsız değişkenin en az iki bağımsız kategori içermesidir. Bu bağlamda bağımsız değişken grup 1 ve grup 2 olmak üzere iki bağımsız kategoriye içermektedir. Üçüncü varsayım ise ölçümlerin bağımsız şekilde gerçekleştirilmesidir. Başka bir deyişle birden fazla gruba giren ölçümün olmamasıdır. Bu duruma ilişkin

olarak tüm öğrenciler sadece bir gruba katılım göstermişlerdir. Yani gözlem bağımsızlığı varsayımı karşılanmıştır.

Dördüncü varsayım ise veri setinin anlamlı sayıda uç değer barındırmamasıdır. Bu noktada veri seti anlamlı olacak şekilde uç değer barındırmamaktadır. Bir başka varsayım ise kovaryans değişkeni ile bağımlı değişken arasındaki anlamlı bir ilişki bulunmasıdır. Bunun için bu değişken arasındaki scatter plot grafiği incelenmiştir (Şekil 19).



Şekil 19:Bağımlı ve kovaryans değişkenleri arası scatter plot grafiği

Buna göre grafik iki değişken arası olası bir ilişkiye işaret etmektedir. İki değişken arasındaki korelasyon katsayısı da .65 olarak hesaplanmış ve anlamlı bir değer olduğu görülmüştür. Normallik varsayımı için de Skewness ve Kurtosis değerleri her bir grup için ayrı ayrı şekilde kontrol edilmiştir. Bu değerler iki grup içinde +2 ve -2 aralığında bulunmuştur. Histogram verileri de normallik varsayımının sağlandığına işaret etmektedir. Bir başka varsayım ise regresyon eğimlerinin eşitliğidir. Bunun

için grup değişkeni ve kovaryans değişkeni arasındaki ilişkiye bakılmış ve anlamlı bir ilişki bulunmamıştır ($p=.85$). Bu da varsayımın sağlandığına işaret etmektedir. Aynı zamanda bağımlı değişkendeki hata varyanslarının eşitliği için Levene's Test kontrol edilmiş ve anlamlılık değeri .26 bulunmuştur. Dolayısıyla bu varsayım da sağlanmıştır. Varsayımlar incelendikten sonra veriler üzerinde ANCOVA gerçekleştirilmiştir (Tablo 10).

Tablo 10: Gruplar arası etki

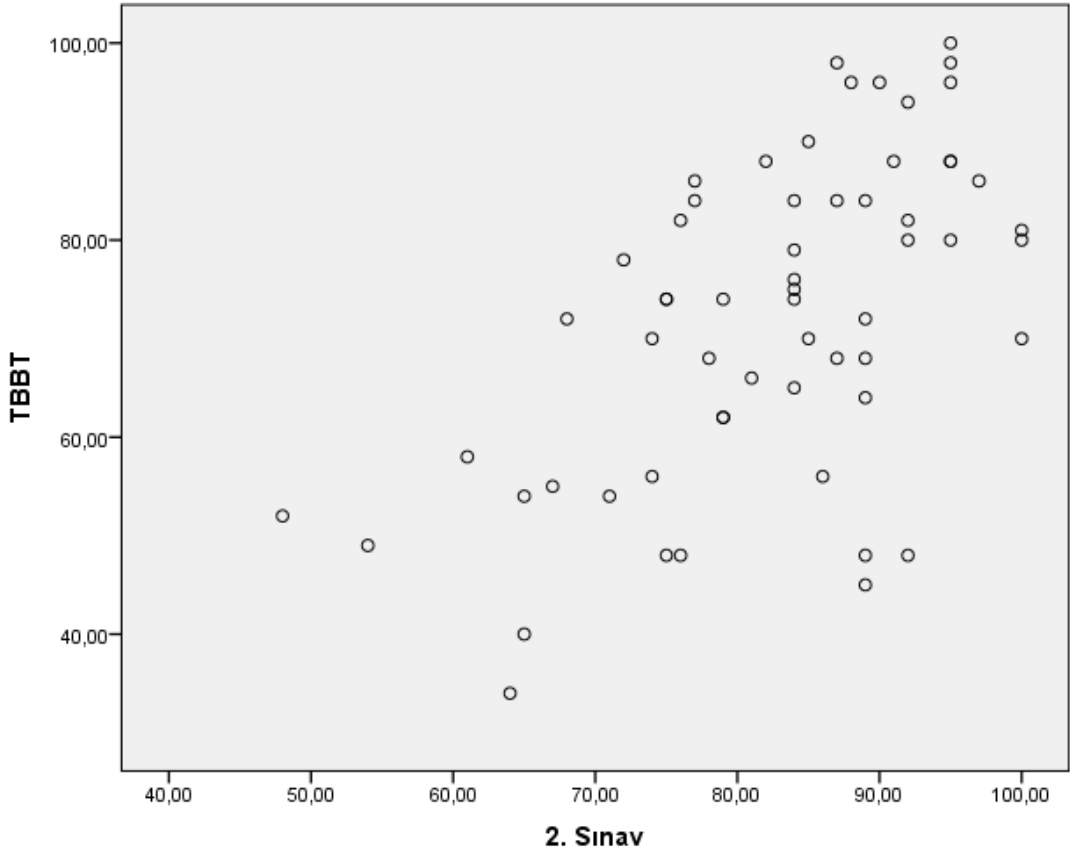
Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	6295.11	2	3147.55	28,388	.00	.50
Intercept	4007.45	1	4007.85	36.14	.00	.39
TBBT	4062.51	1	4062.51	36.63	.00	.39
Grup	1052.19	1	1052.19	9.49	.00	.14
Error	6321.82	57	110.91			
Total	357962.00	60				
Corrected Total	12616.93	59				

Scratch ile programlama eğitimi alan grup 1 ile robotik aracılığı ile programlama eğitimi alan grup 2'nin akademik başarıları arasında TBBT kontrol edilerek anlamlı bir fark olup olmadığı test edildiğinde, gruplar arası anlamlı bir fark olduğu ($F(1, 57) = 9.49, p < 0.5$) gözlemlenmiştir. Yani TBBT kontrol edildiğinde 1. grup 2. gruptan anlamlı şekilde yüksek başarı puanı elde etmiştir.

Çalışmanın ikinci aşamasında gruplara verilen programlama eğitimleri tekrarlanmış ancak içerikleri yer değiştirilmiştir. Bir başka deyişle, Scratch ile programlama eğitimi alan 1. gruba, Robotik programlama ile verilen eğitim verilmiştir. Aynı şekilde 2. gruba da Scratch ile programlama eğitimi verilmiştir. Eğitimin ikinci aşamasından sonra tekrar başarı testi yapılmıştır. Elde edilen veriler üzerinde birinci aşamada yapılan kovaryans (ANCOVA) analizi tekrarlanmıştır. Bu analizdeki tek fark bağımlı değişkenin yapılan son test sonuçları ile elde edilmesidir.

Analizden önce varsayımlar tekrar kontrol edilmiştir. İlk varsayım olan bağımlı ve kovaryans değişkenlerinin sürekli değişken olmasıdır. Çalışmadaki iki değişken de sürekli değişken olarak çalışmada yer almıştır. İkinci varsayım ise bağımsız değişkenin en az iki bağımsız kategori içermesidir. Bu bağlamda bağımsız değişken grup 1 ve grup 2 olmak üzere iki bağımsız kategoriye içermektedir. Üçüncü varsayım ise ölçümlerin bağımsız şekilde gerçekleştirilmesidir. Başka bir deyişle birden fazla gruba giren ölçümün olmamasıdır. Bu duruma ilişkin olarak tüm öğrenciler sadece bir gruba katılım göstermişlerdir. Yani gözlem bağımsızlığı varsayımı karşılanmıştır.

Dördüncü varsayım ise veri setinin anlamlı sayıda uç değer barındırmamasıdır. Bu noktada veri seti anlamlı olacak şekilde uç değer barındırmamaktadır. Bir başka varsayım ise kovaryans değişkeni ile bağımlı değişken arasındaki anlamlı bir ilişki bulunmasıdır. Bunun için bu değişken arasındaki scatter plot grafiği incelenmiştir (Şekil 20).



Şekil 20:Bağımlı ve kovaryans değişkenleri arası scatter plot grafiği

Buna göre grafik iki değişken arası olası bir ilişkiye işaret etmektedir. İki değişken arasındaki korelasyon katsayısı da .58 olarak hesaplanmış ve anlamlı bir değer olduğu görülmüştür. Normallik varsayımı için de Skewness ve Kurtosis değerleri her bir grup için ayrı ayrı şekilde kontrol edilmiştir. Bu değerler iki grup içinde +2 ve -2 aralığında bulunmuştur. Histogram verileri de normallik varsayımının sağlandığına işaret etmektedir. Bir başka varsayım ise regrasyon eğimlerinin eşitliğidir. Bunun için grup değişkeni ve kovaryans değişkeni arasındaki ilişkiye bakılmış ve anlamlı bir ilişki bulunmamıştır ($p=.81$). Bu da varsayımın sağlandığına işaret etmektedir. Aynı zamanda bağımlı değişkendeki hata varyanslarının eşitliği için Levene's Test kontrol edilmiş ve anlamlılık değeri .508 bulunmuştur. Dolayısıyla bu varsayım da sağlanmıştır. Varsayımlar incelendikten sonra veriler üzerinde ANCOVA gerçekleştirilmiştir (Tablo 11).

Tablo 11: Gruplar arası etki

Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	2847,060 ^a	2	1423,530	17,184	,000	,376
Intercept	7185,055	1	7185,055	86,733	,000	,603
TBBT	2813,310	1	2813,310	33,960	,000	,373
Grup	291,738	1	291,738	3,522	,066	,058
Error	4721,923	57	82,841			
Total	416109,000	60				
Corrected Total	7568,983	59				

Robotik aracılığı ile programlama eğitimi alan grup 1 ile Scratch ile programlama eğitimi alan grup 2'nin akademik başarıları ile TBBT arasında anlamlı bir fark olup olmadığı test edildiğinde, gruplar arası anlamlı bir fark olmadığı ($F(1, 57) = 3.52$,

$p>0.5$) gözlemlenmiştir. Yani TBBT kontrol edildiğinde 1. grup 2. gruptan anlamlı şekilde yüksek başarı puanı elde edememiştir.

4.2 Bilgi İşlemsel Düşünme Pratikleri İle İlgili Sonuçlar

Çalışmada öğrenciler dört ana uygulama alanına göre değerlendirilmiştir. Her uygulama alanına ait 4 alt başlık mevcuttur. Bu başlıklar doğrultusunda öğrencilerle yapılan görüşme düşük (1), normal (2), yüksek (3) olmak üzere puanlandırılmıştır (Assessing Development Of Computational Practices,2019).

