



T.C.

İNÖNÜ ÜNİVERSİTESİ

EĞİTİM BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI

**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI**

BLOK TABANLI PROGRAMLAMA ÖĞRETİMİNİN ORTAOKUL  
ÖĞRENCİLERİNİN PROBLEM ÇÖZME BECERİSİNE VE BİLGİSAYARA  
YÖNELİK TUTUMUNA ETKİSİ

YÜKSEK LİSANS TEZİ

**Yunus ALP**

**Malatya-2019**

T.C.  
İNÖNÜ ÜNİVERSİTESİ  
EĞİTİM BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI**

BLOK TABANLI PROGRAMLAMA ÖĞRETİMİNİN ORTAOKUL  
ÖĞRENCİLERİNİN PROBLEM ÇÖZME BECERİSİNE VE BİLGİSAYARA  
YÖNELİK TUTUMUNA ETKİSİ

YÜKSEK LİSANS TEZİ

**Yunus ALP**

**Danışman: Prof. Dr. Olgun Adem KAYA**




**İkinci Danışman: Prof. Dr. Mehmet ÜSTÜNER**

**Malatya-2019**

**T.C.**  
**İnönü Üniversitesi**  
**Eğitim Bilimleri Enstitüsü**  
**Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı**  
**Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı**

Yunus ALP tarafından hazırlanan **Blok Tabanlı Programlama Öğretiminin Ortaokul Öğrencilerinin Problem Çözme Becerisine ve Bilgisayara Yönelik Tutumuna Etkisi** başlıklı bu çalışma, 27.06.2019 tarihinde yapılan sınav sonucunda başarılı bulunarak jürimiz tarafından **Yüksek Lisans** tezi olarak kabul edilmiştir.

İmza

Başkan	: Doç. Dr. Oğuzhan ÖZDEMİR	
Üye (Danışman)	: Prof. Dr. Olgun Adem KAYA	
Üye	: Dr. Öğr. Üyesi Uğur BAŞBOĞAOĞLU	

O N A Y

... / ... / 2019

Doç. Dr. Niyazi ÖZER  
Enstitü Müdürü

## ONUR SÖZÜ

Prof. Dr. Olgun Adem KAYA ve Prof. Dr. Mehmet ÜSTÜNER'in danışmanlıklarında yüksek lisans tezi olarak hazırladığım **Blok Tabanlı Programlama Öğretiminin Ortaokul Öğrencilerinin Problem Çözme Becerisine ve Bilgisayara Yönelik Tutumuna Etkisi** başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün yapıtların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

27/06/2019

Yunus ALP

## ÖN SÖZ

On yıldır ortaokul, lise ve önlisans öğrencilerine yönelik farklı programlama derslerini yürütmekteyim. Öğrencilerime sürekli programlamayı sevdirmek için çabalamak zorunda kaldım. Özellikle küçük yaş gruplarında daha zor olan bu durum, beni programlamada böyle bir çalışma yapmaya yönlendirdi.

Bu çalışmada, blok tabanlı programlamanın amacı, önemi ve avantajları üzerinde durulmuş ve ortaokul altıncı sınıfında okuyan iki farklı grubun, Scratch kullanılarak blok tabanlı programlama öğretiminin deneysel bir örneği uygulanıp, problem çözme becerisi ve bilgisayara yönelik tutumları hakkında elde edilen bulgular açıklanmıştır.

Çalışmam süresince, güler yüzüyle hep yanımda olan ve çalışmama zamanını ayıran danışmanım Prof. Dr. Adem Olgun KAYA'ya, verdiği bilgi ve yönlendirmelerinden faydalandığım değerli hocam Prof. Dr. Mehmet ÜSTÜNER'e, fikirlerini ve desteğini esirgemeyen kıymetli hocam Dr. Öğr. Üyesi ATA PESEN'e ve araştırmanın yapıldığı Kayseri ilinde görev yapan Bilişim Teknolojileri Öğretmeni değerli dostum Yusuf AKYOL'a yardımlarından dolayı teşekkürlerimi sunarım.

Bana her konuda desteklerini ve güvenlerini hiçbir zaman esirgemeyen eşime, anneme, babama, kardeşlerime ve minik gülüşüyle sevgisini her zaman hissettiren kızıma sevgilerimi gönderiyorum.

Son olarak yüksek lisans esnasında dersini aldığım, bilgi ve deneyimlerinden yararlandığım hocalarıma da çok teşekkür ediyorum.

2019, Malatya

Yunus ALP

## ÖZET

### BLOK TABANLI PROGRAMLAMA ÖĞRETİMİNİN ORTAOKUL ÖĞRENCİLERİNİN PROBLEM ÇÖZME BECERİSİNE VE BİLGİSAYARA YÖNELİK TUTUMUNA ETKİSİ

ALP, Yunus

Yüksek Lisans, İnönü Üniversitesi Eğitim Bilimleri Enstitüsü  
Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

Tez Danışmanı: Prof. Dr. Olgun Adem KAYA  
Haziran-2019, XI+108 sayfa

Günümüzde hızlı bir şekilde bilim ve teknolojik gelişmelerin yaşanmasıyla birlikte, bilgisayarın geniş bir kullanım ağının olmasına bağlı olarak, okullardaki kullanımının da sürekli olarak artması, akademik ve sosyal hayatımızdaki yerini gün geçtikçe önemli hale getirmektedir. Gelişen dünyamızda problem çözme becerisinin gelişmesini sağlayacak şekilde çağdaş, günün teknolojisini kullanmada olumlu tutuma sahip olan bireyler yetiştiren, aynı zamanda da uygulama alanı oluşturmayı hedefleyen eğitim sistemlerine ihtiyaç duyulmaktadır. Bu çalışma, blok tabanlı programlama öğretiminin, ortaokul altıncı sınıf öğrencilerinin problem çözme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisinin belirlenmesi amacıyla, ön test-son test kontrol gruplu yarı deneysel desen kullanılarak gerçekleştirilmiştir. Bu amaçla, programlama öğretim aracı olarak Scratch yazılımı kullanılmıştır. Uygulama, 2017-2018 eğitim öğretim yılında Kayseri ilindeki bir ortaokulunda, bilişim teknolojileri ve yazılım dersi alan 6. sınıf öğrencileriyle gerçekleştirilmiştir. Veriler, bilgisayara yönelik tutum ölçeği ve çocuklar için problem çözme envanteri içeren anket formu ile toplanmıştır. Verilerin istatistiksel analizlerinde SPSS v21 programı kullanılmıştır. Elde edilen verilerin analiz sonuçlarına göre; her iki gruptaki öğrencilerin bilgisayara yönelik tutumları ile problem çözme becerileri, uygulama öncesine göre anlamlı derecede, pozitif olarak değişim göstermiştir. Bu sonuca göre; bilişim teknolojileri ve yazılım dersi kapsamında gerçekleştirilen uygulamaların, öğrencilerin bilgisayara yönelik tutumlarını ve problem çözme becerilerini geliştirdiği anlaşılmaktadır. Bunun yanında elde edilen diğer sonuca göre; uygulama sonrasında deney ve kontrol grubu öğrencileri arasında bilgisayara yönelik tutum açısından bir farklılaşma olmazken, problem çözme becerileri açısından deney grubu lehine pozitif bir farklılaşma söz konusudur. Bu sonuca göre; Scratch ile blok tabanlı programlama öğretiminin, öğrencilerin problem çözme becerilerini geliştirmek için etkili bir araç olarak kullanılabileceği anlaşılmaktadır. Öğrencilerin problem

özme becerileri ile bilgi teknolojilerine yönelik tutumlarını geliřtiren farklı yöntemler ve daha geniş gruplarla yeni arařtırmalar yapılması önerilebilir.

**Anahtar Sözcükler:** Bilgisayara yönelik tutum, Problem özme becerileri, Programlama öğretilmi, Scratch, Blok tabanlı programlama



## ABSTRACT

### THE EFFECTS OF BLOCK-BASED PROGRAMMING ON THE PROBLEM SOLVING ABILITY AND ATTITUDE TOWARDS COMPUTER FOR SECONDARY SCHOOL STUDENTS

ALP, Yunus

Master's Thesis, İnönü University Institute of Computer Sciences Education and Instructional Technologies of Science in Education

Advisor: Prof. Dr. Olgun Adem KAYA

June, 2019, XI+108 pages

Today, with rapid development of technology and science, wide usage area of computers and constant growth of computer use make computers important in our academic and social life. Modern educational systems, which bring up individuals who have positive attitude to use of modern technology and have problem solving ability, are needed in our developing world. The aim of the study is to explore the effect of block-based programming on 6th grade student's attitudes towards computer and problem-solving ability. In this research, pretest-posttest nonequivalent control group quasi-experimental design was used. Research group has consisted of 49 6<sup>th</sup> grade students of secondary school in Kayseri province in 2017-2018 education year. The control group consisted of 25 students and experimental group had 24 students. Data have been collected by the attitudes towards computer scale and problem-solving inventory for children. SSPS v21 program has been used to statistical analyses of data. As it shows normal distribution, Independent Samples t-test was used to examine whether the difference between the two means for student's attitudes towards computer scale and problem-solving inventory is statistically significant, after the study. As a result, it has been determined that; students' computer attitude became positive and problem-solving ability developed after the study for both groups. Besides; there is an eloquent difference in students' problem-solving ability of the experimental group in comparison with the control group. The results of the study show that; Scratch can be used effectively to develop students' problem-solving ability as a block-based programming tool for the secondary school students. New researches should be carried out with different methods and bigger samples to improve the students' problem solving abilities and attitudes towards information technologies.

**Key Words:** Computer Attitudes, Problem Solving Ability, Scratch, Teaching programming, Block based programming



## İÇİNDEKİLER

ONUR SÖZÜ.....	ii
ÖN SÖZ.....	iii
ÖZET.....	iv
ABSTRACT.....	vi
İÇİNDEKİLER.....	vii
TABLolar LİSTESİ.....	ix
ŞEKİLLER LİSTESİ.....	xi
BÖLÜM I.....	
GİRİŞ.....	1
1.1. Problem Durumu.....	10
1.2. Araştırmanın Amacı.....	12
1.3. Önem.....	12
1.4. Varsayımlar.....	15
1.5. Sınırlılıklar.....	15
1.6. Tanımlar.....	15
BÖLÜM II.....	
KURAMSAL BİLGİLER ve İLGİLİ ARAŞTIRMALAR.....	17
2.1. Programlama.....	17
2.1.1. Çocuklar için Programlama Öğretimi.....	18
2.1.2. Türkiye’de Programlama Öğretimi.....	20
2.1.3. Çocuklar için Programlama Öğretiminde Kullanılan Araçlar.....	22
2.1.4. Blok Tabanlı Programlama Eğitimi.....	25
2.2. Scratch.....	27
2.3. Problem Çözme ve Problem Çözme Becerisi.....	29
2.3.1. Çocuklarda Problem Çözme.....	30
2.3.2. Problem Çözme ve Programlama.....	30
2.4. Bilgisayara Yönelik Tutum.....	34
2.5. İlgili Araştırmalar.....	36
2.5.1. Türkiye’de Yapılan Araştırmalar.....	36
2.5.2. Yurt Dışında Yapılan Araştırmalar.....	41
BÖLÜM III.....	
YÖNTEM.....	49

3.1. Araştırmanın Modeli .....	49
3.2. Çalışma Grubu.....	50
3.3. Veri Toplama Araçları, Güvenilirlik ve Geçerlik Çalışmaları .....	51
3.3.1. Bilgisayara Yönelik Tutum Ölçeği.....	51
3.3.2. Çocuklar için Problem Çözme Envanteri .....	52
3.4. Verilerin Analizi .....	52
3.5. Uygulama .....	53
BÖLÜM IV .....	
BULGULAR VE YORUM .....	56
4.1. Birinci Alt Probleme İlişkin Bulgular ve Yorumlar .....	56
4.2. İkinci Alt Probleme İlişkin Bulgular ve Yorumlar.....	60
4.3. Üçüncü Alt Probleme İlişkin Bulgular ve Yorumlar.....	64
BÖLÜM V .....	
SONUÇ VE ÖNERİLER.....	69
KAYNAKLAR .....	75
EKLER.....	90
Ek-1. Çocuklar için Problem Çözme Envanteri Kullanım İzni.....	90
Ek-2. Çocuklar için Problem Çözme Envanteri .....	91
Ek-3. Bilgisayara Yönelik Tutum Ölçeği.....	92
Ek-4. Kontrol Grubu Ders İçeriği .....	93
Ek-5. Deney Grubu Ders İçeriği .....	94
Ek-6. Scratch Etkinlikleri .....	97
Ek-7. Bilişim Teknolojileri Sınıfı .....	107
Ek-8. Çalışma Fotoğrafı .....	108

## TABLolar LİSTESİ

<b>Tablo 1.</b> Araştırma Modeli .....	50
<b>Tablo 2.</b> Katılımcıların Demografik Özellikleri.....	51
<b>Tablo 3.</b> Uygulama Süreci.....	55
<b>Tablo 4.</b> Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Betimsel Analiz Sonuçları .....	56
<b>Tablo 5.</b> Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Betimsel Analiz Sonuçları .....	57
<b>Tablo 6.</b> Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Shapiro-Wilk Testi Sonuçları.....	58
<b>Tablo 7.</b> Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Shapiro-Wilk Testi Sonuçları.....	58
<b>Tablo 8.</b> Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Mann-Whitney U Testi Sonuçları .....	59
<b>Tablo 9.</b> Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Mann-Whitney U Testi Sonuçları .....	59
<b>Tablo 10.</b> Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Betimsel Analiz Sonuçları .....	61
<b>Tablo 11.</b> Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Betimsel Analiz Sonuçları .....	62
<b>Tablo 12.</b> Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Shapiro-Wilk Testi Sonuçları .....	62
<b>Tablo 13.</b> Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Shapiro-Wilk Testi Sonuçları .....	63
<b>Tablo 14.</b> Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Bağımsız Örneklem t-testi Sonuçları.....	63
<b>Tablo 15.</b> Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Bağımsız Örneklem t-testi Sonuçları.....	64

<b>Tablo 16.</b> Deney Grubu ve Kontrol Grubunun Kendi İçinde, Problem Çözme Becerileri ve Bilgisayara Yönelik Tutumları Puanları Arasındaki Farkın Verilerinin Shapiro-Wilk Testi Sonuçları .....	65
<b>Tablo 17.</b> Kontrol Grubu İçin Problem Çözme Envanteri Öntest ve Sontest Uygulamasının Wilcoxon İşaretli Sıralar Testi Sonuçları .....	66
<b>Tablo 18.</b> Deney grubu Çocuklar için Problem Çözme Envanteri Öntest ve Sontest Uygulamasının Bağımlı Örneklem t-testi Sonuçları.....	66
<b>Tablo 19.</b> Kontrol Grubu Bilgisayara Yönelik Tutum Ölçeği Öntest ve Sontest Uygulamasının Bağımlı Örneklem t-testi Sonuçları.....	67
<b>Tablo 20.</b> Deney Grubu Bilgisayara Yönelik Tutum Ölçeği Öntest ve Sontest Uygulamasının Bağımlı Örneklem t-testi Sonuçları.....	68

## ŞEKİLLER LİSTESİ

Şekil 1. Bilişim Teknolojileri ve Yazılım Dersinin Tarihi .....	20
Şekil 2. Scratch Ekranı .....	28
Şekil 3. Problem çözme ve programlama süreci bileşenleri.....	33



# BÖLÜM I

## GİRİŞ

Bu bölümde, araştırmanın amacı, problem durumu, önemi, problem cümlesi ile alt problemler, sayılılar, sınırlılıklar ve tanımlar bulunmaktadır.

Tutum, bir kişinin bir topluluğa olaya ya da bireylere yönelik bireysel etkinliklerdeki seçimlerine yön veren içsel bir olgudur (Senemoğlu, 2001). Tutum, doğrudan gözlenemeyen bir olgu olup bireyin sevgi, nefret ve davranışlarında önemli rol oynamaktadır. Anderson ve Gerbing'in (1988) aktarmasına göre Allport (1935) tutumu; "kişinin ilgili durum ya da nesnelere yönelik tepkisi üzerinde yönlendirici ya da dinamik bir etkileme gücüne sahip, yaşantı ve deneyimler sonucu şekillenmiş zihinsel ya da duygusal hazırlık durumu" olarak tanımlamaktadır.

Bilgisayara yönelik tutum; öğrenmede bilgisayarları kullanarak yeni teknolojiler desteğiyle öğrenmeye ilişkin eğilimleri göstermektedir (Mathew Myers ve Halpin, 2002). Eklemek gerekirse bilgisayara karşı tutum bir öğrenme sürecinde etkili bilgisayar kullanımıyla orantılı bir ilişki göstermektedir (Yıldırım, 2012).

Problem, bir kimsenin istenilen hedefe ulaşmasında karşısına çıkan engeller olarak tanımlanmaktadır (Bingham, 2004). Hedefe ulaşmada çıkan problemleri çözmek amacıyla insanlar kişisel deneyimlerine, geleneklere veya konuyla ilgili otorite sahibi olan kişilere başvurarak problemleri aşmak için çözüm aramaktadırlar (Karasar, 2005). Problem çözüme ise insanın gayeye ulaşma sürecinde karşılaştığı probleme yanıt aramada geçirdiği düşünme ve problemi aşma süreci olarak tanımlanabilir (Ülküer, 1988). Bu sayede birey, engelleri azaltarak veya içinde bulunduğu şartlara uyarak dengeye ulaşmanın yollarını aramaktadır. Problem çözüme, öğretmen ve öğrencilerin yanısıra herkesin yararına olduğu için eğitim-öğretim de en önemli şekilde düşünülmektedir. Ortaya çıkan yeni problemleri çözüme kavuşturmak ve uygulamak, çocuklara mutlu yaşamının esası olan özgüveni kazanabilmeleri için fırsat verme anlamında çok önemlidir (Forgan, 2003). Problem çözüme becerisi sadece akademik alanda değil hayatımızın tamamında ihtiyaç duyduğumuz bir yetenektir. Çocuklar, problem çözümü için verilen fırsatlar sayesinde yeteneklerinin farkına varır ve geliştirir. Problem karşısında kendi çözüm yollarını araması için izin verilen çocuklar, kendi deneyimlerini ve bilgilerini kullanarak iç ve dış kaynaklardan faydalanabilmektedir ve

problem çözümü için kendi becerilerini kullanma fırsatı yakalayabilmektedir (Bingham, 2004).

Bilgisayar destekli eğitim, öğrencilerin dikkatlerini toplayarak konuyu anlamak için pek çok özelliği içinde barındırır. Bilgisayarlar içinde animasyonlar, renkli şekiller, grafikler ve ses öğeleri gibi birçok özellik barındırmasını, öğrencilerin derslere farklı açıdan bakabilme yeteneği kazandırmasını, öğrencinin dersten keyif almasını ve tek bir yöntemle dersi anlamakta zorluk yaşayan öğrencilere farklı alternatiflerin bir arada sunulmasını sağlar. (Baki ve Öztekin, 2003). Yurt içi ve yurt dışı kaynaklı, öğrencilerin bilgisayara yönelik tutumları ve problem çözme becerisinin araştırıldığı birçok çalışma yapılmıştır.

Bilgisayar dersine yönelik tutumlarının incelendiği bir araştırmada, cinsiyete göre ve evlerinde bilgisayar olan ve olmayanlar arasında, bilgisayara yönelik tutumlarında farklılaşma olduğu saptanmıştır (Erkan, 2004). Bilgisayarlı öğrenme yöntemini kullanarak öğrencinin matematik dersindeki başarısının etkisini araştıran bir diğer çalışma sonucunda, bilgisayar kullanarak ders işleyen öğrencilerin diğer gruptaki öğrencilere göre matematik dersindeki başarısının daha yüksek olduğu görülmüştür (Shin, Park ve Bae, 2013). Dalton (1986) 7. sınıfta öğrenim göre 97 ortaokul öğrencisi ile, öğretmen gözetiminde problem çözme becerisi geliştirme etkinlikleri grubu, LOGO programı kullanan grup ve geleneksel öğretim gören grup olmak üzere 3 gruplu deneysel bir çalışma yapmıştır. Çalışmanın sonuçlarına göre problem çözme becerisi geliştirme etkinlikleri grubundaki öğrenciler diğerlerine göre daha yüksek problem çözme becerisi puanları elde etmiştir. Ayrıca her iki uygulama grubu, yani problem çözme becerisi geliştirme etkinlikleri grubu ile LOGO programı kullanan gruptaki öğrencilerin matematiğe yönelik tutumlarında geleneksel gruptaki öğrencilere göre anlamlı bir artış olduğu gözlemlenmiştir.

Bilgisayar günümüzde insanların sosyal hayatını, iş hayatını ve boş zamanlarını pek çok yönden etkilemektedir. Bunun yanında, bilgisayar becerilerinin ve deneyiminin yüksek olması mesleki ve kişisel başarıda önemli derecede katkı sağlamaktadır (Hammond ve diğerleri, 2009; Teo, 2008). Günümüzde bilim ve teknolojik gelişmelerin yaşanmasıyla bilgisayar kullanımı giderek artmaktadır. Bu gelişmeler eğitim-öğretime yansımalarıyla bilgisayarların okullardaki kullanımını da sürekli olarak arttırmaktadır (Altun ve Bektaş, 2010; Bottino, 2004). Bilgisayarların kullanımının artmasıyla aradığımız özelliklerin başında gelen okur-yazarlık ifadesinin yerine, bilgisayardaki

temel becerilere sahip olma ifadesi kullanılmaya başlamıştır (Yiğit ve diğerleri, 2009). Bunun sonucunda, öğretim teknolojilerinin geniş bir kullanım ağının olması bilgisayarların akademik ve sosyal hayatımızdaki yerini gün geçtikçe daha da önemli hale getirmektedir. Üretilen bilginin gün geçtikçe artmasının yanı sıra sınıflardaki öğrenci sayısındaki artış öğrencilerin bu bilgilere ulaşılabilmesi için yeni teknolojik ürünlerin ihtiyacını doğurmuştur. Yeni teknolojik gelişmelerin aranmasıyla bilgisayar bulunmuş ve bilgisayarların eğitim kurumlarına girmesi zorunluluk haline gelmiştir (Aktümen ve Kaçar, 2003; OECD, 2008). Günümüzde birçok alanda olduğu gibi bilgisayarlar farklı eğitim kademelerinde kullanılmakta ve eğitim-öğretim sisteminin vazgeçilmez bir parçası olmuştur. Bundan dolayı, eğitim-öğretimde bilgisayarın ve bilgisayara dayalı öğrenmenin ülkemizdeki yeri de gün geçtikçe artış göstermektedir (MEB, 2005).

Programlama eğitimlerine daha erken yaşlarda başlanması, aynı zamanda çocukların seviyesine uygun olacak şekilde tasarlanması bu eğitimin başarılı olmasını sağlamaktadır. Bu süreçte oyun tabanlı ve etkileşimli öğrenme ortamları çocukların eğlenerek öğrenmelerine olanak sağlamaktadır. Kalelioğlu'na göre (2015), programlama eğitimine erken yaşlarda başlanması sürükle-bırak araçlarının kullanılmasıyla öğrencileri kod yazmanın zorluğundan kurtarmaktadır. Öğrencilerin bilgisayar bilimlerine olan ilgisini arttırmanın yanında üst düzey öğrenme becerilerini de geliştirmektedir.

Dünyanın birçok ülkesinde, bireylerin yaratıcı düşüncesinin gelişmesi, problem çözme yeteneklerinin yükselmesi ve daha kalıcı öğrenmenin sağlanması gibi amaçlarla bilgisayar alanındaki ders müfredatları gün geçtikçe daha küçük yaş seviyesine göre düzenlenmektedir (Kalelioğlu, 2015; Grout ve Houlden, 2014; Jones, 2013). Kert ve Uğraş'a göre (2009), programlama eğitimine erken dönemlerde başlanması, öğrencileri programlama sürecinin zorluğundan ve karmaşasından uzaklaştırıp; problemi ortaya koyarak analiz, genel değerlendirme ve yaratıcılık gibi konularda öğrencileri destekleyecek programlama araçlarının tercih edilmesini sağlamıştır.

Ülkemizde ise bilgisayar programlama eğitimleri ortalama olarak 20'li yaşlarda verilmektedir. İçinde bulunduğumuz yaşam şartları göz önüne alındığında bu yaşlarda alınan eğitimlerin verimliliği tartışılmaktadır. Çünkü ilerleyen yaşlar nedeni ile kullanılmayan hayal gücü ve yaratıcılıkta kayıplar meydana gelmektedir. Ancak, gelişmiş ülkelerde bu eğitimler çocuk yaşlarda alınmakta ve böylece 20'li yaşlarda



ortaya Google, Apple, Facebook ve Microsoft gibi büyük projeler ve ürünler ortaya çıkmaktadır.

Türkiye'deki durumu değerlendirecek olursak; 2012'den itibaren Bilişim Teknolojileri ve Yazılım (BTY) dersi kapsamında olmak üzere kodlama eğitimi 5. sınıflara verilmeye başlanmıştır. Son zamanlarda kodlama eğitimi için daha da erken yaşlarda, ilkokul 1. Sınıftan başlayarak tüm diğer sınıflarda kodlama eğitimi verilmesi tartışılmaktadır. Bu tartışma ile kodlama eğitiminin ne kadar önemli olduğu vurgulanmakta, kodlama öğretimine küçük yaşlardan başlanarak, her kademedede yer verilmesi fikri ortaya atılmıştır. Bazı ülkelerde 5 yaşından itibaren kodlama eğitimine başlandığı, eğitimin her kademesindeki tüm yaş grupları için programlama eğitimleri verilmesi planlandığı görülmektedir (Saygıner ve Tüzün, 2017). Buna paralel olarak bazı çalışmalarda da kodlama eğitiminin erken yaşta olmasının önemi vurgulanmakta, öğrencilerin çeşitli açılardan gelişimlerini destekleyerek eğitime olumlu katkılarda bulunabileceği ifade edilmektedir (Demirer ve Sak, 2008; Özçınar, Yecan ve Tanyeri, 2016).

Bazı çalışma sonuçlarına göre; kodlama eğitimi sonucunda sadece anlamlı kodlar topluluğu olarak bir programı oluşturulmaktan daha öte bir durum söz konusu olup, bunun yanı sıra öğrencilerin karşılaştıkları problemler için kendilerine özgü çözümler üretmelerine yardımcı olması durumu söz konusudur (Shin ve diğerleri, 2013; Karabak ve Güneş, 2013). Yapılan çeşitli çalışmalarda öğrencilerin kodlama eğitimi sayesinde, matematik kavramlarının yanında bilişim kavramlarını da daha rahat öğrenebildikleri (Shin ve diğerleri, 2013; Monroy-Hernandez ve Resnick, 2008) belirtilmektedir. Ayrıca, bilgi işlemsel düşünme becerisinin yanısıra (Lee vd., 2011; Grover ve Pea, 2013; Lye ve Koh, 2014; Penmetcha, 2012; Oluk ve Korkmaz, 2016; Repenning, Webb, Ioannidou, 2010; Yıldız ve Çiftçi, 2017) eleştirel düşünme, yaratıcı düşünme, sistematik düşünme, analiz yapma (Monroy-Hernandez ve Resnick, 2008) gibi becerilerinin de geliştirdiği bildirilmektedir. Baz (2018) yaptığı çalışmada, kodlama öğrenimi alan çocukların hatalarını tespit edip çözme becerisi ve analiz yeteneklerinin geliştiğini belirtmektedir. Saygıner ve Tüzün (2017) programlama eğitiminin dünya genelinde ilköğretim düzeyinde incelemişler ve 2015 yılından itibaren çoğu Avrupa ülkesinin programlama kodlama eğitimi müfredatlarına neden ekledikleri incelendiğinde, öğrencilerin problem çözme becerilerini ve mantıksal düşünme becerilerini geliştirmesi fikrinin ön plana çıktığı anlaşılmıştır.

Dünya genelinde programlama eğitimine verilen önem gün geçtikçe artmaktadır. Özellikle son yıllarda gelişmiş ülkeler kategorisinde ele alınan ülkelerin genellikle küçük yaşlardan başlamak üzere kodlama ve programlama eğitimini müfredatlarına ekledikleri görülmektedir. Bu ülkelerden biri olan İngiltere, beş yaşından itibaren çocukların kodlama öğrenmesi için 2014 yılından itibaren 5-14 yaş aralığındaki öğrenciler için kodlama öğretimine başlamıştır. Bu uygulama ile öğrencilerin algoritmalar ve temel seviyede programlama mantığını edinmeleri amaçlanmıştır (Öndeş, 2016). İngiltere'nin yanı sıra ABD, Avustralya, Tayvan ve Güney Kore de programlama öğretimini küçük yaşlardan itibaren başlatan ülkeler arasındadır (Sayın ve Seferoğlu, 2016; Demirer ve Sak, 2016). Ülkemizde programlama öğretiminin durumuna bakılacak olursa; 2012 yılında müfredata giren "Bilişim Teknolojileri ve Yazılım" dersinde, ilk olarak "*Problem çözme, Programlama ve Özgün Ürün Geliştirme*" ünitesiyle bu konu işlenmeye başlamıştır. 5. sınıftan 8. Sınıfa dek "*Bir problemi çözmek ve projeyi gerçekleştirmek için strateji geliştirebilir, çözüm üretirken farklı bakış açılarını ve yaklaşımları kullanabilir*" ve "*Yazarlık ve programlama dillerini tanıyabilir, en az bir yazarlık/ programlama dilini etkili biçimde kullanabilir*" gibi kazanımlar hedeflenmiştir (TTKB, 2012). Ülkemizde de programlama öğretimine giderek daha fazla önem verilmekle birlikte, hala programlama öğretimi sırasında kullanmak üzere yeterince kaynağa sahip olunmaması ve bu nedenle her öğretmenin kendi anlayışına göre programlama öğretimi yapması, bu konunun olması gerektiği gibi ele alınıp alınmadığına dair tartışmalara yol açmaktadır (Dinçer, 2018).

Programlama öğrenimi süreci, problem çözme ve analitik düşünme gibi beceriler gerektiren bir süreçtir. Bu sürecin sonunda ortaya anlamlı, işe yarayan, bir probleme çözüm üreten bir yazılım ürünü çıkması beklenir. Bilgi ekonomisinden söz edilen günümüzde problem çözme becerisine sahip, analitik düşünebilen, tüketmenin yanında üretebilen ve teknolojiyi kullanarak üretim yapabilen bireylere ihtiyaç duyulmaktadır. Birçok ülke bu niteliklere sahip yurttaşlara sahip olabilmek için eğitim müfredatlarında çeşitli teknoloji derslerine yer vermektedir (Öztürk ve Alper, 2019).

Bilgi toplumunda, bilginin içselleştirilip karşılaşılan problemlere uygulanmasıyla ilgili problemin çözülebilmesi ön plana çıkmaktadır. Kısaca, sahip olduğu bilgiyi yeni durumlara uygulayarak karşılaşılan problemleri çözebilme becerisine problem çözme becerisi denilebilir. Problem çözme becerisi insanın çalışma, düşünme ve hayat tarzını etkileyebilen çok temel bir faktördür. Programlama süreci

temelde, çözüm bekleyen bir problem için çözüm arayışı sürecinde yazılım dilleri aracılığıyla yürütülen soyut bir düşünme sürecidir ve bu da daha fazla soyut düşünme ve problem çözme becerisine ulaştırabilir. Yapılan pek çok araştırmaya göre; ilköğretim seviyesindeki ve hatta ortaöğretimdeki öğrencilerin, problem çözme becerisinde oldukça zayıf kaldıkları görülmüştür. Programlama öğrenimi sürecinde çoğu öğrencinin zorlandıkları kısım, problem çözme becerileri gerektiren bir süreç olmasından kaynaklanmaktadır (Gomes ve Mendes, 2007).