Tablo 12: Bilgi İşlemsel Düşünme Pratikleri Tanımlayıcı İstatistikleri

	Kontrol Türü	Mean	Std. Deviation	N
Deneme ve Yanılma	1. Grup	7.40	2.56	30
	2. Grup	7.13	2.55	30
	Total	7.26	2.54	60
Test Etme ve Hata Ayıklama	1. Grup	7.06	2.36	30
	2. Grup	6.70	2.87	30
	Total	6.88	2.61	60
Yeniden Kullanma ve Karıştırma	1. Grup	6.66	2.33	30
	2. Grup	6.40	2.06	30
	Total	6.53	2.18	60
Özetleme ve Modüler Hale Getirme	1. Grup	5.53	1.92	30
	2. Grup	5.60	1.94	30
	Total	5.56	1.91	60

Tablo 12 incelendiğinde bilgi işlemsel düşünme becerisi pratiklerinin değerlendirilmesi için belirlenen dört ana uygulama grubundan birincisi olan deneme ve yanılmada 1. grubun ortalaması 7.40 ve 2. grubun ortalaması ise 7.13'tür. Test etme ve hata ayıklamada 1. grubun ortalaması 7.06 ve 2. grubun ortalaması ise 6.70'dir. Yeniden kullanma ve karıştırmada 1. grubun ortalaması 6.66 ve 2. grubun

ortalaması ise 6.40'dır. Özetleme ve modüler hale getirmede 1. grubun ortalaması 5.53 ve 2. grubun ortalaması ise 5.60'dır.

Bilgi-işlemsel pratiklerin ölçümünden elde edilen bilgiler ile çok değişkenli varyans analizi (MANOVA) gerçekleştirilmiştir. Analizden önce ilgili varsayımlar test edilmiştir. Birinci varsayım olarak en az iki sürekli bağımsız değişkenin varlığıdır. Bu çalışmada bilgi-işlemsel pratiklerin dört alt boyutu sürekli değişken olarak ölçülmüş ve analizde bağımsız değişken olarak yer almıştır. İkinci varsayım, bağımsız değişkenin en az iki kategori içermesidir. Analizdeki bağımsız değişkenin 1. grup (Scratch grubu) ve 2. grup (robotik grubu) olmak üzere iki kategorisi mevcuttur. Üçüncü varsayım olan bağımsız gözlem, iki grubun birbirinden bağımsız olması ve etkileşim içinde olmamaları nedeniyle sağlanmıştır. Ayrıca her bir grupta 30'ar öğrencinin verisi olduğu için de analiz için gerekli olan minimum sayı da aşılmıştır.

Uç değerler ise Mahalanobis değerleri kontrol edilerek test edilmiştir. Dört bağımlı değişken için kullanılan geçerli olan kritik değerden (18.47) büyük değere rastlanmamıştır. Bu nedenle veri setinde uç değer olmadığı sonucuna varılmıştır. Her bağımsız değişken ikililerinin arasındaki ilişkiler Scatter-Plot matrisleri ile kontrol edilmiş ve tüm ikililer arasında doğrusal bir ilişki olduğu gözlemlenmiştir. Bu nedenle her bir grup için bağımlı değişkenler arasındaki doğrusal ilişki varsayımı da sağlanmıştır.

Çok değişkenli normallik varsayımı da MANOVA için kontrol edilmesi gereken varsayımlardan biridir. Ancak SPSS ile doğrudan test edilemediği için birinci ve ikinci gruplar bağlamında ayrı ayrı olarak her bir bağımlı değişken için normallikler kontrol edilmiştir. Yani birinci grup için bağımsız değişkenin, yine aynı şekilde ikinci grup için de dört bağımlı değişkenin normallikleri test edilmiştir.

Tablo 13: Box's Test

Box's M	7.85
F	.72
df1	10
df2	16082.86
Sig.	.70

Kovaryans matrislerinin eşitliği varsımı için ise Box's M test değeri kontrol edilmiştir. Tablo 13'te belirtildiği şekilde anlamlılık değeri .72 olarak hesaplanmıştır. Dolayısıyla bu varsayım da yerine getirilmiştir. Son varsayım olarak çoklubağlantılılık (multicollinearity) varsayımı test edilmiştir. MANOVA için bağımlı değişkenler arasında orta düzeyde bir ilişki olması gerekmektedir. İlişki düzeyi anlamsız ise ya da .90'dan yüksek bir ilişki olması durumlarında MANOVA'nın bu varsayımı karşılanamamaktadır. Bu çalışmadaki dört bağımlı değişkenin korelasyon matrisine bakıldığında her bir ikili arasında anlamlı olan ancak .90'ı aşmayan ilişki olduğu gözlemlenmiştir.

Tablo 14: Çoklu deęişken testi

Effect	Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power ^c	
Intercept	Pillai's Trace	.91	146.64 ^b	4.00	55.00	.00	.91	586.58	1.00
	Wilks' Lambda	.08	146.64 ^b	4.00	55.00	.00	.91	586.58	1.00
	Hotelling's Trace	10.66	146.64 ^b	4.00	55.00	.00	.91	586.58	1.00
	Roy's Largest Root	10.66	146.64 ^b	4.00	55.00	.00	.91	586.58	1.00
Tur	Pillai's Trace	.01	.20 ^b	4.00	55.00	.93	.01	.83	.09
	Wilks' Lambda	.98	.20 ^b	4.00	55.00	.93	.01	.83	.09
	Hotelling's Trace	.01	.20 ^b	4.00	55.00	.93	.01	.83	.09
	Roy's Largest Root	.01	.20 ^b	4.00	55.00	.93	.01	.83	.09

($p = 0.93$. $p > 0.05$) anlamlı bir ilişki yoktur.

Varsayımlar test edildikten sonra MANOVA analizi çalıştırılmıştır. Tüm varsayımlar sağlandığı için Wilk's Lambda değeri analizin anlamlı sonuç verip vermediğini anlamak için kontrol edilmiştir. Elde edilen sonuçlara göre gruplar arasında bilgi işlemsel düşünme pratiklerinin herhangi bir boyutunda anlamlı bir farklılığa rastlanmamıştır ($p = 0.91$. $p > 0.05$).

Her bir durum için basıklık ve çarpıklık değerlerine bakıldığında -1.5-+1.5 sınırlarının dışında bir değer gözlemlenmemiştir. Bu nedenle çok deęişkenli normallik varsayımının karşılandığı kabul edilmiştir.

Tablo 15: Levene Testi 2

	F	df1	df2	Sig.
Deneme ve Yanılma	.23	1	58	.63
Test Etme ve Hata Ayıklama	4.16	1	58	.04
Yeniden Kullanma ve Karıştırma	.87	1	58	.35
Özetleme ve Modüler Hale Getirme	.00	1	58	.96

BEŞİNCİ BÖLÜM

V. SONUÇ, TARTIŞMA VE ÖNERİLER

Bu bölümde araştırma için yapılan literatür taraması ve araştırmada elde edilen bulgular özetlenmiş, araştırmada elde edilen sonuçlar iki başlık altında verilerek tartışılmış ve araştırma sonucuna göre önerilerde bulunulmuştur.

Bilgi işlemsel düşünme 21. Yüzyılın temel becerilerinden biridir (Wing, 2006). Pek çok ülkenin müfredatında yer almaktadır. Bilgi işlemsel düşünme becerisinin kazandırılmasında programlama öğretimi büyük rol oynamaktadır (The Partnership for 21st Century Skill, 2009). Scratch uygulaması ise programlama öğretiminde yaygın olarak tercih edilen bir yazılımdır. STEM'in yaygınlaşmasıyla önemi ve kullanımı artan robotik de programlama öğretiminde tercih edilmektedir. Yapılan araştırmada bilgi işlemsel düşünme becerisinin kazandırılmasında ve akademik başarının artırılmasında Scratch'ın mı yoksa robotiğin mi daha etkili bir öğrenme aracı olduğu sorusuna cevap aranmıştır.

Bundy (2007) 21. Yüzyılı anlamak için bilgi işlemsel düşünmeyi anlamalısınız diyerek çağımız için bu becerinin önemini vurgulamıştır. Bilgi işlemsel düşünme bir bilgi işlemcisi (bilgisayar, makine, insan) tarafından etkili bir biçimde gerçekleştirilecek şekilde problemlerin formüle edildiği düşünme sürecidir. (Cuny, Snyder ve Wing, 2010). ISTE (2015) bilgi işlemsel düşünme becerisinin eleştirel düşünme, yaratıcı düşünme, problem çözme, algoritmik düşünme, iletişim becerileri ve işbirlikli öğrenmeyi kapsadığını ifade etmekte ve bu becerilerle tanımlanabileceğini vurgulamaktadır. Kodlama soyut kavramları somutlaştırabilme, yaratıcılık, takımla çalışma ve problem çözme gibi becerileri kazandırabilir. Bu beceriler kodlamayı bilgi işlemsel düşünme becerisinin kazanılmasında etkili bir araç yapabilir (ET 2020, 2016.).

ISTE'nin 2016 yılında öğrenciler için belirlediği standartlardan biri de bilgi işlemsel düşünür (Computational Thinker) olmalıdır. Her öğrencinin sahip olması gereken 21. yüzyıl becerilerinden biri olarak karşımıza çıkan bilgi işlemsel düşünmenin

kazandırılmasında kullanılan yöntemlerden biri olan programlama birçok ülkenin ilk ve ortaöğretim programında zorunlu olarak yer alırken bazı ülkelerin öğretim etkinliklerinin bir parçası olarak desteklenmiştir (Grover ve Pea, 2013; Mannila ve diğerleri, 2014).

Ülkemizde bilgi işlemsel düşünme becerisine ilk ve ortaöğretim programında yer vermiştir. Öğretim programının genel amaçlarınının 5. maddesi “Problem çözme ve bilgi işlemsel düşünme becerileri edinmelerini ve geliştirmelerini amaçlamak” şeklindedir (Milli Eğitim Bakanlığı [MEB], 2017:ss 5). Ayrıca “Bilgi teknolojileri alanında her öğrenciye teknolojiye yararlanma ve bilgi-işlemsel düşünme becerisi kazanabilme fırsatı sağlanmalıdır” öğretim programının uygulanmasında dikkat edilecek hususlardan biri olarak belirtilmiştir (MEB, 2017:ss 11). Bu durum ülkemizde 21. yüzyıl becerilerinden biri olan bilgi işlemsel düşünmenin önemsendiği ve temel becerilerden biri olarak görüldüğü şeklinde yorumlanabilir.

Farklı amaçları gerçekleştirmek amacıyla elektronik aygıtların programlanmasında özel kelime ve semboller içeren komutlardan oluşan programlama dilleri kullanılır (Van-Roy ve Haridi, 2004). Komutların yazılmasıyla ortaya belli bir amaca hizmet eden programlar ve yazılımlar çıkar (TDK, 2017). Farklı teknolojiler için hazırlanmış pek çok programlama dili vardır ve her dilin kendine has yazım kuralları ve deyimleriyle hedefe uygun komutlar oluşturularak çalıştırılır (Van-Roy ve Haridi, 2004). Dile has kodların yazılması işlemine kodlama veya programlama, denir. Ortaya çıkan ürüne ise program veya uygulama denir Programlama yazma, test etme ve hata ayıklama gibi süreçlerden oluşan ve üst düzey bir bilişsel beceri gerektiren karmaşık bir problem çözme süreci olarak ifade edilebilir (Kert ve Uğraş, 2009; Fesakis ve Serafeim, 2009).

İlköğretim düzeyindeki öğrencilerin programlama ile karşılaşması onların bilişsel gelişimine (Papert, 1980; Clements ve Gullo, 1984; Lewis, 2010; Akpınar ve Altun, 2014; Atmatzidou ve Demetriadis, 2016) ve programlamaya dair algılarına (Lewis, 2010; Kalelioğlu ve Gülbahar, 2014) olumlu katkı sağlamaktadır.

Görsel programlama yazılımları küçük yaş grubundaki öğrencilere bilgi işlemsel düşünme becerisi kazandırmak için kullanım kolaylığı sağlarken deneyimi az olanlara bile kod yazım hatalarına düşmeden tasarım ve üretim imkânı sunar (Grover

ve Pea, 2013). Kodlama öğretiminde tercih edilen araçlardan biridir (Armoni ve diğerleri, 2015; Brennan ve Resnick, 2012).

Programlama öğretimine yardımcı yazılım ve araçlar olarak ya da görsel yazılım araçları olarak adlandırılan kod bloklarının sürükleyip bırakılması gibi kolay bir yöntemle kullanıma olanak sunan yazılımlar yaygınlaşmıştır. Elektronik devre elemanlarının programlayarak programlama eğitiminde kullanılmasını sağlayan, somut ürünlerin ortaya konulmasını sağlayan yazılımlar da mevcuttur.

Araştırmada görsel yazılım araçlarından olan Scratch ve mBlock kullanılmıştır. Ücretsiz bir görsel programlama uygulaması olan Scratch, Massachusetts Teknoloji Enstitüsü (MIT) Medya Laboratuvarında geliştirilen blok programlama yapısına sahip, sürükle bırak etkileşimi ile kullanılan bir yazılımdır. Animasyon, oyun ve etkileşimli sunular oluşturma ve geliştirme olanağı sunan Scratch, özellikle küçük yaştaki öğrencilerin dikkatini çekecek programlama olaylarını içeren bir yapıya sahiptir. mBlock ise Scratch benzeri grafiksel programlama arayüzüdür. Program, günlük hayatta kullandığımız dile en yakın ve en anlaşılır şekilde kullanılan kod bloklarını barındırır. mBlock ile animasyon, oyun ve etkileşimli sunular yapabileceğimiz gibi mbot adlı robot kitini programlayabiliriz. Ayrıca arduino kart programlamaya da olanak tanımaktadır.

Araştırma 5. Sınıf öğrencilerinin programlama öğretiminde robotik ve Scratch uygulamalarının öğrencilerin bilgi işlemsel düşünme becerilerine ve akademik başarılarına etkisinin karşılaştırılması ve incelenmesi için yapılmıştır. Nicel yöntemler tercih edilmiştir. Araştırmada etkinlikler sonunda Harvard Üniversitesi tarafından Scratch projelerinde öğrencilerin bilgi işlemsel düşünme becerisi pratiklerini ölçmek için hazırlanmış olan ölçek kullanılmıştır. Görüşme tekniği kullanılarak öğrencilerin bilgi işlemsel düşünme pratikleri ölçülmeye çalışılmıştır. Alan uzmanı tarafından hazırlanan Scratch ve mBlock testleri kullanılarak akademik başarı ölçülmeye çalışılmıştır. Öğrencilerin başarısını ölçmek için kullanılan ölçeklerde Scratch ve mbot eğitimi alan gruplara sorulan sorular hemen hemen benzer olup genelde sorulardaki şekillerin arayüzü (robotik eğitimi alan gruba mBlock, Scratch eğitimi alan gruba Scratch arayüzü) değişmiştir.

Araştırmada yer alacak öğrenciler dört farklı sınıftan kura yöntemiyle seçilmişlerdir. Öğrencilerin yarısı kız, yarısı erkek olacak şekilde kura çekilmiştir. Dört sınıftan 15'er öğrenci olmak üzere toplam 60 öğrenci belirlenmiştir.

Çalışmada veri toplama aracı öğrencilerin ders içerisindeki etkinlikler doğrultusunda yapmış oldukları çalışma ile ilgili görüşme tekniği ve alan uzmanı tarafından hazırlanmış olan test sonuçları olacaktır.

Öğrenciler 2 gruba ayrılmıştır. 1. gruba Scratch, 2. gruba ise robotik etkinlikleri ile bilgi işlemsel düşünme becerisinin kazandırılması hedeflenmiştir. Her iki gruba dört haftalık eğitim verilmiştir.

Öğrenci başarısını arttırmak için önce Scratch mı yoksa robotik mi öğretilmeli sorusuna cevap bulabilmek için aynı öğrenci gruplarına Scratch öğrenenlere robotik, robotik öğrenenlere ise Scratch eğitimi verilmiştir. Eğitim sonundaki sınav sonuçları karşılaştırılmıştır. Öğrencilerin temel bilgisayar becerisini ölçen testte 1. grubun puan ortalaması, 2. grubun puan ortalamasından 7 puan daha yüksek çıkmıştır. Gruplara dört hafta süreyle verilen eğitimden sonra yapılan 1. sınavda 1. grubun puan ortalaması, 2. grubun puan ortalamasından 12 puan daha yüksek çıkmıştır. Dört hafta süreyle 1. gruba robotik ve 2. gruba Scratch eğitimi verildikten sonra yapılan 2. sınavda 1. grubun puan ortalaması, 2. grubun puan ortalamasından 2 puan daha az çıkmıştır.

Sınavlar (TBBT, 1. ve 2. sınav sonuçları) arasında anlamlı bir fark bulunmuştur. Sınavlar ve tür (1. ve 2. Grup ve TBBT, 1. ve 2. sınavlar) arasında ise anlamlı bir fark bulunmamıştır.

Bilgi işlemsel düşünme becerisi pratiklerinin değerlendirilmesi için belirlenen dört ana uygulama grubundan birincisi olan deneme ve yanılmada 1. grubun ortalaması, 2. grubun ortalamasından 0.3 daha yüksek çıkmıştır. Test etme ve hata ayıklamada 1. grubun ortalaması, 2. grubun ortalamasından ise 0.4 daha yüksek çıkmıştır. Yeniden kullanma ve karıştırmada 1. grubun ortalaması, 2. grubun ortalamasından 0.3 daha yüksek çıkmıştır. Özetleme ve modüler hale getirmede 1. grubun ortalaması, 2. grubun ortalamasından 0.1 daha az çıkmıştır. Bilgi işlemsel düşünme pratiklerinde 1. ve 2. grup arasında anlamlı bir fark bulunmamıştır.

5.1 Akademik Başarının Karşılaştırılması

Elde edilen bulgular Scratch ile programlama veya mBlock ile robot programlama arasındaki akademik başarıda anlamlı bir fark olmadığını göstermiştir. Hem Scratch hem de robotik gruplarında yapılan sınavlar sonrası akademik başarıda artış gözlemlenmiştir. Bu verilerden yola çıkarak programlama öğretiminde bu iki öğrenme aracının akademik başarıya etkisinde anlamlı bir fark olmadığını fakat her iki araçta akademik başarıyı olumlu yönde etkilediğini söyleyebiliriz.

Öğrenciler bir ay süren Scratch ve robotik eğitimlerin öncesinde ve sonrasında akademik başarı testine tabi tutulmuşlardır. Bu iki ölçüm arasında 5 haftalık bir süre vardır. Etkinliklerin 3 ay sonrasında gruplar yer değiştirerek aynı eğitimler tekrarlanmıştır. Eğitim sonunda ise üçüncü kez akademik başarıları ölçülmüştür. Sınav ortalamaları incelendiğinde 2. sınavın sonuçlarının TBBT ve 1. sınavdan daha yüksek olduğu gözlemlenmiştir.

TBBT ve 1. sınav sonuçlarına göre gruplararası puanların karşılaştırılması ile 1. grubun 2. gruptan anlamlı bir şekilde yüksek puan aldığı gözlemlenmiştir. Fakat TBBT ve 2. sınav sonuçlarına göre gruplararası puanların karşılaştırılması ile 1. grup ve 2. grubun puanları arasında anlamlı bir puan farkının olmadığı gözlemlenmiştir. Öğrencilerin Scratch ile programlamaya başlamasının robotik ile başlamasından akademik olarak daha verimli sonuçlar doğurduğunu göstermiştir. Fakat grupların değişmesi ve robotik eğitiminin ardından Scratch, Scratch eğitiminin ardından robotik eğitimi verilmesiyle akademik başarının dengelendiği görülmüştür. Böylece programlama öğretim sürecinde hem Scratch hem de robotiği kullanacaklar için tercih sırasının önemli olmadığı, sadece birini tercih edeceklerin Scratch'ı tercih etmesinin akademik başarıyı daha olumlu etkileyeceği sonucuna varılmıştır.

Araştırmada gruplara yer değiştirilerek robotik ve Scratch eğitimi verilmiştir. Ardından aldıkları eğitime uygun ölçek kullanılarak ölçüm yapılmıştır. Bu ölçüm her iki öğrenme aracını da kullanmayı tercih edecek bir eğitmenin hangi aracı önce kullanırsa akademik başarıyı daha çok arttırabileceğini görebilmek için yapılmıştır. Araştırma sonucu iki grupta da akademik başarıda artış olduğunu göstermiştir. Fakat puan ortalamalarına bakıldığında 1. grubun 1. sınav ile 2. sınav puan ortalamalarında 0.2 puan azalma olmuştur. Fakat 2. grubun 1. sınav ile 2. sınav ortalamalarında 13.5 puan artma olmuştur. 1. ve 2. gruplardaki 1. sınav ile 2. sınav arasındaki puan

ortalamalarına bakılarak önce Scratch sonra mBlock eğitiminin verilmesinin akademik başarıyı daha olumlu etkilediği görülmüştür. Bu da öğrencilerin 2. sınava girdiklerinde programlama konusunda daha çok bilgi ve tecrübe sahibi olduğunu gösterebilir.

Programlama öğretiminde karşılaşılan zorluklardan biri de soyut düşünme yetersizliğidir (Cevahir ve Özdemir, 2017). Öğrenciler Piaget'e (1955) göre somut işlemler (9-10 yaş) döneminden soyut işlemler dönemine (11-12 yaş) geçiş sürecindedirler. 1. sınav ile 2. sınav arasında yaklaşık 5 aylık bir süreç vardır. 5 aylık süreç onların zihinsel gelişim yaşadığı süreci içerebilir. Bu durum da 2. sınavda ortalamanın yüksek olmasını açıklayabilir.

Hem 1. hem de 2. grup 1. sınavda ilk kez almış oldukları kodlama eğitimi sonrası sınav olmuşlardır. 2. sınavda ise almış oldukları kodlama eğitimini farklı öğrenme araçlarını kullanarak tekrar almışlardır. İkinci eğitimde her ne kadar kullanılan araç farklı olsa da aynı eğitimi almışlardır. Yani aldıkları ilk eğitimi farklı araçlar kullanarak tekrar etmişlerdir. Bu durum da 2. sınavdaki başarı artışının nedenlerinden biri olabilir.