Öğrencilerin üniversite seviyesinde programlama öğrenirken temel mantığı edinmekte sorun yaşamamaları için programlama eğitimine daha erken yaşlarda başlanması gerekmektedir. Görsel bir programlama öğrenme aracı olan Scratch, bu amaçla geliştirilmiştir (Karabak ve Güneş, 2013). Scratch programı erken yaşlardaki çocuklardan başlayarak, gençlerde ve yeni başlayan tüm yaşlardaki kişiler için programlama öğrenimini kolaylaştırmak ve cazip hale getirmek amacıyla tasarlanmıştır. Scratch ile programlama sürecinde söz dizimi hataları olmaz. Ayrıca bu ortam sayesinde Java programlama diline giriş yapılabilmesi, belli bir aşinalık oluşması için bir yol olarak görülmektedir. Genç yaşlardaki öğrencilerin programlama öğrenimi sürecinde ilk olarak görsel bir ortam sunan Scratch tavsiye edilmektedir (Malan ve Leitner, 2007).

2012 yılında yayınlanan öğretim programında “Programlama, Özgün Ürün Geliştirme ve Problem Çözme” ünitesinde Scratch programının kullanımı uygun olabilir. Millî Eğitim Bakanlığı <http://scratch.eba.gov.tr> sitesinde bu konuda eğitim videoları yayınlamaya başlayarak Scratch kullanımını teşvik etmektedir. Öğrenciler, Scratch ortamını kullanarak kendi projelerini yapabilmekte ve bu sayede gerçek hayatta karşılaştıkları problemlere uygun çözümler üretme becerisi kazanabilmektedirler (Karabak ve Güneş, 2013; Lee, 2011).

Scratch, bir öğretim aracı olarak akademik performansı arttırmak için kullanılabilir. Scratch ile ders işlenmesi sayesinde öğrencilerin problem çözme becerilerinin geliştirilebileceği öngörülmektedir (Brown ve diğerleri, 2008). Program yazma becerisi edinebilmek için bilgi işlemsel düşünme becerisine sahip olmak gerekir. Çünkü bilgi işlemsel düşünme becerisi sayesinde, bir problem ne kadar zor olursa olsun parçalara ayrılabilir, önemli noktalar tespit edilip ve algoritmik düşünme becerisi ile her bir adımı incelenip çözülebilir (ISTE, 2018). Programlama öğretimine verilen önem giderek artmakta ve bu konudaki çalışmalara daha fazla ihtiyaç duyulmaktadır. Bu sebeple, Scratch ve Kodu Game Lab gibi görsel programlama dilleri kullanılarak

yapılan öğretimin etkili olup olmadığı, öğrencilerin hangi becerilerini ne seviyede geliştirdiğinin araştırılması gerekmektedir (Dinçer, 2018).

Programlama öğretimi için kullanılan araç programlama başarısını etkileyebilir (Brusilovsky ve diğerleri, 1997; Dinçer, 2018). Özellikle giriş seviyesindeki programlama derslerinde kullanılan dilin karmaşık olması öğrencileri olumsuz etkileyebilir. Bu derslerde öncelikle programlama dilinden bağımsız olarak, algoritma mantığı verilmelidir. Daha sonrasında ise öğrenciler, yazdıkları algoritmaları deneyerek ilgili programlama dilinde tecrübe kazanabilir (Ersoy, Madran ve Gülbahar, 2011).

Programlama öğretimindeki güçlükler düşünüldüğünde genellikle daha basit diller (mini language) tavsiye edilmiş ve kullanılmıştır. Bu diller profesyonel yazılımlar geliştirmekten ziyade, programlama öğretimini kolaylaştırmak amacıyla, daha basit komutlar ve etkileşim teknikleri kullanılarak geliştirilmiştir (Ersoy ve diğerleri, 2011).

Çok sayıda öğrenci programlama derslerini sıkıcı ve zor dersler olarak nitelendirmekte ve bu derslerde zorlanmaktadır (Bennedsen ve Caspersen, 2008; Kurland, Pea, Clemaent ve Mawby, 1989; Kinnunen ve Malmi, 2008; Robins, Rountree ve Rountree, 2003; Monroy- Hernandez ve Resnick, 2010). Programlama öğretiminde karşılaşılan zorlu ve karmaşık süreç bir sorun oluşturmakta ve öğrencilerde programlamaya ilişkin olumsuz bir tutuma yol açabilmektedir. Öğrencilerin programlamaya ilişkin sonuç olarak başarısızlığa götüren böylesi olumsuz bir algıya sahip olmalarına sebep olan faktör, temel olarak programlama öğretimi sürecinde yaşanan çeşitli problemlerdir (Schulte ve Bennedsen, 2006; Soloway, 1986; Deek ve Espinosa, 2005). Programlama öğretimi derslerinde özellikle yeni başlayanlar için amaç, öğrencilere programlamaya ilişkin temel mantığın yanı sıra önemli kavramların öğretilmesi olmalıdır (Bennedsen ve Caspersen, 2008; Winslow, 1996; Linn ve Dalbey, 1985; Özden, 2008). Bununla birlikte çoğu durumda, programlama mantığı yerine, normalde bir yazılım geliştirmek için kullanılan bir programlama dili ile doğrudan programlama öğretimi yapılmaktadır (Schulte ve Bennedsen, 2006; Deek ve Espinosa, 2005). Başlangıç düzeyinde olup programlama kavramlarına henüz aşına olmayan acemi programcılar olarak bu öğrenciler, bir yandan kullanılan programlama dilinde bulunan kurallar ve kodlarla başa çıkmaya çalışırken, bir yandan da programlamaya dair temel kavramları (döngü, dizi, fonksiyon vb.) öğrenme durumunda kalmaktadırlar. Üzerinde çalışılan programlama diline ait bütün kuralları ve kodları öğrenseler de programlama mantığını edinemedikleri için, kodları doğru kullanmakta ve programlama

sırasında yaşanan bir problemi çözmekte güçlük çekmektedirler (Cooper, Dann ve Pausch, 2003; Porter ve Calder, 2004; Garner, Haden ve Robins, 2005). Bu yüzden normal bir programlama dili kullanılarak yapılan programlama işlemine başlamadan önce temel mantığın ve temel kavramların çok iyi öğrenilmesi son derece önemlidir (Futschek, 2006).

Programlama sürecinde ele alınan problemi bölümlere ayırarak modellemek ve programlama kavramlarını kullanarak çözmek, soyut bir işlem olarak nitelendirilebilir. Bir diğer deyişle öğrencilerin çoğu bu süreci somutlaştırma aşamasında güçlük çekmektedirler (Cooper ve diğerleri, 2003; Soloway ve Spohrer, 1989; Futschek, 2006). Soyut ifadelerin, metin tabanlı ifadelerin ve teknik kavramların kullanıldığı bu süreç, öğrencilerin programlama öğrenirken zorluk yaşamalarına ve sıkılmalarına sebep olmaktadır (Ginat, 2003; Mayer, 1981; Lahtinen, Ala-Mutka ve Järvinen, 2005). Öğrencilerin programlama öğrenirken algoritma öğrenmedeki soyut süreç problemler yaşamalarına sebep olmaktadır. Algoritma öğrenme sürecini daha anlaşılır, somut ve kolay hale getirmek için, öğrencilerin algoritmayı tasarlama sürecini görselleştirmelerine yardımcı olan Scratch ya da Alice gibi ortamlar kullanılabilir (Baldwin ve Kuljis, 2000; Cooper ve diğerleri, 2003; Cooper ve diğerleri, 2006; Maloney ve diğerleri, 2010).

Öğrencilerin küçük yaşlardan itibaren edindikleri bilgileri kullanarak problemlere çözümler üretebilen bireyler olarak yetişmeleri amacıyla da kullanılabilen programlama öğretimi, ülkemizde “Bilişim Teknolojileri ve Yazılım” dersi kapsamında 5. sınıftan itibaren ele alınmaktadır. Bununla birlikte öğrenciler, programlama süreçlerinin karmaşık olması nedeniyle programlama öğrenirken zorlanmaktadırlar (Bennedsen ve Caspersen, 2008). Öğrencilerin dersin başlarında programlama sürecinde karşılaştıkları zorluklarla mücadele ettikleri, fakat sonraki aşamalarda sıkıldıkları görülmektedir (Ersoy ve diğerleri, 2011). Saygıner ve Tüzün (2017), programlama eğitimde çekilen güçlüklerin, Programlama öğretiminde kullanılan öğretim yöntemlerinin uygun olmaması, kullanılan programlama dilinin karmaşık olması, programlama yaparken yabancı dil gereksinimi gibi faktörlerden kaynaklandığını belirtmişlerdir. Bunun yanında, programlama sırasındaki soyut işlemlerin öğrenilmesinde güçlük yaşandığı, ekranda görüntülenen verilerin odaklanma problemine yol açtığı belirtilmektedir (Fields, Kafai ve Giang, 2016). Bu sorunlar ise öğrencilerin derse olan ilgisini azaltmakta, akademik başarılarının düşüşüne neden

olmaktadır (Jenkins, 2002). Programlama öğrenimine ilişkin edinilen bu olumsuz algı ise öğrencilerde konuya ilişkin olumsuz bir tutum gelişimine neden olmaktadır (Anastasiadou ve Karakos, 2011), çünkü bilindiği gibi tutumlar çoğu kez algıların sonucunda oluşmaktadır (Akande, 2009).

Programlama eğitimi sürecinde edinilen çeşitli beceri ve kazanımlar vurgulanmakla birlikte, programlama öğretiminde kullanılan yöntem, ortam, programlama dili ve hedef kitleye uygunluğu açısından bazı problemler de ortaya çıkmaktadır. Programlama eğitimi sürecinde özellikle yeni başlayanlar için ortaya çıkan en önemli sorunun, geleneksel programlama dillerinin karmaşık yapılarından dolayı zor öğrenilmeleridir. Bu problemi çözmek için, görsel olarak kodlama yapma imkânı sunarak programlama öğrenmeyi kolaylaştıran Scratch, Alice, Microsoft Small Basic, Stagecast Creator ve Toontalk benzeri programlama ortamları oluşturulmuştur. Özellikle genç yaşta çocukların yararlanabileceği bu tür ortamların yardımıyla onlar da erken yaşta edindikleri programlama becerisinin getirdiği avantajları ve yaratıcılıklarını kullanarak harika projeler ve fikirler ortaya koyabilirler.

Scratch gibi görsel programlama yapma ortamı sunan araçlar, öğrencilerin daha somut düşünerek daha özgün ürünler çıkarmalarına imkân tanımakta ve programlama sürecini daha cazip hale dönüştürmektedir (Utting ve diğerleri, 2010). Özellikle bu ortamlardaki tasarım kısmı öğrencilerin problem çözme becerileri geliştirmelerine katkı sağlamaktadır (Cooper ve diğerleri, 2003; Moreno, 2012). Scratch benzeri programlama öğretimi ortamları, öğrencilere hikâye ve oyun tasarlama imkânı sunarak programlama öğrenme sürecini anlaşılır kılıp ve basitleştirmekte ve bu sayede programlama sürecinin sonunda somut bir ürün ortaya koymalarına yardımcı olmaktadır (Utting ve diğerleri, 2010; Ramadhan, 2000). Tasarım odaklı görsel programlama öğretim ortamları söz konusu olduğunda son yıllarda giderek daha da popüler hale gelerek sıklıkla yararlanılan ortamlardan birinin de Scratch olduğu görülmektedir (Resnick ve diğerleri, 2009).

Programlama öğretimi sırasında Scratch kullanıldığında yapılanlar kaydedilmekte ve istenildiğinde tekrar açılabilir. Bu yönüyle de oldukça pratik ve yararlı programlama öğretim araçlarından biridir. Scratch, ABD’de bulunan MIT Medya Laboratuvarı ve UCLA Üniversitesi araştırmacılarının birlikte çalışarak geliştirdikleri bir programlama öğretimi dilidir. Öğrenciler ve öğretmenler çok zaman alan mevcut programlama dilleri ile programla öğretiminde çeşitli zorluklar

yaşamaktadırlar. Bu açıdan, yaştan bağımsız olarak, anlaşılması ve öğrenmesi daha kolay olan Scratch gibi eğitimsel programlama dilleri sezgisel programlamayı mümkün kılmaktadır. Ayrıca Scratch, çeşitli multimedya türlerine ve farklı dillere destek vermektedir. Bu nedenle, bir programlama dili olarak harika bir araç olup birçok faydasından söz edilebilir (Nam, Kim ve Lee, 2010).

Scratch sayesinde öğrenciler, normalde zor olan programlama kavramlarıyla daha kolay başa çıkabilmekte ve zorlanmadan programlama başlangıç yapabilmektedirler. Hata mesajları olmaması, sürükle bırak yöntemiyle anında görsel geribildirim içeren çıktılar vermesi, çalışılan projeyi paylaşma imkânı sunması, programlamanın yanında tasarım sürecini de içermesi gibi avantajlarıyla Scratch, programlama öğretiminde öğrencilere ve öğretmenlere oldukça yararlı bir araçtır (Kert ve Uğraş, 2009; Maloney, 2008). Öğrenciler Scratch sayesinde geleneksel metin tabanlı programlama sırasında karşılaştıkları güçlüklerin çoğunu yaşamamış, programlamanın çok zor olduğunu düşünmemiş, böylece programlamaya yönelik olumsuz bir tutum geliştirmemişlerdir. Scratch ile oyun programlama süreci Lego oynamak gibidir. Tıpkı Lego’da olduğu gibi Scratch’te de bloklar birleştirilerek ortaya bir ürün konmaktadır. Eğer bir amaca yönelik olarak doğru bloklar uygun bir şekilde birleştirilirse, Scratch ile basit veya kısmen profesyonel düzeyde bir program, animasyon veya oyun geliştirmek mümkündür. Scratch, günümüzde programlama eğitiminde en çok kullanılan programlama dillerinden birisidir. Scratch programının ilk 12 yıllık eğitim sürecinde kullanımı hızla artmaktadır ve Harvard Üniversitesi de dâhil bazı üniversitelerde programlamaya geçiş aşamasında kullanılmaktadır (Resnick ve diğerleri, 2009). Scratch, kullanıcının kod yazmasını gerektirmeyen sadece kod bloklarını bir araya getirerek program geliştirmesini sağlayan görsel programlama dillerinden biridir.

### **1.1. Problem Durumu**

Öğrencilerde bulunan programlamanın zor olduğuna ilişkin algı, onların programlamaya ilişkin olumsuz bir tutum edinmelerine sebep olmaktadır (Anastasiadou ve Karakos, 2011). Öğrencilerde oluşan bu olumsuz tutum ise, onların programlama başarılarını olumsuz bir şekilde etkilemektedir (Lahtinen, ve diğerleri, 2005). Bu açıdan ele alındığında Scratch, kodlama bilgisi gerektirmemesi, basit bir ara yüze sahip olması, her yaştan insanın programlama öğrenmesi için uygun olması ve programlama öğretiminde karşılaşılan olumsuz durumları ortadan kaldırması sayesinde, kullanışlı bir programlama öğretimi aracı olarak giderek daha fazla tercih edilmektedir. Scratch

kullanılarak; programlamanın karmaşık ve zor yapısından kaynaklanan olumsuz tutumlar giderilerek daha başarılı bir programlama öğretimi gerçekleştirilebilir. Bu sayede öğrenciler, temel programlama mantığını öğrenerek, bu bilgilerini yeni bir programlama dili ile çalışırken kullanabilirler (Erol, 2015). Bu nedenle Scratch gibi blok tabanlı programlama öğretiminde etkililiğinin incelenmesi gerekir. Bu öğretimin problem çözme becerisine etkisinin araştırılması ve bununla birlikte öğrencilerin bilgisayara yönelik tutum gibi duyuşsal özelliklerine etkisinin de incelenmesi gerektiği söylenebilir.

Smith, (1968) tutumu; “bireye atfedilen ve bireyin psikolojik bir obje ile ilgili düşünce, duygu ve davranışlarını düzenli bir biçimde oluşturan eğilim” şeklinde tanımlar. Petty ve Cacioppo (1996) ise tutumu; “insanların herhangi bir nesne, insan veya konulara ilişkin olumlu veya olumsuz duyguları” olarak tanımlamaktadır. Ülgen ise (1994) konuyu “tutum; öğrenmeyle kazanılan, bireyin davranışlarına yön veren, karar verme sürecinde yanlılığa neden olan bir olgudur” şeklinde ele almaktadır. Duyuşsal, davranışsal ve bilişsel olmak üzere üç boyutu olan tutuma ilişkin “bilişsel boyut, bireyin inançlarından ve tutum objeleri hakkındaki gerçeğe dayanan bilgi ve değerlendirmelerinden” oluşmaktadır (Tavşancıl, 2002). Tutumun duyuşsal boyutu ise, “bireyin önceki deneyimlerine bağlı olarak değişen, kişinin değerler sistemi ile yakından alakalı olan, ilgili tutum nesnesinden hoşlanma-hoşlanmama yönünü oluşturan bir bileşendir” (Tavşancıl, 2002). Davranışsal boyut ise, “bireyin ilgili tutum nesnesine yönelik takındığı tavır, söz veya hareketlerinden oluşur” (İnceoğlu, 2010). Öğrencilerin sahip oldukları tutumlar, ilgili konudaki akademik başarılarını etkileme gücüne sahip olması nedeniyle, genellikle çalışmalarda ele alınarak incelenen konulardandır. Bu çalışma kapsamında ele alınan programlama öğretimi de genellikle bilgisayar ortamında gerçekleştirilmesi nedeniyle, öğrencilerin sahip olduğu bilgisayara yönelik tutumlarının ele alınarak mevcut durumdaki düzeyi ile deneysel uygulamadan nasıl etkilendikleri incelenmiştir.

Ayrıca, lisans seviyesinde işlenen programlama derslerinde bile öğrencilerin zorluk yaşadıkları düşünüldüğünde, ilköğretim ve ortaöğretim seviyesindeki öğrencilere programlama öğretimi amacıyla kullanılacak olan teknik, strateji ve araçlar konusunun önemle ele alınması gerekliliği ortaya çıkmaktadır. Bu bağlamda, programlama öğretiminde blok tabanlı araçlar kullanılmasının, öğrencilerin problem çözme becerilerine nasıl bir katkıda bulunduğu araştırılmıştır.



İçinde yaşadığımız yeniçağda, öğrencilerin karşılaştığı problemlerin üstesinden gelebilmesi için problem çözme aşamalarını bilmesinin yanı sıra geniş bir bilgi ve hizmet kaynağı olan bilgisayar kullanımı becerisine ve bilgisayara yönelik olumlu bir tutuma sahip olması çok önemlidir. Bu sebeple “Blok tabanlı programlama öğretiminin kullanılmasının ortaokul altıncı sınıf öğrencilerinin bilgisayara yönelik tutumu ile problem çözme becerileri üzerinde bir etkisi var mıdır?” sorusu çalışmanın problem cümlesini oluşturmaktadır.

### **1.2. Araştırmanın Amacı**

Bu araştırmanın amacı, “Blok tabanlı programlama öğretiminin, altıncı sınıf öğrencilerinin problem çözme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisini belirlemektir.” Bu genel amaç çerçevesinde şu sorulara yanıt aranmıştır:

1. Kontrol grubu ile deney grubu öğrencilerinin, uygulama öncesinde Problem Çözme Becerileri Envanteri ve Bilgisayara Yönelik Tutum Ölçeği’nden aldıkları ön test puanları arasında istatistiksel olarak anlamlı bir farklılık var mıdır?
2. Kontrol grubu ile deney grubu öğrencilerinin, uygulama sonrasında Problem Çözme Becerileri Envanteri ve Bilgisayara Yönelik Tutum Ölçeği’nden aldıkları son test puanları arasında istatistiksel olarak anlamlı bir farklılık var mıdır?
3. Kontrol grubu ve deney grubunun kendi içinde, problem çözme becerileri ve bilgisayara yönelik tutum puanları arasındaki farkın (gelişim düzeyi) ortalamaları arasında istatistiksel olarak anlamlı bir farklılık (değişim) var mıdır?

### **1.3. Önem**

Yenilenen öğretim programlarında üst düzey düşünme becerilerinin tüm derslerin omurgasında yer alması gerektiği belirtilmiştir. Bilgisayar dersinin ilkökul ve ortaokul döneminde gerekli olduğu, diğer tüm dersler kadar öneme sahip olduğu, programlama eğitimi gibi kazanımları ile öğrencilere düşünme becerileri kazandırma potansiyeli biliniyorken bununla ilgili çalışma yapılmaması bu gibi araştırmalara ihtiyaç duyulduğunu göstermektedir.

Programlama sürecinin her bir bileşeni karmaşık bilişsel beceriler gerektirmektedir. Özellikle hata ayıklama ve doğrulama aşaması buna en çok ihtiyaç duyulan kısımdır. Bu aşamalarda bir takım üst düzey düşünme becerilerine sahip olması gerekmektedir. Burada programlama sürecine tam olarak hâkim olmak için gerekli olan

becerilerin, problem çözüme becerileriyle benzer oldukları söylenebilir. Programlama eğitimiyle de bu becerilerin kazandırılacağı ve program yazmada ilerleme kaydedilebileceği öngörülmektedir.

Günümüzdeki teknolojik gelişmeler düşünüldüğünde öğrencilerin bu gelişmelere uyum sağlamaları ve değişime ayak uydurabilmeleri için bazı standartlara sahip olmaları önemlidir (ISTE, 2018). Buna göre öncelikle öğrenciler kendilerine özel hedefler belirlemeli ve bu hedefleri gerçekleştirmek için teknolojiden de yararlanarak daha etkili öğrenme süreçleri yaşamalıdır. Ayrıca, bu çağın öğrencileri dijital dünyaya uyum sağlamalı, iyi birer dijital vatandaş olmalı, en uygun araştırma teknik ve stratejilerini kullanarak doğru bilgiye ulaşmalıdırlar. Bunun yanında, yaratıcı ve yenilikçi çözümler bularak karşılaştıkları problemleri doğru tanımlamalı ve çözüm için uygun teknolojiler kullanmalıdırlar. Daha ötesi, öğrenciler kendilerini etkili ve doğru bir şekilde ifade edebilmek için en uygun araçları kullanmalı, çok iyi birer iletişimci olarak küreselleşen dünyada iş birliği yapabilen bireyler olmalıdırlar (ISTE, 2018). Programlama öğretiminin, yukarıda bahsedilen becerilerden bazılarının kazandırılmasında kullanılabileceği anlaşılmaktadır. Örneğin, problem çözüme becerilerinin geliştirilmesinde programlama öğretiminin etkili olduğunu gösteren çok sayıda çalışma bulunmaktadır (Nam ve diğerleri, 2010; Taylor, Harlow ve Forret, 2010; Shin ve Park, 2014; Yıldırım, 2016; Yüncül, Durak, Çankaya ve Mısırlı, 2017).

Son yıllarda programlama öğretimi zorunlu olarak okutulan bir ders olarak çeşitli ülkelerin öğretim programlarına girmiştir (Balanskat ve Engelhardt, 2015; Sayın ve Seferoğlu, 2016; Demirer ve Sak, 2016). Programlama öğrenme süreci, öğrencilerin problem çözüme becerisi, yaratıcı ve algoritmik düşünme gibi beceriler kazanmalarına yardımcı olmakla birlikte kavramların soyut olması nedeniyle bu becerilerin kazanılması her zaman kolay olmamaktadır. Programlama sürecini somutlaştırmak için Scratch gibi görsel programlama dilleri kullanılması öğrencilere yardımcı olabilir (Ersoy ve diğerleri, 2011; Resnick ve diğerleri, 2009). Öğrenciler blok tabanlı programlama dillerini kullanarak programlama öğrenirken sürece aktif bir şekilde katılarak daha anlamlı ve kalıcı öğrenmeler gerçekleştirebilirler. Bu ortamlarda öğrenciler çeşitli senaryolar oluşturabilmekte ve bu sayede de sahip oldukları bilgileri örgütleyerek daha anlamlı ilişkiler kurabilirler (Erekmekçi ve Fidan, 2012).

Programlama eğitiminde yaşanan sorunlar özellikle başlangıç düzeyindeki öğrencilerin programlama mantığını anlamada ve somutlaştırmada zorlanmalarına ve

dolaylı olarak olumsuz bir tutum geliřtirmelerine sebep olmaktadır (Kinnunen ve Malmi, 2008). Yařanan bu sorunun çözümlü için bařlangıç seviyesindeki programlama öđretimi derslerinde temel programlama mantıđı, programlama kavramları ile yapıları basit ve etkili bir řekilde öđretilmelidir (Rajaravivarma, 2005; Gee, 2005). Blok tabanlı programlama dillerinin yardımıyla küçük yařtaki çocukların da eđlenceli bir řekilde yaratıcılık, algoritmik düşünme, problem çözme ve bilgi işlemsel düşünme becerileri gibi becerileri erken dönemde kazanmaları sağlanabilir (Shin ve diđerleri, 2013; Çatlak, Tekdal ve Baz, 2015). Bu nedenle Scratch kullanılarak yapılan programlama öđretimi etkinlikleri; öđrencilerin problem çözme becerilerinin geliřmesini ve bilgisayara yönelik tutumlarını olumlu bir řekilde etkileyebilir. Bu sayede, öđrenciler temel programlama mantıđını daha iyi öđrenerek daha iyi bir öđrenme süreci yařayabilirler.

Bireylere günümüz dünyasında ihtiyaç duyulan çeřitli beceriler kazandırma potansiyelinden dolayı programlama öđretimi konusu giderek daha fazla gündem olmaktadır. Bununla birlikte bu konuda nitelikli ve yeterli kaynak bulma konusunda sıkıntı yařanmaktadır. Buna paralel olarak, ülkemizde ortaokul öđrencileriyle Scratch kullanımına iliřkin çalışmaların sayısının az olduđu görölmüřtür. Ayrıca daha önce yapılan çalışmalarda programlama dili eğitimi alanında daha çok öđretim yöntemleri ile ilgilenilmiş, problem çözme ve eleřtirel düşünme becerilerine yönelik lisans ve ortaöđretim düzeyinde çalışmalar yapılmıř, fakat ilköđretim düzeyinde blok tabanlı programlamanın hem problem çözme becerisine hem de bilgisayara yönelik tutuma etkisi ile ilgili bir çalışma yapılmamıřtır. Bu nedenle, bu arařtırmanın ilgili literatüre katkıda bulunacađı öngörülmektedir. Buna ek olarak, programlama öđretiminin ilgili müfredata eklenmesinin ardından, programlama öđretimi için uygun bir programlama öđretimi aracı seřebilmeleri konusunda öđretmenlere yardımcı olması beklenmektedir.

Bu çalışma, ortaokul öđrencileri ile Scratch programlama öđretimi aracı kullanarak yapılan programlama öđretiminin;

- ✓ Öđrencilerin problem çözme becerilerine ve bilgisayara yönelik tutumlarına olan etkisi,
- ✓ İlkokul ve ortaokulda programlama öđretimi amacıyla Scratch aracının kullanılmasının etkili olabilme durumunun tespiti,

- ✓ Her seviyede programlama öğreten eğitimciler için görsel programlama imkânı sunan blok tabanlı programlama öğretimi araçlarının kullanımı konusunda farkındalık sağlama potansiyeli,
- ✓ Bilişim Teknolojileri ve Yazılım derslerinde işlenen programlama öğretiminde kullanılacak içerik konusunda katkı sağlama, açılarından önemlidir.

#### **1.4. Varsayımlar**

1. Örneklemedeki katılımcıların soruları içten ve doğru cevaplayacağı varsayılmıştır.
2. Veri toplama araçlarındaki soruların, öğrencilerin bilgisayara yönelik tutumlarını ve problem çözme becerilerini ortaya koyacak düzeyde olduğu düşünülmektedir.

#### **1.5. Sınırlılıklar**

1. Araştırma 2017-2018 eğitim-öğretim yılında, Kayseri il merkezinde yer alan bir devlet ortaokulunda öğrenim gören altıncı sınıf öğrencileri ile sınırlıdır.
2. Araştırma, öğrencilerin bilgisayara yönelik tutum ve problem çözme becerileri düzeylerinin ortaya koyulması için, ölçeklere verilen cevaplar ile sınırlandırılmıştır.
3. Veri toplama aracı olarak ölçekler dışında herhangi bir veri toplama aracından yararlanılmamıştır. Elde edilen veriler bu ölçekler ile sınırlıdır.

#### **1.6. Tanımlar**

**Tutum**, bir bireyin herhangi bir konuya ilişkin duygu, düşünce ve davranışlarını düzenli olarak biçimlendiren eğilimidir (Kağıtçıbaşı, 1998, s.84.). Tutum, bireyin yaşadığı deneyimlerin bir sonucu olarak ortaya çıkan, insan davranışlarını belli bir şekilde yönlendirme gücüne sahip, belli bir durum, kavram, nesne ya da kişiye yönelik duygusal ve düşünsel bir tavır alma biçimidir (Allport,1935).

**Bilgisayara yönelik tutum;** öğrenmede bilgisayarları kullanarak yeni teknolojilerle öğrenmeye ilişkin eğilimleri göstermektedir (Mathew Myers ve Halpin, 2002)

**Problem**, bir kimsenin istenilen hedefe ulaşmasında karşısına çıkan engeller olarak tanımlanmaktadır (Bingham, 2004).

**Problem çöme**, insanın gayeye ulaşma sürecinde karşılaştığı probleme yanıt aramada geçirdiği düşünme ve problemi aşma sürecidir (Ülküer, 1988).

**Scratch**, blok tabanlı bir programlama aracıdır. Scratch, özellikle erken yaştaki çocukların ve programlama öğrenmeye yeni başlayan diğer yaşlardaki bireylerin kolay bir şekilde programlama öğrenebilmeleri amacıyla UCLA Üniversitesi ve MIT Media Lab çalışanlarının işbirliğiyle geliştirerek ücretsiz bir şekilde kullanıma açtıkları görsel ve blok tabanlı bir programlama öğretim aracıdır. Scratch çoğu görsel programlama ortamında olduğu gibi kod yazmanın aksine kod bloklarını sürükleyip bırak mantığı çerçevesinde kullanarak program oluşturmaya yarayan bir yazılımdır (Resnick ve diğerleri, 2009).



## BÖLÜM II

### KURAMSAL BİLGİLER ve İLGİLİ ARAŞTIRMALAR

Bu bölüm, araştırmanın kuramsal bilgilerini ve konuyla ilgili yapılan önceki araştırmaları içermektedir.

#### 2.1. Programlama

Programlama dillerindeki komutların, bir problem çözümü için, bilgisayar ortamında bir araya getirilerek çalıştırılması sürecinde gerçekleşen işlemler bütününe programlama denir (Kesici ve Kocabaş, 2007). Bu süreç sonunda ortaya çıkan ürüne de program ya da yazılım denir. Başka bir açıdan programlama, bir programcının, bilgisayarın hedeflenen bir görevi tamamlayarak hedef sorunu veya sorunları çözmesi için bir takım komut setleri oluşturup uygulamaya koyduğu bir süreçtir. Bilgisayarın sorunsuz bir şekilde çalışarak belli bir görevi yerine getirmesini sağlayan bu komutlar topluluğu, bir programlama dili kullanılarak geliştirilir ve ortaya çıkan ürün ise bir bilgisayar programı olarak nitelendirilir (Balanskat ve Engelhardt, 2014).