Scratch iki boyutlu (x,y) bir uygulama olduğundan 1. gruptakiler karakterlerini hareket ettirmek için x ve y değerlerini kullanmak zorunda kalmışlardır. Bu da soyut bir konu olan koordinat sistemi bilgisini gerektirmektedir. Ülkemiz müfredatında koordinat sistemi 7. Sınıfta öğrencilere anlatılmaktadır (Araştırmada 5. Sınıf öğrencileri ile çalışılmıştır.). Dolayısıyla öğrenciler bu kavramlarla ilk kez bu derste karşılaşmışlardır. Örneğin penguen dansı etkinliğinde seçilen karakter iki sağa iki sola bir ileri bir geri üç ileri şeklinde hareket ettirilmelidir. Öğrenciler sağa ve sola hareketlerde karakterlerin sağa sola bakan kostümlerine geçiş yaparak bu hareketi yaptırmışlardır. Fakat ileri ve geri hareket ettirmede sadece y ekseninde arttırma ve azaltma yapması gerektiğini kavramakta oldukça zorlanmıştır. Çünkü x ve y kavramı onlar için oldukça yeni ve soyut bir kavramdır. Bu durum da Scratch ile programlama öğrenmede öğrenciler için artı bir yük oluşturmuş olabilir. Ayrıca Scratch ile çalışırken öğrenciler kodlamanın yanı sıra karakter seçimi, karakter tasarımı, ortam seçimi, ortam tasarımı gibi süreçlerden geçmişlerdir. Bu süreçler öğrencilerin asıl amaçları olan programlama öğrenmeyi ikinci plana atmış olmalarına sebep olmuş olabilir.

2. gruptaki öğrenciler ise motor, kapı, sensör, led gibi kavramlarla ilk kez bu derste karşılaşmışlardır. mBlock adlı yazılımla birlikte mBot adlı robotun donanımsal özelliklerini de tanımaya çalışmışlardır. mBotun bilgisayara ve mBlock adlı yazılıma nasıl bağlanacağı, sensörlerin hangi kapıya bağlı olduğu, çalışma esnasında yaşanacak donanımsal problemlerin ne olduğunu ve nasıl çözümler üretilmesi gerektiğini öğrenmişlerdir. Bunlara robot sensörlerinin duyarsızlaşması ya da motor performansının düşmesinin nedeninin robotun azalan şarjının olması ya da verilen komutlara göre hareket etmeyen robotun bilgisayarla bağlantısında problemin olmuş olması örnek olarak verilebilir. Bu durumlar da mBlock ile programlama öğrenmede öğrenciler için artı bir yük oluşturmuş olabilir. Ayrıca zaman kaybına da neden olduğu için etkinliğin tamamlanamaması, etkinlikle kazanılacak kazanımların elde edilememesi gibi sonuçlara neden olmuş olabilir.

Çalışmada öğrenci ister Scratch isterse mBlock arayüzü ile robot programlayarak öğrensin akademik başarısının artış göstereceği sonucuna ulaşılmıştır. Akademik başarıyı hedef alan bir öğretmen her iki öğrenme aracını da tercih edebilir. Fakat seçimini kendini geliştirdiği ve bir öğretmen olarak konuyu daha iyi aktarabileceğini düşündüğü öğrenme aracından yana yapması daha uygun olacaktır.

Scratch ile programlama öğretiminde sadece kodlama yapılacak cihaza ihtiyaç varken robot ile programlama öğretiminde hem kodlama yapılacak bir cihaza hem de kodlanacak robota ihtiyaç vardır. Bu da robotla programlama öğretimini daha masraflı kılmaktadır. O yüzden öğretmen çalıştığı kurumun ya da öğrencilerinin maddi imkanını da tercih yaparken göz önünde bulundurmalıdır. Yeterli sayıda alınmayan ya da öğrencilerin kullanımına uygun olmayan (sağlam, kolay kodlanabilir, kullanışlı arayüze sahip, yeterli miktarda sensörü ve motoru bulunan vs.) robotlar ile verilecek programlama öğretimi öğretimin kalitesini düşürebilir.

Çalışmada 3 öğrenciye bir bilgisayar düşerken. 6 öğrenciye bir robot düşmüştür. Bu da çalışmanın sonuçlarını etkileyen durumlardan biri olabilir. Robotla çalışan öğrenciler kendi içlerinde robotu programlama, robotu platforma yerleştirme, takip etme, kodlama esnasında robotu elinde tutup koruma gibi görevler aldılar. Bu yüzden çalışma sürecinde bazı öğrenciler kodlamadan çok kodlamayla hareket eden robotla ve robotun yanlış ya da doğru bir şekilde kodlanıp kodlanmadığıyla ilgilendiler. Kodlamayla uğraşanlara robotun istenilen hareketi yapıp yapmadığıyla ilgili bilgi

verdiler ve kodlamayla uğraşanlar onlardan gelen dönütlerle kodları düzelttiler. 1. grupta ise kodlama olayı ve kodlamanın sonucu ekranda olduğu için gruplar sadece ekrana odaklandılar. Daha az sayıda oldukları için daha rahat çalıştılar. mBlock ile robot programlayanlardan farklı olarak verilen görevler doğrultusunda sahneleri kendileri tasarladılar ve çalışmak istedikleri karakterleri kendileri seçtiler. Bu durum Scratch ile programlama öğrenen öğrencilerin dijital görsel tasarım becerilerine olumlu katkı sağlamış olabilir. Ayrıca öğrencinin programlama öğrenme amacını ikinci plana atarak dijital görsel tasarım becerilerini geliştirme isteği uyandırmış da olabilir.

Scratch dijital görsel tasarım becerilerini geliştirmenin yanında yaratıcı düşünme becerilerini de geliştirmektedir (Kobsiripat, 2015). Scratch ile öğrenciler verilen görevleri dijital görsel tasarım becerilerini de kullanarak birbirlerinden farklı şekilde yerine getirirken robotikle sadece öğrenciler birkaç ufak değişiklikle görevlerini yerine getirmiştir. Örneğin penguen dansı etkinliğinde Scratch ile çalışan öğrenciler farklı karakterleri seçmiş, arka planları farklı kullanmış, dans figürlerini seçtikleri karakterlere özgü hale getirmişlerdir. Robotikte ise öğrenciler farklı olarak sadece farklı dans figürlerinde farklı nota çaldılar ya da farklı renkteki led ışıklarını yaktılar. Scratch sunduğu dijital görsel tasarım aracıyla ve öğrencilere sunduğu geniş karakter yelpazesıyla öğrencilerin yaratıcı düşünme becerilerini desteklemiş olabilir.

Araştırmada robot programlama ortamı olarak robotların (mBotların) programlanmasını sağlayan mBlock tercih edilmiştir. Scratch ile yapılan kodlama ve tasarımların çıktısı ekranda, mBlock ile ise yapılan kodlamaların çıktıları robotun hareketlerinde gözlemlenecek şekilde etkinlikler oluşturulmuştur. MBlock ile robot programlama Scratch'tan farklı olarak motor, sensör ve led kavramlarını içermektedir. Süreçte öğrencilerin bu kavramların nasıl kodlanması gerektiğini öğrenmeleri gerekmektedir. Kodlarıyla fiziksel nesnelere yönlendirmeye çalışmaları öğrencilerin elektronik bilgi ve becerilerini arttırmış ve desteklemiş olabilir. Dijital görsel tasarım ile uğraşmadıkları için de kodlamaya daha çok odaklanmış olabilirler.

Yenilik etkisi (novelty effect) öğrenmenin öğretim yöntemine bağlı olarak değil yenilik etkisinden dolayı gerçekleştirildiği düşüncesidir (Clark, 1989). Buna göre öğrencideki öğrenme performansının artışının nedeni öğretimin kalitesi değil öğretim yöntemi uygulanırken kullanılan teknolojinin daha önce

karşılaşılmaş olduđu için güdülenmenin artmasıdır. Ama bu durum geçicidir (Merchant, Goetz, Cifuentes, Keeney-Kennicutt ve Davis, 2014). Çalışma öncesinde öğrencilerin programlama deneyimi yoktur. Öğrenciler için hem Scratch hem de robotik yeni bir öğrenme ortamıdır. Çalışma sonrasında öğrencilerin akademik düzeyleri arasında anlamlı bir fark olmadığı için öğrenme ortamlarındaki yenilik etkisiyle ilgili bir yorum yapılamamaktadır. Fakat grupların sınavlardan almış oldukları not ortalamalarının artışının nedenlerinden biri olarak yenilik etkisi gösterilebilir.

5.2 Bilgi İşlemsel Düşünme Pratiklerinin Karşılaştırılması

21. yüzyılın temel becerilerinden biri olarak kabul edilen bilgi işlemsel düşünme becerisinin değerlendirilmesiyle ilgili çok az fikir birliği olduğunu söyleyen Brennan ve Resnick (2012) genç öğrenenlerin bilgi işlemsel düşünme becerisi gelişimini değerlendirebilecek bilgi işlemsel kavramlar, pratikler ve bakış açıları olmak üzere üç aşamalı bir yöntem geliştirmişlerdir.

Bilgi işlemsel kavramlar sıra, döngüler, paralellik, olaylar, durumlar, operatörler ve veri olmak üzere yedi tanedir. Deneme-yineleme, test etme-hata ayıklama, yeniden kullanma-karıştırma, özetleme ve modüler hale getirme olarak dört ana uygulama grubu belirlenmiştir. Bakış açısı (derinlik) ise tanımlama, ilişkilendirme ve sorgulama olmak üzere üç unsuru içermektedir (Brennan ve Resnick, 2012).

Brennan ve Resnick (2012) ürün tabanlı görüşmeler, tasarım senaryoları ve öğrenci dosyaları olmak üzere üç farklı yaklaşımı kullanarak bilgi işlemsel düşünme becerisinin daha güvenilir bir şekilde ölçülebileceğini vurgulamıştır. Bu çalışmada ürün tabanlı görüşmeler kullanılarak öğrencilerin bilgi işlemsel düşünme pratikleri ölçülmeye çalışılmıştır.

Çalışmadaki öğrenciler bilgi işlemsel pratikler aşamasının dört ana uygulama alanına göre değerlendirilmiştir. Her uygulama alanına ait 4 alt başlık mevcuttur. Bu başlıklar doğrultusunda öğrencilerle yapılan görüşme düşük, normal, yüksek olmak üzere puanlandırılmıştır (Assessing Development Of Computational Practices,2019).

Çalışma sonuçları SPSS ortamına girilip değerlendirildikten sonra elde edilen sonuç incelendiğinde iki grubun puan ortalamaları arasında çok büyük bir farkın olmadığı

görülmüştür. İki grup arasında bilgi işlemsel düşünme pratikleri açısından anlamlı bir farklılık olmadığı sonucuna ulaşılmıştır.

İki grubun ortalamaları incelendiğinde 1. grubun deneme-yanılma, test etme-hata ayıklama ve yeniden kullanma ve karıştırmada 2. gruba göre yüksek puana sahip olduğu fakat özetleme ve modüler hale getirmede 2. grubun ortalamasının daha yüksek olduğu görülmüştür. Bunun nedeni Scratch’da öğrencilerin deneme-yanılma ve test etme-hata ayıklama sürecinde ekranda görmüş olduğu eksiklikleri ve yanlışları kodların sıralamasını değiştirerek ya da kod ekleyip çıkararak giderebilmesi ve düzeltebilmesi olarak gösterilebilir. Fakat 2. grupta yaşanan eksiklikler ve aksaklıklar sadece kodların düzenlenmesi, eklenmesi ya da çıkarılması işlemleriyle düzelmemektedir. Şarjı azalan, bilgisayar bağlantısı kopan ya da zayıflayan robot, yanlış kapıya bağlanan sensör gibi donanımsal hatalar olabilmektedir. Bunlar da 2. grubun kodlama kısmında deneme-yanılma ve test etme-hata ayıklama yapması dışında donanımsal sorunlar ve çözümleriyle uğraşmasını gerektirmiş olabilir.

Birinci gruptakilerin 2. gruptakilere göre kullanabileceği kod miktarı daha fazladır. 2. gruptakiler sadece robot sekmesindeki mBot altındaki kodları ve mBlock programının izin verdiği birkaç kod bloğunu kullanabilirken 1. gruptakiler robot sekmesi dışındaki tüm sekmelerde yer alan kodları kullanabilmektedir. 2. gruptakiler robotun sahip olduğu özellikler ve sensörler dışındaki kodları kullanamadıkları için kullanabilecekleri kod miktarı 1. gruptakilere göre oldukça azdır. O yüzden yeniden kullanma ve karıştırma kısmında 2. grup 1. gruptakilere göre daha sınırlı ve kısır bir alanda çalışmıştır. Fakat 1. gruptakiler yaratıcılıklarını ve problem çözme becerilerini daha çok kullanmışlardır. Bu da yeniden kullanma ve karıştırma kısmında daha başarılı olmalarının nedeni olabilir.

Özetleme ve modüler hale getirmede 2. grup 1. gruptan daha yüksek bir ortalamaya sahiptir. Bunun nedeni olarak 1. gruptaki bazı öğrencilerin çalışmanın tasarım kısmına odaklanarak çalışmadan uzaklaşmış olması olabilir. Örneğin “Yangın Var” adlı etkinlikte 1. gruptaki öğrenciler sarı, kırmızı, turuncu renkler arasında geçiş yapan bir dikdörtgenin içinde karakterleri dikdörtgenin kenarlarına değmeyecek şekilde dikdörtgenin içinde hareket ettireceklerdi. Fakat bazı öğrenciler bu çalışmada

dikdörtgenin boyutlarını, renklerini, kalınlığını ayarlamakla veya karakter tasarlamakla uğraşmışlardır. Tasarım süreçlerini uzatarak kodlama kısmında verilen görev dışında farklı bir çalışma yapamamışlardır. Sonuç olarak projelerini geliştirememişlerdir. Aynı etkinlikte 2. gruptaki öğrenciler kutularla çevrili dikdörtgen bir alanda robotları kutulara değdirmeyecek şekilde alanın içinde rastgele hareket ettireceklerdi. Çalışmada öğrenciler sürekli ileri gidecek ama engel gördüğünde sağa ya da sola dönecek şekilde robotlarını kodlamışlardır. Kalan zamanlarında ise engel gördüğünde kırmızı led yakacak, engel görmediğinde yeşil led yakacak ya da engel gördüğünde notaları kullanarak uyarı verecek şekilde robotlarını programlayabilmişlerdir. Bu da 2. grubun özetleme ve modüler hale getirmede daha yüksek bir ortalamaoya sahip olmasının nedeni olabilir.

Elde edilen bulgular çağımızın temel becerilerinden biri olan bilgi işlemsel düşünme becerisine Scratch ile programlamanın veya mBlock ile robot programlamanın etkileri arasında anlamlı bir fark olmadığını göstermiştir. Verilerden yola çıkarak öğrencilerin bilgi işlemsel becerisini geliştirmek için her iki programlama öğretimi yöntemi de tercih edilebilir diyebiliriz.

5.3 Öneriler

Bilgi işlemsel düşünme becerisini programlama öğretimiyle kazandırmak isteyen eğitimciler bu çalışmadan yola çıkarak şunları gözönünde bulundurmalıdır:

- Robot ile programlama öğretiminin maliyeti daha fazladır. Eğer bütçe yeterli değilse Scratch ile programlama öğretimi seçilebilir.
- Bilgi işlemsel düşünme becerisi kazanması hedeflenen öğrenci grubunda dijital görsel tasarımın mı yoksa temel elektronik bilgisinin mi daha fazla bilgi yükü oluşturacağı araştırılarak daha az bilgi yükü oluşturacak yöntem tercih edilebilir.
- TBBT ile 1. sınav ve 2.sınav sonuçlarının gruplararası karşılaştırılması sonucundan yola çıkarak programlama öğretimi sürecinde akademik başarıyı göz önünde bulundurarak her iki yönteme de yer verecek olanlar için istedikleri yöntemi tercih ederek süreci başlatmaları ama bu iki yöntemden

sadece birini tercih edecek olanlara ise TBBT ve 1. sınav arasındaki gruplararası puan durumuna göre Scratch eğitimi vermesi tavsiye edilebilir.

- Scratch ve robotik gruplarındaki 1. sınav ile 2. sınav arasındaki not ortalaması artışlarına bakılarak her iki öğrenme aracını da kullanacak bir eğitime önce robotik daha sonra Scratch eğitimini vermesi önerilebilir.
- Her iki öğrenme aracından birini tercih edecek olan öğretmenlere araştırmamızın sonucuna göre iki grubun bilgi işlemsel düşünme pratikleri ve akademik başarıları arasında anlamlı bir fark olmadığından eğitimi hangi öğrenme aracında daha iyi vereceklerine inanıyorlarsa o yöntemi seçmeleri önerilebilir. Eğer çalışılan kurum ekonomik olarak robotlarla çalışmak için uygun değilse Scratch tercih edilebilir.
- Farklı özelliklerdeki robotların ve görsel programlama araçlarının kullanımı araştırma sonuçlarımızdan daha farklı sonuçlar verebilir. Lego, Ozobot, Vex, Dash ve Dot, Primo ve Robo Mind gibi robotlar ile Kodulab, Alice, Blockly, App Inventor gibi görsel yazılım araçları bunlara örnek olarak verilebilir. Çalışma farklı öğrenme araçları (farklı robotlar, farklı görsel programlama araçları) kullanılarak tekrar edilebilir.
- 21. yüzyıl becerileri öğrencilere etkili bir yaşam boyu öğrenim süreci ve başarılı bir mesleki yaşam imkânı sunması adına son derece önemlidir. Bu yüzden çalışma seçilecek öğrenme araçlarında problem çözme, yaratıcılık, eleştirel düşünme, işbirliği gibi farklı 21. yüzyıl becerilerini değerlendirecek şekilde tekrar edilebilir.

KAYNAKÇA

- Akpınar, Y. ve Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*. 13(1),1-4.
- Ala-Mutka, K. (2004). Problems in learning and teaching programming – a literature study for developing visualizations in the Codewitz-Minerva project. *Institute of Software Systems, Tampere University of Technology*.
- Allan, W., Coulter, B., Denner, J., Erickson, J., Lee, I., Malyn-Smith, J. ve Martin, F. (2010). Computational thinking for youth. *White Paper for the ITEST Small Working Group on Computational Thinking (CT)*.
- Assessing Development Of Computational Practices (2019). 13 Mart 2018 tarihinde http://scratched.gse.harvard.edu/ct/files/Student_Assessment_Rubric.pdf sayfasından erişilmiştir.
- Aşkar, P. ve Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8(1), 26-32.
- Ilias, B. ve Evripidis, H.(2016, Mart). Changing students' attitude towards stem by educational robotics & digital games programming. *V. New Perspectives in Scienze Education'nda sunulan bildiri*. Libreria Universitaria. Florence. Italy.
- Barr, D., Harrison, J. ve Conery, L. (2011). Computational thinking: a dijital age skill for everyone. 13 Mart 2018 tarihinde <http://files.eric.ed.gov/fulltext/EJ918910.pdf> sayfasından erişilmiştir.
- Barr, V. ve Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1). 48- 54
- Bethune, J. (2016). Programming makes its way into Japanese schools. 13 Mart 2018 tarihinde <https://metropolisjapan.com/classroom-coding/> sayfasından erişilmiştir.
- Brennan, K. ve Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (ss. 1-25).
- Bundy, A. (2007). Computational thinking is pervasive. 13 Mart 2018 tarihinde <http://www.inf.ed.ac.uk/publications/online/1245.pdf> sayfasından erişilmiştir.
- Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4). 9–14.
- Clark. R. (1989). Reconsidering research on learning from media. *Review of Educational Research*, 53(4), 445–459.

- Code.org.(2013, Aralık 8). *President Obama asks America to learn computer science.* [Video dosyası]. 13 Mart 2018 tarihinde <https://www.youtube.com/watch?v=6XvmhE1J9PY> sayfasından erişilmiştir.
- Colleen M. Lewis.(2010, Mart) How programming environment shapes perception. learning and goals: logo vs. scratch. *Proceedings of the 41st ACM technical symposium on Computer science education.* Milwaukee, Wisconsin, USA. [doi>10.1145/1734263.1734383]
- Cuny, J., Snyder, L. ve Wing, J. M. (2010). *Demystifying computational thinking for non-computer scientists.* 13 Mart 2018 tarihinde <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> sayfasından erişilmiştir.
- Curzon, P. (2015). Computational thinking: Searching to speak. 13 Mart 2018 tarihinde <http://teachinglondoncomputing.org/free-workshops/computational-thinking-searching-to-speak/> sayfasından erişilmiştir.
- Çatlak, Ş., Tekdal, M. ve Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3),13-25.
- Demirkan, T. (2016). *Bir 'internet cumhuriyeti' Estonya.* 13 Mart 2018 tarihinde http://www.bbc.com/turkce/haberler/2016/05/160504_estonya_internet sayfasından erişilmiştir.
- Dg Connect.(2014). Coding - the 21st century skill. *Digital Skills and Jobs Coalition.* 13 Mart 2018 tarihinde <https://goo.gl/9jjeE9> sayfasından erişilmiştir.
- Eguchi, A. (2010). What is educational robotics? Theories behind it and practical implementation. *Proceedings of Society for Information Technology & Teacher Education International Conference 2010*, 4006-4014. Chesapeake, VA: AACE.
- European Commission (Eylül, 2016). *Coding the Computational thinking on the curriculum.* Avrupa komisyonu web sayfasından erişildi: https://ec.europa.eu/education/sites/education/files/2016-pla-coding-computational-thinking_en.pdf
- Fesakis, G. ve Serafeim, K. (2009). Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ict in education. *In proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE-2009)*, 2, 258-262. Newyork. NY. USA
- Feurzeg, W. (2006). Educational technology at BBN. *IEEE Annals of the History of Computing*, 28, 18-31.
- Frankel, R. M. ve Devers, K. J. (2000). Study design in qualitative research. *Education for health: Change in learning and practice*, 13 (2), 251-261.
- Fraenkel, J. R., Wallen, N. E., ve Hyun, H. H.(2012). How to design and evaluate research in education.