Programlama, esasen belli bir görevin gerçekleştirilmesi için bir mekanizmanın yönlendirilmesidir. Günlük hayatta kullanılan kahve makinesi veya fırın gibi mekanizmaları ara yüzde bulunan talimatlar kullanılarak programlanabilir. Buna benzer olarak bilgisayarlar da programlanabilen cihazlardır. Bir bilgisayarın işlediği talimatlar topluluğuna bilgisayar programı veya kısaca program denir. Bu programları yazan insanlara ise programcı adı verilir. İnsanlar nasıl kendi aralarında İngilizce, Almanca gibi diller yardımıyla iletişim kuruyorsa, bilgisayarla iletişim kurarak ona belli görevleri yaptırabilmek için programcılar da çeşitli programlama dilleri kullanırlar (Parker, 2016; Zak, 2013).

Bir programlama dili, programcılarının komut ya da talimat setlerini, bir diğer deyişle program geliştirmek için kullandıkları özel bir dildir. Farklı alanlarda kullanılan yüzlerce programlama dili bulunmaktadır. Bilgisayarda programlama yapmak için ilk zamanlar makine dili kullanılmıştır. Aslında, bilgisayarın anladığı tek dil makine dilidir. Fakat makine dili maliyetli, karmaşık ve zor olduğu için makine dilini temsil etmek üzere programlama dilleri ortaya çıkmıştır. Programlama dilleri makine dilinin üzerine inşa edilmiştir (Sebesta, 2012). İlk önceleri, makine dili komutlarını temsil etmek üzere assembly dilleri kullanılmaktaydı. Daha sonra bunlar bir derleyici tarafından makine

diline dönüştürüldü. Sonraki aşamada ise COBOL ve FORTRAN gibi üst düzey programlama dilleri ortaya çıktı. Bu dillerle çalışmak, makine kodu ve assembly dillerine göre daha kolaydı. Bu diller sayesinde, programlama süreci daha kaliteli ve verimli hale geldi (Regan, 2008). 1980'lerden sonra ise bu dillerin daha gelişmiş versiyonları olan, nesneye yönelik programlama yapılabilen Visual Basic, Java, Delphi, C++, Python gibi üst düzey programlama dilleri ortaya çıkmış ve giderek daha popüler hale gelmişlerdir (Eryılmaz, 2003).

Son dönemde oldukça yaygın bir şekilde kullanılmaya başlanan Python, programlama öğrenmek için iyi bir seçimdir. Sözdizimi basittir ve dilin içine yerleştirilmiş çok güçlü bazı özelliklere sahiptir. Çok basitten son teknolojiye kadar, Nesne Yönelimli tekniklere vb. birçok programlama stilini destekler. Unix / Linux, MS Windows, Macintosh gibi çok sayıda platformda çalışır. Ayrıca çok kullanıcı dostu ve yardımsever bir kullanıcı topluluğuna sahiptir. Bunların hepsi yeni başlayanlar için çok önemli özelliklerdir. Ancak Python sadece bir başlangıç dili değildir. Programlama deneyiminiz arttıkça Python'u kendi içinde bir amaç olarak veya hızlı bir prototip dili olarak kullanmaya devam edebilirsiniz (Gauld, 2000, s.5).

### **2.1.1. Çocuklar için Programlama Öğretimi**

Programlama öğreniminin zor olmasının en başta gelen sebeplerinden birisi, programlamayı öğrenecek kişinin birden fazla tür bilgiye sahip olmasının gerekmesidir. Programlama öğrenmeye başlayan bir kişi sadece verilen problemi çözme ile uğraşmamakta, buna ek olarak programa ait sözdizimsel ve kavramsal bilgiye de sahip olmak durumunda kalmaktadırlar (Mannila, Peltomäki ve Salakoski, 2006). Bu ise programlama eğitimini bir kat daha fazla zor hale getirmektedir.

Edirisinghe (2008) ve Cooper ve diğerleri (2000) yeni öğrenenler için programlamayı zorlaştıran sebepler arasında, yazılan kodun kavranması ve analiz edilmesi için gerekli olan becerilerin yetersizliğini öne sürmektedir. Bu tür becerilerden yoksun olmak kod yazımı sırasında yapılan hataları saptamayı ve sonrasında da düzeltmeyi zor bir hale getirmektedir. Bu tür öğrenciler hatayı tam olarak belirleyemediklerinden, hatayı gidermek için bilinçli bir şekilde değil rastgele değişikliklere başvurmaktadırlar (Polo, 2013). Bunun sonucunda da hatayı düzeltmeler dahi hatanın nasıl ve niçin düzeldiğini tam olarak anlayamamakta ve ileride de bu hataları tekrarlamaktadırlar (Edirisinghe, 2008).

Programlamanın eğitimle olan bağı sadece ekonomik değildir, beraberinde bir öğrenme ve beceri edinme ortamı olarak da değerli olmasındandır. Bunun yanında, bazı programlama ortamları yapılandırmacı öğrenme ile uyumlu öğrenme ortamları olarak kullanılabilir (Papert, 1993). Programlama, aynı zamanda öğrencilerin yaratıcı, mantıksal ve sistematik düşünme ve problem çözme becerisi gibi becerileri geliştirmelerinde yardımcı olma potansiyeline sahiptir (Clements ve Gullo, 1984; Çetin, 2012; Fesakis ve Serafeim, 2009; Jenkins, 2002).

Programlamayı öğrenmek ve bu konuda uzmanlaşmak oldukça yorucu ve zorlayıcı bir süreçtir (Eckerdal, 2009; Gomes ve Mendes, 2007; Genç ve Karakuş, 2011; Hadjerrouit, 2008; Jenkins, 2002). Çünkü programlama öğrenmeye yeni başlayan öğrenciler öncelikle mantıksal yapılar içerisinde karşılaşma ihtimalleri çok yüksek olan algoritma hatalarını çözmek zorundadırlar. Programlama öğreniminde giriş seviye derslerindeki başarısızlık oranlarının % 15 ila % 30 arasında olduğu düşünülmektedir (Fesakis ve Serafeim, 2009). Programlama öğretiminde kullanılan ortamlar çoğunlukla metin tabanlı ortamlardır. Acemi programcı olan öğrenciler ise bu tarz ortamlarda program geliştirmek amacıyla ilgili programlama diline ait çok sayıda kodu öğrenmek zorundadırlar. Bu nedenle metin tabanlı sistemlerde kod yazmak, okumak, yazılanları anlamak oldukça zorlayıcı olabilmektedir (Ortiz, 1997). Programlama öğrenmeye yeni başlayan, daha önce bir programlama alt yapısı olmayan öğrencilerin karşılaştıkları sorunları azaltmak ve daha kolay öğrenmelerini sağlamak amacıyla bazı yeni yaklaşımlar bulunmaktadır. Blok tabanlı programlama öğretimi araçlarının, bu yaklaşımlar arasında olduğu bilinmektedir (Guzdial, 2004).



### 2.1.2. Türkiye’de Programlama Öğretimi

Yıl	Karar	Ders Adı	Durum	Sınav Durumu	Haftalık Ders Saati							
					1	2	3	4	5	6	7	8
1997	143	Bilgisayar	Seçmeli	Not var				1-2	1-2	1-2	1-2	1-2
2005	192	Bilgisayar	Seçmeli	Not yok	1	1	1	1	1	1	1	1
2007	111	Bilişim Teknolojileri	Seçmeli	Not yok	1	1	1	2	2	1	1	1
2010	75	Bilişim Teknolojileri	Seçmeli	Not yok						1	1	1
2012	69	Bilişim Teknolojileri ve Yazılım	Seçmeli	Not yok					2	2	2	2
2013	22	Bilişim Teknolojileri ve Yazılım	5 ve 6. sınıflarda zorunlu 7 ve 8. sınıflarda seçmeli	Not var					2	2	2	2

**Şekil 1.** Bilişim Teknolojileri ve Yazılım Dersinin Tarihi (BTE, 2019).

Şekil 1’de görüldüğü gibi “Bilgisayar” dersi ilk kez 1997 yılında 143 sayılı karar ile 4.5.6.7. ve 8. sınıflar 1 veya 2 ders saati seçmeli olarak ders çizelgesinde yerini almıştır. 1998’de dersin öğretim programı 180 sayılı karar ile yayınlanmıştır. Dersler için formatör öğretmenler yetiştirilmiştir. 2002 yılında BÖTE bölümleri mezun vermeye başlamıştır. 2005 yılında 192 sayılı kararla 1. sınıftan 8. sınıfa kadar bilgisayar dersleri haftalık 1 ders saati olarak düzenlenmiş ve notla değerlendirme kaldırılmıştır. 2007 yılında 111 sayılı karar ile dersin adı "Bilişim Teknolojileri" olarak değiştirilmiştir. 1.2.3. sınıflarda haftalık 1 ders saati, 4.5. sınıflarda 2 ders saati ve 6.7.8. sınıflarda 1 ders saati olarak düzenlenmiş, yeni öğretim programıyla birlikte 8 basamaklı çalışma kitapları dağıtılmıştır. Kitaplarda; Bilgi Teknolojileri Kavramları, Ofis Yazılımları Bilgisayarın Kullanımı ve Dosyaların Yönetimi konuları yer almıştır. 2010 yılında 75 sayılı karar ile öğrencilerin bilgisayar bilgisi yeterli görülerek, 6.7.8 sınıflarda seçmeli olarak haftalık 1 ders saati olarak düzenlenmiştir (BTE, 2019).

Programlama eğitiminin önemi tüm dünyada yükselirken, ülkemizde de 2012 yılında 69 sayılı kararla dersin adı “Bilişim Teknolojileri ve Yazılım” olarak güncellenmiştir. Yazılım, ders isminde ilk defa kullanılmıştır. Ortaokulda 5.6.7.8.

sınıflarda seçmeli olarak haftada 2 ders saatine çıkarılmıştır. Dersin yayınlanan programında problem çözme, algoritma ve programlama konuları yer almıştır (TTKB, 2012). 2013 yılında yapılan değişiklikle 5. ve 6. sınıflarda ders zorunlu hale getirilmiştir. Tekrardan notla değerlendirme geri getirilmiştir. 2017 yılında yenilenen öğretim programında blok tabanlı programlama araçlarına ağırlık verilmiştir (MEB, 2017). Millî Eğitim Bakanlığı ile Google Türkiye arasında yapılan iş birliği sayesinde, öğrencilerin BTY dersinde programlama, kodlama ve dijital okur yazarlık konularında akademisyenlerden oluşan ekip tarafından; çalışma kâğıtları, interaktif sınıf içi etkinlik uygulamaları, sunumlar ve öğretmen rehberi materyalleri oluşturulmuştur. Bunun yanında ortaöğretimde de programlamanın ağırlıklı olduğu “Bilgisayar Bilimi Dersi (Kur 1 - Kur 2) Öğretim Programı” yayınlamıştır (MEB, 2016). 2018’de programda yapılan güncellemeyle; problem çözme, algoritmalar, programlama, robot programlama, mobil programlama ve web programlama konuları yer almıştır.

Ülkemizde son yıllarda programlama öğretimi ve konuyla ilgili farkındalık geliştirilmesi amacıyla çok sayıda etkinlik gerçekleştirilmiştir. Bunlardan birisi olan Kodlarize etkinliği, ortaokul düzeyindeki öğrencilere yönelik Rize Valiliği tarafından koordine edilen bir kodlama projesidir. Kodris ise, 8 ile 16 yaş arasındaki öğrencilerin programlama öğrenebilmeleri amacıyla oluşturulan, yaş gruplarına göre öğrenme düzeyi ayarlanmış bir öğretim programına sahip, çevrimiçi bir öğrenme ortamıdır. Kodlayap, Karadeniz Teknik Üniversitesi, Trabzon Valiliği ve Trabzon İl Millî Eğitim Müdürlüğü’nün iş birliğiyle gerçekleştirilen bir kodlama öğretimi projesidir. Bilge Kunduz kodlama projesi ise, her yaştan öğrencide bilgi işlemsel düşünme becerisi ve bilgisayar bilimlerine ilişkin farkındalık oluşturma amacıyla oluşturulmuştur. Kodla(MA)nisa Proje kapsamında ise, Manisa’da bulunan ortaokullarda ve ayrıca açılan atölyelerde robotik ve kodlama eğitimi etkinlikleri gerçekleştirilmektedir. Bilgisayar Programlama Çocuk Oyunağı etkinliğinde ilkokul, ortaokul ve lise seviyesindeki öğrenciler yer almış ve bu etkinlik sayesinde bu öğrenciler ilk defa bir bilgisayar programı yazmışlar ve bir web sitesi tasarlamışlardır. Maker Kamp etkinliğinde ise öğrenciler algoritmalar geliştirmekte ve geliştirdikleri algoritmaları kullanarak program oluşturabilmektedirler. İstanbul Çocuk Üniversitesi “bilgisayar programlıyorum” etkinliği ile 5. 6. ve 7. sınıfları kapsayan bir programlama etkinliği gerçekleştirmiştir. Hacker Can, öğrencilere Türkçe kod yazarak program geliştirme becerisi kazandırmak amacıyla oluşturulan ilk programlama öğretimi platformudur. Gebze Teknik

Üniversitesi “çocuklar için kodlama atölyesi” adı ile ilkokul ve ortaokul öğrencileri için kodlama öğretimi etkinlikleri düzenlemiştir. İlkokul öğrencileri ile çalışılırken Scratch programlama aracı kullanılmış ortaokul öğrencileri ile çalışılırken Microsoft Small Basic aracı kullanılmıştır. (Şahin, 2018).

### 2.1.3. Çocuklar için Programlama Öğretiminde Kullanılan Araçlar

Programlama öğretirken en önemli konu temel programlama mantığının öğretilmesidir. Bunun için ise ilk olarak algoritmalar ele alınmalıdır. Algoritmalar, hedef programın tasarım aşamasında yararlanılan ve programın geliştirilmesi aşamasının uygun bir şekilde anlatıldığı yapılar olarak ele alınabilir. Algoritma, sonlu bir işi ifade etmek için kullanılan, net olarak tanımlanan ve belirli aşamaları olan yapılardır (Arabacıoğlu ve diğerleri, 2007). Tespit edilen sorunun çözümüne yönelik gerçekleştirilecek işlemler sırasında ihtiyaç duyulan kodlar, kullanılan programlama diline göre farklılık gösterebilir. Diğer yandan programlamada kullanılan algoritma mantığı çoğu durumda aynıdır. Bu açıdan programlama öğretirken öncelikle programlama mantığının öğretilmesi üzerinde durulmalıdır (Gökoğlu, 2017). Öğrencilere programlama mantığı öğretilirken en önemli aşama algoritmaların kullanılarak daha kolay ve sade programlama yapabilme becerisi kazandırmak olduğu için programlama öğretimi sırasında öncelikle algoritmalar konusunun çok iyi işlenmesi uygun olacaktır. Son zamanlarda, bu amaçla blok tabanlı programlama eğitimi ortamlarının yaygın olarak kullanıldığı görülmektedir.

Son yıllarda programlama öğretimine olan ilgi artmış, buna paralel olarak çocuklara yönelik programlama dili ve görsel ortamların sayısında da bir artış olmuştur. Bu programlama dilleri genellikle grafik tabanlıdır ve bu sayede öğrenciler sözdizimine takılmadan doğrudan bir program yazmaya odaklanabilmektedir. Grafik tabanlı programlama dillerine örnek olarak son dönemlerde daha da yaygın bir şekilde kullanılan blok tabanlı programlama dilleri verilebilir. Bu dilleri kullanan öğrenciler, komut setlerini bir araya getirirken tıpkı bir yapbozun parçalarını birleştirir gibi sürükleyip bırak yöntemiyle Lego benzeri blokları sürükleyip uygun şekilde birleştirerek programlama komutları oluştururlar. Oluşturulan komut dosyaları çoğu kez bir hikâyeyi anlatan ya da bir oyun oluşturan, bazen de çeşitli karakterleri veya arka plan görüntülerini oluşturan ya da kontrol eden komut setleri olarak işlev görür. Bu diller, sözdizimi hatalarından kaynaklanan sorunları azaltarak ve metin tabanlı programlama dillerinin çoğuna göre daha kolay oluşturulan bir komut setine imkân tanıyarak

öğrencilerde oluşması muhtemel programlamaya yönelik bilişsel ve duyuşsal engellerin aşılmasına yardımcı olmaktadır (Hill, 2015). Programlama öğrenmeye yeni başlayanların programlama öğrenirken karşılaşılan zor yapıları kolaylıkla öğrenebilmeleri amacıyla Hyperstudio, Scratch, Lego Mindstorm, Alice, Etoys, Code Academy, Khan Academy, Code Club, Code.org ve Coder Dojo gibi görsel olmalarıyla ön plana çıkan programlama araçları ortaya çıkmıştır (Yükseltürk ve Altıok, 2016). Scratch, App Inventor, Alice, LEGO Mindstorms, GameMaker, Microsoft Small Basic ve LOGO programlama öğretim araçları blok tabanlı programlama dilleri arasında en çok tanınan ve kullanılan araçlardandır (Demirer ve Sak, 2016). Bu programlardan Scratch, Alice, Game Maker ve LOGO biraz daha küçük yaş gruplarına kod ortamından uzak, görsel bir programlama ortamı sunan yazılımlardır. LEGO Mindstorms, bilgisayardaki bir yazılım aracılığıyla LEGO robotlara hareket komutları vermektedir (Çetin, 2012).

Çocuklara programlamayı öğretmek adına dünya genelinde farklı ara yüzlere sahip, kullanımı kolay ve görsel yönden zengin pekçok programlama aracı tasarlanarak hazırlanmıştır. 5- 6 yaş grubu anasınıfı öğrencileri için kullanılan Scratch Jr ile ilköğretim ve ortaöğretimin ilk basamaklarında kullanılabilecek Scratch programı, iş birlikçi yapısıyla kolay ulaşılabilirliği sayesinde programlama öğretiminde ilk sıralarda bulunmaktadır (Yüksel, 2017).

Daha önce Türkiye’de programlama öğretimi lisans düzeyinde ve meslek liselerinde ele alınmakta idi. Buna karşın, son yıllarda, birçok ülke, programlama öğrenimini sırasında çocukların eğlenebilecekleri bir ortam sağlayarak öğrencilerin daha erken yaşlarda programlama yapmaya başlamalarına yönelik düzenlemeler yapmaktadır. Ülkemizde 2012 yılında değişen eğitim müfredatı ile 5. ve 6. sınıflarda “Bilişim Teknolojileri ve Yazılım” dersi kapsamında “problem çözme, programlama ve özgün ürün geliştirme” amacıyla blok tabanlı programlama öğretimi araçları kullanılmaktadır (Berksoy, Sözcü, Armağan ve Arslan, 2016).

Çocuklara yönelik programlama dilleri ile temel olarak programlama öğrenimi sırasında sıklıkla öğrencilerin karşısına bir sorun olarak çıkan, zor bir yapıya sahip olmasından kaynaklanan öğrenme zorluğunu azaltmak ve programlama öğrenmeye yeni başlamış öğrenenleri bilişim teknolojilerine ısındırarak öğrenmelerini kolaylaştırmak hedeflenmiştir (Schwartz, Stagner ve Morrison, 2006). Öğrencilerin görsel programlama öğretimi ortamlarında yapılan programlama eğitimini eğlenceli buldukları

belirtilmektedir (Fidan, 2016; Lopez, Gonzalez ve Cano, 2016; Ozoran ve diğeri, 2012; Yıldırım, 2016). Görsel bir ortamda kod blokları ile çalışma imkânı sunan blok tabanlı programlama araçlarının kullanım amacı, asıl olarak bunlarla gelişmiş programlar oluşturmak değil, daha ziyade başlangıç seviyesindeki programcılar için ilgili çekici, kolaylaştırıcı ve eğlenceli bir programlama ortamları sunmak ve onları daha gelişmiş programlama dilleri kullanarak daha gelişmiş programlar yazmaya cesaretlendirmektir (Papert, 1993). Blok tabanlı programlama dillerinin yardımıyla küçük yaşta çocukların da eğlenceli bir şekilde yaratıcılık, algoritmik düşünme, problem çözme ve bilgi işlemsel düşünme becerileri gibi becerileri erken dönemde kazanmaları sağlanabilir (Shin ve diğeri, 2013; Çatlak ve diğeri, 2015).

Öğrencilere erken yaşlarda programlama becerileri kazandırabilmek için bilişim derslerinden yararlanılmakla birlikte çoğu eğitimci bu derslerde, öğrencilere programlamanın mantığını öğretmekte zorluk yaşamaktadır. Yükseköğretimde lisans seviyesinde işlenen programlama derslerinde bile zorluk yaşandığı göz önüne alındığında, ilköğretim ve ortaöğretim seviyesindeki öğrencilere programlama öğretimi amacıyla kullanılacak olan teknik, strateji ve araçlar konusunun önemle ele alınması gerekliliği ortaya çıkmaktadır. Ayrıca, programlama öğrenme sürecinde çocukların blok tabanlı araçlar kullanmalarının, onların problem çözme becerilerine katkı sağladığını gösteren çalışmalar bulunmaktadır (Brown ve arkadaşları, 2008; Kalelioğlu ve Gülbahar, 2014; Nam ve diğeri, 2010; Shin ve Park, 2014; Vatansever, 2018). Programlama öğretimi üzerine çalışan Erol (2015), ortam olarak blok tabanlı araçları seçtiği, temel programlama mantığı ve yapılarını öğretmeyi hedefleyen araştırmasının sonucunda; deney grubundaki öğrencilerin bu araçlar kullanılarak yapılan oyun tasarlama sürecini kolay ve eğlenceli bulduklarını, yapılan uygulama ve etkinliklerin ise öğrencilere temel programlama mantığını kazandırarak, konuyu öğrenmeye ilişkin isteklerini artırmada başarılı olduğunu bildirmiştir.

Bir diğeri çalışmada ise programlama öğretiminde “blok tabanlı robot kodlama platformunun” kullanılabilirlik durumunu, programlama öğretimine ilişkin uygulamaların nasıl geliştirilebileceği araştırılmıştır (Numanoğlu ve Keser (2017). Araştırma sonucunda; blok tabanlı programlama araçları sayesinde programlama öğretiminde soyut kavramların somutlaştırılabileceği ve öğrencilerin yazdıkları kodların ve algoritmaların sonucunu hemen görebilmeleri sayesinde bilgi işlemsel düşünme ve problem çözme becerilerinin geliştirilebileceği belirtilmiştir.

Öğrencilere matematiksel ve mantıksal düşünme becerileri kazandırmak amacıyla geliştirilen Logo programlama dili, 1969 yılında Papert ve MIT araştırmacıları tarafından geliştirilmiştir. Bunun yanında, son yıllarda çocuklarda problem çözme ve bilgi işlemsel düşünme becerilerini arttırmanın yanı sıra, programlama öğrenmeyi eğlenceli bir hale getirerek kolaylaştıran çeşitli programlama araçları vardır. Code Monkey, Small Basic, Code Combat, Code.org, Kodu, Appinventor, Blockly ve Scratch bu araçlara örnek olarak verilebilir. Bunlardan, Microsoft tarafından geliştirilen Small Basic, programlamaya yeni başlayan kişiler tarafından tercih edilen araçlardandır. Bir maymun karakterinin kodlar yazılarak verilen görevleri yerine getirmesi sağlanan Code Monkey, daha çok erken yaşlardaki çocuklar için kullanılmaktadır. Hareketli oyunları seven çocuklara yönelik olan Code Combat ortamında, başlangıçta bir karakter seçilerek haritada belirlenen yerlerdeki görevler uygun kodlamalarla yerine getirilmektedir. Bu kodlama oyununda Javascript, Python, Lua ve CoffeeScript programlama dilleri arasından seçim yapılabilmektedir (Erdem, 2018).

Çocuklara kolay ve basit bir programlama dili ile görsel olarak Xbox'ta ve PC'de oyun oluşturma imkânı sunan Kodu, Microsoft'un geliştirdiği bir araçtır. Code.org web sitesi ise Google, Microsoft, Facebook gibi büyük teknoloji firmalarının da desteklediği, erken yaşlardan başlayarak programlama öğretmeyi amaçlayan bir oluşumdur. Google tarafından oluşturulan Blockly çok sayıda farklı kodlama etkinlikleri içeren bu araç, küçük yaştaki çocukların programlama öğrenmesine yardımcı olacak şekilde blok tabanlı bir programlama öğretimi aracıdır. Blok tabanlı kodlama platformu olan Appinventor kullanılarak Android sistemler için çeşitli uygulamalar geliştirilebilir (Erdem, 2018). Bu çalışmada kullanılmak üzere seçilen Scratch ise, son yıllarda özellikle ilkökul, hatta ortaokul seviyesindeki öğrencilere programlama öğretimi amacıyla sıklıkla tercih edilen (Kukul ve Gökçearslan, 2014; Oluk ve Korkmaz, 2016; Moreno-León, Robles ve Román-González, 2016) blok tabanlı bir kodlama öğretimi aracıdır.

#### **2.1.4. Blok Tabanlı Programlama Eğitimi**

Programlama sürecinde bireyler, bilgi işlemsel ve matematiksel düşünme becerilerini ve problem çözme becerilerini geliştirirken, bir yandan da proje tasarımı ve düşünceler arasında bağlantılar kurarak fikirler oluşturmayı öğrenebilirler. Günümüzde bu ve benzeri beceriler sadece programcıların değil her meslekten kişiler için ihtiyaç duyulmaya başlanan becerilerdir (Wing, 2006; Resnick, 2013).

21. yüzyıl becerileri arasında belirtilen programlama becerisi, yeni nesil teknolojilerin günlük hayatta kullanılmasıyla birlikte birçok sektör çalışanları için gerekli ve önemli bir beceri olarak karşımıza çıkmaktadır (Balanskat ve Engelhardt, 2015).

Akpınar ve Altun'a (2014) göre, programlama öğrenmek aşağıda belirtildiği üzere; bireylerin günümüz dünyasında ihtiyaç duyulan çeşitli becerilerini geliştirme potansiyeline sahiptir. Programlama, problemde analiz yeteneğini arttırabilmekte, problemin daha rahat çözümlenmesi için alt parçalara ayırmak ve genel bir çözüm elde etmede kullanılabilir bu yüzden programlama, problem çözme becerisini geliştirmeyi destekler niteliktedir. (Saeli ve diğerleri, 2011; Feurzeig ve diğerleri, 1970; Papert, 1993; Dasso ve diğerleri, 2005).

Scratch, Alice, code.org, Kodu game lab blok temelli programlama araçları, gibi görsel programlama dilleri olması nedeniyle öğrencilerin küçük yaşlardan itibaren geleneksel programlama dillerindeki karmaşık yapıları öğrenmelerine ihtiyaç duyulmaksızın, oyun, animasyon ve programlar oluşturmalarına olanak sunmaktadır (Resnick ve diğerleri, 2009). Bu araçlarda yer alan kodlar, Lego benzeri blok şeklinde bir yapıya sahip olduğu için özellikle başlangıç seviyesindekiler için kolaylık sağlamaktadır (Wilson ve Moffat, 2010). Son yıllarda giderek daha popüler hale gelen robotik kodlama aracı olarak kullanılabilen Scratch for Arduino programı kullanılarak, Arduino sistemine bağlı sensörleri ve eklentileri yönetmek mümkün hale gelmiştir. Böylece Scratch ortamına aşina olan öğrenciler kolaylıkla hazırladıkları projelerini bir robot üzerinde deneme ve görme imkânına kavuşmaktadırlar (Scratch for Arduino, 2015). Sıklıkla tercih edilmesi ve blok tabanlı bir programlama öğretimi yapısına sahip olması nedeniyle bu çalışmada da kullanılan Scratch üzerine yapılan çalışmalar incelenerek sonraki bölümde verilmektedir.

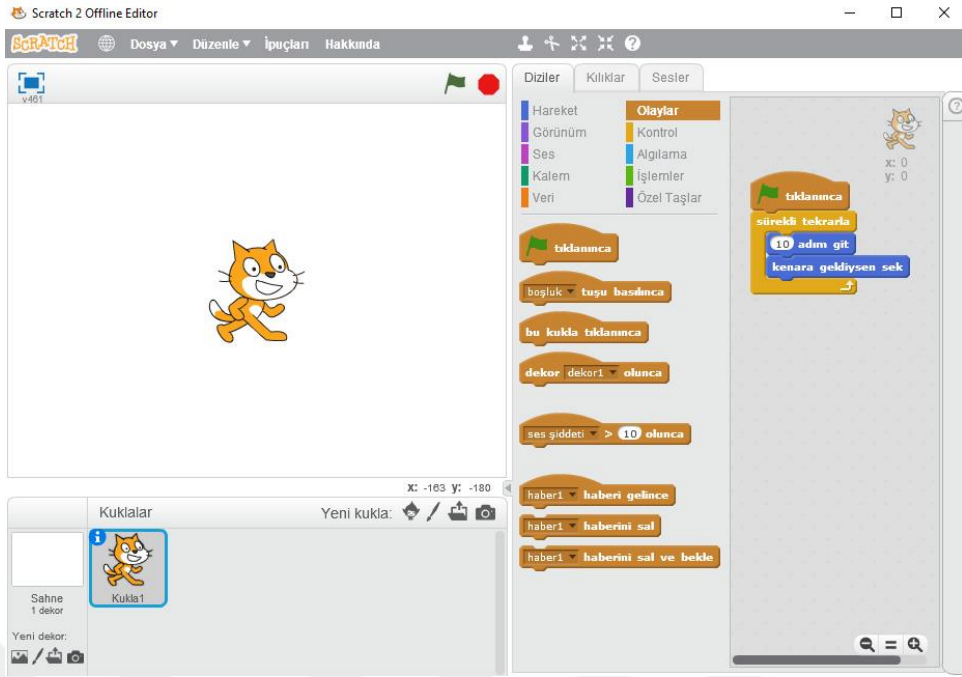
Görsel programlama öğretimi ortamlarında yapılan programlama eğitiminin öğrenme sürecini kolaylaştırdığı (Genç ve Karakuş, 2011; Malan ve Leitner, 2007; Lewis, 2010; Nam ve diğerleri, 2010; Maloney ve diğerleri, 2010; Yıldırım, 2016), öğrencileri programlamaya yönelik olumlu yönde etkilediğini (Fesakis ve Serafeim, 2009; Gülbahar ve Kalelioğlu, 2014; Fidan, 2016; Kobsiripat, 2015; Ozoran ve diğerleri, 2012; Malan ve Leitner, 2007; Yıldırım, 2016), bu sayede programlamaya yönelik olumlu bir tutuma sahip oldukları (Baltalı, 2016; Genç ve Karakuş, 2011; Fesakis ve Serafeim, 2009; Ozoran ve diğerleri, 2012; Malan ve Leitner, 2007;

Yıldırım, 2016) ve programlama için pozitif olarak güdülediği (Erol, 2015; Fowler, 2012; Fidan, 2016; Gülbahar ve Kalelioğlu, 2014; Genç ve Karakuş, 2011; Lopez ve diğerleri 2016; Ruf, Mühling ve Hubwieser, 2014; Osman, Zakaria, Loke ve Downe, 2012) belirtilen çok sayıda çalışma bulunmaktadır. Benzer bir şekilde programlama yaparken başarılı olan öğrencilerin, başarıma duygusuyla birlikte bilgisayara yönelik tutumlarının da olumlu olarak gelişebileceği söylenebilir (Yükseltürk ve Altıok, 2016). Bu gibi nedenlerle özellikle programlamaya yeni başlayan acemi programcılar için görsel programlama ortamlarının kullanılması daha uygun olacağı söylenebilir.

## **2.2. Scratch**

Programlama eğitimi sürecinde özellikle yeni başlayanlar için ortaya çıkan, görsel olarak kodlama yapma imkânı sunan Scratch, Alice, Microsoft Small Basic, Stagecast Creator ve Toontalk benzeri programlama ortamları oluşturulmuştur. Özellikle genç yaştaki çocukların yararlanabileceği bu tür ortamların yardımıyla onlar da erken yaşta edindikleri programlama becerisinin getirdiği avantajları ve yaratıcılıklarını kullanarak harika projeler ve fikirler ortaya koyabilirler. Bunlar gibi görsel programlama öğretimi araçlarının programlama öğretimindeki yerini araştıran pek çok çalışma bulunmaktadır (Gülbahar ve Kalelioğlu, 2014; Ozoran, Cagiltay ve Topalli, 2012; Nam ve diğerleri, 2010; Shin ve Park, 2014; Wang, Huang ve Hwang, 2014). Scratch, Massachusetts Teknoloji Enstitüsü (MIT) Medya Laboratuvarı'nda bulunan Lifelong Kindergarten ekibinin bir projesi olarak 2003 yılında oluşturulmaya başlanıp, 2007'de kullanıcılar için yazılım halinde sunulmuştur. Programlama öğrenme için kullanılan ilk ortam olarak özellikle 8-16 yaş grubundaki gençler ve çocuklar için tasarlanan Scratch, giderek daha da popüler hale gelmiş ve hemen her yaştan insanın programlama öğrenmek için kullanabileceği görsel bir programlama ortamı haline gelmiştir (Maloney ve diğerleri, 2008).





**Şekil 2.** Scratch Ekranı

Şekil 2’de görüldüğü gibi metin tabanlı kod yazmayı gerektirmeyen yapısı sayesinde Scratch ortamında programlama öğrenenler, sürükle bırak işlemiyle bir kod bloğunu sürükleyip uygun bir yere bırakarak çeşitli projeler oluşturabilmektedir. Algoritma mantığını öğrenmekte zorluk yaşayanlar veya programlamaya yeni başlamış olan acemi programcılar için büyük kolaylıklar sunan Scratch, basit ve kullanıcı dostu yapısı ile milyonlarca insan evde, okulda veya kütüphanelerde kolaylıkla Scratch projeleri oluşturabilmektedir. Oluşturulan bu projeler ise çevrimiçi ortamda kaydedilmekte ve böylece dünyanın her tarafındaki Scratch kullanıcıları ile paylaşılabilir. Scratch kullanıcıları birbirleriyle kolaylıkla iletişim kurabilmekte ve iş birliği halinde çeşitli projeler üzerinde çalışabilmektedirler. Böylelikle kullanıcılar, çevrimiçi ortamda paylaşılan bir projeyi detaylı olarak inceleyebilmekte, isterse üzerinde düzeltmeler yaparak kendine özgü bir hale dönüştürebilmektedir (Scratch About, 2019). Bu sayede bir başkasının yaptığı bir proje, bir diğer kullanıcıya ilham verebilmekte ve onları daha gelişmiş ya da çok daha farklı bir projeye ulaştırma potansiyeli taşımaktadır.

Scratch aracının programlama öğretimindeki yerine ilişkin bir doküman incelemesi çalışması yapan Çatlak ve diğerleri (2015), bu aracın öğrenciler için ilginç, kolay kullanılabilen ve merak uyandırıcı bir ortam sunduğunu, programlamanın temellerini öğretmek için etkili bir araç olduğunu, ayrıca öğrencilerde yaratıcı düşünme

becerilerini geliştirdiği ve öğrenmeye ilişkin isteklerini arttırdığını belirtmişlerdir. Konuya ilişkin bir diğer araştırmada ise, Scratch'ın bir öğrenme aracı olarak kullanılabilir olduğu ve Scratch kullanılarak yapılan eğitim faaliyetleri sonucunda öğrencilerin yaratıcılıklarının geliştiğini bildirilmiştir (Kobsiripat, 2015).

Scratch ortamında bulunan öğeler sadece kod blokları değildir. Kodlama esnasında çeşitli çoklu ortam unsurları da kullanılabilir. Scratch, resim ve müzik öğeleri, simülasyonlar, sunular, oyunlar, animasyonlar ve videolar gibi pek çok öğe kullanılarak çok geniş yelpazede projeler oluşturulmasına olanak tanınması sayesinde, diğer derslerde de kullanılabilir. Bilişim dersleri dışında, Fen, Matematik, Sosyal Bilgiler ve Yabancı Dil gibi derslerde de öğrencilerin zorlandıkları konular Scratch projeleri oluşturularak oyunlaştırılıp eğlenceli hale getirilebilir ve öğrenciler tarafından daha kolay bir şekilde öğrenilebilir. Tamamen eğitim amaçlı oluşturulan ve kullanımı ücretsiz olan Scratch yazılımı ile öğrenciler, bazı gerçek nesnelere kodlar sayesinde yönetebilir, yeni komut setleri geliştirerek elektronik devrelere istediklerini yaptırabilirler. Picoboard adı verilen elektronik bileşen bunlardan birisi olup, öğrenciler Scratch ortamında geliştirilen bir program sayesinde etkileşim kurularak bu devreye hükmedebilirler. Devrede kullanılan çeşitli sensörler yardımıyla kendilerinin yazdığı bir programı çalıştırarak gerçek dünyada somut nesnelere çalışabilirler. Scratch programı bu sayede öğrencilerde yaratıcılık, analiz yapabilme ve farklı düşünme becerisi, problem çözme ve sistematik düşünme becerisi gibi bazı becerilerini geliştirmek amacıyla kullanılabilir (Çatlak ve diğerleri, 2015).

Son zamanlarda ortaya çıkan ve giderek kullanımı yaygınlaşarak daha da popüler hale gelen görsel özelliklere sahip programlama dilleri sayesinde programlama öğretiminin kolaylaşması, ilkökul ve ortaokul seviyesinde programlama öğretimine başlanabileceği düşüncesine yol açmıştır. Geniş bir kullanıcı kitlesine ve çok sayıda dil desteğine sahip olması, farklı yaş seviyelerine hitap edebilmesi ve basit ara yüze sahip olması ile gibi özellikleri nedeniyle bu çalışmada Scratch yazılımı kullanılmıştır.

### **2.3. Problem Çözme ve Problem Çözme Becerisi**

John Dewey problemi, “insan zihnini karıştıran, insana meydan okuyan ve insanın inancını belirsizleştiren bir olgu” şeklinde tanımlanmaktadır. Dolayısı ile problemin çözümü de zihinde karışıklık yaratan belirsizliklerin ortadan kaldırılması olarak anlamlandırılabilir. Bir problemi çözmek için problemin tespitinin yapılması, çözüm için gerekli verilerin toplanması, bu verilerden de bir eleme yaparak çözüme

götürecek olanların seçilmesi ve elde kalan verilerin uygun bir şekilde kullanılması gerekir (Baykul, 2005).

### **2.3.1. Çocuklarda Problem Çözme**

Öğrenci ve öğretmenlerin yararına olduğu kadar herkesin de yararına olan problem çözme becerisi, öğretim programlarında yer alması gereken oldukça önemli bir beceridir. Karşılaşılan yeni durumlara ve sorunlara uygun çözümler bulmak ve bu çözümleri doğru şekilde uygulamak, yalnızca ihtiyaçların karşılanması anlamında değil, çocuklara mutlu bir yaşamın için oldukça gerekli olan bir beceri kazanma fırsatı vermesi açısından da önemlidir (Forgan, 2003). Bir problemle karşılaşmak sadece okula, derslere ya da bu araştırmada da olduğu üzere programlama dillerine özgü değildir. Bu sebeple problem çözme becerisi bireyin tüm yaşamı boyunca gereksinim duyacağı bir yetenektir.

Yapılan çalışmalar, problem çözme becerisi gelişmiş olan kişilerin sosyal ilişkilerde daha aktif oldukları, daha iyi iletişim becerilerine sahip oldukları, kendilerine ve karşılaştıkları duruma uygun çalışma yöntemleri sergileyerek akademik açıdan daha başarılı olma potansiyeline sahip olduklarını göstermektedir (Şahin, Şahin ve Heppner, 1993).

Çocuklar, problem çözme becerilerini kullanma olanağı buldukça bu konudaki yeteneklerini keşfeder ve becerilerini geliştirirler. Karşılaşılan zorluklar ve sorunlar üzerinde bir başkasının karar vermesi yerine kendi çözüm yollarını bulması için teşvik edilen çocuk, bunu yaparken kendi düşüncelerini, bilgilerini, deneyimlerini ve becerilerini kullanma fırsatı bularak iç ve dış kaynaklardan nasıl faydalanacağını öğrenmiş olur (Bingham, 2004). Bu ise bir çocuk için hayatı boyunca karşılaşacağı sorunları çözmeye kullanabileceği ve çok işine yarayacak becerileri geliştirmesine yardımcı olacaktır.

### **2.3.2. Problem Çözme ve Programlama**

Çok sayıda araştırmada ilköğretim ve ortaöğretim seviyesindeki öğrencilerin özellikle problem çözme becerisi geliştirmede sorun yaşadıklarını göstermektedir. Problem çözme becerileri, öğrencilere eğitim hayatları boyunca kazandırılması gereken en kritik becerilerdir. Problem çözme becerisi, insanların sorunlarla başa çıkabilmek için uyumlu çözümler bulma arayışında, tecrübe, bilgi ve becerilerini kullanmalarını gerektiren karmaşık bir süreçtir (Lai ve Yang, 2011). Problem çözme ve programlama

kavramları günümüzde de birbirlerini etkileyen olgulardır. Bir problemle karşı karşıya kalındığında, bu problemin çözümü için programın planlı bir şekilde yürütülmesi gerekir. İmar edilen bütün teknolojiler aynı zamanda insan yaşamının yansımasıdır. Bilgisayar bilimindeki Programlama olgusu, mevcut problemi çözmenin ilk adımıdır. Çünkü insanoğlu var olan probleme nasıl çözüm önerisi getiriyorsa, bir bilgisayar sistemindeki problemi çözmek adına yapılan kodlamanın da bilgisayar diliyle bir programlama olduğu söylenebilir. Calder (2010) görsel programlama kullanarak yapılan programlama öğretiminin, öğrencilerin problem çözme becerilerini geliştirdiğini ve öğrenmeye ilişkin isteklerini artırdığını belirtmiştir. 6. sınıf öğrencileriyle yapılan bir çalışmanın sonuçlarına göre problem çözme becerisi, programlama öğrenen öğrencilerde gelişmiştir (Lai ve Yang, 2011). 5-6 yaş grubundaki öğrencilerle yaptıkları çalışmada Fessakis, Gouli ve Mavroudi (2013), programlama öğrenen öğrencilerin problem çözme becerilerinin geliştiğini bildirmişlerdir. Bu çalışmaların sonuçlarında da görüldüğü üzere, programlama yaparken sıklıkla problem çözmek zorunda kalan öğrencilerin problem çözme becerilerinin arttığı anlaşılmaktadır.

Konuya ilişkin bir diğer araştırmada ise ilköğretim öğrencileriyle çalışılmış, Scraeth'in bir öğrenme aracı olarak kullanılabilir olduğu ve Scraeth kullanılarak yapılan eğitim faaliyetleri sonucunda öğrencilerin yaratıcılıklarının, girişkenlik ve akıcı düşünme becerilerinin geliştiği bildirilmiştir (Kobsiripat, 2015).

Programlama öğretiminin bilgi işlemsel ve matematiksel düşünmenin geliştirilmesinde etkisi olup olmadığını araştıran bir çalışmada; programlama öğretiminin matematik konularının öğretiminde, sistematik, işbirlikçi ve yaratıcı düşünme becerileri ile problem çözme becerilerinin geliştirilmesinde etkili olduğu belirtilmektedir (Taylor ve diğerleri, 2010).

Programlama öğreniminde bir diğer etkili faktör ise problem çözme becerisidir (Pillay ve Jugoo, 2005; Yurdugül ve Aşkar, 2013). Gomes ve Mendes (2007) problem çözme becerisinin programlama öğreniminde oldukça önemli olduğunu ve programlama öğreniminde yaşanan zorlukların çoğunun problem çözme becerisi eksikliğinden kaynaklandığını belirtmektedir. Ayrıca problem çözme becerisinin programlama başarısının etkisini inceleyen çalışmalara bakıldığında ise bu ikisi arasında bir ilişki olduğu ve bu ilişkinin anlamlı olduğu bildirilmektedir (Pillay ve Jugoo, 2005; Yurdugül ve Aşkar, 2013).

2000'li yıllarda ise Gundurao, Manjunath ve Nachappa (2010) benzer bir çerçeve ile program yazma basamaklarını 3 temel adıma ayırmakta ve bu üç temel adımı aşağıdaki gibi alt basamaklarla ifade etmektedir:

- Problemin
  - o Analizi
  - o Çözümü (programın tasarımı)
    - Algoritma
    - Akış şeması
- Bilgisayar ortamındaki çözüm
  - o Kodlama
  - o Test etme ve hata ayıklama
  - o Belgeleme ve onarım

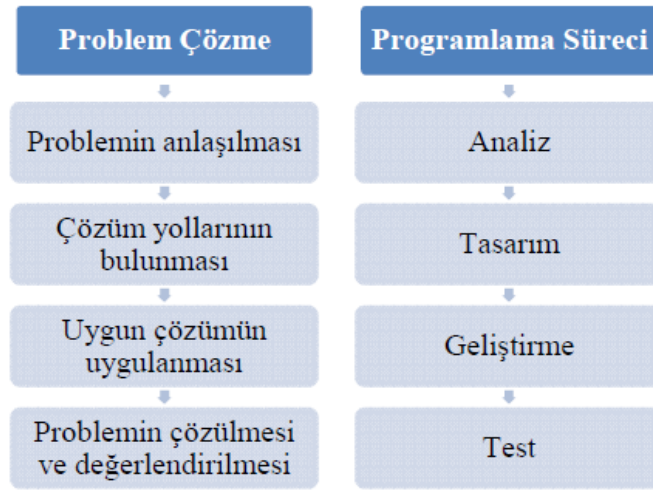
Bu basamaklar incelendiğinde de anlaşılacağı gibi programlama sürecinde program yazmak ve problem çözmek konusunda acemi programcıların sadece algoritma tasarlamayı öğrenmesi yetmemekte, algoritma basamaklarını belirli bir programa da çevirebilmeleri gerekmektedir (Cooper ve diğerleri, 2003). Programlama dili öğrenirken, kullanılan programlama diline ait model ve çeşitli programlama yapıları ile veri akışının nasıl ilişkili olduğu iyi anlaşılmalıdır. Dahası, program yazmak, test etmek ve hata bulmak gibi programlamanın farklı aktivitelerini yönetmek amacıyla çeşitli yazılım araçlarını öğrenmeyi gerektirmektedir (Helminen ve Malmi, 2010).

Bunun yanında yukarıda bahsedilen süreci öğrenmek, başka bir deyişle programlama becerisi edinmek önemli yararlar da sağlamaktadır. Özelde programlama, programlama ile alakası olmayan alanlara transfer edilen önemli problem çözme ve tasarlama becerilerini öğrenmeye yardım ederek sayısal düşünmeyi desteklemektedir. Programlama işlemi, bireyin zihninde yürüttüğü problem çözme süreçlerinin dış dünyadaki bilgisayar gibi bir ortamda kodlar ve çeşitli yapılar kullanarak temsilini oluşturmaktır. Bu nedenle programlama için; kişinin kendi düşünmesi hakkında derinlemesine düşünmesi denilebilir (Resnick ve diğerleri, 2009). Bunların yanında bilgisayar programlama ile ilişkili belirli becerileri öğretmek, öğrencilerin üst düzey düşünme becerilerinin yanında; mantıksal, eleştirel ve yaratıcı düşünme becerilerini geliştirmektedir (McMahon, 2009).

Kneeland (2001) problem çözmenin zekâ ile ilgili olmadığını belirtmektedir. Ona göre problem çözme becerisi, “düşünme ve çözüm sürecinin doğru olarak uygulanmasına bağlıdır”. Problem çözme aşamalarına ve programlama mantığına birlikte bakılacak olursa aşamalar şu şekilde ilerlemektedir:

1. Problemin tam olarak anlaşılması ve tanımlanması
2. Gerekli bilgi
3. Çözüm yollarının ortaya konulması
4. En iyi çözüm yolunun seçilmesi
5. Problemi çözme

Şekil 3’de programlama süreci bileşenleri ile problem çözme basamakları birlikte gösterilmiştir.



**Şekil 3.** Problem çözme ve programlama süreci bileşenleri (Çetin, 2012).

Şekil 3’e bakılarak durum programlama için düşünüldüğünde ilk aşama, ihtiyaç duyulan programın ve özelliklerinin tam olarak belirlenmesidir. Bir programı sıfırdan başlayarak oluşturma durumunda veya mevcut bir programda ihtiyaç duyulan değişiklikler yapılırken, öncelikle problem net olarak tespit edilmelidir. İkinci aşamada çözüme nasıl başlanacağına karar vermeden önce problemle ilgili bilgilerin toplanması gerekir. Bunlar problemi oluşturan, problemi çözmeye yardımcı olacak olan bilgiler olabilir. Mümkün olduğunca fazla çözüm yolunun ortaya konulması gerekmektedir. Programlama dillerinde zaten bir işin farklı yöntemlerle yapılma seçenekleri bulunmaktadır. İlk iki aşama, programlama süreci bileşenleri bölümünde bulunan

bileşenlerden ilki olan Analiz basamağını oluşturmaktadır. Analiz basamağında problemin net olarak tanımlanması, anlaşılması gerekir. Gerekli bilgiler de toplanınca tasarım için programcı hazır hale gelecektir. Üçüncü aşamada öğrencinin çözüm yollarını belirlemesi ve aralarından en doğru olanını uygulamaya koyması gerekmektedir. Bu da Tasarım ve Geliştirme aşamalarını oluşturmaktadır. Bu aşamada doğru ve etkili karar verme süreci işin içerisine girmektedir. Ortaya konulan çözüm yollarından en uygun olanı, problemi çözen kişi tarafından seçilmelidir. Bu araştırma için düşünülecek olursa ihtiyaca yönelik program yazmaya hazırlanan öğrenci, içerisinde döngü komutları bulunacak olan programı için hangi döngünün kullanılacağını seçmek durumundadır. Örneğin For döngüsü belirli değişkenler arasındaki işlemleri yerine getirirken, While döngüsü ise belirli bir koşul sağlanıncaya kadar istenen işlemleri yapar. Yazılacak programa göre hangisi daha uygunsa, onun seçilmesi makul olacaktır. Programlamada genellikle kodun daha az satırda olması tercih edilen bir durumdur, bu sebeple muhtemelen ilk bakılacak olan faktör kod sayısını hangi döngü modelinin azaltacağı yönünde olacaktır. Sonuç olarak problemin çözümünde sarf edilen çabalar bazen neyin yapılması gerektiği yerine neyin yapılmaması gerektiğini bildirir, öğrenciye bunu öğretir. Tüm bu çabalar, öğrenciyi etkili problem çözme sürecinin sonuna ulaştırır. Verilen kararın planlanan şekle uygun olarak uygulamaya koyulması ve yapılan değerlendirmelerin olumlu sonuçlanması ile problem çözme süreci tamamlanmış olur (Çetin, 2012).

Programcı, programı yazdıktan sonra doğru çalışıp çalışmadığını kontrol eder. Programlama süreci bileşenlerinden sonuncusu olan Test aşaması da böylece tamamlanmış olur. Yapılan değerlendirme sonucunda olumlu yanıt alıyorsa süreç tamamlanmış olur. Yazılan programlama dili son kullanıcılara kullanılır ve varsa hata ayıklama süreci işletilir. Son değerlendirmede program istenen şekilde çalışıyorsa problem çözülmüş olur. Programda düzeltilmesi gereken hatalar varsa bu sürecin her bir problem için yeniden işlemesi gerekmektedir (Çetin, 2012).

#### **2.4. Bilgisayara Yönelik Tutum**

Tutumun davranışları etkileyebilen bir özelliği olduğu bilinmektedir. Tutum, doğrudan gözlenememekle birlikte dolaylı olarak ölçülebilen bir değişkendir. Öneminden dolayı, birçok alanda o alana yönelik tutumlar ölçülerek tutumun derecesi tespit edilmeye çalışılır (Tezbaşaran, 1997). Smith'e (1968) göre "tutum, kişinin bir obje, bir kavram ile ilgili duygu, düşünce ve davranışlarını düzenli bir şekilde ortaya

çıkarıcı bir eğilimdir". Türk Dil Kurumu'na göre ise "tutum; bireyin insanlar, cansız varlıklar ve olaylar karşısında takındığı davranış biçimidir" (TDK, 1974). Chapman ise tutumu; "bireyin kendi dışındaki dünyada kullandığı zihinsel odak olup, içinde bulunduğu ruh halini diğer bireylere anlatma ve aktarma şekli" olarak ele alır (Chapman, 1999).

Günümüzde teknolojik gelişmelerin çok önemli bir aşamasını temsil eden bilgisayar ve ilgili teknolojilerin, günlük hayatımızda artan bir yere sahip oldukları ve giderek daha fazla önem kazandıkları, bilgisayara yönelik tutumun ise geçmişe kıyasla olumlu yönde geliştiği görülmektedir. Bununla birlikte bireylerin olumlu ya da olumsuz tutum durumları kişiden kişiye değişebileceği için her bir çalışma için ayrıca incelenmelidir.

Bu çalışmada ele alınan bilgisayara yönelik tutum ifadesinde ise bireylerin bilgisayara ilişkin duygu ve düşünceleri kastedilmektedir. Bilgisayara yönelik tutumun yaş, cinsiyet, önceki bilgisayar deneyimi ve bilgisayar sahibi olma gibi durumlarla ilişkili olduğu belirtilmektedir (Erkan, 2004). Bilgisayara yönelik tutumların nelerden etkilendiği ya da hangi faktörlerle etkilendiğinin tespit edilmesi için yeni çalışmalar yapılmalıdır. Bu çalışmada bilgisayara yönelik tutumun blok tabanlı programlama faaliyetlerinden etkilenme durumu tespit edilmeye çalışılmıştır.

Bilgisayar yönelik tutum "bireyin bilgisayara, bilgisayar kullanımına, bilgisayar kullananlara ve bilgisayarların toplumsal ya da kişisel etkilerine yönelik olarak sahip olduğu duygu, düşünce ve davranışları içeren bir eğilim" şeklinde tanımlanabilir (Üstündağ, 2001, 50). Ektiren'e (2014) göre "Bireyin değişime, bilime, teknolojiye ve bilgisayar kullanmaya ilişkin tutumu, bu olgulara ilişkin doğru-yanlış inanç, duygu, düşünce ya da bilgilerinin toplamından oluşur".

Meta analiz yöntemi kullanarak bir çalışmada Topçu (2009), bilgisayar yönelik tutumu ele almış, cinsiyetin bilgisayar tutumuna olan etkisini araştıran 47 çalışma üzerinden cinsiyet ile bilgisayara yönelik tutumu arasındaki ilişkiyi incelemiştir. Elde edilen verilerin analizi sonucuna göre, erkeklerin bilgisayara yönelik tutumları kadınlara oranla daha yüksek ve olumlu olduğu ortaya çıkmıştır (Topçu, 2009). Bu sonuca göre, bilgisayara yönelik tutum ele alınırken ya da bilgisayarla ilgili bir çalışma yapılırken cinsiyetin dikkate alınması, gerekirse kız öğrencilerin tutumlarının olumlu yönde gelişmesi için fazladan etkinlikler yapılması durumu göz önünde bulundurulmalıdır.



Ortaokul öğrencilerinin bilgisayara yönelik tutumlarını inceleyen bir diğer çalışmada ise üstün yetenekli öğrencilerin bilgisayara yönelik tutumlarının yüksek oldukları belirtilmiştir (Taşdemir, 2017).

Öğrencilerin bilgisayara yönelik tutumlarının ölçülmesi ile mevcut durum tespit edilerek incelenebilir. Eğitimde kullanılan teknolojilerin önemli bir bölümünün bilgisayar donanımları ya da çeşitli bilgisayar yazılımlar içermesi bakımından, öğrencilerin eğitim hayatları süresince bilgisayara yönelik tutumları onların öğrendikleri tüm konuları ve aldıkları tüm dersleri etkileyebilecek bir öneme sahiptir. Bu açıdan daha verimli ve kalıcı bir eğitim için şayet tespit edilen olumsuz tutumlar varsa bunlarla ilgili önlem almak mümkün hale gelecektir.

## **2.5. İlgili Araştırmalar**

Bu bölümde, bu çalışmada programlama öğretimi uygulaması yapılarak özellikle öğrencilerin problem çözme becerileri ve tutumları ile diğer bazı faktörlerle ilişkilerini elen alan çalışmalar bulunmaktadır.

### **2.5.1. Türkiye’de Yapılan Araştırmalar**

Öztürk ve Alper, (2019) ortaokul öğrencileriyle yaptıkları çalışmada, programlama öğretimi amacıyla kullanılan Ters Yüz Öğrenme Yönteminin, öğrencilerin bilgisayara yönelik tutumlarına, akademik başarılarına ve öz-yönelimli öğrenme beceri düzeylerine etkisini incelemişlerdir. Ön test-son test eşleştirilmiş kontrol gruplu yarı deneysel bir tasarım kullanılan çalışmada, toplamda 192 öğrenci yer almıştır. Araştırmada elde edilen bulgulara göre, Ters Yüz Öğrenme Yöntemi kullanılan deney grubu öğrencilerinin teknolojiyle öz-yönelimli öğrenme düzeyleri, akademik başarıları ve bilgisayara yönelik tutumları, kontrol grubu öğrencilerine göre istatistiksel olarak anlamlı ve yüksek bulunmuştur.

Dinçer’in (2018) 6. Sınıf öğrencileriyle yaptığı çalışmasının amacı, programlama öğrenmek için Kodu Game Lab ve Scratch araçlarını kullanan öğrencilerin programlama başarıları ile programlama öğrenmeye yönelik tutumları ve programlamaya yönelik öz-yeterlik algılarındaki değişim ve iki grup arasındaki farkları incelemektir. Araştırmada çalışma grubu olarak bir ortaokuldaki iki farklı şubede öğrenim gören 27 6. sınıf öğrencisi belirlenerek iki deney grubu ile çalışılmıştır. Deney gruplarındaki öğrencilerin seçimi için küme örnekleme yöntemi seçilmiştir. Bilişim Teknolojileri Öğretmenleri ile yapılan mülakatlar ile uygulamanın içeriği oluşturulmuş

ve ders planı yapılmıştır. Uygulama süreci toplam 20 ders saati sürmüştür. Bu sürede Deney-1 Grubu Scratch kullanırken, Deney-2 Grubu Kodu Game Lab kullanmıştır. Her iki gruba Programlama Dilleri Akademik Başarı Testi, Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğretimine Yönelik Tutum Ölçeği ve Programlamaya İlişkin Öz-Yeterlik Ölçeği ön-test ve son test olarak uygulanmıştır. Ayrıca kalıcılığın incelenmesi amacıyla son-testten sekiz hafta sonra Programlama Dilleri Akademik Başarı Testi tekrar-test olarak uygulanmıştır. Araştırmanın sonuçlarına göre; Scratch ile programlama öğrenen öğrencilerin, Kodu Game Lab ile öğrenim görenlere göre programlama başarılarının istatistiksel olarak anlamlı derecede arttığı tespit edilmiştir. Scratch programlama dili kullanılan öğrenciler ile Kodu Game Lab kullanılan öğrencilerin öz-yeterlik algıları ve tutumları arasında anlamlı bir fark saptanmamıştır. Diğer yandan, Kodu Game Lab ve Scratch kullanılarak programlama öğretimi yapılan öğrencilerin programlama dillerine yönelik öz-yeterlik algıları anlamlı derecede artmıştır. Ayrıca, Kodu Game Lab kullanılan öğrencilerin programlamaya yönelik tutumları olumlu olarak değişmiş ve artmıştır. Buna göre öğrencilerin oyun ortamlarında öğrenmelerinin onların ilgili konuya yönelik tutumlarının değiştirilmesinde etkili olabileceği söylenebilir.

Vatansver'in (2018) ortaokul 5. ve 6. sınıf öğrencileriyle yaptığı araştırmanın amacı, Scratch kullanılarak gerçekleştirilen programlama öğretimi etkinliklerinin öğrencilerin bu sürece ilişkin görüşlerini incelemek ve onların problem çözme becerilerine etkisini araştırmaktır. Karma yöntem araştırmalarından açıklayıcı desen kullanılan bu araştırmanın nicel kısmında tek grup ön test - son test model, nitel kısmında ise durum çalışması kullanılmıştır. Araştırmaya bir devlet ortaokulunda öğrenim gören 117 kız, 109 erkek olmak üzere toplam 226 öğrenci dâhil edilmiştir. Öğrencilerin problem çözme becerilerini belirlemek amacıyla ön test ve son test olarak yapılandırılmamış problemlerin çözümüne yönelik geliştirilen problem çözme becerisi ölçeği uygulanmıştır. Çalışma sırasında, öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımlarını belirlemek için, 27 öğrenciye ölçekteki problem çözme adımlarına ilişkin görüşme soruları yöneltilmiştir. Öğrenciler Scratch ile oyun tasarımında ağırlıklı olarak sırasıyla; oyunu zihinlerinde gerçek hayat şartlarına uygun kurgulamaktadırlar, kurguladıkları oyunun tasarımını metne dökmektedirler, tasarıma uygun olarak kod bloklarını yerleştirmektedirler, programı çalıştırıp bir sonraki aşamaya geçmektedirler ve hata durumunda ilgili bölümü tasarıma göre kontrol

etmektedirler. Ortaya çıkan sonuçlara göre; öğrencilerin problem çözme becerileri cinsiyete ve sınıf seviyesine göre farklılık göstermemekle birlikte, ölçeğin birinci ve ikinci ölçümleri arasında istatistiksel olarak anlamlı bir fark bulunmuştur. Buna göre sonuçlar Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerinde orta düzeyde bir etkisi olduğunu göstermektedir ( $d=0,39$ ). Buradan hareketle, problem çözme becerilerine sahip öğrenciler yetiştirmek için alternatif bir metot olarak Scratch ile programlama öğretiminde oyun tasarımının gerçekleştirilmesi önerilmektedir. Öğrencilerin problem çözme adımlarını incelediğimizde Scratch ortamında yapılandırılmamış problemleri çözebilmeleri ve oyun projesini başarı ile tamamlamaları için tasarım odaklı yaklaşım uygulanması önerilmektedir.