- Flanagan, S. (2015). Introduce programming in a fun. Creative way. *Teach Digital Citizenship and Literacy*. 13 Mart 2018 tarihinde <http://goo.gl/NBWzVG> sayfasından erişilmiştir.
- Cevahir, H. ve Özdemir, M. (2017, Mayıs). Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu'nda sunulan bildiri*. İnönü Üniversitesi. Malatya. Türkiye.
- Hiltunen, T. (2016). Learning and Teaching Programming Skills in Finnish Primary Schools – The Potential of Games. Yüksek Lisans Tezi. *University of Oulu Department of Information Processing Science*. Finlandiya.
- How Do I Assess The Development Of Ct (2019). 13 Mart 2018 tarihinde <http://scratched.gse.harvard.edu/ct/assessing.html> sayfasından erişilmiştir.
- International Society for Technology in Education (ISTE). (2015). CT Leadership toolkit. 13 Mart 2018 tarihinde <http://www.iste.org/docs/ct-documents/ctleadershiptoolkit.pdf?sfvrsn=4> sayfasından erişilmiştir.
- International Society for Technology in Education (ISTE).(2016). Standards for Students. 13 Mart 2018 tarihinde <http://www.iste.org/standards/standards/for-students-2016> sayfasından erişilmiştir.
- International Society for Technology in Education (ISTE) ve Computer Science Teachers Association (CSTA). (2011). Computational thinking in k–12 education leadership toolkit. 13 Mart 2018 tarihinde <http://www.iste.org/docs/ct-documents/ct-leadershiptoolkit.pdf?sfvrsn=4> sayfasından erişilmiştir.
- Moreno-León, J. ve Robles, G. (2015). Analyze your scratch projects with dr. scratch and assess your computational thinking skills". *Scratch Conference*. ss. 12-15.
- Jenkins, T. (2002). On the difficulty of learning to program. *In 3rd annual Conference of LTSN-ICS*. Loughborough University. Leicestershire. UK.
- Gal-Ezer, J. ve Stephenson, C.(2014). A tale of two countries: successes and challenges in k-12 computer science education in Israel and the United States. *ACM Transactions on Computing Education (TOCE)*. 14(2), 1-18. [doi>10.1145/2602483]
- Kalkınma Bakanlığı. (2015). 9. Strateji ve eylem planı. *2015-2018 Bilgi Toplumu Stratejisi ve Eylem Planı*. Ankara: Kalkınma Bakanlığı. 13 Mart 2018 tarihinde <http://goo.gl/nbPWAZ> adresinden alındı
- Karabak, D. ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3),21.
- Kelleher, C. ve Pausch, R. (2005). Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.* 37(2),83– 137
- Kert, S. B. ve Uğraş, T.(2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. *1st International Congress of Educational Research*.

- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia-Social and Behavioral Sciences*. 174, 227-232.
- Korkmaz, Ö. ve 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.
- Küçük, S. ve Şişman, B.(2016). Birebir robotik öğretiminde öğreticilerin deneyimleri. *İlköğretim Online*, 16(1), 1305-3515. 13 Mart 2018 tarihinde <http://ilkogretim-online.org.tr/index.php/io/article/view/1131> sayfasından erişilmiştir.
- Lee , I., Martin, F., Denner, J., Coulter, B., Allan, W. ve Erickson, J. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2, 32–37.
- Lewis, C. M. (2010). How programming environment shapes perception. learning and goals: logo vs. scratch. In *Proceedings of the 41st ACM technical symposium on computer science education*, 41, 346-350. ACM.
- Lye, S. Y. ve Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- Maloney, J., Resnick, M., Rusk. N., Silverman. B. ve Eastmond, E. (2010). The scratch programming language and environment. *ACM Trans. Comput. Educ.*, 10(4), 16-15. DOI = 10.1145/1868358.1868363.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. ve Settle, A. (2014). Computational thinking in k-9 education *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference* (ss. 1-29). New York. USA: ACM.
- Mayer, R. E. (1976).Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order. *Journal of Educational Psychology*, 68 (2), 143-150.
- Milli Eğitim Bakanlığı (MEB). (2017). Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. Sınıflar). 13 Mart 2018 tarihinde https://can.meb.gov.tr/meb_iys_dosyalar/2017_08/24134656_BiliYim_Teknolojileri_ve_YazYIYm_Dersi_YYretim_ProgramY.pdf sayfasından erişilmiştir.
- Meerbaum-Salant, O., Armoni, M. ve Ben-Ari, M. (2010). Learning computer science concepts with scratch. In *Proceedings of the Sixth International Workshop on Computing Education Research (ICER '10)*. ss. 69-76. New York. USA.
- Mejia, Z. (2017). *Here's how Barack Obama just surprised hundreds of kids who are learning to code*. 13 Mart 2018 tarihinde <http://www.cnbc.com> sayfasından erişilmiştir.
- Merchant, Z., Goetz, E. T., Cifuentes, L., Keeney-Kennicutt, W. ve Davis, T. J. (2014). Effectiveness of virtual reality-based instruction on students' learning outcomes in K-12 and higher education: A metaanalysis. *Computers & Education*, 70, 29-40.

- Moreno, J. A. (2014). Automatic detection of bad programming habits in scratch: A preliminary study. (IEEE) *Frontiers in Education Conference*. 2014 IEEE.
- National Academies of Science. (2010). Report of a workshop on the scope and nature of computational thinking. *Washington DC: National Academies Press*.
- National curriculum in England: computing programmes of study. (2013). *National curriculum in England: computing programmes of study*. 13 Mart 2018 tarihinde [https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study#key-stage-1](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study#key-stage-1) sayfasından erişilmiştir.
- Numanoğlu, M. ve Keser, H. (2017). Programlama öğretiminde robot kullanımı-mbot örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*; Bartın. 6(2), 497-515.
- Özçınar, H. (2017). Hesaplamalı Düşünme Araştırmalarının Bibliyometrik Analizi. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(2), 149–171. 13 Mart 2018 tarihinde <https://doi.org/10.17943/etku.288610> sayfasından erişilmiştir.
- Özden, M. Y. (2015). Computational thinking. 13 Mart 2018 tarihinde <http://myozden.blogspot.com.tr/2015/06/computational-thinking-bilgisayarca.html> sayfasından erişilmiştir.
- Özdiñç, F. ve Altun, A. (2014). Bilişim teknolojileri öğretmeni adaylarının programlama sürecini etkileyen faktörler. *İlköğretim Online*, 13(4), 1534-1541.
- Özyurt, Ö. ve Özyurt, H. (2015). Bilgisayar programcılığı öğrencilerinin programlamaya karşı tutum ve programlama öz-yeterliklerinin belirlenmesine yönelik bir çalışma. *Eğitimde Kuram ve Uygulama Dergisi*, 11(1), 51-67.
- Papert, S. (1993). *The Children's Machine*. New York: Basic Books.
- Pfeiffer, J. (1962). *The thinking machine*. Philadelphia, PA: Lippincott.
- Piaget, J. (1955). Les stades du developpement intellectuel de l'enfant et de l'adolescent. In P. Osterrieth et al. *Le probleme des stades en psychologie de Penfant*. Paris: Presses Universitaires de France, 1955. ss. 3342.
- Resnick, M., Kafai, Y. ve Maeda, J. (2003). A networked media-rich programming environment to enhance technological fluency at after-school centers in economically-disadvantaged communities. *Proposal to the National Science Foundation* (project funded 2003–2007).
- Resnick, M. ve Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the Interaction Design and Children conference*. Boulder, CO. 13 Mart 2018 tarihinde <http://ilk.media.mit.edu/papers/IDC-2005.pdf> sayfasından erişilmiştir.
- Robertson, A. (2012). *Estonia to make coding part of first-grade education*. 13 Mart 2018 tarihinde <https://www.theverge.com/2012/9/7/3300354/estonia-progetiiger-coding-pilot-program> sayfasından erişilmiştir.

- Robins, A., Rountree, J. ve Rountree, N. (2003). Learning and teaching programming: a review and discussion. *Comput. Sci. Educ.* 13 (2), 137–172
- Rogers, C. B., Wendell, K. ve Foster, J. (2010). The academic bookshelf: A review of the NAE report. "engineering in k-12 education." *Journal of Engineering Education*, 99(2), 179-181. 13 Mart 2018 tarihinde http://findarticles.com/p/articles/mi_qa3886/is_201004/ai_n53931016/ sayfasından erişilmiştir.
- Settle, A. ve Perkovic, L. (2010). Computational thinking across the curriculum: A conceptual framework. *Technical Reports College of Computing and Digital Media Technical Report*.
- Smith, D. C., Cypher, A. ve Tesler, L. (2000). Novice programming comes of age. *Communications of the ACM*, 43(3), 75–81.
- Sünbül, A., M. (2014). *Öğretim ilke ve yöntemleri*. Konya: Eğitim Yayınevi.
- Türk Dil Kurumu (TDK). (2017. 03 20). *Türk Dil Kurumu*. 13 Mart 2018 tarihinde <http://www.tdk.gov.tr> sayfasından erişilmiştir.
- The Partnership for 21st Century Skill. (2009). *Curriculum and instruction: A 21st century skills implementation guide*. 13 Mart 2018 tarihinde <http://goo.gl/Ona2jN> sayfasından erişilmiştir.
- Tolunay, A. ve Akyol, A. (2006). Kalkınma ve kırsal kalkınma: Temel kavramlar ve tanımlar. *Süleyman Demirel Üniversitesi Orman Fakültesi Dergisi*, 2, 116-127.
- Turkle, S. (2005). The second self – computers and the human spirit. *MIT Press*. 20th Anniversary Edition.
- UK Digital Skills Taskforce. (2014). *Digital skills for tomorrow's world*. 13 Mart 2018 tarihinde <https://policy.bcs.org/sites/policy.bcs.org/files/Interim%20report.pdf> sayfasından erişilmiştir.
- Van-Roy, P. ve Haridi, S. (2004). Concepts, techniques and models of computer programming. London. England: *MIT Press*.
- Weinberg, A. E. (2013). *Computational Thinking: An investigation of the existinig scholarship and research*. Colorado State University, School of Education, Colorado. 11 Temmuz 2018 tarihinde <http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-Existing-Scholarship-Research-Dissertation.pdf> sayfasından erişilmiştir.
- Wilson, A., Hainey, T. ve Connolly, T. M. (2013). Using Scratch with primary school children: an evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning*. 3(1). 93-109.
- Wing, J. M. (2006). Computational thinking. *Communication of ACM*, 49,33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Phil. Trans. R. Soc. A*. 366: 3717–3725 doi:10.1098/rsta.2008.0118

Yolcu, V. ve Demirer, V. (2010). A Review on the Studies about the use of robotic technologies in education. *SDU International Journal of Educational Studies*, 4 (2), 127-139. 13 Mart 2018 tarihinde <http://dergipark.gov.tr/sduijes/issue/32846/340897> sayfasından erişilmiştir.



EKLER

EK 1. ETİK KURUL



T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
SOSYAL VE BEŞERİ BİLİMLER ETİK KURUL KARARLARI

KARAR TARİHİ	TOPLANTI SAYISI	KARAR SAYISI
21.03.2018	3	2018 / 73

KARAR NO: 2018 - 73
Üniversitemiz Eğitim Bilimleri Enstitüsü yüksek lisans öğrencisi Elif ŞİMŞEK'in Dr. Öğr. Üyesi Polat ŞENDURUR danışmanlığında "Programlama Öğretiminde Robotik ve Scratch Uygulamalarının Öğrencilerin Bilgi İşlemsel Düşünme Becerilerine ve Akademik Başarıya Etkisinin Karşılaştırılması ve İncelenmesi" konulu yüksek lisans tezine ilişkin gözlem ve akademik başarı testi çalışmaları okunarak görüşüldü.