Taşdemir'in (2017) yapmış olduğu tarama modelindeki betimsel araştırmada, özel yetenekli öğrencilerin teknoloji kullanarak kendi kendine öğrenmeleri ile bilgisayara yönelik tutumları arasındaki ilişki incelenmiştir. Araştırmaya Ankara ilinde destek eğitime devam eden 322 özel yetenekli öğrenci katılmıştır. Veri toplama aracı olarak "Ortaokul ve Lise Öğrencileri İçin Bilgisayara Yönelik Tutum Ölçeği", "Çocuklar İçin Teknolojiyle Kendi Kendine Öğrenme Ölçeği" ve demografik bilgiler için bir bilgi formu kullanılmıştır. Çalışma sonuçları, özel yetenekli öğrencilerin teknoloji kullanarak kendi kendine öğrenme düzeyleri ile bilgisayara yönelik tutumları arasında pozitif bir ilişki olduğunu göstermektedir. Ayrıca teknoloji ile kendi kendine öğrenme düzeyleri de bilgisayara yönelik tutumları da ortalamanın üzerinde bulunmuştur.

Yapmış oldukları çalışmada programlama öğretimi için Scratch kullanımına yönelik öğrenci görüşlerini inceleyen Özcan ve diğerleri (2017), Scratch ortamının programlama öğretimi için yararlı bir araç olduğunu, eğlenceli ve kolay anlaşılır bir yapıda olduğu, programlama mantığını geliştirmeye ve algoritmaları kavramaya yardımcı olduğu, yaratıcılığı geliştirdiği gibi öğrenci görüşleri tespit ettiklerini bildirmişlerdir.

Scratch aracının programlama öğretimindeki yerine ilişkin bir doküman incelemesi çalışması yapan Çatlak ve diğerleri (2015), bu kapsamda ulaşılan 53 makaleden çalışma kriterlerini sağlayan 32 tanesi çalışmaya dâhil etmişlerdir. Araştırmanın bulgularına göre, programlama öğretiminde Scratch aracının etkili olduğu ve programlama öğrenme sürecini, kod bloklarının sürükle - bırak yöntemiyle rahatça

kullanımı sayesinde daha kolay anlaşılır hale getirdiği belirtilmiştir. Ayrıca bu aracın özünde oyun teması barındırması sayesinde öğrenciler için ilginç, kolay kullanılabilen ve merak uyandırıcı bir ortam sunduğunu, programlamanın temellerini öğretmek için etkili bir araç olduğunu, öğrencilerde yaratıcı düşünme becerilerini geliştirdiği ve öğrenmeye ilişkin isteklerini arttırdığını belirtmişlerdir.

Yükseltürk ve Altıok (2016) 169 Bilişim Teknolojileri Öğretmen adayı ile çalışmıştır. Öğrencilerden bir eğitsel oyun konusu seçip, Scratch programlama aracını kullanarak o konuda eğitsel bir oyun oluşturmalarını istemişlerdir. Çalışmada Scratch aracının seçilme nedeni olarak öğretmen adaylarının programlama öğretiminde kullanılan güncel yöntemler ve araçlara ilişkin fikir sahibi olmaları gerekliliğinin yanı sıra; programlama sürecinin karmaşık ve zor olmasında rağmen, Scratch aracının başlangıç seviyesindekiler için kolay, ilgi çekici, eğlenceli ve heyecan verici bir ortam sunarak programlama öğretimi kolaylaştırması belirtilmiştir. Sonuç olarak hazırlanan oyunlar belirlenen kriterlere göre değerlendirilince, oyunların “Orta veya İyi kalitede” oldukları görülmüştür. Ayrıca öğretmen adaylarının yaptıkları öz-değerlendirme sonuçlarına göre, öğretmen adayları “bir projeyi tamamlayarak kendine ait bir ürün ortaya koymanın çok güzel bir duygu” olduğunu belirtmişlerdir.

Gülbahar ve Kalelioğlu (2014) çalışmalarında Scratch programlama öğretimi aracının, öğrencilerin problem çözme becerilerine etkisini araştırmışlardır. İlköğretim 5.sınıfta bulunan 49 öğrencinin katıldığı çalışma sonucunda, öğrencilerin problem çözme becerilerinde anlamlı bir değişim gözlenmemiştir. Ancak, öğrencilerin Scratch kullanarak programlama öğrenmeye ilişkin düşünceleri incelendiğinde, öğrenciler Scratch uygulamasını kolay kullandıklarını, programlama yapmayı sevdiğini ve bir program yazmak konusunda istekli olduklarını belirtmişlerdir.

Hava'nın (2012), 4. Sınıfta bulunan 34 öğrenciyle öntest-sontest kontrol gruplu deneysel modele göre yaptığı çalışmada öğrencilere “eğitsel bilgisayar oyunu tasarlama” etkinliği yaptırılarak bu uygulamanın, öğrencilerin akademik başarılarına etkisi araştırılmıştır. Kontrol grubu öğrencileri eğitsel oyun oynarken deney grubu öğrencileri eğitsel bir oyun tasarlamışlardır. Araştırma sonucunda, deney grubu ile kontrol grubunun akademik başarıları arasında anlamlı bir fark ortaya çıkmamış, ancak her iki gruptaki öğrencilerin uygulama sonrasında akademik başarılarının arttığı tespit edilmiştir.

Programlama öğretiminin öğrencilerin güdülenmelerine ve programlama becerilerine etkisini inceleyen bir çalışmada; 2D Programmable PyGame, Visual Basic Express ve Scratch araçlarının her biri bir grupta kullanılmıştır. 591 ortaokul öğrencisinin katıldığı çalışmanın analizlerine göre; 2D Programmable PyGame ve Scratch kullanan öğrenciler programlama becerisi ve güdülenme açısından Visual Basic Express kullanan öğrencilere göre daha iyi bir performans sergilemişlerdir. Ayrıca her 3 aracın da ortaokul seviyesinde programlama öğretimi amacıyla kullanılabilir uygun ortamlar oldukları belirtilmiştir (Osman vd., 2012).

6. sınıf seviyesindeki 60 öğrenci ile yapılan bir diğer çalışmada Scratch kullanılmasının programlama öğretimi yapılmasının algoritma öğretiminde etkililiği incelenmiştir. Araştırmada deney grubundaki öğrenciler bir oyun tasarlayarak algoritma öğrenimi gerçekleştirirken, kontrol grubundaki öğrenciler için düz anlatım yöntemi kullanılmıştır. Çalışma sonunda deney ve kontrol grubu arasında akademik başarı açısından anlamlı bir farklılık bulunmazken, her iki gruptaki öğrencilerin uygulama sonrasında başarılarının arttığı tespit edilmiştir (Tekerek ve Altan, 2014).

Solmaz'ın (2014) yaptığı çalışmada; programlama öğretimi amacıyla Alice ortamının kullanılmasının, üniversite öğrencilerinin problem çözme becerileri, eleştirel düşünme eğilimleri, üstbilişsel farkındalık düzeyleri ve ders başarıları üzerine etkisini incelemiştir. Karma yöntem kullanılan çalışmaya “Gazi Üniversitesi Gazi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı'nda” öğrenim gören 39 ikinci sınıf öğrencisi katılmıştır. PHP programlama dilinde temel programlama kavramları öğretilmek istenen deney grubundaki öğrencilerle 7 hafta süresince Alice yazılımı kullanılırken, kontrol grubundaki öğrenciler için yardımcı bir araç kullanılmamıştır. Uygulama sonunda deney ve kontrol grupları arasında akademik başarı açısından anlamlı bir farklılık belirlenmemiştir. Bu sonuca göre; Alice aracı kullanılarak PHP programlama dili öğretimi yapılmasının ders başarısı üzerinde istatistiksel olarak anlamlı bir etkisi olmadığı görülmüştür. Ayrıca, problem çözme becerisi, eleştirel düşünme eğilimi ve üstbilişsel farkındalık değişkenlerine ait analiz sonuçlarına göre deney ve kontrol grubundaki öğrenciler arasında anlamlı bir farklılık gözlenmemiştir. Sonuç olarak; programlama dili öğretiminde kullanılan görsel öğrenme çevresinin değil, kullanılan yöntemin eleştirel düşünme üzerinde etkili olduğu; programlama öğretiminin, görsel bir dilin kullanılması veya kullanılmaması, ya da

eğitimde kullanılan ortamın türüne göre öğrencilerin problem çözme becerileri üzerinde çok farklı etkilere yol açabileceği ortaya çıkmıştır.

Genç ve Karakuş (2011), 109 öğrenci ile yapmış oldukları çalışmada öğrencilerden Scratch ortamında bir oyun tasarımlarını istemişlerdir. Uygulama sonunda; öğrencilerin %79' u Scratch programının kullanımını kolay ve eğlenceli bulduklarını, aracın keyifli bir çalışma ortamı sunduğunu ve bu araç ile çalışmaktan hoşlandıklarını belirtmişlerdir. Öğrencilerin sadece %20'si Scratch ile programlama yapmaktan hoşlanmadıklarını belirtirken, %73'ü ise Scratch'in programlama öğretiminde kullanılabileceğini, Scratch ile programlamanın temel kavramlarının diğer birçok ortama göre daha kolay olduğunu belirtmişlerdir.

### 2.5.2. Yurt Dışında Yapılan Araştırmalar

Öğrencilere programlama öğretmek ve programlamaya yönelik öğrenmeye ilişkin isteklerini incelemek amacıyla Scratch yazılımını kullanan Ouahbi ve arkadaşları (2015), 69 lise öğrencisi ile çalışmışlardır. 3 gruba ayrılan öğrencilerden birinci gruptakiler, öğretmenleri rehberliğinde Scratch kullanarak oyunlaştırma yöntemiyle programlama öğrenmişler, ikinci gruptakiler, bir öğretmenden yüz yüze eğitim almışlar, üçüncü gruptakiler ise, yüz yüze eğitim almışlar fakat derslere farklı öğretmenler gelmiştir. Öğrencilerin %95'inin programlama konusunda herhangi bir ön bilgiye sahip olmadığı çalışmanın sonucunda, Scratch kullanılan gruptaki öğrencilerin güdülenmeleri ve başarıları pozitif olarak farklılaşırken, diğer iki gruptaki öğrenciler için programlamanın sıkıcı, zor ve monoton bir süreç olduğu izlenmiştir.

İlköğretim öğrencileri ile çalışan Shin ve Park (2014) araştırmalarında Scratch kullanılarak yapılan programlama öğretimi uygulamasının öğrencilerin problem çözme becerileri üzerine etkisini incelemişlerdir. Toplam 46 öğrenci ile yapılan çalışma sonucunda Scratch kullanılarak yapılan programlama öğretiminin, öğrencilerin problem çözme becerilerine olumlu etkisi olduğunu belirtmişlerdir.

Maloney ve arkadaşlarının (2008) yaptıkları bir çalışmada, blok temelli ve görsel bir programlama dili olması nedeniyle Scratch, programlamaya yeni başlayan öğrenciler için kolaylaştırıcı bir ortam olarak kullanılmıştır. 8-18 yaş arası gençler, 18 ay süresince okuldan sonra gittikleri bir kulüp merkezinde, deneyimli bir danışmanın rehberliği olmadan ve hiçbir eğitim almadan 536 Scratch projesi geliştirmişlerdir. Çalışmada, bu projelerin incelenerek katılımcıların kullanabildikleri temel programlama kavramlarının

ortaya çıkarılması hedeflenmiş, ayrıca çalışmaya katılan kulüp üyesi öğrencilerin süreç boyunca programlamaya ve Scratch'e ilişkin düşünceleri gözlemlenmiştir. Böylece gençlerin programlama öğrenmek için neden başka bir yazılım değil de Scratch'i seçtikleri tespit edilmeye çalışılmıştır. Öğrenciler "bilgisayar programlaması senin için ne ifade ediyor?" sorusunu genellikle "Bilgisayar programlaması da nedir?" şeklinde cevaplamışlardır. Bu cevaptan yola çıkarak araştırmacılar, öğrencilerin programlama ile Scratch arasında bir ilişki kurmadıkları, Scratch'i bir programlama aracı olarak değil bir oyun oluşturma aracı olarak gördükleri için, Scratch'in bu denli popüler olmuş olabileceği şeklinde yorumlamışlardır. Kulüp etkinliklerine katılan bu öğrencilerin kendi başlarına kullanıcı etkileşimi, döngüler, koşullu ifadeler, iletişim ve senkronizasyon kavramlarını gösteren komutları keşfedip ve kullandıkları görülmüştür. Değişkenler, Boolean mantıksal operatörleri ve rasgele sayılar gibi daha zor keşfedilebilen kavramların kullanılması daha az yaygındı, ancak zamanla arttı. Bu bulguların, resmi talimatların olmadığı ve mentorların önceden programlama tecrübesinin olmadığı bir çalışma ortamı olması nedeniyle özellikle şaşırtıcı olduğu belirtilmiştir. Rasgele numara üretme blokları ile değişkenlerin projelerde daha az kullanılmakla birlikte, proje üzerinde çalışma süresi arttıkça bunların kullanımının da arttığı tespit edilmiştir. Bu durumun, öğrencilere resmi bir eğitimin verilmemesi ya da programlama tecrübesi olmayan danışmanlardan kaynaklanabileceğini belirtmişlerdir. Diğer yandan araştırmacılar, bu çalışmadaki bulgulara göre bu yaş grubundaki öğrencilerin performansının, merkeze uzak bölgelerdeki öğrencilerin, programlamanın temellerini ve programlama sırasında kullanılan pek çok önemli ifadeleri Scratch ortamını kullanarak kendi kendilerine öğrenebileceklerini göstermesi bakımında oldukça önemli olduğunu belirtmişlerdir (Maloney ve diğerleri, 2008).

Scratch kullanılarak yapılan programlama öğretiminin öğrencilerin güdülenmelerine, tutumlarına, programlama becerilerine ve üst düzey düşünme becerilerine olumlu yönde etkisi olduğu, ayrıca programlama öğretimi için uygun bir araç olduğu ifade edilmektedir (Malan ve Letiner, 2007).

6. sınıfta eğitim gören 60 öğrenci ile çalışan Nam ve diğerleri (2010) yapmış oldukları araştırmada, öğrencilere Scratch programlama öğretimi aracı kullanılarak programlama eğitimi vermenin, öğrencilerin problem çözme becerileri ve programlama başarılarına etkisini incelemişlerdir. Deney grubunda bulunan öğrenciler Scratch kullanarak eğitim alırken, kontrol grubunda bulunan öğrenciler geleneksel yöntemlerle

eđitim almıřlardır. Toplam 4 hafta sren alıřma sonucunda arařtırmacılar, deney grubundaki đrencilerin problem özme becerilerinde ve programlama bařarılarında pozitif olarak bir deęiřim tespit ettiklerini belirtmiřlerdir.

Scratch programlama dili kullanılarak hazırlanan bir program ile yabancı dil eđitiminin etkililięine iliřkin bir alıřmada, Scratch programlama ortamında İngilizce ve Tamil dilini đretmeye yönelik bir program oluřturulmuřtur. Arařtırmanın sonucunda, ders ortamı ve test ortamı olmak üzere iki bölümden oluřan programın yabancı dil eđitiminde kullanılabileceęi bulgusuna ulařtıklarını belirtmiřlerdir (Sanjanaashree, Anand Kumar ve Soman 2014). Buradan hareketle, Scratch kullanılarak hazırlanan programların yabancı dil derslerinde etkili bir řekilde kullanılabileceęi söylenebilir.

Scratch programının đrencilere bilgisayar bilimi kavramlarını đretmede bařarılı olup olmadıęını arařtıran bir alıřmada 204 ortaokul đrencisi ve sekiz đretmen yer almıřtır. 20 hafta sren alıřmada haftada iki ders saati ders iřlenmiřtir. Gözlem ve görüřmeler ile nitel veriler toplanmıřtır. đrencilerin tekrarlı yapılar, deęiřkenler gibi önemli bilgisayar bilimi kavramlarını đrendikleri tespit edilmiřtir (Meerbaum-Salant, Armoni ve Ben-Ari (2013).

Scratch ile yapılan bir dięer alıřmada, proje tabanlı yöntem kullanılarak Scratch ortamında oluřturulan problem özme senaryoları ieren bir program yardımıyla đrencilere matematik konuları đretilmiřtir. Bu uygulamanın, đrencilerin đrenmeye yönelik tutumlarını, problem özme becerilerini ve güdülenmelerini nasıl etkiledięi incelenmiřtir. Matematik dersi aısından 43 normal seviyede ve 48 bařarılı olmak üzere toplam 91 ortaokul đrencisi ile alıřılmıřtır. alıřmanın sonuçlarına göre; Scratch ile yapılan đrenme etkinliklerinin her iki gruptaki đrencilerin akademik bařarılarını anlamlı düzeyde artırdıęı, ancak, matematikte iyi olan đrencilerin tutumlar, problem özme becerileri ve güdülenme aısından dięer đrencilere göre daha iyi performans sergiledikleri sonucuna ulařılmıřtır (Wang ve dięerleri, 2014).

Scratch ve Karel programlama araçlarını karřılařtıran Ruf ve arkadařları (2014) ortaokul đrencileri ile alıřmıřlardır. Programlama araçlarından her biri bir sınıfta kullanılarak đrencilere programlama đretilmeye alıřılmıřtır. alıřma sonucunda, Scratch kullanılan gruptaki đrencilerin đrenmeye iliřkin isteklerinin ve programlama becerilerinin, Karel kullanılan đrencilere nispeten daha iyi olduęu belirlenmiřtir.



Adams ve Webster (2012)'ın araştırmasında, 2008-2011 yılları arasındaki Scratch kampına katılan öğrenciler, video oluşturmaktan ziyade oyun oluşturmaya tercih etmişlerdir. Kamp temel programlama kavramlarını ortaokul ve lise öğrencilerine tanıtmak amacıyla tasarlanmıştır. 322 ortaokul ve lise öğrencisinin 300 farklı Scratch projesindeki kod bloklarının istatistiğini çıkartarak, t-test ile analiz etmişlerdir. Değişkenler, döngü deyimleri, koşul ifadeleri vb. temel öğelerin nasıl kullanılacağı öğrencilere gösterilmiştir. Öğrencilerin hikâye oluştururken koşul ve döngü ifadeleri kullanmasına ihtiyaç olmamakla birlikte oyun tasarımları sürecinde koşul ifadeleri ve döngüleri kullanmaları gerekmektedir. Dolayısıyla oyun tasarımı projeleri, öğrencileri programlama deyimlerini öğrenmeye zorlamaktadır. Yaz kampında 209 Alice projesi, 103 Scratch oyun projesi, 10 Scratch video projesi yapılmıştır. Scratch oyun projelerinde daha fazla değişken, kod bloğu, koşullu ifade, döngü deyimleri kullanıldığı görülmüştür. Hikâyelerde en çok diyaloglar kullanılmaktadır. Videolarda neredeyse oyunlar kadar döngüler kullanılmış ama değişkenler ve koşullu ifadeler çok daha az kullanılmıştır.

Adams (2010) araştırmasında haftanın 5 günü 9:30-14:30 arası 30 lisans öğrencisi ile Scratch çalışması gerçekleştirmiştir. Öğrenciler olumlu bir bilgisayar deneyimi yaşadıklarını, programlamada temel fikirleri öğrendiklerini, bilgisayar alanında kariyer yapma isteği duyduklarını ifade etmişlerdir. Katılımcılar Scratch programının zorluğunu orta olarak nitelemişlerdir. Öğrencilerin %93'ü Scratch programı ile oyun, geri kalanlar ise video oluşturmuşlardır. Oyunlarda koşul ve mantık ifadeleri yoğun olarak kullanılmıştır. Videolarda ise hiçbir koşul ifadesi kullanılmamıştır. Kamp öğrenciler için olumlu geçmiştir ve programlama hakkında daha fazla bilgi edinmek istemişlerdir.

Lewis (2010) Logo ve Scratch programlarının tutum ve öğrenme çıktılarını karşılaştırmıştır. Öğrenciler iki gruba ayrılarak, ilk grup ilk 6 gün Scratch ile devamında Logo ile ikinci grup ilk 6 gün Logo devamında Scratch ile çalışmıştır. Öğrencilerin ilk altı gündeki algıları, hedefleri ve öğrenme çıktıları araştırılmıştır. İkinci ve beşinci günde yazılı değerlendirme yapılmıştır. Anketler yedinci günde doldurulmuştur. İlk grup 26, ikinci grup ise 24 6. sınıf öğrencisinden oluşmaktadır. Scratch programını öğrenen öğrenciler Logo programını öğrenen öğrencilere göre koşullu ifadeleri yorumlamakta daha başarılı olmuşlardır. Bu sonuç Scratch programının kompüsyonel kavramları öğretmekte başarılı olduğunu göstermektedir (Adams, 2010; Adams ve

Webster, 2012; Claypool, 2013). Bununla birlikte Logo öğrenen öğrencilerin programlama yeteneğine güven düzeyleri Scratch programını öğrenen öğrencilere göre daha yüksek çıkmıştır.

Başka bir araştırmada Nascimento vd. (2010), Brezilyalı lise öğrencileri için programlama dili öğretiminde bir uzaktan eğitim deneyimi hazırlamışlardır. Biri özel okul, biri devlet okulu olmak üzere iki okuldan seçilen 15-18 yaş aralığındaki 182 öğrenciden 40 öğrenci seçilmiş, bu öğrencilere bir ay boyunca Python programlama dili öğretilmiştir. Her konu bir deste öğretilmiş, her ders; videolar, komut dosyaları, alıştırmalar ve okuma referanslarını içermiştir. Sonuçlar dersi tamamlayan öğrencilerin %94'ünün sınavlarda başarılı olduğunu göstermiştir. Bununla birlikte dersi tamamlamayan öğrencilerin oranı da yüksektir. Bunun sebebi olarak da öğrencilerin genç olması ve öz yönetimlerinin zayıf olması gösterilmiştir.

Nascimento ve diğerleri (2010) Brezilyalı lise öğrencileri için programlama dili öğretiminde uzaktan öğrenme deneyimi geliştirmiştir. Özel okul ve devlet okulu olmak üzere iki okuldan seçilen, yaşları 15 ile 18 arasında değişen 182 öğrenciden 40 öğrenci seçilerek bir ay boyunca programlama dili olarak Python öğretildi. Her derste bir konu videolar, senaryolar, alıştırmalar ve okuma referansları ile işlenmiştir. Sonuçlar öğrencilerin %94'ünün kursu başarıyla tamamladığını göstermiştir. Ancak, öğrencilerin genç olmaları ve özyönetim eksiklikleri nedeniyle kursu tamamlamayan öğrencilerin oranının da yüksek olduğu bildirilmiştir.

Malan ve Leitner (2007) araştırmalarında, lisenin yaz okulunda bilgisayar bilimi dersinde ilk defa öğrencilere Scratch programı göstermişlerdir. Amaç daha sonra Java programına geçişi kolaylaştırmaktır. Öğrencilerin %76'sı sonraki Java deneyimlerinde Scratch ile çalışmalarının faydalı olduğunu ifade etmişlerdir, %16'sı ise pozitif veya negatif herhangi bir geribildirimde bulunmamıştır. Öğrencilere daha detaylı incelendiğinde geribildirim vermeyen öğrencilerin daha öncesinde programlama tecrübesi olan öğrenciler oldukları, faydalı oldu diyenlerin ise programlama tecrübesi olmayan öğrenciler oldukları belirtilmiştir.

Calder (2010) altıncı sınıf 26 öğrenci ile Scratch programlama dilinde çalışırken matematiksel düşünme yollarını araştırmıştır. Araştırma sürecinin alt amaçları problem çözme sürecinde ilişkisel düşünme, durumu yorumlama ve değerlendirme için mantık ve sebep-sonuç ilişkisi kurma aşamalarını araştırmaktır. Araştırma nitel araştırma örnek olay incelemesidir. Öğrencilerin süreç ve yansımalar hakkındaki web günceleri, öğrenci

ve öğretmen görüşmeleri ve sınıf gözlemleri içerik analizi ile analiz edilmiştir. Araştırmanın sonuçları Scratch yazılımının matematiksel kavramları keşfetmek için değerli ve motive edici bir programlama ortamı sağladığını, yaratıcı problem çözme sürecini ve matematiksel düşünmeyi kolaylaştırdığını ve problem çözme için ilgi çekici ve nispeten kullanımı kolay ve motive edici bir program olduğunu göstermiştir.

Brown ve diğerleri (2008) ortaokul öğrencilerine problem çözme becerilerini öğretmek için Scratch kullanımını araştırmışlardır. 113 ortaokul 5. ve 6. sınıf öğrencisine bir ay süresince haftada 45 dk. olmak üzere toplamda dört Scratch dersi verilmiştir. Deney grubu öğrencilerinin problem çözme becerileri kontrol grubu öğrencilerine göre daha yüksek çıkmıştır. Araştırmacılar, bir öğretim aracı olarak Scratch'in problem çözme için matematiksel akıl yürütme becerilerini uygulamada etkili bir yol olacağını, bu nedenle Scratch'in akademik performansı arttırmak için kullanılabileceğini ve Scratch ile işlenen derslerin öğrencilerin problem çözme becerilerini öğrenmesinde yararlı olabileceğini önermişlerdir.

Hsu (2014), programlama öğrenmek amacıyla Scratch ortamında bir oyun tasarlayan ilköğretim öğrencilerinin, programlamanın temel mantığını kavrama durumları ve programlamanın temel yapılarını oluşturdukları oyunda kullanma durumlarını cinsiyet değişkeni açısından incelemiştir. 46 öğrenci ile yapılan çalışmanın analiz sonuçlarına göre; programlama yapılarını anlama durumlarının cinsiyete göre farklılaşmadığı, ancak kız öğrencilerin döngü yapılarını daha iyi anladıkları belirlenmiştir. Ayrıca öğrencilerin tasarladıkları oyunlarda, benzer oranda blok kullandıkları, sahne, kostüm, karakter benzeri yapıların kullanımında da cinsiyet açısından bir anlamlı bir farklılaşma olmadığı sonucuna ulaşılmıştır.

Öğretmenlere Scratch kullanarak programlama öğretimi yapılmasının programlama becerilerine nasıl bir etkisi olacağını inceleyen Chiu (2014), çalışmasında öğretmenlerden Scratch ortamında bir oyun tasarlama istemiştir. Uygulama sonunda öğretmenlerin programlama becerilerinde anlamlı bir farklılık bulunmamakla birlikte öğretmenler, Scratch'in programlama öğretimi amacıyla kullanılacak uygun bir ortam olduğunu, sürükle bırak yöntemiyle blokları kolayca kullandıklarını ve bu sürecin eğlenceli olduğunu ifade etmişlerdir.

Lise öğrencileriyle yapılan bir çalışmada, Scratch ve App Inventor programlama araçlarının öğrencilerin programlama öğrenmelerine etkisini inceleyen Nikou ve

Economides (2014), uygulama sonunda her iki aracın da öğrencilerin programlama öğrenmeye ilişkin isteklerini artırdığını belirtmişlerdir.

Scratch ortamında programlama yapmanın öğrencilerin mantıksal matematiksel düşünme becerileri ile problem çözme becerileri üzerine etkisi üzerine yapılan bir çalışmada 11-13 yaşlarındaki öğrenciler 3 ay süreyle Scratch ortamında çalışmışlardır. Uygulama sonunda elde edilen bulgulara göre; Scratch'te koşul, karşılaştırma ve döngü operatörlerini kullanarak etkinlikler yapan öğrencilerin mantıksal matematiksel düşünme becerileri ile problem çözme becerilerinin geliştiği bildirilmiştir (Begosso ve Da Silva, 2013).

Ortaokul öğrencilerine Scratch kullanarak dijital oyun ve hikâye tasarlatılan Burke'nin çalışmasında (2012), öğrencilerin teknoloji okuryazarlığını artırmak, programlama becerilerini geliştirmek, dijital hikâye tasarımı ile genel okuryazarlık çalışmalarını farklılaştırmak amaçlanmıştır. Çalışmanın sonuçlarına göre; öğrencilerin hikâye oluşturma ve eş zamanlama gibi dijital hikâye becerilerinin geliştiği ve döngü, karar verme gibi programlama yapılarını öğrendikleri tespit edilmiştir.

Ortaokul öğrencileriyle çalışan Fadjo (2012), Scratch kullanımının öğrencilerin bilgi işlemsel düşünme becerileri üzerine etkisini araştırmak amacıyla yaptığı çalışmada birinci gruptaki 56 öğrencisi ile Scratch kullanarak dijital hikâye tasarımı yaptırılırken, ikinci gruptaki 78 öğrenci Scratch kullanarak oyun tasarlamışlardır. Bulgulara göre; öğrenciler olaylar, koşullu ifadeler gibi temel programlama kavramlarını öğrenerek bilgi işlemsel becerilerini geliştirmişlerdir. Sonuç olarak, Scratch'in öğrencilerin bilgi işlemsel düşünme becerilerini geliştirmede etkili bir araç olarak kullanılabileceğini vurgulanmıştır.

5. sınıf öğrencileri ile bir çalışma yapan Baytak ve Land (2011), öğrencilerin programlama becerilerini geliştirmek için Scratch ortamında onlara oyun tasarımı etkinlikleri yaptırmışlardır. Uygulama sonunda öğrenciler; tasarım, planlama, test etme ve paylaşma gibi fonksiyonların kullanarak çeşitli oyunlar tasarlayabilmişlerdir. Öğrencilerin blok tabanlı görsel programlama araçları ile programlama kavramlarını daha iyi öğrendikleri ve bu sayede kolaylıkla çeşitli seviyelerde bilgisayar oyunları oluşturabileceklerinin anlaşıldığı belirtilmiştir.

Bu bölümde ele alınan çok sayıda ulusal ve uluslararası çalışmada bir blok tabanlı görsel programlama öğretimi aracı olarak Scratch ele alınmıştır. Bu çalışma

sonularına gre Scratch aracı, oėu kere ėrencilerin bilgi iřlemsele dřnme becerileri, problem zme becerileri, st dzey dřnme becerileri gibi zihinsel becerilerini olumlu anlamda etkileyerek geliřtirmiřtir. Ayrıca ėrencilerin dřncelerini inceleyen alıřmalara gre ise erken yařtaki ėrencilerden ileri yařlardaki ėretmenlere dek hemen her seviyede insan, Scratch ortamında alıřmaktan zevk aldıklarını, basit ve kolay anlaşılır bir yapıda olması nedeniyle programlamanın temel kavramlarını kolaylıkla anlayabildiklerini belirtmiřlerdir. Programlama ve biliřim dersleri dıřında diėer derslerde de kullanılabileėi anlařılan Scratch'in, farklı disiplinlere ait konuların ėretilmesinde de bařarıyla yararlanılabilecek web tabanlı, proje alıřmasına uygun bir ortam olduėu anlařılmaktadır.



## BÖLÜM III

### YÖNTEM

Bu bölümde arařtırmada yapılan deneysel iřlem sürecindeki uygulamalar, arařtırmanın modeli, alıřma grubu, verilerin toplanmasında kullanılan araçlar ile verilerin toplama süreci ile verilerin analizine dair açıklamalar bulunmaktadır.

#### 3.1. Arařtırmanın Modeli

Bu alıřmada, blok tabanlı programlama öđretiminin, ortaokul altıncı sınıf öđrencilerinin problem özme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisinin belirlenmesi amacıyla, ön test-son test kontrol gruplu (eřitlenmemiř kontrol gruplu model) yarı deneysel desen kullanılmıřtır. Deneysel arařtırmalar deđiřkenlerin kontrol altına alınabildiđi ve etkilerinin incelendiđi bilimsel yöntemler içerisinde en kesin sonuçların elde edilebildiđi arařtırmalardır. Yarı deneysel desende kullanılacak gruplara rastgele seçkiyle denek atamasının yapılamaması, deneysel desen ile arasındaki esas farktır. Ancak, grupların benzer niteliklere sahip katılımcılardan oluşturulmasına önem verilir (Karasar, 2005). alıřmada yer alan her iki sınıf öđrencilerinin deđiřme imkânı olmadıđından, sınıflardan biri deney grubu diđerisi ise kontrol grubu olarak rastgele seçilmiřtir. Her iki gruba deneysel iřlem öncesi ve sonrasında “ocuklar için Problem özme Envanteri” ve “Bilgisayara Yönelik Tutum Öleđi” uygulanmıřtır. Deney grubunda Scratch kullanılarak Blok tabanlı programlama ile eđitim verilirken, kontrol grubunda ise uygulamadaki mevcut öđretim programı uygulanmıřtır. alıřmanın bađımlı deđiřkenleri bilgisayara yönelik tutum ve problem özme becerisi; bađımsız deđiřkeni ise Scratch ile programlama öđretimi yöntemidir.

**Tablo 1.** Araştırma Modeli

Grup	Öntest	Yapılan Öğretim	Sontest
Deney	ÇPÇE*, BYTÖ**	SCRATCH ile Programlama Öğretimi	ÇPÇE*, BYTÖ**
Kontrol	ÇPÇE*, BYTÖ**	Uygulamadaki Mevcut Öğretim	ÇPÇE*, BYTÖ**

\*ÇPÇE: Çocuklar için Problem Çözme Envanteri

\*\*BYTÖ: Bilgisayara Yönelik Tutum Ölçeği

### 3.2. Çalışma Grubu

Bu araştırma kapsamında, 2017-2018 eğitim öğretim yılında Kayseri ili Kocasinan ilçesindeki bir ortaokulunda bilişim teknolojileri ve yazılım dersi kapsamındaki 6. sınıflarında bulunan %48'i kız (n=59), %52'si erkek (n=64) olmak üzere toplam 123 öğrenci mevcuttur. Çalışma grubu oluşturulmadan önce sınıfların mevcudu ve sınıfların hazır bulunuşluk düzeyleri dikkate alınmıştır. Çalışma grubu oluşturulurken okulda bulunan 4 adet 6. Sınıf şubesine “Bilgisayara Yönelik Tutum Ölçeği” ile “Çocuklar için Problem Çözme Envanteri” öntest olarak uygulanmıştır. Elde edilen veriler bağımsız örneklem t-testi ile analiz edilmiş, öntest puanları düşük ve birbirine yakın olan iki sınıf olan 6-K ve 6-H sınıfı çalışma grubu olarak seçilmiştir. Daha sonra yansız atama yoluyla ile bu sınıflardan biri deney (6-K) diğeri ise kontrol grubu (6-H) olarak belirlenmiştir. Araştırmada, çalışma grubunun problemle ilgili olarak belirlenen niteliklere sahip kişiler, olaylar, nesnelere ya da durumlardan oluşturulmasını sağlayan ölçüt örnekleme yaklaşımı kullanılmıştır (Büyüköztürk ve diğerleri, 2010, p. 91). Problem çözme becerileri ve bilgisayara yönelik tutum ortalama puanları standart puanlara dönüştürülerek z puanları -3 ile 0 arasında olan öğrenciler deney ve kontrol gruplarının belirlenmesi amacıyla kullanılmıştır. Ayrıca 6. sınıf öğrencileri daha önce programlama dersi almadığından dolayı programlama konusundaki önbilgileri eşit varsayılmıştır. Tablo 2’de çalışmada yer alan katılımcıların demografik özellikleri yer almaktadır.

**Tablo 2.** Katılımcıların Demografik Özellikleri

Cinsiyet	Grup (N=49)	Deney (N=24)	Kontrol (N=25)
Kız	f	10	11
	%	42	44
Erkek	f	14	14
	%	58	56

Tablo 2'de görüldüğü gibi kontrol grubunun %44'ü kız öğrencilerden, %56'sı ise erkek öğrencilerden oluşurken, deney grubunun %42'si kız öğrencilerden, %58'i ise erkek öğrencilerden oluşmaktadır.

### 3.3. Veri Toplama Araçları, Güvenilirlik ve Geçerlik Çalışmaları

Blok tabanlı programlama öğretiminin, ortaokulu altıncı sınıf öğrencilerinin problem çözme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisinin belirlenmesi amacıyla, “Bilgisayara Yönelik Tutum Ölçeği (Ek-3)” ile “Çocuklar için Problem Çözme Envanteri (Ek-2)” veri toplama aracı olarak kullanılmıştır.

#### 3.3.1. Bilgisayara Yönelik Tutum Ölçeği

Bilgisayara Yönelik Tutum Ölçeği, Aşkar ve Orçan (1987) tarafından geliştirilmiş 14 olumlu ve 10 olumsuz ifadenin bulunduğu toplamda 24 maddelik 5'li Likert tipli tek boyutlu bir ölçektir. Ölçekteki ifadeler 1= “Katılmıyorum”, 2= “Az Katılıyorum”, 3= “Orta Düzeyde Katılıyorum”, 4= “Çoğunlukla Katılıyorum”, 5= “Tamamen Katılıyorum” şeklinde puanlandırılmıştır. Olumlu maddeler “tamamen katılıyorum” seçeneğinden başlayarak 5, 4, 3, 2, 1 olarak puanlanmış, olumsuz maddeler ise 1,2,3,4,5 şeklinde kodlanmıştır. Ölçeğin Cronbach alpha güvenilirlik katsayısı 0.89 olarak hesaplanmış, bu çalışmada ise bu değer 0.83 olarak tespit edilmiştir. Ölçekten en yüksek 120 puan, en düşük 24 puan alınabilmektedir. 72'den yüksek puanlar pozitif tutumları, 48'den düşük puanlar ise negatif tutumları belirlemektedir.



### 3.3.2. Çocuklar için Problem Çözme Envanteri

Serin, Bulut Serin ve Saygılı (2010) tarafından geliştirilen Çocuklar için Problem Çözme Envanteri, 3 alt boyutu (Problem çözme becerisine güven:12 olumlu madde, Özdenetim: 7 olumsuz madde ve Kaçınma: 5 olumsuz madde) içeren, toplamda 24 maddeden oluşmaktadır. Ölçekteki ifadeler “Hiçbir zaman böyle davranmam (1)”, “Ender olarak böyle davranırım (2)”, “Arada sırada böyle davranırım (3)”, “Sık sık böyle davranırım (4)”, “Her zaman böyle davranırım (5)” olacak şekilde derecelendirilmiş iken olumsuz maddeler ise yukarıdaki puanlamanın tersi yönde kodlanmıştır. Ölçeğin Cronbach Alpha güvenirlik katsayısı 0.80 olarak hesaplanmış, bu çalışmada ise bu değer 0.77 olarak tespit edilmiştir.

### 3.4. Verilerin Analizi

İstatistiksel analize başlamadan önce, öğrencilere uygulanan ölçek ve envanter puanlanmıştır. Devamında hazırlanan verilerin istatistiksel analizleri ise bilgisayar paket programlarından IBM SPSS Statistics v21 programı aracılığıyla bilgisayar ortamında gerçekleştirilmiştir. Bulgular aritmetik ortalama ( $\bar{X}$ ), standart sapma (S) ve p değerleri kullanılarak yorumlanmıştır. Deney ve kontrol gruplarının, düzey belirleme testi puanlarının normal dağılım gösterip göstermediğini saptamak amacıyla Shapiro-Wilk testi ile incelenmiş ve normal dağılıma sahip olan puanlarda parametrik testler, normal dağılıma sahip olmayan puanlarda ise parametrik olmayan testler kullanılmıştır. Ayrıca, karşılaştırma yapılan verilerin tümü için anlamlılık düzeyi .05 olarak kabul edilmiştir.

Araştırmanın birinci ve ikinci alt problemlerine yanıt bulmak için deney ve kontrol gruplarının problem çözme becerileri ve bilgisayara yönelik tutumları arasında ön test puanı ortalamalarında anlamlı bir farkın olup olmadığı Mann-Whitney U testi ile analiz edilmiştir. Sontestlerin tamamlanmasının ardından problem çözme becerileri ve bilgisayara yönelik tutumları arasında son test puanı ortalamalarında anlamlı bir farkın olup olmadığı kontrol etmek amacıyla bağımsız örneklem t-testi uygulanmıştır. Bu testin güvenilir sonuçlar verebilmesi için şu koşullar gerekir (Can, 2013, s. 116) :

1.Ortalamaları kıyaslanacak verilerin her birisinin dağılımı normal dağılım özelliklerini taşımalıdır.

2.Grupların varyansları eşittir.

3.Her bir veri diğerinden bağımsızdır.

Yukarıdaki varsayımlar doğrulandıktan sonra t testi yapılmıştır. İki değer arasındaki ortalamaları arasında anlamlı fark olup olmadığını belirlemek için uygulanan bağımsız örneklem t-testi, farkın derecesinin ne olduğunu anlamamıza yardımcı olmaz. Bu yüzden istatistiksel anlamlılık bulunduğundan sonra etki büyüklüğüne de bakılması gerekmektedir (Can, 2013, s. 121). Bunun sonucunda bağımsız örneklem t-testi için etki büyüklüğü altta belirtilen formül ile hesaplanabilir:

$$d = t \times \sqrt{\frac{N_1 + N_2}{N_1 \times N_2}}$$

Sırasıyla etki d nin alabileceği, 0.2, 0.5 ve 0.8 gibi değerler, küçük, orta ve büyük etki olarak değerlendirilir (Can, 2013, s. 121).

Araştırmanın üçüncü alt problemine yanıt bulmak için deney grubu ve kontrol grubunun kendi içinde, problem çözme becerileri ve bilgisayara yönelik tutum puanları arasındaki farkın (gelişim düzeyi) ortalamaları arasında anlamlı bir farkın olup olmadığını belirlemek için hem deney hem de kontrol grubunun öntest ve sontest skorlarının ortalamaları karşılaştırılmıştır. Normal dağılım gösteren verilerin karşılaştırılmasında bağımlı örneklem t-testi analizi, normal dağılım göstermeyen verilerin karşılaştırılmasında ise Wilcoxon İşaretli Sıralar Testi analizi kullanılmıştır. Bu testin güvenilir sonuçlar verebilmesi için şu koşullar gerekir (Can, 2013, s. 136) :

1.Ortalamaları kıyaslanacak verilerin (en az aralık ölçeğinde olan), farklarının oluşturduğu veri dizisi normal dağılım özelliği taşımalıdır.

2.Fark puanları birbirinden bağımsızdır.

Yukarıdaki varsayımlar doğrulandıktan sonra yapılan iki değer arasındaki ortalamaları arasında anlamlı fark olup olmadığını belirlemek için uygulanan bağımsız örneklem t-testi, farkın derecesinin ne olduğunu anlamak için test ile hesaplanan t değerinin, örneklem mevcudunun kareköküne oranı etki büyüklüğünü verir.

$$d = \frac{t}{\sqrt{N}}$$

### 3.5. Uygulama

Uygulama Scratch ile yapılan blok tabanlı programlama öğretiminin öğrencilerin problem çözme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisinin incelenmesi amacıyla 2017-2018 eğitim öğretim yılında Kayseri ilindeki bir ortaokulda

bilişim teknolojileri ve yazılım dersi kapsamındaki 6. Sınıflarında bulunan öğrencilerle gerçekleştirilmiştir. Uygulamada biri deney (6-K) diğeri ise kontrol grubu (6-H) olmak üzere iki grup bulunmaktadır. Uygulama 8 hafta boyunca, haftada iki ders saati olarak devam etmiştir. Ders içeriği ile ilgili kazanımlar uzman görüşü doğrultusunda ve MEB' in Bilişim Teknolojileri ve Yazılım dersi öğretim programına göre belirlenmiş ve bu kazanımlara uygun olarak Millî Eğitim Bakanlığı tarafından hazırlanmış olan ders kitabı/öğretim materyalleri incelenerek etkinlikler oluşturulmuştur. Deney grubu öğrencileri, blok tabanlı programlama öğretim aracı olan Scratch 2.0 Çevrimdışı Editörü ile “Problem Çözme ve Programlama” ünitesinin kazanımları ve içerikleri (Ek-5) göz önünde bulunarak işlenmişken, kontrol grubu öğrencileri ise “Etik ve Güvenlik” ve “İletişim, Araştırma ve İş Birliği” ünitelerinin kazanımları ve içerikleriyle (Ek-4) müfredatta belirtildiği şekilde derslerini sürdürmüşlerdir. Deney grubuna yönelik hazırlanan scratch etkinlikleri (Ek-6) ile çocukların programlama mantığını anlamaları ve yazılım geliştirebilmeleri için programlamanın ortak kavramları olan algoritma ve akış şeması, girdi-çıkı, değişken, operatörler, döngüler, fare olayları (mouse events), prosedür (procedure), nesne kontrolü, karar verme ve kontrol işlemleri gibi bileşenleri öğrenmeleri amaçlanmıştır. Öğrenciler, etkinlerin uygulamasını 24+1 adet bilgisayarı bulunan bilişim teknolojileri sınıfında (Ek-7), programlama adımlarının çıktıları üzerinden takip ederek bireysel olarak gerçekleştirmiş olup, gerekli dönütleri anında almaları sağlanmıştır.

Uygulamanın ilk haftasında deney ve kontrol grubu öğrencilerine “Bilgisayara Yönelik Tutum Ölçeği” ile “Çocuklar için Problem Çözme Envanteri” öntest olarak uygulanmıştır. Deney ve kontrol grubundaki öğrenciler 8 hafta süresince Tablo 3'te gösterilen eğitimleri almışlardır. Sonraki haftada ise her iki gruptaki öğrencilere “Bilgisayara Yönelik Tutum Ölçeği” ile “Çocuklar için Problem Çözme Envanteri” sontest olarak uygulanmıştır.

**Tablo 3.** Uygulama Süreci

Hafta	Deney Grubu	Kontrol Grubu
	<b>1. Öntestlerin uygulanması</b> a. Çocuklar için Problem Çözme Envanteri b. Bilgisayara Yönelik Tutum Ölçeği <b>2. Deney ve kontrol gruplarının rastgele seçilmesi</b>	
Hafta 1	1. Problem Çözme ve Algoritma 2. Akış Şeması, Tekrarlı Yapı, Karar Yapı, Doğrusal Yapı 3. Yönergeleri Takip Etme, İşlem Basamakları	1. Etik, İnternet Etiği 2. Siber Zorbalık, Dijital Zorbalık
Hafta 2	1. Scratch Arayüz ve Yapısı: Proje Ekranı (Sahne), Blok Paketi (Diziler), Kodlama Alanı, Menü Çubuğu, Kuklalar (Karakterler), Kılıklar (Kostümler), Sesler 2. Karaktere Komutlar Verme	1. Telif Hakları, Korsan Ürün 2. Lisans, Lisans Türleri, Beta Sürüm, Freeware, Shareware, Demo 3. Açık Erişim Felsefesi, Yaratıcı Birliktelik (Creative Commons)
Hafta 3	1. Sahne ve Karakter 2. Olaylar ve Hareket Blokları	1. Bilişim Suçları, İnternet Bilgi İhbar Merkezi 2. Siber Suç, Siber Dünya
Hafta 4	1. Sıralama, Döngü 2. Dekor, Kukla ve Kılık Kütüphanesi 3. Hareket, Görünüm ve Olaylar Blokları	1. Dijital Yurttaşlık, Dijital Ayak İzi, Kültürel Yozlaşma 2. Oyun, Oyun Türleri, Oyun Platformları
Hafta 5	1. Habersal blokları 2. Sahne Dekorları ve Kılıklar (Kostümler) Arasında Geçiş 3. Olaylar, Kontrol, Görünüm, Ses Blokları	1. EBA, Bilgi Gizliliği, Güvenlik, 2. Bilgi Koruma, Erişilebilirlik, Veri Bütünlüğü, Gizlilik 3. Şifreleme, Veri Güvenliği
Hafta 6	1. Değişken Oluşturma, Koşul kullanma, Girdi-Çıktı Kontrolü, Prosedür (Procedure) 2. Olaylar, Kontrol, Hareket, Görünüm, Veri, Algılama, İşlemler ve Özel Taşlar Blokları	1. Virüs, Zararlı Yazılım, Yayılma, Bulaşma, Zarar Verme 2. Virüsten Korunma, Güvenlik Duvarı, Antivirüs, Kare Kod, Zararlı Yazılım, Koruyucu Yazılım
Hafta 7	1. Fare Olayları (Mouse Events), Sahne koordinat düzlemi. x ve y Konumları 2. Olaylar, Kontrol, Hareket, Algılama, İşlemler ve Kalem Blokları	1. Bilgisayar Ağları, Ağ Bileşenleri, Donanım ve Yazılım, Ağ Kartı, Ağ Kablosu
Hafta 8	1. Nesne Kontrolü, Grafik Kontrol, Olay Kontrolü, Çoklu Ortam 2. Operatörler, Karar Verme ve Kontrol İşlemleri 3. Olaylar, Kontrol, Hareket, Algılama, İşlemler, Ses, Veri ve Görünüm Blokları	1. Arama Motorları 2. Bilgi Yönetimi, Bilgi Kirliliği 3. EBA
	<b>1. Sontestlerin uygulanması</b> a. Çocuklar için Problem Çözme Envanteri b. Bilgisayara Yönelik Tutum Ölçeği	

## BÖLÜM IV

### BULGULAR VE YORUM

Bu bölümde, araştırma kapsamında olan çalışma grubundaki öğrencilerden elde edilen verilere ilişkin analiz sonuçları tablolar halinde sunulmuştur.

#### 4.1. Birinci Alt Probleme İlişkin Bulgular ve Yorumlar

Araştırmanın birinci alt problemi “Müfredata göre ders işlenen kontrol grubu ve blok tabanlı programlama uygulamaları ile ders işlenen deney grubu öğrencilerinin, uygulama öncesindeki problem çözme becerileri ve bilgisayara yönelik tutumları arasında istatistiksel olarak anlamlı bir farklılık var mıdır?” olarak belirlenmiştir.

Ortaokul altıncı sınıf öğrencilerinin problem çözme becerileri; çocuklar için problem çözme envanterindeki sorulara verdikleri yanıtlarla incelenmiştir. Öğrencilerin çocuklar için problem çözme envanterinden aldıkları puanlara göre; aritmetik ortalama, standart sapma, en küçük ve en büyük değerleri belirlenmiş ve Tablo 4’te gösterilmiştir.

**Tablo 4.** Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Betimsel Analiz Sonuçları

Grup	Puan	N	$\bar{X}$	S	En küçük	En büyük
Kontrol	Problem çözme becerisi	25	78.59	8.04	64.08	89.04
	Problem çözme becerisine güven	25	80.89	13.32	50.00	104.00
	Öz denetim	25	70.63	17.23	34.29	96.00
	Kaçınma	25	84.10	16.75	38.40	115.20
Deney	Problem çözme becerisi	24	80.03	7.56	55.92	89.04
	Problem çözme becerisine güven	24	76.32	15.29	38.00	116.00
	Öz denetim	24	78.86	19.12	37.71	120.00
	Kaçınma	24	85.06	20.72	38.40	115.20

Öğrencilerin çocuklar için problem çözme envanterinden aldıkları puanlar; kontrol grubunda 64 ile 89, deney grubunda ise 56 ile 89 arasında değişmektedir. Kontrol grubu katılımcılarına ait problem çözme becerisi puanlarının ortalaması ( $\bar{X}=78.59$ ), problem çözme becerisine güven puanlarının ortalaması ( $\bar{X}=80.89$ ), öz denetim puanlarının ortalaması ( $\bar{X}=70.63$ ) ve kaçınma puanlarının ortalaması ( $\bar{X}=84.10$ ) olarak bulunmuştur. Deney grubu katılımcılarına ait problem çözme becerisi puanlarının ortalaması ( $\bar{X}=80.03$ ), problem çözme becerisine güven puanlarının ortalaması ( $\bar{X}=76.32$ ), öz denetim puanlarının ortalaması ( $\bar{X}=78.86$ ) ve kaçınma puanlarının ortalaması ( $\bar{X}=85.06$ ) olarak bulunmuştur. Bulgular, katılımcıların problem çözme puanlarının orta düzeyde olduğu sonucunu vermektedir.

Araştırmanın, ortaokul altıncı sınıf öğrencilerinin bilgisayara yönelik tutumları; bilgisayara yönelik tutum ölçeğindeki sorulara verdikleri yanıtlarla incelenmiştir. Öğrencilerin bilgisayara yönelik tutum ölçeğinden aldıkları puanlara göre aritmetik ortalama, standart sapma, en küçük ve en büyük değerleri belirlenmiş ve Tablo 5'te gösterilmiştir.

**Tablo 5.** Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Betimsel Analiz Sonuçları

Grup	Puan	N	$\bar{X}$	S	En küçük	En büyük
Kontrol	Bilgisayara yönelik tutum	25	70.67	7.55	53.04	81.12
Deney	Bilgisayara yönelik tutum	24	71.65	7.84	49.92	81.12

Öğrencilerin bilgisayara yönelik tutum ölçeğinden aldıkları puanlar; kontrol grubunda 53 ile 81, deney grubunda ise 50 ile 81 arasında değişmektedir. Bilgisayara yönelik tutum puanlarının ortalaması; kontrol grubunda ( $\bar{X}=70.67$ ), deney grubunda ise ( $\bar{X}=71.65$ ) olarak hesaplanmıştır. Bu veriler, araştırmaya katılan öğrencilerin bilgisayara yönelik tutumlarının nötr olduğunu göstermektedir.

Grupların öntest verilerinin normal dağılım gösterip göstermediğine bakmak için yapılan Shapiro-Wilk testi sonuçları Tablo 6 ve Tablo 7'de verilmiştir.

**Tablo 6.** Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Shapiro-Wilk Testi Sonuçları

Grup	Puan	İstatistik	sd	p
Kontrol	Problem çözme becerisi	.92	25	.046
Deney	Problem çözme becerisi	.89	24	.014

Tablo 6 incelenecek olursa deney ve kontrol gruplarından çocuklar için problem çözme envanteri kullanılarak elde edilen öntest verilerinin Shapiro-Wilk Testi sonuçlarına göre normal dağılım göstermediği ( $p < 0.05$ ) görülmektedir.

**Tablo 7.** Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Shapiro-Wilk Testi Sonuçları

Grup	Puan	İstatistik	sd	p
Kontrol	Bilgisayara yönelik tutum	.93	25	.078
Deney	Bilgisayara yönelik tutum	.86	24	.003

Tablo 7 incelendiğinde çocuklar için problem çözme envanteri kullanılarak elde edilen öntest verilerinin Shapiro-Wilk Testi sonuçlarına göre kontrol grubunun normal dağılım gösterdiği ( $p > 0.05$ ), deney grubunun ise normal dağılım göstermediği ( $p < 0.05$ ) görülmektedir.

Öntest skorlarının normal dağılım göstermediği gruplarda ortalamalar arasındaki fark parametrik olmayan testlerin uygulanmasının daha doğru olacağı anlamına geldiğinden, deney ve kontrol grubu öğrencilerinin çocuklar için problem çözme envanteri öntest ortalama puanları arasında anlamlı bir farkın olup olmadığı Mann-Whitney U testi kullanılarak incelenmiş ve sonuçlar Tablo 8’de verilmiştir.

**Tablo 8.** Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Öntest Verilerinin Mann-Whitney U Testi Sonuçları

Grup	N	Sıra Ortalaması	Sıra Toplamı	U	p
Kontrol	25	23.76	594.00	269.00	.53
Deney	24	26.29	631.00		

Tablo 8’de görüldüğü gibi deney grubundaki öğrenciler ile kontrol grubundaki öğrencilerin problem çözme becerisi puanları arasında anlamlı bir farklılık olmadığı bulunmuştur ( $U=269$ ,  $p=0.53$ ). Sıra ortalamaları dikkate alındığında, deney grubunun problem çözme becerileri ( $\bar{X}=26.29$ ), kontrol grubuna ( $\bar{X}=23.76$ ) göre az da olsa yüksek olmakla birlikte, bu fark istatistiksel olarak anlamlı değildir ( $p>0.05$ ). Buna göre; uygulama öncesinde deney ve kontrol gruplarının problem çözme becerileri açısından denk oldukları söylenebilir.

Araştırmada ele alınan diğer değişken olan bilgisayara yönelik tutumları açısından karşılaştırma yapmak için öncelikle normal dağılım durumu incelenmiş ve normal dağılım olmadığı tespit edilmiştir. Öntest puanları normal dağılım göstermediği gruplarda ortalamalar arasındaki farkın anlamlılığının incelenmesi için parametrik olmayan testlerin uygulanması daha uygun olacağından, deney ve kontrol grubu öğrencilerinin bilgisayara yönelik tutum ölçeği öntest puanları arasında anlamlı bir farkın olup olmadığı Mann-Whitney U testi kullanılarak incelenmiş ve sonuçlar Tablo 9’da verilmiştir.

**Tablo 9.** Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Öntest Verilerinin Mann-Whitney U Testi Sonuçları

Grup	N	Sıra Ortalaması	Sıra Toplamı	U	p
Kontrol	25	23.46	586.50	261.50	.44
Deney	24	26.60	638.50		



Tablo 9’da görüldüğü gibi deney grubundaki öğrenciler ile kontrol grubundaki öğrencilerin bilgisayara yönelik tutum puanları arasında anlamlı bir farklılık olmadığı bulunmuştur ( $U=261.50$ ,  $p=0.44$ ). Sıra ortalamaları dikkate alındığında, deney grubunun bilgisayara yönelik tutumları ( $\bar{X}=26.60$ ), kontrol grubuna ( $\bar{X}=23.46$ ) göre az da olsa yüksek olmakla birlikte bu fark istatistiksel olarak anlamlı değildir. Buna göre uygulama öncesinde deney ve kontrol gruplarının, bilgisayara yönelik ön tutumlar açısından denk oldukları söylenebilir.

Buradan hareketle deney ve kontrol gruplarının uygulama öncesinde, bu çalışmada ele alınan iki değişken olan problem çözme becerileri ve bilgisayara yönelik tutum açısından denk oldukları söylenebilir.

#### **4.2. İkinci Alt Probleme İlişkin Bulgular ve Yorumlar**

Araştırmanın ikinci alt problemi “Müfredata göre ders işlenen kontrol grubu ve blok tabanlı programlama uygulamaları ile ders işlenen deney grubu öğrencilerinin, uygulama sonrasındaki problem çözme becerileri ve bilgisayara yönelik tutumları arasında istatistiksel olarak anlamlı bir farklılık var mıdır?” olarak belirlenmiştir.

Araştırmanın, ortaokul altıncı sınıf öğrencilerinin problem çözme becerileri; çocuklar için problem çözme envanterindeki sorulara verdikleri yanıtlarla incelenmiştir. Öğrencilerin çocuklar için problem çözme envanterinden aldıkları puanlara göre; aritmetik ortalama, standart sapma, en küçük ve en büyük değerleri belirlenmiş ve Tablo 10’da gösterilmiştir.

**Tablo 10.** Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Sontest Verilerinin Betimsel Analiz Sonuçları

Grup	Puan	N	$\bar{X}$	S	En küçük	En büyük
Kontrol	Problem çözme becerisi	25	81.25	6.37	72.00	91.92
	Problem çözme becerisine güven	25	83.68	17.68	50.00	108.00
	Öz denetim	25	88.32	14.52	61.71	109.71
	Kaçınma	25	93.31	13.22	67.20	115.20
Deney	Problem çözme becerisi	24	87.89	10.97	63.12	108.00
	Problem çözme becerisine güven	24	80.32	16.87	54.00	118.00
	Öz denetim	24	81.05	19.72	41.14	116.57
	Kaçınma	24	93.12	20.88	52.80	120.00

Öğrencilerin çocuklar için problem çözme envanterinden aldıkları puanlar; kontrol grubunda 72 ile 92, deney grubunda ise 63 ile 108 arasında değişmektedir. Kontrol grubu katılımcılarına ait; problem çözme becerisi puanlarının ortalaması ( $\bar{X}$ =81.25), problem çözme becerisine güven puanlarının ortalaması ( $\bar{X}$ =83.68), öz denetim puanlarının ortalaması ( $\bar{X}$ =88.32) ve kaçınma puanlarının ortalaması ( $\bar{X}$ =93.31) olarak bulunmuştur. Deney grubu katılımcılarına ait; problem çözme becerisi puanlarının ortalaması ( $\bar{X}$ =87.89), problem çözme becerisine güven puanlarının ortalaması ( $\bar{X}$ =80.32), öz denetim puanlarının ortalaması ( $\bar{X}$ =81.05) ve kaçınma puanlarının ortalaması ( $\bar{X}$ =93.12) olarak bulunmuştur. Bulgular, katılımcıların problem çözme puanlarının orta düzeyde olduğu sonucunu vermektedir.

Araştırmanın diğer değişkeni, ortaokul altıncı sınıf öğrencilerinin bilgisayara yönelik tutumları; bilgisayara yönelik tutum ölçeğindeki sorulara verdikleri yanıtlarla

incelenmiştir. Öğrencilerin bilgisayara yönelik tutum ölçeğinden aldıkları puanlara göre; aritmetik ortalama, standart sapma, en küçük ve en büyük değerleri belirlenmiş ve Tablo 11’de gösterilmiştir.

**Tablo 11.** Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Sontest Verilerinin Betimsel Analiz Sonuçları

Grup	Puan	N	$\bar{X}$	S	En küçük	En büyük
Kontrol	Bilgisayara yönelik tutum	25	77.06	12.91	48.00	105.12
Deney	Bilgisayara yönelik tutum	24	82.83	12.26	65.04	109.92

Öğrencilerin bilgisayara yönelik tutum ölçeğinden aldıkları puanlar; kontrol grubunda 48 ile 105, deney grubunda ise 65 ile 110 arasında değişmektedir. Bilgisayara yönelik tutum puanlarının ortalaması; kontrol grubunda ( $\bar{X}=77.06$ ), deney grubunda ise ( $\bar{X}=82.83$ ) olarak hesaplanmıştır. Bu veriler, araştırmaya katılan öğrencilerin bilgisayara yönelik tutumlarının pozitif olduğunu göstermektedir.

Grupların sontest verilerinin normal dağılım gösterip göstermediğine bakmak için yapılan Shapiro-Wilk testi sonuçları Tablo 12 ve Tablo 13’te verilmiştir.

**Tablo 12.** Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Sontest Verilerinin Shapiro-Wilk Testi Sonuçları

Grup	Puan	İstatistik	sd	p
Kontrol	Problem çözme becerisi	.930	25	.086
Deney	Problem çözme becerisi	.974	24	.769

Tablo 12 incelenecek olursa deney ve kontrol gruplarındaki öğrenciler için problem çözme envanteri kullanılarak elde edilen sontest verilerinin Shapiro-Wilk Testi sonuçlarına göre normal dağılım gösterdiği ( $p>0.05$ ) görülmektedir.