Üniversitemiz Eğitim Bilimleri Enstitüsü yüksek lisans öğrencisi Elif ŞİMŞEK'in Dr. Öğr. Üyesi Polat ŞENDURUR danışmanlığında "Programlama Öğretiminde Robotik ve Scratch Uygulamalarının Öğrencilerin Bilgi İşlemsel Düşünme Becerilerine ve Akademik Başarıya Etkisinin Karşılaştırılması ve İncelenmesi" konulu yüksek lisans tezine ilişkin gözlem ve akademik başarı testi çalışmalarının kabulüne oybirliği ile karar verildi.

ASLI GİBİDİR.

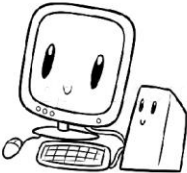
EK 2. TBBT Sınav Soruları

TEKKEKÖY İMAM HATİP ORTAOKULU
Bilişim Teknolojileri Ve Yazılım Dersi
5. Sınıflar II. Dönem I. Yazılı Sınav Soruları

Öğrencinin Adı/Soyadı:
Öğrencinin Numarası/Sınıfı:

<p>Aşağıdaki şifrelerden hangisi en güvenilirdir?(2 Puan)</p> <p>a) 327654 b) SamanYolu c) SamanYolu55 d) Saman_Yolu_55</p>	<p>Aşağıdaki eşleştirmelerden hangisi yanlıştır? (2 Puan)</p> <p>a) org-Kar amacı gütmeyen kuruluş siteleri b) edu-Hükümet siteleri c) mil-Askeri siteler d) com-Ticari kuruluşların siteleri</p>																				
<p>Günlük yaşamdaki mektubun elektronik ortamdaki karşılığı aşağıdakilerden hangisidir? (2 Puan)</p> <p>a)E-Devlet b) E-Posta c) İnternet d) E-Kitap</p>	<p>Bilgisayarda <u>internete</u> girmemizi sağlayan kart aşağıdakilerden hangisidir ? (2 Puan)</p> <p>a)Ses Kartı b)Ana Kart c)Ethernet Kartı d)Ekran Kartı</p>																				
<p>Bir bilgisayardan, Flash bellek veya internetten yayılıp bilgisayarlara zarar veren programlara ne denir? (2 Puan)</p> <p>a)Ücretli yazılım b) Güvenlik Duvarı c) Virüs d)Demo Yazılım</p>	<p>Verilen masaüstü işletim sistemlerini boşluklara yazınız. (Windows - MacOS - Pardus) (6 Puan)</p> <p>a. Apple şirketine ait işletim sistemidir. b. Microsoft tarafından piyasaya sürülmüş olup en çok kullanılan işletim sistemidir. c. Türklerin yaptığı işletim sistemidir.</p>																				
<p>Aşağıdaki yargılardan doğru olanların başına D; yanlış olanların başına Y koyunuz. (12 Puan)</p> <p>a) (.....) Excel elektronik tablolar ve hesaplama programıdır. b) (.....) Excelde satır ve sütunun birleştiği yere hücre denir. c) (.....) Excelde formüller ? ile başlar. d) (.....) Excelde birden fazla çalışma sayfasıyla çalışabiliriz. e) (.....) Excel hücreleri genişletilemez, hücreye sığacak boyutta yazılar yazmalıyız. f) (.....) B1*C1 formülündeki * işareti bölme yapacağını gösterir.</p>	<p>Aşağıdaki tabloya bakarak kelimelerin bulunduğu hücre isimlerini yazınız. (10 Puan)</p> <table border="1"><tr><td></td><td>C</td><td>D</td><td>E</td></tr><tr><td>4</td><td>lila</td><td></td><td>5 ergun</td></tr><tr><td>5</td><td>elif</td><td>kırmızı</td><td>zeynep</td></tr><tr><td>6</td><td>ahmet</td><td></td><td>7 gülden</td></tr><tr><td>7</td><td>canan</td><td>13</td><td>mor</td></tr></table> <p>a) Ahmet kelimesinin bulunduğu hücrenin adı b) 13 sayısının bulunduğu hücrenin adı c) Kırmızı kelimesinin bulunduğu hücrenin adı d) Mor kelimesinin bulunduğu hücrenin adı e) 5 sayısının bulunduğu hücrenin adı</p>		C	D	E	4	lila		5 ergun	5	elif	kırmızı	zeynep	6	ahmet		7 gülden	7	canan	13	mor
	C	D	E																		
4	lila		5 ergun																		
5	elif	kırmızı	zeynep																		
6	ahmet		7 gülden																		
7	canan	13	mor																		
<p>Karışık olarak verilen kapasite ölçü birimlerini küçükten büyüğe doğru sıralayınız. (10 Puan)</p> <p>1 KB Bvte Bit 1GB 1MB</p> <p></p> <p><input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p>	<p>Verilen mobil (tablet ve telefonlarda kullanılan) işletim sistemlerini boşluklara yazınız. (Android - iOS - Windows Phone) (6 Puan)</p> <p>a) Bugün birçok tablet, akıllı telefon ve hatta kol saatlerinde bulunan Google'a ait işletim sistemidir. b) Microsoft'un akıllı telefonlar ve tabletler için ürettiği işletim sistemidir. c) Apple'in kendi ürettiği tablet ve telefonlar için çıkardığı işletim sistemidir.</p>																				

TEKKEKÖY İMAM HATİP ORTAOKULU
Bilişim Teknolojileri Ve Yazılım Dersi
5. Sınıflar II. Dönem I. Yazılı Sınav Soruları

<p>Aşağıda karışık halde verilmiş yazılım ve donanım adlarını ait oldukları bölümdaki boş kutulara yazalım. (24 Puan)</p> <p>Fare, İnternet Explorer, Paint Programı, Yazıcı, Windows 8, Mikrofon, Ram, Pardus, Ekran, PowerPoint, Windows Media Player, Harddisk</p> <table border="1"><thead><tr><th>Donanım</th><th>Yazılım</th></tr></thead><tbody><tr><td> </td><td> </td></tr></tbody></table>	Donanım	Yazılım			<p>Aşağıdakilerden hangisi bilgisayarın tanımı değildir? (2 Puan)</p> <p>a) Kendisine verdiğimiz bilgileri istediğimizde saklayabilen, istediğimizde geri verebilen cihazdır. b) Bilgisayar aritmetiksel mantıksal işlemler yapabilen, bilgi depolayan, hatasız ve hızlı işlem yapabilen bir makinedir. c) İnsanoğlunun gereklerine uygun yardımcı alet ve araçların yapılması ya da üretilmesi için gerekli bilgi ve yetenektir. d) Çok hızlı işlem yapma özelliğine sahip, büyük bilgileri çok küçük alanlarda saklayabilen ve istendiğinde bu bilgilere çok kısa zamanda ulaşabilen elektronik bir cihazdır.</p> <p>Hangisi bir sosyal medya aracı değildir? (2 Puan)</p> <p>a) Google+ b) Televizyon c) Tweeter d) Facebook</p>
Donanım	Yazılım				
<p>Microsoft Office Excel'de (Elektronik hesaplama tablosu) hücre kavramı neyi ifade eder? (2 Puan)</p> <p>a) Sayfa ile sütun kesişimine hücre denir. b) Satır ile sayfa kesişimine hücre denir. c) Satır ve sütunun kesişimine hücre denir. d) Satırların başlangıcına hücre denir.</p>	<p>Bilgisayar kullanımı ile ilgili aşağıdaki ifadelerden doğru olanların başına "D" yanlış olanların başına "Y" koyunuz. (10 Puan)</p> <p>() İnternette iyi ve anlaşılır bir dil kullanmalıyız. () Bilmediğimiz siteleri ziyaret edebiliriz. () İnternette tanımadığımız kişilerle sohbet edebiliriz. () İnternette özel bilgilerimizi hiç korkmadan paylaşabiliriz. () Başkaları adına e-posta adresi almak yanlış bir davranıştır.</p>				
<p>Aşağıdakilerden hangisi bir tarayıcı değildir? (2 Puan)</p> <p>a) Facebook b) Mozilla c) Google Chrome d) İnternet Explorer</p>	<p>Microsoft Office Word programı temel olarak ne amaçla kullanılmaktadır? (2 Puan)</p> <p>a) Kelime işlemci programıdır. Belge ve dokümanlar hazırlanır b) Resim hazırlanır. c) Hesap tabloları oluşturulur. d) Sürücüler üzerinde kopyalama ve taşıma işlemleri yapılır.</p>				
	<p>Elif ŞİMŞEK Bilişim Teknolojileri ve Yazılım Öğretmeni Sınavda Herkese Başarılar Dilerim.</p>				

EK 3. Scratch Sınav Soruları

Öğrencinin Adı/Soyadı:

Sınıfı/Numarası:

```

tıklandığında
5 kez tekrarla
Merhaba! diye konuş
3 saniye bekle

```

1) Yukarıdaki kod bloğuna tıklandığında ekranda kaç kez "Merhabal" yazısı görünür?
a)2 b)3 c)4 d)5

```

tıklandığında
5 kez tekrarla
Merhaba! diye konuş
3 saniye bekle

```

2) Yukarıdaki kod bloğuna tıklandığında ekranda kaç saniye aralıkla "Merhabal" yazısı görünür?
a)2 b)3 c)4 d)5

```

tıklandığında
eğer rengine değişiyor mu? ise
1 saniyede x: 5 y: -110 konumuna git

```

3) Yukarıdaki kod bloğuna göre aşağıdakilerden hangisi yanlıştır?
a) Blok tıklandığında çalışmaktadır.
b) Kod bloğu sürekli çalışmaktadır
c) Eğer beyaz rengi algıladıysa x:5 y:-110 konumuna gider.
d) İstenen konuma gitmesi 1 saniyede gerçekleşir.

```

boşluk tuşu basıldığında
90° yönüne dönün
10 adım gidin

```

4) Yukarıdaki kod bloğunun açıklaması aşağıdakilerden hangisidir?
a) Klavyede sağ yön tuşuna basıldığında sağa doğru dön ve 10 adım git.
b) Klavyede boşluk tuşuna basıldığında sağa doğru dön ve 10 adım git.
c) Klavyede aşağı yön tuşuna basıldığında aşağı doğru dön ve 10 adım git.
d) Klavyede yukarı yön tuşuna basıldığında sağa dön ve 10 adım git.

```

ARI yaklaşıyor mu? olana kadar bekleyin

```

5) Yukarıdaki kod bloğunun işlevi aşağıdakilerden hangisidir?
a) Karakterin oyunun başlangıcında sahnedeki konumunu belirler.
b) Arı karakterine yaklaşıncaya kadar bekler.
c) 1 ile 10 arasında rasgele bir sayı seçer.
d) Sahnedeki kalem izlerini temizler.

```

tıklandığında
eğer sayı1 = sayı2 ise
2 saniye boyunca Tebrikler diye konuş
bütün blokları durdur
öyle değilse
2 saniye boyunca Yanlış cevap diye konuş
Gong sesini çal

```

6) Yukarıdaki bloğa göre aşağıdakilerden hangisi yanlıştır?
a) sayı1 ve sayı2 değişkenleri birbirine eşitse bütün blokların çalışması durur.
b) sayı1 ve sayı2 değişkenleri birbirine eşit değilse bütün blokların çalışması durur
c) sayı1 ve sayı2 değişkenleri birbirine eşit değilse ekranda "Yanlış cevap" yazısı görünür.
d) sayı1 ve sayı2 değişkenleri birbirine eşit değilse Gong sesi çalar.
7) "Eğer kedi ağaca yaklaşıyorsa miyav desin, yaklaşıyorsa beni yürüt desin." Bu ifadeyi anlatan komut bloğunda aşağıdaki komutlardan hangisi kesin vardır?

```

Sürekli

```

a)

```

3 saniyede x: 0 y: 0 konumuna gidin

```

b)

```

Eğer ise
Başka

```

c)

d)

8)

```

tıklandığında
Söyle: 1 ile 65 arası rastgele seçin süre: 2 saniye

```

Yukarıdaki Scratch uygulaması hangi işlemi gerçekleştirmektedir?

a) Karakterimiz 2 saniye süreyle 1 der
b) Karakterimiz 1 saniye süreyle 2 der
c) Karakterimiz 2 saniye süreyle 1 ile 65 arasında rastgele bir sayı söyler
d) Karakterimiz 1 saniye süreyle 2 ile 65 arasında rastgele bir sayı söyler

9) "Yeşil bayrak tıklandığında başlangıç konumu olarak belirlenmiş x: -203 y: -101 konumuna gidip ordan 8 sn de x: 227 y: -107 konumuna gider."