**Tablo 13.** Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Sontest Verilerinin Shapiro-Wilk Testi Sonuçları

Grup	Puan	İstatistik	sd	p
Kontrol	Bilgisayara yönelik tutum	.993	25	.999
Deney	Bilgisayara yönelik tutum	.954	24	.335

Tablo 13 incelenecek olursa deney ve kontrol gruplarından bilgisayara yönelik tutum ölçeği kullanılarak elde edilen sontest verilerinin, Shapiro-Wilk Testi sonuçlarına göre normal dağılım gösterdiği ( $p>0.05$ ) görülmektedir.

Sontest skorlarının normal dağılım gösterdiği gruplarda ortalamalar arasındaki fark parametrik testlerin uygulanabileceği anlamına geldiğinden, deney ve kontrol grubu öğrencilerinin çocuklar için problem çözme envanteri sontest ortalama puanları arasında anlamlı bir farkın olup olmadığı, bağımsız örneklem t-testi kullanılarak incelenmiş ve sonuçlar Tablo 14’te verilmiştir.

**Tablo 14.** Deney ve Kontrol Gruplarından Çocuklar için Problem Çözme Envanteri Kullanılarak Elde Edilen Sontest Verilerinin Bağımsız Örneklem t-testi Sonuçları

Grup	N	$\bar{X}$	S	sd	t	p
Kontrol	25	81.25	6.37	47	-2.603	.012
Deney	24	87.89	10.97			

Tablo 14’de görüldüğü gibi deney grubundaki öğrenciler ile kontrol grubundaki öğrencilerin problem çözme becerisi puanları arasında istatistiksel olarak anlamlı bir farklılık olduğu bulunmuştur ( $t(47)=- 2.603$ ,  $p<0.05$ ). Deney grubunun problem çözme becerileri ( $\bar{X}=87.89$ ), kontrol grubuna ( $\bar{X}=81.25$ ) göre daha yüksek olup, bu fark

istatistiksel olarak anlamlıdır. Hesaplanan etki büyüklüğü ( $d=0.7$ ) bu etkinin orta düzeyde olduğunu göstermektedir. Bu analiz sonuçlarına göre, problem çözme becerisi açısından deney ve kontrol gruplarının sontest skorları arasında deney grubu lehine anlamlı bir farklılaşmanın olduğu sonucu çıkarılabilir.

Sontest skorlarının normal dağılım gösterdiği gruplarda ortalamalar arasındaki fark parametrik testlerin uygulanabileceği anlamına geldiğinden, deney ve kontrol grubu öğrencilerinin çocuklar için bilgisayara yönelik tutum ölçeği sontest ortalama puanları arasında anlamlı bir farkın olup olmadığı bağımsız örneklem t-testi kullanılarak incelenmiş ve sonuçlar Tablo 15’te verilmiştir.

**Tablo 15.** Deney ve Kontrol Gruplarından Bilgisayara Yönelik Tutum Ölçeği Kullanılarak Elde Edilen Sontest Verilerinin Bağımsız Örneklem t-testi Sonuçları

Grup	N	$\bar{X}$	S	sd	t	p
Kontrol	25	77.06	12.91	47	-1.479	.146
Deney	24	82.38	12.26			

Tablo 15’te görüldüğü gibi deney grubundaki öğrenciler ile kontrol grubundaki öğrencilerin bilgisayara yönelik tutum puanları arasında anlamlı bir farklılık olmadığı bulunmuştur ( $t(47)= -1.479$ ,  $p>0.05$ ). Deney grubunun problem çözme becerileri ( $\bar{X}=82.38$ ), kontrol grubuna ( $\bar{X}=77.06$ ) göre daha yüksek olmakla birlikte bu fark istatistiksel olarak anlamlı değildir. Bu analiz sonuçlarına göre, bilgisayara yönelik tutumları açısından deney ve kontrol gruplarının sontest skorları arasında anlamlı bir farklılaşmanın olmadığı sonucu çıkarılabilir.

### 4.3. Üçüncü Alt Probleme İlişkin Bulgular ve Yorumlar

Araştırmanın üçüncü alt problemi “Müfredata göre ders işlenen kontrol grubu ve blok tabanlı programlama uygulamaları ile ders işlenen deney grubu öğrencilerinin, uygulama öncesi ve sonrasında kendi içinde, problem çözme becerileri ve bilgisayara yönelik tutumları puanları (gelişim düzeyi) arasında istatistiksel olarak anlamlı bir değişim var mıdır?” olarak belirlenmiştir. Bu araştırma problemi ile her iki gruptaki öğrencilerin uygulama sonrasında tutum ve problem çözme becerilerinde bir değişim olup olmadığı sorgulanmaktadır.

Grupların son test puanları ile öntest puanları arasındaki farkların oluşturduğu fark puan dizisinin normal dağılım gösterip göstermediğine bakmak için yapılan Shapiro-Wilk testi sonuçları Tablo 16’da verilmiştir.

**Tablo 16.** Deney Grubu ve Kontrol Grubunun Kendi İçinde, Problem Çözme Becerileri ve Bilgisayara Yönelik Tutumları Puanları Arasındaki Farkın Verilerinin Shapiro-Wilk Testi Sonuçları

Puan	Grup	İstatistik	sd	p
Problem çözme becerisi puanları arasındaki fark	Kontrol	.874	25	.005
	Deney	.938	24	.137
Bilgisayara yönelik tutum puanları arasındaki fark	Kontrol	.964	25	.503
	Deney	.937	24	.129

Tablo 16 incelendiğinde çocuklar için problem çözme envanteri kullanılarak elde edilen fark puanlarının Shapiro-Wilk Testi sonuçlarına göre kontrol grubu verilerinin normal dağılım göstermediği ( $p < 0.05$ ), deney grubu verilerinin ise normal dağılım gösterdiği ( $p > 0.05$ ) görülmektedir. Deney ve kontrol gruplarından bilgisayara yönelik tutum ölçeği kullanılarak elde edilen fark puanlarının normal dağılım gösterdiği ( $p > 0.05$ ) görülmektedir.

Fark puanlarının normal dağılım göstermediği gruplarda ortalamalar arasındaki farkın istatistiksel olarak anlamlı olup olmadığını belirlemek için parametrik olmayan testler tercih edilir. Kontrol grubu öğrencilerinin uygulama öncesi ve uygulama sonrası yapılan çocuklar için problem çözme envanteri puanlarının ortalamaları arasında anlamlı bir farkın olup olmadığı Wilcoxon İşaretli Sıralar Testi kullanılarak incelenmiş ve sonuçlar Tablo 17’de verilmiştir.

**Tablo 17.** Kontrol Grubu İçin Problem Çözme Envanteri Öntest ve Sontest Uygulamasının Wilcoxon İşaretli Sıralar Testi Sonuçları

Sontest- Öntest	N	Sıra Ortalaması	Sıra Toplamı	z	p
Negatif Sıra	4	17.50	70.00	-2.288	.022
Pozitif Sıra	20	11.50	230.00		
Eşit	1				

\*Negatif sıralara dayalı

Tablo 17'deki sonuçlara göre kontrol gurubundaki öğrencilerin problem çözme envanterinden aldıkları uygulama öncesi ve sonrası puanları arasında anlamlı bir fark olduğunu göstermektedir ( $z=-2.288$ ,  $p<0.05$ ). Fark puanlarının sıra ortalaması ve toplamına bakıldığında, gözlenen bu farkın pozitif sıralar, yani sontest puanı lehinde olduğu sonucunu göstermektedir. Bu sonuca göre çalışma sonunda, kontrol grubu öğrencilerinin problem çözme becerilerinin geliştiği söylenebilir.

Fark puanlarının normal dağılım gösterdiği gruplarda ortalamalar arasındaki farkın istatistiksel olarak anlamlı olup olmadığını belirlemek için parametrik testler tercih edilir. Deney grubu öğrencilerinin uygulama öncesi ve uygulama sonrası yapılan çocuklar için problem çözme envanteri puanlarının ortalamaları arasında anlamlı bir farkın olup olmadığı bağımlı örneklem t-testi kullanılarak incelenmiş ve sonuçlar Tablo 18'de verilmiştir.

**Tablo 18.** Deney grubu Çocuklar için Problem Çözme Envanteri Öntest ve Sontest Uygulamasının Bağımlı Örneklem t-testi Sonuçları

Grup	N	$\bar{X}$	S	sd	t	p
Öntest	24	78.91	9.27	23	-4.246	.000
Sontest	24	86.67	12.38			

Tablo 18'deki sonuçlara göre deney grubundaki öğrencilerin uygulama sonrasında problem çözme becerilerinde anlamlı bir artış olduğu bulunmuştur ( $t(23)=-4.246$ ,  $p<0.05$ ). Öğrencilerin uygulama öncesi problem çözme becerileri puanlarının

ortalaması  $\bar{X}=78.91$  iken, blok tabanlı (Scratch) programlama öğretimi sonrasında  $\bar{X}=86.67$ 'e yükselmiştir. Hesaplanan etki büyüklüğü ( $d=0.8$ ) bu etkinin büyük olduğunu göstermektedir. Bu durum, blok tabanlı programlama öğretim yönteminin, deney grubu öğrencilerinin problem çözme becerilerini artırmada önemli bir etkiye sahip olduğunu gösterir.

Fark puanlarının normal dağılım gösterdiği gruplarda ortalamalar arasındaki farkın istatistiksel olarak anlamlı olup olmadığını belirlemek için parametrik testler tercih edilir. Kontrol grubu öğrencilerinin uygulama öncesi ve uygulama sonrası toplanan bilgisayara yönelik tutum ölçeği puanlarının ortalamaları arasında anlamlı bir farkın olup olmadığı bağımlı örneklem t-testi kullanılarak incelenmiş ve sonuçlar Tablo 19'da verilmiştir.

**Tablo 19.** Kontrol Grubu Bilgisayara Yönelik Tutum Ölçeği Öntest ve Sontest Uygulamasının Bağımlı Örneklem t-testi Sonuçları

Grup	N	$\bar{X}$	S	sd	t	p
Öntest	25	70.67	7.55	24	-2.309	.030
Sontest	25	77.06	12.91			

Tablo 19'daki sonuçlara göre kontrol grubundaki öğrencilerin uygulama sonrasında bilgisayara yönelik tutumlarında anlamlı bir artış olduğu bulunmuştur ( $t(24) = -2.309$ ,  $p < 0.05$ ). Öğrencilerin uygulama öncesi bilgisayara yönelik tutum puanlarının ortalaması  $\bar{X}=70.67$  iken, uygulamadaki öğretim yöntemi sonrasında  $\bar{X}=77.06$ 'e yükselmiştir. Hesaplanan etki büyüklüğü ( $d=0.5$ ) bu farkın orta düzeyde olduğunu göstermektedir. Bu durum, uygulama sonrasında, kontrol grubu öğrencilerinin bilgisayara yönelik tutumlarının olumlu olarak değiştiğini gösterir.

Fark puanlarının normal dağılım gösterdiği gruplarda ortalamalar arasındaki farkın istatistiksel olarak anlamlı olup olmadığını belirlemek için parametrik testler tercih edilir. Deney grubu öğrencilerinin uygulama öncesi ve uygulama sonrası yapılan bilgisayara yönelik tutum ölçeği puanlarının ortalamaları arasında anlamlı bir farkın olup olmadığı bağımlı örneklem t-testi kullanılarak incelenmiş ve sonuçlar Tablo 20'de verilmiştir.



**Tablo 20.** Deney Grubu Bilgisayara Yönelik Tutum Ölçeği Öntest ve Sontest Uygulamasının Bağımlı Örneklem t-testi Sonuçları

Grup	N	$\bar{X}$	S	sd	t	p
Öntest	24	70.63	9.22			
Sontest	24	81.96	12.18	23	-4.688	.000

Tablo 20'deki sonuçlara göre deney grubundaki öğrencilerin uygulama sonrasında bilgisayara yönelik tutumlarında anlamlı bir artış olduğu bulunmuştur ( $t(23) = -4.688$ ,  $p < 0.05$ ). Öğrencilerin uygulama öncesi bilgisayara yönelik tutum puanlarının ortalaması  $\bar{X} = 70.63$  iken, blok tabanlı programlama öğretimi sonrasında  $\bar{X} = 81.96$ 'e yükselmiştir. Hesaplanan etki büyüklüğü ( $d = 0.9$ ) bu etkinin büyük olduğunu göstermektedir. Bu durum, blok tabanlı programlama öğretim yönteminin, deney grubu öğrencilerinin bilgisayara yönelik tutumlarını artırmada önemli bir etkiye sahip olduğunu gösterir.

## BÖLÜM V

### SONUÇ VE ÖNERİLER

Bu bölümde, araştırma kapsamında toplanan verilerden elde edilen bulgulara dayanılarak ulaşılan sonuçlara ve önerilere yer verilmiştir.

İlgili literatürde de görülebileceği üzere, programlama öğretimi araçları sadece programlama öğretmek amacıyla değil, öğrencilerin farklı bilişsel ve duyuşsal becerilerini belli bir yönde değiştirmek (kaygıyı azaltmak, tutumu olumlu hale getirmek gibi), geliştirmek (mantıksal düşünme yeteneğini, eleştirel düşünme becerisini geliştirmek gibi) veya belli bir alana ait bir konuyu öğretmek (matematik dersi kesirler konusunu öğretmek gibi) amacıyla kullanılmaktadır (Grout ve Houlden, 2014; Jones, 2013; Kalelioğlu, 2015; Kert ve Uğraş, 2009). Bu çalışmada da öğrencilerin programlama öğretimi ya da matematik gibi belli bir konudaki akademik başarılarından ziyade onların problem çözme becerileri ve bilgisayara yönelik tutumları ele alınmıştır. Öğrencilerin ya da yetişkinlerin, programlama ile uğraşan herkesin programlama sürecinde belli bir oranda karşılaşacakları problemleri çözmeleri gerekeceği ve programlama yaparken genellikle bir bilgisayar ortamı kullanıldığı için bu iki değişken bu çalışmada tercih edilmiştir. Problem çözme becerisini geliştirmeyi amaçlayan bir programlama eğitiminin amacı, sadece programlama aracının kullanımını ya da programlama yapmayı öğretmek değil, aynı zamanda öğrencilerin programlama sürecinde problem çözme konusunda gerçek deneyimler kazanarak problem çözme becerilerini geliştirmek olmalıdır. Programlama süreci, öğrencilerin çeşitli durumlarla bütünleştirebilecekleri ve problem durumlarıyla esnek bir şekilde başa çıkabilmeleri için, stratejik bilgi edinerek öğrenebilecekleri karmaşık bir problem çözme sürecidir. Bununla birlikte, programlama yapmak genellikle, öğrenilmesi zor ve zaman alıcı bir süreç olarak bilinmektedir. Bu nedenle, öğrencilere programlama eğitimine kolayca nasıl yaklaşılacağını göstermek gerekir. Bunun en iyi yollarından birisi de görsel ve blok tabanlı bir yapıya sahip olan Scratch uygulamasını kullanmaktır.

Bu araştırma, ortaokul öğrencilerinin programlama eğitimi almalarının, bilgisayara yönelik tutumları ve problem çözme becerilerine etkisini araştırmak amacıyla yapılmıştır. Araştırmanın uygulama süreci, 2017-2018 eğitim öğretim yılında bir devlet ortaokulunda, 6. sınıfta öğrenim gören, bilişim teknolojileri ve yazılım dersi

alan öğrencilerle gerçekleştirilmiştir. Araştırma kapsamında, deney (N=24) ve kontrol grubu (N=25) olmak üzere, iki sınıfta bulunan toplam 49 öğrenci ile çalışılmıştır. Uygulama, haftada iki ders saati olmak üzere toplam 8 hafta boyunca yapılmıştır. Deney grubu öğrencileriyle dersler, blok tabanlı programlama öğretim aracı olan Scratch 2.0 Çevrimdışı Editörü ile işlenirken, kontrol grubu öğrencileri ise “Etik ve Güvenlik” ve “İletişim, Araştırma ve İş Birliği” ünitelerinin kazanımları ve içerikleriyle müfredata göre işlenmiştir. Uygulama öncesinde ve sonrasında her iki gruptaki öğrencilere “bilgisayara yönelik tutum ölçeği” ile “problem çözme becerileri envanteri”, ön test ve son test olarak uygulanmış, elde edilen veriler ile ulaşılan sonuçlar yorumlanarak açıklanmıştır.

Bu çalışmada birinci problemde ele alınan durum, uygulama öncesinde grupların denkleğinin sorgulanmasıdır. Deneysel bir işlem yapılmadan önce genellikle hem kontrol grubunun hem deney grubunun ölçülmek istenen değişkenler açısından denk olmaları beklenir. Böyle olmadığı durumlarda kullanılan çeşitli istatistik tekniklerle, grupların denk olmamasından kaynaklanabilecek sorunlar kontrol altına alınabilmekle birlikte, başlangıçta grupların denk olması tercih edilir. Bu çalışmada yapılan istatistik analizlere göre, araştırma kapsamında çalışılan her iki gruptaki öğrencilerin, ele alınan değişkenler olan problem çözme becerisi ve bilgisayara yönelik tutum açısından denk oldukları tespit edilmiştir. Böylece, uygulama sonrasında yapılan ölçümlerin, büyük oranda deneysel çalışmadan kaynaklandığı söylenebilir.

İkinci problemde ele alınan durumda ise, uygulanan deneysel işlem sonrasında yani; deney grubunda blok tabanlı bir programlama öğretimi aracı kullanılarak programlama öğretimi yapılmasının sonrasında, deney grubundaki öğrencilerin bilgisayara yönelik tutumlarında kontrol grubundaki öğrencilere göre anlamlı bir farklılık bulunmamıştır. Bununla birlikte, blok tabanlı programlama öğretiminin problem çözme becerisi açısından kontrol grubundaki öğretime göre daha etkili olduğu ortaya çıkmıştır. Hem deney hem de kontrol gruplarındaki öğrencilerle 8 hafta boyunca bilişim dersi kapsamındaki konular üzerinde çalışılmakla birlikte, blok tabanlı programlama öğrenmek amacıyla Scratch kullanan deney grubu öğrencileri, problem çözme becerileri açısından daha iyi bir ilerleme göstermişlerdir. Problem çözme becerileri açısından ölçülen bu farkın istatistiksel açıdan anlamlı olduğu tespit edilmiştir. Bu durumun literatürdeki pek çok çalışma ile uyumlu olduğu görülmektedir.

Problem çözüme becerileri hakkında yapılan çalışmalara bakıldığında pek çok çalışmada; yaş ve cinsiyetin öğrencilerin problem çözüme becerileri üzerinde herhangi bir etkisinin görülmediği sonucu bulunmuştur (Çilingir, 2006; Dündar, 2009; Gültekin 2006, Özkütük, Silkü, Orgun ve Yalçınkaya, 2003; Saygılı, 2000; Tümkaya ve İflazoğlu, 2000). Buradan hareketle bu becerinin doğuştan gelen ya da kendiliğinden gelişen bir beceri olmadığı, problem çözüme becerilerinin geliştirilmesi için özel etkinlikler yapılması gerektiği anlaşılmaktadır. Bu çalışma da dâhil olmak üzere ilgili literatürdeki pek çok çalışma sonuçlarının gösterdiği üzere; programlama öğretimi sürecinin bu becerinin gelişimini desteklediği düşünülürse, öğrencilerin problem çözüme becerilerini geliştirmek için programlama öğretiminin kullanılabilceği görülmüştür.

Bu çalışmadaki çalışma grubuna benzer şekilde ortaokul öğrencileriyle yapılan programlama öğretiminin, öğrencilerin problem çözüme becerilerine etkisini araştıran çok sayıda çalışmada, Scratch ile programlama öğretiminin öğrencilerin problem çözüme becerilerini geliştirmede etkili olduğu rapor edilmiştir (Brown ve arkadaşları, 2008; Nam, Kim ve Lee, 2010; Shin ve Park, 2014; Vatansever, 2018).

Ayrıca, ilgili literatürde bu çalışmadaki problem çözüme becerilerinin programlama öğretimi uygulaması sayesinde gelişmesi bulgusunu destekleyen pek çok çalışma bulunmaktadır. Aşağıda verilen çalışmalarda da belirtilen, programlama eğitiminin çocukların problem çözüme becerilerine olumlu yönde katkı sağlaması bulgusu, çalışmanın bulgusunu destekler niteliktedir.

Scratch ile yapılan uygulamalarda özellikle yaş grubu olarak erken yaşlardaki öğrenciler ile çalışılmış olması dikkat çekmektedir. Çalışma grupları genelde ilkököl ve ortaoköl seviyesindedir. Bu durum da bizim çalışmamız ile uyumludur. Bu yaş grubundaki öğrenciler genellikle dikkat süresi daha kısa olduğu için çabuk sıkılabilmekte ve ileri yaşlardaki öğrencilere göre daha fazla zorlanmaktadırlar. Bu duruma bir de programlama yapmak için gerekli olan algoritma ve diğer programlama bilgilerini kavramanın ve yerinde doğru olarak kullanmanın zorluğu eklendiğinde, bu öğrencilere özel, kullanımı daha kolay ve anlaşılır yöntem, teknik ve araçlara ihtiyaç duyulduğu anlaşılmaktadır. İlgili araştırmalara bakıldığı zaman, geleneksel programlama dilleri ile yapılan programlama öğretimi sırasında özellikle programlama geçmişi olmayan öğrencilerin, genellikle çok zorlandıkları belirtilmektedir. Bu nedenle bu çalışmada blok tabanlı ve görsel yapısı sayesinde bu yaş grubundaki öğrencilerin zorlanmadan çalışabilecekleri bir araç olarak Scratch kullanılmıştır. Yukarıda da

belirtildiği üzere çok sayıda araştırma, herhangi bir araç ya da ortam değil de özellikle Scratch kullanılarak yapılan bir programlama öğretiminin, öğrencilerin problem çözme becerilerini geliştirebileceği bulgusunu desteklemektedir.

Diğer yandan, az sayıda olmakla birlikte bazı çalışmalarda ise programlama öğretimi uygulamalarının, öğrencilerin problem çözme becerilerine olumlu ya da olumsuz herhangi bir etkisi olmadığı, problem çözme becerilerini geliştirmediği görülmektedir. Bu çalışmalardan birisi olan Solmaz'ın (2014) Alice yazılımını kullanarak programlama öğretimi yaptığı çalışmada, öğrencilerin problem çözme becerilerinde anlamlı bir değişim gözlenmemiştir. Bu durum programlama öğretiminde kullanılan yazılım türü ve ortamından, uygulama süresinden ya da çalışma grubundaki öğrencilerin üniversite öğrencileri olmalarından kaynaklanmış olabilir. Bir diğer çalışmada ise Logo aracı kullanılarak yapılan etkinlikler sonrasında öğrencilerin problem çözme becerilerinde bir değişim olmamıştır (Dalton, 1986). Bu çalışmada da kullanılan programlama öğretimi aracının oldukça eski araçlardan birisi olduğu göze çarpmaktadır. Bunlara benzer bir çalışmada Gülbahar ve Kalelioğlu (2014), Scratch programlama öğretimi aracının, 5. Sınıf öğrencilerinin problem çözme becerilerine etkisini araştırmışlar ve çalışma sonucunda, öğrencilerin problem çözme becerilerinde anlamlı bir değişim gözlenmediğini bildirmişlerdir. Bu durum ise çalışılan yaş grubundan, kullanılan yöntemden ya da uygulama süresinin sınırlı olmasından kaynaklanmış olabilir.

Araştırmada ele alınan diğer değişken olan bilgisayara yönelik tutum ele alınacak olursa; eğitimde kullanılan teknolojilerin önemli bir bölümünün bilgisayar donanımları ya da çeşitli bilgisayar yazılımları içermesi bakımından, öğrencilerin eğitim hayatları süresince bilgisayara yönelik tutumları, onların öğrendikleri tüm konuları ve aldıkları tüm dersleri etkileyebilecek bir öneme sahiptir. Bu açıdan daha verimli ve kalıcı bir eğitim için tespit edilen olumsuz tutumlar varsa bunlarla ilgili önlem almak mümkün hale gelecektir. Özellikle bilgisayar kullanılarak yapılan uygulamalar için bilgisayara yönelik tutumun pozitif olması tercih edilen bir durumdur. Tutumlar eğitim sürecinde öğrencilerin başarısını doğrudan etkilemektedir. Bilgisayara yönelik tutumu yüksek olan öğrenciler, öğretim ortamına aktif olarak katılım göstermektedir. Bilgisayar konusunda deneyim yükseldikçe bilgisayara yönelik tutum düzeyi de yükselmektedir (Erkan, 2004). Yapılan çalışmalarda genelde öğrencilerin bilgisayara yönelik tutumlarının olumlu bulunduğu ve bu tutumları ne kadar yükselirse, başarının da o

oranda yükseldiği belirtilmektedir (Öztürk ve Alper, 2019; Anderson, 1985; Aşkar, Yavuz ve Köksal, 1992; Koochang, 1989). Bizim çalışmamızda deneysel işlem sonrasında kontrol ve deney grubundaki öğrenciler arasında bilgisayara yönelik tutumları açısından anlamlı bir fark gözlenmemiştir. Bununla birlikte, her iki gruptaki öğrencilerin uygulama öncesinde nötr olan bilgisayara yönelik tutumlarının, uygulama sonrasında pozitif olarak değiştiği tespit edilmiştir ( $\bar{X}$  Kontrol grubu=77,  $\bar{X}$  Deney grubu =82). Diğer bir deyişle her iki gruptaki öğrencilerin tutumları olumlu olarak değişmiştir. Tutumlar gibi duyuşsal durumların kısa süreli çalışmalarla değişmediği genel olarak bilinmektedir. Diğer yandan uygulama sürecinde özellikle öğrencilerin çok ilgisini çeken oyun temelli uygulamalar ile çalışıldığı durumlarda, bilgisayara yönelik tutumlarda olumlu anlamda bir değişim söz konusu olabilmektedir. Tutumlar davranışları etkileme gücüne sahip olduğundan, eğitim ortamlarında önem verilmesi gereken faktörlerden birisi olarak karşımıza çıkmaktadır.

Araştırmanın üçüncü probleminde ise deney ve kontrol gruplarının uygulamalar bittikten sonraki durumları gruplar içi istatistik teknikleriyle analiz edilerek her iki grubun kendi içindeki gelişim durumları incelenmiştir. Bulgulara göre hem deney hem de kontrol gruplarının her iki değişken olan problem çözme becerileri ve bilgisayara yönelik tutumları açısından olumlu olarak değişim gösterdikleri anlaşılmıştır. Bu ise bilgi teknolojileri dersleri kapsamında işlenen konuların, öğrencilerin ilgisini çektiğini, problem çözme becerilerini geliştirebileceğini, bunun yanında bilgisayara yönelik tutumlarını da olumlu etkileyebileceğini göstermektedir.

Çocukluk döneminde öğrenilmeye başlayan problem çözme becerileri, okul çağında gelişim gösterir. Bilimsel, yaratıcı ve eleştirel düşünme gibi üst düzey düşünme becerilerinin gelişimi açısından eğitim-öğretim etkinlikleri arasında problem çözme sürecinin bulunması olumlu katkı sağlayacaktır. Karşılaştığı sorunların üstesinden gelebilen, mücadeleci ve çözüm üreten bireyler günümüz dünyasına daha kolay uyum göstereceklerdir. Günümüz “bilgi toplumu” yapısına bakıldığında genel “okuryazarlık” kavramının yerini “bilgisayar okuryazarlığı” aldığı ifade edilebilir. Tüm dünya üzerinde bağlantı kurmasından dolayı, bilgiye erişmek için kullandığımız araçların başında bilgisayarlar yer almaktadır. Bilgisayarlar, bireylerin akademik yaşantılarının dışında iş yaşamında ve sosyal iletişimde de büyük bir paya sahiptirler. Bu nedenle kişilerin bilgisayar yaşantısı arttıkça bilgisayara yönelik tutumlarının yükselmesini ve günlük yaşamlarında karşılaştıkları problemlerin üstesinden gelebilmelerini sağlayacaktır.

Araştırmanın sonuçları doğrultusunda,

- ✓ Günümüz dünyasında çok ihtiyaç duyulan programlama becerilerinin geliştirilmesi için programlama öğretimi uygulamalarının öğretim programlarına dâhil edilmesi,
- ✓ Programlama becerilerinin geliştirilmesi için en uygun araçların kullanılması konusunda ülkemiz şartlarında daha çok araştırma yapılması,
- ✓ Öğrencilerin problem çözme becerilerinin geliştirilmesi için öğretim programlarına uygun derslerin ve etkinliklerin konulması,
- ✓ Öğrencilerin bilgisayarlara yönelik tutumunda önemli bir yeri olan oyun dünyası olarak gördükleri bilgisayarları eğitim aracı olarak görmelerinin sağlanması,
- ✓ Bilgisayarlar aracılığıyla algoritmik düşünme yapısını kullanan yöntemler ile öğrencilerin çözüm bulma ve keşfetme becerilerini eğlenceli bir ortamda geliştirmelerinin sağlanması,
- ✓ Öğrencilerin problem çözme becerileri ile bilgi teknolojilerine yönelik tutumu etkileyen farklı boyutlar aracılığıyla daha geniş gruplarda veya farklı yöntemlerle araştırma yapılması önerilebilir.

## KAYNAKLAR

- Adams, J. C. (2010). *Scratching middle schoolers' creative itch*. Proceedings of the 41st ACM Technical Symposium on Computer Science Education (pp. 356-360). New York: ACM.
- Adams, J. C., & Webster, A. R. (2012, February). *What do students learn about programming from game, music video, and storytelling projects?* Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (pp. 643-648). New York: ACM.
- Akande, S. O. (2009). Knowledge, perception, and attitudes of library personnel towards preservation of information resources in Nigerian federal university libraries. *Library Philosophy and Practice (e-journal)*, 303.
- Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *Elementary Education Online*, 13(1), 1-4.
- Aktümen, M., & Kaçar, A. (2003). İlköğretim 8. sınıflarda harfli ifadelerle işlemlerin öğretiminde bilgisayar destekli öğretimin rolü ve bilgisayar destekli öğretim üzerine öğrenci görüşlerinin değerlendirilmesi. *Kastamonu Eğitim Dergisi*, 11(2), 339-358.
- Anderson, JC, & Gerbing, DW (1988). Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin*, 103(3), 411.
- Altun, T., & Bektaş, E. (2010). Views of Regional Boarding School teachers about the use of ICT in education. *Procedia-Social and Behavioral Sciences*, 9, 462-467.
- Anastasiadou, S. D., & Karakos, A. S. (2011). The beliefs of electrical and computer engineering students' regarding computer programming. *The International Journal of Technology, Knowledge and Society*, 7(1), 37-51.
- Anderson, R. E. (1985). But can she program? Equal opportunity for women does not compute. *Computer User*, 4, 20.
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). *Bilgisayar programlama öğretiminde yeni bir yaklaşım*. IX. Akademik Bilişim Konferansı, 193-197.
- Aşkar, P., & Orçan, H. (1987). The development of an attitude scale toward computers. *Journal of Human Sciences*, 6(2), 19-23.
- Aşkar, P., Yavuz, H., & Köksal, M. (1992). Students' perceptions of computer assisted instruction environment and their attitudes towards computer assisted learning. *Educational Research*, 34(2), 133-139.
- Baki, A., & Öztekin, B. (2003). Excel yardımıyla fonksiyonlar konusunun öğretimi. *Kastamonu Eğitim Dergisi*, 11(2), 325-338.