Yukarıdaki açıklama hangi kod bloğuna aittir?

```

tıklandığında
x: -203 y: -101 konumuna gidin
8 saniyede x: 227 y: -107 konumuna gidin

```

a)

```

tıklandığında
x: 153 y: 201 konumuna gidin
8 saniyede x: 227 y: -107 konumuna gidin

```

b)

```

tıklandığında
6 saniyede x: -203 y: -101 konumuna gidin
x: 227 y: 107 konumuna gidin

```

c)

```

tıklandığında
8 saniyede x: 227 y: 101 konumuna gidin
x: -203 y: 107 konumuna gidin

```

d)

10) Scratch programında karakterin girilen saniyede girilen x ve y koordinatına gitmesini sağlayan blok aşağıdakilerden hangisidir?

```

1 saniyede x: 200 y: 120 konumuna git

```

a)

```

x: 200 y: 120 konumuna git

```

b)

```

y'yi 10 değiştir

```

c)

```

x'i 10 değiştir

```

d)

11,12 ve 13. Soruları aşağıdaki şekle göre cevaplandırınız.



11) Sahnede kaç karakter vardır?

- a)2 b)3 c)4 d)5

12) Kaç numaralı simge programın çalışmasını durdurur?

- a)1 b)2 c)3 d)4

13) Kaç numaralı simge programı çalıştırır?

- a)1 b)2 c)3 d)4

14) Kedimiz başlangıç konumuna gider. 1 saniye beklemeden sonra kelebeğe doğru koşar ve "Seni yakaladım diyet" bağırır. Bu ifadeye uyan komut bloğu aşağıdakilerden hangisidir?

```

tıklandığında
x: 0 y: 0 konumuna gidin
kelebek: 'e gidin
Söyle: Seni yakaladım! süre: 2 saniye

```

a)

```

tıklandığında
1 saniye bekleyin
x: 0 y: 0 konumuna gidin
kelebek: 'e gidin
Söyle: Seni yakaladım! süre: 2 saniye

```

b)

```

tıklandığında
x: 0 y: 0 konumuna gidin
1 saniye bekleyin
kelebek: 'e gidin
Söyle: Seni yakaladım! süre: 2 saniye

```

c)

```

tıklandığında
kelebek: 'e gidin
Söyle: Seni yakaladım! süre: 2 saniye
1 saniye bekleyin
x: 0 y: 0 konumuna gidin

```

d)

Aşağıdaki 15,16,17,18. soruları yandaki tabloya göre cevaplandırınız. Size göre doğru olan kutucukların numaralarını soruların altındaki kutucuğa yazınız.

10 adım git	bu blokları durdur	miyav sesini çal
1	2	3
Gizleyin	miyav sesini bitene kadar çal	tıklandığında
4	5	6
x'i 0 yapın	x: 0 y: 0 konumuna git	Sürekli
7	8	9

15)Yandaki tabloya göre kaç numaralı kutucuktakiler ses kodlarındandır?

16) Yeşil bayrağa tıklandığında sürekli 10 adım gitmesini sağlayan kodlar hangi kutucuktaki kodlardır?

17) Yukarıdaki tabloya göre kaç numaralı kutucuklar programı başlatan ve bitiren kodlardır?

18) Yeşil bayrak tıklandığında karakteri gizleyen kodlar kaç numaralı kutucuktaki kodlardır?

Not Baremi: 15,17 ve 18. Sorular 8'er puan 16. Soru 6 puandır diğer sorular ise 5'er puandır.

EK 4. mBlock Sınav Soruları

Öğrencinin Adı/Soyadı:

Sınıfı/Numarası:

```

tiklandığında
5 defa tekrarla
  ileri git hızı 100 yap
  3 saniye bekle
  ileri git hızı 0 yap
  
```

- 1) Yukarıdaki kod bloğunda yeşil bayrağa tıkladığında robot kaç kez durup hareket eder ?
a)2 b)3 c)4 d)5

```

tiklandığında
5 defa tekrarla
  ileri git hızı 100 yap
  3 saniye bekle
  ileri git hızı 0 yap
  
```

- 2) Yukarıdaki kod bloğuna tıkladığında robot kaç saniye aralıkla hareket edip durur ?
a)2 b)3 c)4 d)5

```

tiklandığında
eğer Kapa1 mesafe algılayıcı değeri < 5 ise
  ileri git hızı 100 yap
  
```

- 3) Yukarıdaki kod bloğuna göre aşağıdakilerden hangisi yanlıştır?
a) Blok tıkladığında çalışmaktadır.
b) Kod bloğu sürekli çalışmaktadır.
c) Eğer engelin mesafesi 5'den küçük ise ilerler.
d) Motor hızı 100'dür.

```

boşluk tuşu basılınca
sağa dön hızı 100 yap
ileri git hızı 100 yap
  
```

- 4) Yukarıdaki kod bloğunun açıklaması aşağıdakilerden hangisidir?
a) Klavyede sağ yön tuşuna basıldığında robot sağa doğru döner ve geri gider.
b) Klavyede boşluk tuşuna basıldığında robot sağa doğru döner ve ileri gider.
c) Klavyede sağ yön tuşuna basıldığında robot sağa doğru döner ve ileri gider.
d) Klavyede boşluk tuşuna basıldığında robot sola doğru döner ve ileri gider.

```

tiklandığında
Kapa3 mesafe algılayıcı değeri < 5 olana kadar bekle
  
```

- 5) Komutunun işlevi aşağıdakilerden hangisidir?
a) Robotun engelle arasındaki mesafeyi belirler.
b) Engelle robot arasındaki mesafe 5 olana kadar bekler.
c) Engelle robot arasındaki mesafe 5'den küçük olana kadar bekler.
d) Robot engelle arasındaki mesafe 5 olunca bekler.

```

tiklandığında
eğer Kapa3 mesafe algılayıcı değeri < 5 ise
  kart ledler hepsi kor 255 yeş 0 mav 0
  hepsi durdur
değilse
  ileri git hızı 100 yap
  kart ledler hepsi kor 0 yeş 255 mav 0
  
```

- 6) Yukarıdaki bloğa göre aşağıdakilerden hangisi yanlıştır?
a) Mesafe 5'den küçükse hepsi durur.
b) Mesafe 5'den küçükse kırmızı yanar.
c) Mesafe 5'den büyükse ileri gider kırmızı yanar.
d) Mesafe 5'den küçükse ileri gider kırmızı yanar.
7) "Eğer robot engelle yaklaşıyorsa C4 notası çalsın yaklaşıyorsa F4 notası çalsın." Bu ifadeyi anlatan komut bloğunda aşağıdaki komutlardan hangisi kesin vardır?

```

Eğer ise
  Sürekli
  Başka
  
```

- a)  b) 
c)  d) 

```

tiklandığında
kart ledler hepsi kor 0 ile 255 arasında bir sayı (tut) yeş 0 mav 0
  
```

- 8) Yukarıdaki mBlock uygulaması hangi işlemi gerçekleştirmektedir ?
a) Kırmızı, yeşil ve mavi ışık yanar.
b) Yeşil ışığın farklı tonlarından biri yanar.
c) Kırmızı ışık yanmaz.
d) Kırmızı ışığın farklı tonlarından biri yanar.

- 9) "Yeşil bayrak tıkladığında motor hızı 100 olarak bir saniye ileri gider, motor hızı 100 olarak bir saniye sağa döner ve robot durur."

Yukarıdaki açıklama hangi kod bloğuna aittir?

```

tiklandığında
ileri git hızı 100 yap
1 saniye bekle
sağa dön hızı 100 yap
1 saniye bekle
ileri git hızı 0 yap
  
```

- a)  b) 

```

tiklandığında
ileri git hızı 100 yap
1 saniye bekle
sola dön hızı 100 yap
1 saniye bekle
  
```

- c)  d) 

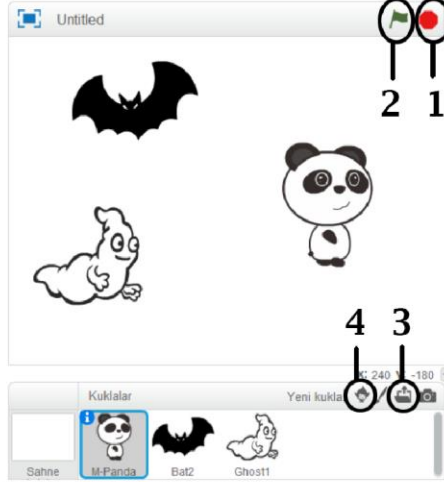
- 10) mBlock programında robotun rastgele bir hızla ileri gitmesini sağlayan kod aşağıdakilerden hangisidir?

```

ileri git hızı 0 yap
  
```

- a)  b)  c)  d) 

11,12 ve 13. Soruları aşağıdaki şekle göre cevaplandırınız.



- 11) Sahnede kaç karakter vardır?
a)2 b)3 c)4 d)5
- 12) Kaç numaralı simge programın çalışmasını durdurur?
a)1 b)2 c)3 d)4
- 13) Kaç numaralı simge programı çalıştırır?
a)1 b)2 c)3 d)4
- 14) Robotumuz yeşil bayrağa tıkladığında ileri gider. 1 saniye bekler. Engelle arasındaki mesafe 5'den büyükse yeşil ışık değilse kırmızı ışık yanar.

Bu ifadeye uyan komut bloğu aşağıdakilerden hangisidir?

- a)
- b)
- c)
- d)

Aşağıdaki 15,16,17,18. soruları aşağıdaki tabloya göre cevaplandırınız. Size göre doğru olan kutucukların numaralarını soruların altındaki kutucuğa yazınız.

1	2	3
4	5	6
7		
8		
9		

15)Yukarıdaki tabloya göre kaç numaralı kutucuktakiler ses ve renk kodlarındandır?

16)Robotun sürekli C4 notasını çaldırmasını sağlayan kodlar hangi kutucuktaki kodlardır?

17) Yukarıdaki tabloya göre kaç numaralı kutucuklar programı başlatan ve bitiren kodlardır?

18) Hangileri işlem bloğunda yer almaktadır?

Not Baremi: 16,17 ve 18. Sorular 6 şar puan 15. Soru 12 puandır diğer sorular ise 5'er puandır.