- Balanskat, A., & Engelhardt, K. (2015). Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe. *European Schoolnet*.
- Baldwin, L.P. ve Kuljis, J. (2000). Visualization techniques for learning and teaching programming. *Journal of Computing and Information Technology*, 8(4), 285–291.
- Baltalı, S. (2016). *Programlama öğretiminde kullanılabilecek yazılımlara ilişkin öğretmen görüşleri*. Yayınlanmamış Yüksek Lisans Tezi, Uludağ Üniversitesi, Bursa.
- Baykul, Y. (2005). *İlköğretimde Matematik Öğretimi 1- 5 sınıflar*. Ankara: PegemA Yayıncılık.
- Baytak, A. & Land, S. (2011) An investigation of the artifacts and process of constructing computers games about environmental science in a fifth-grade classroom. *Educational Technology Research and Development*, 59(7), 765–782.
- Baz, F. Ç. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Curr Res Educ*, 4(1), 36-47.
- Begosso, L. C., & da Silva, P. R. (2013). Teaching computer programming: A practical review. *2013 IEEE Frontiers in Education Conference (FIE)*, Oklahoma City, OK, 2013, pp. 508-510.
- Bennedsen, J. ve Carpersen, M. E. (2008). Exposing the programming process. Bennedsen, J.,Carpersen, M. E. ve Kolling, M. (Eds.). *Reflection on the theory of programming: Methods and implementation* (s.6-16) içinde. Springer Berlin Heidelberg, New York.
- Berksoy, İ., Sözcü, Ö., Armağan, E., & Arslan, A. (2016) *Algoritma ve Programlama Eğitiminde Scratch Programı Üzerine Yapılan Çalışmaların Analizi*. Web: <http://ab.org.tr/ab16/bildiri/74.docx> adresinden 18 Kasım 2018’de alınmıştır.
- Bingham, A. (2004). *Çocuklarda problem çözme yeteneklerinin geliştirilmesi* (çev. A.F. Oğuzkan). İstanbul: Millî Eğitim Bakanlığı Yayınları.
- Bottino, R. M. (2004). The evolution of ICT- based learning environments: which perspectives for the school of the future? *British Journal of Educational Technology*, 35(5), 553-567.
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer aided instruction as a vehicle for problem solving: Scratch boards in the middle years classroom. *ASEE Annual Conference and Exposition, Conference Proceedings* (pp. 22-24).
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., & Miller, P. (1997). Mini-languages: a way to learn programming principles. *Education and information technologies*, 2(1), 65-83.

- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *The Journal of Media Literacy Education*, 4(2), 121-135.
- Büyüköztürk, Ö. (2011). *Sosyal bilimler için veri analizi el kitabı (14. basım)*. Ankara: Pegem Akademi Yayın Dağıtım.
- Büyüköztürk, Ş., Çakmak, E., Akgün, Ö., Karadeniz, Ş., & Demirel, F. (2010). *Bilimsel Araştırma Yöntemleri (7. basım)*. Ankara: Pegem Akademi Yayın Dağıtım.
- BTE, (2019). *Bilişim Teknolojileri Eğitimcileri Derneği*, BT Derslerinin Tarihçesi. URL:<http://bte.org.tr/bt-derslerinin-tarihcesi/> [Erişim Tarihi: 04.05.2019]
- Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Can, A. (2013). *SPSS ile bilimsel araştırma sürecinde nicel veri analizi*. Ankara: Pegem Akademi.
- Chapman, E. N. (1999). *Tutum: En Değerli Varlığımız*, (Çev.: Durmuş, A.), Alfa Yayınevi, İstanbul.
- Chiu, C. F. (2014). Teaching programming concepts to K-12 teachers with Scratch. *Journalism and Mass Communication*, 4(2), 125-132.
- Claypool, M. (2013). Dragonfly: strengthening programming skills by building a game engine from scratch. *Computer Science Education*, 23(2): 112-137.
- Clements, D. H., ve Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051-1058.
- Cooper, S., Dann, W. and Pausch, R. (2003). Using Animated 3D Graphics to Prepare Novices for CS1. *Computer Science Education*. 13(1), 3-30.
- Cooper, S., Dann, W., & Pausch, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15(5), 107-116.
- Cooper, S., Powers, K., McNally, M., Goldman, K.J. Proulx, V. ve Carlisle M. (2006). Tools for teaching introductory programming: What works? *ACM SIGCSE Bulletin*, 38, 560-561.
- Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3), 13-25.
- Çetin, E. (2012). *Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi*. Yayımlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi. Ankara.

- Çilingir, A. (2006). *Fen lisesi ile genel lise öğrencilerinin sosyal becerileri ve problem çözme becerilerinin karşılaştırılması*. Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü, Erzurum.
- Dalton, D. W. (1986). *A Comparison of the Effects of LOGO Use and Teacher-Directed Problem-Solving Instruction on the Problem-Solving Skills, Achievement, and Attitudes of Low, Average, and High Achieving Junior High School Learners*. Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Las Vegas, NV, USA.
- Dasso, A., Funes, A., Riesco, D., Montejano, G., Peralta, M. ve Salgado, C. (2005). Teaching programming. *Proceedings of JEITICS*, 183-186. Educación en Informaticay TICs en Argentina
- Deek F. ve Espinosa, I. (2005). An evolving approach to learning problem solving and program development: The distributed learning model. *International Journal on E-Learning*, 4, 409-426.
- Demirer, V. ve Sak, N. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Dinçer (2018). *6.sınıf öğrencilerine Scratch ve Kodu Game Lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarılarının karşılaştırılması*, Yayınlanmamış Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü, İzmir.
- Dündar, S. (2009). Üniversite öğrencilerinin kişilik özellikleri ile problem çözme becerileri arasındaki ilişkinin incelenmesi. *Dokuz Eylül Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 24(2), 139-150.
- Eckerdal, A. (2009). *Novice programming students' learning of concepts and practise*. Unpublished Doctoral dissertation, Acta Universitatis Upsaliensis.
- Edirisinghe, E. M. (2008). *Teaching students to identify common programming errors using a game*. In Proceedings of the 9th ACM SIGITE conference on Information technology education (95-98).
- Ektiren, M. T. (2014). *Okul yöneticilerinin bilgisayar kullanımına dair tutumları ve öğretimde bilgisayar kullanımını yönlendirme düzeyleri*. Yayınlanmamış Yüksek Lisans Tezi, İstanbul Aydın Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.
- Erdem, E. (2018). *Blok tabanlı ortamlarda programlama öğretimi sürecinde farklı öğretim stratejilerinin çeşitli değişkenler açısından incelenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Başkent Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.
- Erekmekçi, M.ve Fidan, Ş. (2012). Oyunun tasarım platformları: Oyunun eğitim ve kültüre etkisi. *Yaşam Bilimleri Dergisi*, 1(1), 851-861. Web: <http://www.yasambilimleridergisi.com/makale/pdf/1356289798.pdf>
- Erkan, S. (2004). Öğretmenlerin Bilgisayara Yönelik Tutumları Üzerine Bir İnceleme, *Manas Üniversitesi Sosyal Bilimler Dergisi*, 12, 141-145.

- Erol, O. (2015). *Scratch ile programlama öğretiminin bilişim teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisi*. Yayınlanmamış Doktora Tezi, Anadolu Üniversitesi, Eğitim Bilimleri Enstitüsü, Eskişehir.
- Ersoy, H., Madran, R. O. ve Gülbahar, Y. (2011). *Programlama dilleri öğretimine bir model önerisi: Robot programlama*. Akademik Bilişim Konferansı 2011' de sunulan bildiri, İnönü Üniversitesi, Malatya. Web: [ab.org.tr/ab11/kitap/ersoy\\_madran\\_AB11.pdf](http://ab.org.tr/ab11/kitap/ersoy_madran_AB11.pdf).
- Eryılmaz, S. (2003). *Algoritma tasarlama ve programlamaya giriş*. Ankara: Detay Yayıncılık.
- Fadjo, C. L. (2012). *Developing computational thinking through grounded embodied cognition*. Yayınlanmamış Doktora Tezi, Columbia University, ABD.
- Fesakis, G., and Serafeim, K. (2009). *Influence of the familiarization with "Scratch" on future teachers' opinions and attitudes about programming and ICT in education*. Paper presented at the Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education da sunulan bildiri. Paris, Fransa. Web:[doi.org/10.1145/1595496.1562957](https://doi.org/10.1145/1595496.1562957).
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.
- Fidan, A. (2016). *Scratch ile programlama öğretiminde oyunlaştırmanın öğrenci katılımına etkisi*. Yayınlanmamış Yüksek Lisans Tezi, Uludağ Üniversitesi, Bursa.
- Fields, D. A., Kafai, Y. B., & Giang, M. T. (2016). Coding by choice: A transitional analysis of social participation patterns and programming contributions in the online Scratch community. In *Mass collaboration and education*, 209-240. Springer, Cham.
- Forgan, J. W. (2003). *Teaching problem solving through children' s literature*. Connecticut: Teacher Ideas Press.
- Fowler, A. (2012). *Enriching student learning programming through using Kodu*. 3rd Annual Conference of Computing and Information Technology Research and Education New Zealand (CITREnz2012)'da sunulan bildiri. Christchurch, New Zealand. Web: [http://www.citrenz.ac.nz/conferences/2012/pdf/2012CITREnz\\_Fowler01-Kodu.pdf](http://www.citrenz.ac.nz/conferences/2012/pdf/2012CITREnz_Fowler01-Kodu.pdf)
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., ve Solomon, C. (1970). Programming language as a conceptual framework for teaching mathematics. *Newsletter SIGCUE Outlook*, 4(2), 13–17
- Futschek, G. (2006, November). Algorithmic thinking: the key for understanding computer science. *International conference on informatics in secondary schools- evolution and perspectives* (pp. 159-168). Springer, Berlin, Heidelberg.

- Gamgam, H. (1989). *Parametrik olmayan istatistik teknikler*. Ankara: Gazi Üniversitesi Fen-Edebiyat Fakültesi Yayınları.
- Garner, S., Haden, P., & Robins, A. (2005, January). My program is correct but it doesn't run: a preliminary investigation of novice programmers' problems. *Proceedings of the 7th Australasian conference on Computing education-Volume 42* (pp. 173-180). Australian Computer Society, Inc.
- Gauld, A. (2000). *Learn to Program Using Python: A Tutorial for Hobbyists, Self-Starters, and Those Who Want to Learn the Art of Programming with Cdrom*. Addison-Wesley Longman Publishing Co. Inc.
- Gee, J. P. (2005). Learning by design: Good video games as learning machines. *ELearning*, 2, 5-16.
- Genç, Z., & Karakuş, S. (2011, September). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımı Scratch kullanımı*. 5th International Computer & Instructional Technologies Symposium (ICITS), Elazığ, Turkey.
- Ginat, D. (2003). The novice programmers' syndrome of design-by-keyword. *ACM SIGCSE Bulletin*, 35(3), 154-157. Thessaloniki, Greece.
- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. *International Conference on Engineering Education-ICEE* (Vol. 2007).
- Gökoğlu, S. (2017). Programlama eğitiminde algoritma algısı: Bir metafor analizi. *Cumhuriyet International Journal of Education*, 6(1), 1-14.
- Grout, V. ve Houlden, N. (2014). Taking computer science and programming into schools: The Glyndwr/BCS Turing Project. *Procedia – Social and Behavioral Sciences*, 141(25), 680–685.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Gundurao, H. K., Manjunath, N. S., & Nachappa, M. N. (2010). *Computer Technology and Computer Programming*. Mumbai, IND: Global Media.
- Guzdial, M. (2004). Programming environments for novices. *Computer science education research*, 2004, 127-154.
- Gülbahar, Y. ve Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 13(1), 33-50.
- Gültekin, A. (2006). *Psikolojik danışmanlık ve rehberlik öğrencilerinin problem çözme becerilerinin incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi, Sosyal Bilimler Enstitüsü, Erzurum.
- Hadjerrouit, S. (2008). Towards a blended learning model for teaching and learning computer programming: a case study. *Informatics in Education-An International Journal*, 17(3), 181-210.

- Hammond, M., Younie, S., Woollard, J., Cartwright, V., & Benzie, D. (2009). *What does out past involvement with computers in education tell us? A view from the research community*. Warwick University Press.
- Hava, K. (2012). *Eğitsel bilgisayar oyunu tasarlama yönteminin, ilköğretim 4. sınıf öğrencilerinin akademik başarısına etkisi*. Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi, Ankara.
- Helminen, J., & Malmi, L. (2010, October). *Jype-a program visualization and programming exercise tool for Python*. Proceedings of the 5th international symposium on Software visualization, 153-162. ACM.
- Hill, C. (2015). *Programming environments for children: Creating a language that grows with you*. Doctoral dissertation, University of California, Santa Barbara.
- Hsu, H. J. (2014). *Gender differences in Scratch game design*. 2014 International Conference on Information, Business and Education Technology (ICIBET), Taiwan, Web: [http://www.atlantispress.com/php/download\\_paper.php?id=11390](http://www.atlantispress.com/php/download_paper.php?id=11390) adresinden 11 Kasım 2018'de alınmıştır.
- ISTE (2018). *ISTE Standards for Students*. Web: <http://www.iste.org/standards/for-students>.
- İnceoğlu, M. (2010). *Tutum algı iletişim*. İstanbul: Beykent Üniversitesi Yayınevi.
- Jenkins, T. (2002). *On the difficulty of learning to program*. Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 4(2002), 53-58.
- Jones, S. P. (2013). *Computing at school in the UK*. Retrieved from <http://research.microsoft.com/enus/um/people/simonpj/papers/cas/computingatschoolcacm.pdf>.
- Kağıtçıbaşı, Ç. (1988). *İnsan ve insanlar*, Evrim Basım Yayım, İstanbul.
- Kalelioğlu, F. & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 33-50.
- Kalelioğlu, Filiz. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210.
- Karabak, D., & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 163-169.
- Karasar, N. (2005). *Bilimsel araştırma yöntemi*. Ankara: Nobel Yayın Dağıtım.
- Kert, S. B., & Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. *The First International Congress of Educational Research, Çanakkale, Turkey*.

- Kesici, T., & Kocabaş, Z. (2007). *Bilgisayar 2 Ders Kitabı* (2. Baskı). Ankara.
- Kinnunen, P. ve Malmi, L. (2008). *CS minors in a CS1 course*. Proceeding of the Fourth International Workshop on Computing Education Research, ICER (s.79 -90). New York, USA, Web: <http://dl.acm.org/citation.cfm?id=1404529> adresinden 11 Aralık 2018'de alınmıştır.
- Kneeland, S. (2001). *Problem Çözme (cev. N. Kalaycı)*. Ankara: Gazi Kitabevi.
- 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. Web:doi.org/10.1016/j.sbspro.2015.01.651.
- Koohang, A. A. (1989). A study of attitudes toward computers: Anxiety, confidence, liking, and perception of usefulness. *Journal of Research on computing in Education*, 22(2), 137-150.
- Kukul, V., & Gökçearslan, Ş. (2014). *Scratch ile Programlama Eğitimi Alan Öğrencilerin Problem Çözme Becerilerinin İncelenmesi*. 8. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu Edirne. Web: <https://goo.gl/6ucqn1>, adresinden 12 Ağustos 2018'de alınmıştır.
- Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (1986). A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research*, 2(4), 429-458.
- Lahtinen, E., Ala-Mutka, K. ve Järvinen, H. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3), 14-18.
- Lai, A. F., & Yang, S. M. (2011). The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities. *2011 International Conference on Electrical and Control Engineering* (pp. 6940-6944). IEEE.
- Lee, Y.-J. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today*, 34(2), 26-31.
- Lewis, C. M. (2010, March). *How programming environment shapes perception, learning and goals: logo vs. scratch*. Proceedings of the 41st ACM technical symposium on Computer science education (pp. 346-350). ACM.
- Linn, M. C. ve Dalbey, J. (1985). Cognitive consequences of programming instruction: Instruction, access, and ability. *Educational Psychologist*, 20, 191-206.
- Lopez, J. M., Gonzalez, M. R. ve Cano, E. V. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using “Scratch” in five schools. *Computers & Educations*, 97, 129-141. Web: doi.org/10.1016/j.compedu.2016.03.003
- Lye, S. Y., & 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.

- Malan, D. J., & Leitner, H. H. (2007). Scratch for Budding Computer Scientists. *ACM SIGCSE Bulletin*, 39(1): 223-227.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., ve Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. *ACM SIGCSE Bulletin*, 40(1), 367-371.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B. ve Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4).
- Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, 16(3), 211-227.
- Mathew Myers, J., & Halpin, R. (2002). Teachers' attitudes and use of multimedia technology in the classroom: Constructivist-based professional development training for school districts. *Journal of Computing in Teacher Education*, 18(4), 133-140.
- Mayer, R. E. (1981). The psychology of how novices learn computer programming. *Computing surveys*, 13, 121-141.
- McMahon, G. (2009). Critical Thinking and ICT Integration in a Western Australian Secondary School. *Journal of Educational Technology & Society*, 12(4).
- MEB. (2005). *İlköğretim Programı*. Ankara: MEB Yayınları.
- MEB. (2016). *Bilgisayar bilimi dersi öğretim programı, 2015-2016*. Ankara: Milli Eğitim Basımevi.
- MEB. (2017). *Bilişim teknolojileri ve yazılım dergisi öğretim programı, 2016-2017*. Ankara: Milli Eğitim Basımevi.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239-264.
- Monroy-Hernández, A. ve Resnick, M. (2008). Empowering kids to create and share programmable media. *Interactions*, 15(2).
- Moreno, J. (2012). Digital competition game to improve programming skills. *Educational Technology & Society*, 15(3), 288-297.
- Moreno-León, J., Robles, G., & Román-González, M. (2016). Comparing computational thinking development assessment scores with software complexity metrics. *2016 IEEE global engineering education conference (EDUCON)* (pp. 1040-1045). IEEE.
- Nam, D., Kim, Y., & Lee, T. (2010). The effects of scaffolding-based courseware for the Scratch programming learning on student problem solving skill. *ICCE2010*, 723-727.



- Nascimento, M. R., Mendonça, A. P., Guerrero, D. D. S., Figueiredo, J. C. A. (2010, October 27-30). *Teaching Programming for High School Students: A Distance Education Experience*. Paper presented at 40th ASEE/IEEE Frontiers in Education Conference, Washington, DC.
- Nikou, S. A., & Economides, A. A. (2014). Transition in student motivation during a scratch and an app inventor course. *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1042-1045). IEEE.
- Numanoğlu, M. & Keser, H. (2017). Programlama Öğretiminde Robot Kullanımı - Mbot Örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*, 6 (2), 497-515. DOI: 10.14686/buefad.306198
- OECD. (2008). *Trends shaping education*. Centre for Educational Research and Innovation Publications. Retrieved from [http://www.oecd-ilibrary.org/education/trends-shaping-education-2008\\_9789264046627-en](http://www.oecd-ilibrary.org/education/trends-shaping-education-2008_9789264046627-en)
- Oluk, A. ve Korkmaz, Ö. (2016). Comparing student's scratch skills with their computational thinking skills in terms of different variables. *Modern Education and Computer Science*, 11, 1-7.
- Ortiz L.A. (1997). *A framework for visual programming languages*. Yayınlanmamış Yüksek Lisans Tezi, University of Puerto Rico.
- Osman, M. A., Loke, S. P., Zakaria, M. N., & Downe, A. G. (2012, December). Secondary students' perfectionism and their response to different programming learning tools. In *2012 IEEE Colloquium on Humanities, Science and Engineering (CHUSER)*(pp. 584-588). IEEE.
- Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., ve Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, 1479-1482.
- Ozoran, D., Cagiltay, N., & Topalli, D. (2012). Using scratch in introduction to programming course for engineering students. *2nd International Engineering Education Conference (IEEC2012)*, 2, 125-132.
- Öndeş, Ö., (2016, 29 Şubat). *İngiltere ve ABD' de kodlama eğitimi*. Hürriyet. Web: <http://www.hurriyet.com.tr/egitim/ingiltere-ve-abdde-kodlama-egitimi-40061604>.
- Özcan, S., Ergün, K., Köse, Ö., Emir, N. & Gezgin, D. (2017, Nisan). *Bilgisayar programlama eğitiminde scratch programi kullanımına ilişkin lise öğrencilerinin görüşleri*. 2nd International Scientific Researches Congress on Humanities and Social Sciences (IBAD-2017), İstanbul.
- Özçınar, H., Yecan E. ve Tanyeri, T. (2016). Öğretmen Gözüyle Görsel Programlama Öğretimi. *3. Uluslararası Eğitimde Yeni Yönelimler Konferansı*, 71-79.
- Özdener, N. (2008). A comparison of the misconceptions about the time efficiency of algorithms by various profiles of computer programming students. *Computers and Education*, 51, 1094–1102.

- Özkütük, N., Silkü, H. A., Orgun, F., & Yalçınkaya, M. (2003). Öğretmen adaylarının problem çözme becerileri. *Ege Eğitim Dergisi*, 3(2), 1-9.
- Öztürk, S. & Alper, A. (2019). Programlama öğretimindeki ters-yüz öğretim yönteminin öğrencilerin başarılarına, bilgisayara yönelik tutumuna ve kendi kendine öğrenme düzeylerine etkisi. *Bilim, Eğitim, Sanat ve Teknoloji Dergisi (BEST Dergi)*, 3(1), 13-26.
- Penmetcha, M. R. (2012). *Exploring the effectiveness of robotics as a vehicle for computational thinking*. Yayınlanmamış Doktora Tezi, Purdue University.
- Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Parker, J. R. (2016). *Python: An introduction to programming*. Stylus Publishing, LLC.
- Petty, R.E., & Cacioppo, J.T. (1996). *Attitudes and persuasion: Classic and contemporary approaches*. Colorado: Westview Press.
- Pillay, N., & Jugoo, V. R. (2005). An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bulletin*, 37(4), 107-110.
- Polo, B. J. (2013). *SBL-online: Implementing studio-based learning techniques in an online introductory programming course to address common programming errors and misconceptions*. Yayınlanmamış Doktora Tezi, University of Hawaii.
- Porter, R. ve Calder, P. (2004). *Patterns in learning to program - An experiment?* Australian Computing Education Conference. Dunedin: NewZeland.
- Rajaravivarma, R. (2005). A games-based approach for teaching the introductory programming course. *ACM Inroads - The SIGSCE Bulletin*, 37, 98-102.
- Ramadhan, H. A. (2000). Programming by discovery. *Journal of Computer Assisted Learning*, 16, 83-93.
- Regan, G. O. (2008). *A brief history of computing*. Springer Science & Business Media.
- Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. *In Proceedings of the 41st ACM technical symposium on Computer science education*, 265-269.
- Resnick, M. (2013). *Learn to code - code to learn*. Web: <https://goo.gl/K5EN0v> adresinden 21 Aralık 2018'de alınmıştır.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. ve Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- Robins, A., Rountree, J. ve Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.

- Ruf, A., Muhling, A., ve Hubwieser, P. (2014). *Scratch vs. Karel: impact on learning outcomes and motivation*. Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education. Berlin, 50-59. Almanya.
- Saeli, M., Perrenet, J., Jochems, W. M. G. ve Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, 10(1), 73-88
- Sanjanaashree, P., Anand Kumar, M. ve Soman, K. P. (2014). *Language learning for visual and auditory learners using scratch toolkit*. In Computer Communication and Informatics (ICCCI), 2014 International Conference on IEEE. 1-5.
- Saygılı, H. (2000). *Problem çözme becerisi ile sosyal ve kişisel uyum arasındaki ilişkinin incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi, Sosyal Bilimler Enstitüsü Eğitim Bilimleri Anabilim Dalı, Erzurum.
- Saygıner, Ş., ve Tüzün, H. (2017). *Programlama eğitiminde yaşanan zorluklar ve çözüm önerileri*. 5th International Instructional Technologies & Reacher Education Symposium. İzmir.
- Sayın, Z., & Seferoğlu, S. S. (2016). *Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi*. Akademik Bilişim Konferansı, 3-5.
- Schulte, C. ve Bennedsen, J. (2006). *What do teachers teach in introductory programming?* The Second International Computing Education Research Workshop, University of Kent, Canterbury, United Kingdom.
- Schwartz, J., Stagner, J., and Morrison, W. (2006). *Kid' s Programming Language (KPL)*. The 33rd International Conference and Exhibition on Computer Graphics and Interactive Techniques. Boston, Massachusetts.
- Scratch. (2019). *Scratch About*. Web: <https://scratch.mit.edu/about/>, adresinden 11 Kasım 2018'de alınmıştır.
- Scratch for Arduino. (2015). *About S4A*. Web: <http://s4a.cat> adresinden 22 Aralık 2018'de alınmıştır.
- Sebesta, R. W. (2012). *Concepts of programming languages*. Boston: Pearson.
- Senemoğlu, N. (2001). *Gelişim öğrenme ve öğretim*. Ankara: Gazi Kitabevi.
- Shin, S. ve Park, P. (2014). A study on the effect affecting problem solving ability of primary students through the scratch programming. *Advanced Science and Technology Letters*, 117-120. Web: [doi.org/10.14257/astl.2014.59.27](https://doi.org/10.14257/astl.2014.59.27).
- Shin, S., Park, P. ve Bae, Y. (2013). The Effects of an information-technology gifted program on friendship using Scratch programming language and clutter. *International Journal of Computer and Communication Engineering*, 2(3), 246-249.

- Smith, M. B. (1968). Attitude change. *International encyclopedia of the social sciences*, 1, 458–67.
- Smith, M. K. (2002). Howard Gardner and multiple intelligences. *The encyclopedia of informal education*, 15, 2012.
- Serin, O., Bulut Serin, N., & Saygılı, G. (2010). İlköğretim düzeyindeki çocuklar için problem çözme envanteri'nin (ÇPÇE) geliştirilmesi. *İlköğretim Online*, 9(2), 446–458.
- Solmaz, E. (2014). *Programlama dili öğretiminde Alice yazılımının ders başarısı, eleştirel düşünme ve problem çözme becerileri ile üstbilişsel farkındalık düzeyine etkisi*. Yayınlanmamış Doktora Tezi, Gazi Üniversitesi, Ankara.
- Soloway, E. (1986). Learning to program, learning to construct mechanisms and explanations. *Communications of the ACM*, 29, 850-858.
- Soloway, E.ve Spohrer, J. C. (1989). *Studying the novice programmer*. Hillside, N.J. Erlbaum.
- Şahin, (2018). *Ortaokul seviyesinde programlama öğretimi için bir yöntem önerisi*. Yayınlanmamış Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi, Eğitim Bilimleri Enstitüsü, Trabzon.
- Şahin, N., Şahin, N. H., & Heppner, P. P. (1993). Psychometric properties of the problem solving inventory in a group of Turkish university students. *Cognitive Therapy and Research*, 17(4), 379-396.
- Taşdemir, L. (2017). *Özel yetenekli öğrencilerin bilgisayara yönelik tutumları ve teknoloji ile kendi kendine öğrenmeleri arasındaki ilişkinin incelenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Tavşancıl, E. (2002). *Tutumların ölçülmesi ve SPSS ile veri analizi*. Ankara: Nobel Yayıncılık.
- Taylor, M., Harlow, A. ve Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences*, 8, 561-570.
- TDK, (1974). *Türk Dil Kurumu Eğitim Terimleri Sözlüğü*. Web: <http://www.tdk.com.tr> adresinden 22 Aralık 2018'de alınmıştır.
- Tekerek, M. & Altan, T. (2014). The effect of Scratch environment on student's achievement in teaching algorithm. *World Journal on Educational Technology*, 6(2), 132-138.
- Teo, T. (2008). Pre-service teachers' attitudes towards computer use: A Singapore survey. *Australasian Journal of Educational Technology*, 24(4).
- Tezbaşaran, A. A. (1997). *Likert Tipi Ölçek Geliştirme Kılavuzu*, 2. Baskı, Türk Psikologlar Derneği Yayınları, Ankara.

- Topçu, P. (2009). *Cinsiyetin bilgisayar tutumu üzerindeki etkisi: Bir meta analiz çalışması*. Yayınlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü, İstanbul.
- TTKB, (2012). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı*. Web: [https://www.bilgisayarbilisim.net/attachments/bilisimteknolojileri\\_ortaokul-rar.29815/](https://www.bilgisayarbilisim.net/attachments/bilisimteknolojileri_ortaokul-rar.29815/) adresinden 19 Aralık 2018’de alınmıştır.
- Tümkaya, S., ve İflazoğlu, U. A. (2000). Çukurova Üniversitesi Sınıf Öğretmenliği öğrencilerinin otomatik düşünce ve problem çözme düzeylerinin bazı sosyo demografik değişkenlere göre incelenmesi. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 6(6).
- Utting, I., Cooper, S., Kölling, M., Maloney, J. ve Resnick, M. (2010). Alice, greenfoot and scratch –A discussion. *ACM Transactions on Computing Education*, 10(4).
- Ülgen, G. (1994). *Eğitim psikolojisi: Kavramlar, ilkeler, yöntemler, kuramlar ve uygulamalar*. Ankara: Bilim Yayınları.
- Ülküer, N. (1988). Çocuklara problem çözme becerisi nasıl kazandırılır. *Yaşadıkça Eğitim*, 5, 28-31.
- Üstündağ, N. (2001). *Müfredat laboratuvar okullarında görev yapan yönetici ve öğretmenlerin bilgisayar-tutumları ile kaygı düzeyleri arasındaki ilişkinin incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Vatansever, Ö. (2018). *Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerisi üzerindeki etkisinin incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Uludağ Üniversitesi Eğitim Bilimleri Enstitüsü, Bursa.
- Wang, H. Y., Huang, I., & Hwang, G. J. (2014). *Effects of an Integrated Scratch and Project-Based Learning Approach on the Learning Achievements of Gifted Students in Computer Courses*. In *Advanced Applied Informatics (IIAIAI)*, 2014 IIAI 3rd International Conference on (pp. 382-387).
- Wilson, A. and Moffat, D. C. (2010). *Evaluating scratch to introduce younger school children to programming*. *Proceedings of the 22nd Annual Psychology of Programming Interest Group*, Madrid.
- Wing, J. M. (2006). Computational thinking. *Communications of The Acm*, 49(3), 33-35.
- Winslow, L. E. (1996). Programming pedagogy – A psychological overview. *SIGCSE Bulletin*, 28, 17-22.
- Yıldırım, E. (2016). Dijital oyun tasarım programlarının eğitimde önemi. *Mesleki Bilimler Dergisi*, 5(2),12 – 19. Web: <http://dergipark.gov.tr/mbd/issue/34074/377099>.

- Yıldırım, N. (2012). *Yabancı dil eğitiminde eğitsel oyunlar aracılığıyla mobil öğrenme*. Yayınlanmamış Yüksek Lisans Tezi. Fırat Üniversitesi, Eğitim Bilimleri Enstitüsü, Elazığ.
- Yıldız, M. ve Çiftçi, E. (2017). *Bilişimsel Düşünme ve Programlama*.
- Yiğit, N., Alev, N., Altun, T., Özmen, H., ve Akyıldız, S. (2009). *Öğretim teknolojileri ve materyal tasarımı* (Geliştirilmiş 4 Basım). Trabzon: Akademi Kitabevi.
- Yurdugül, H., ve Aşkar, P. (2013). Learning programming, problem solving and gender: A longitudinal study. *Procedia-Social and Behavioral Sciences*, 83, 605-610.
- Yüksel, S., (2017). *Scratch programı öğretiminde ayrılıp birleşme tekniği kullanımının öğrencilerin derse yönelik tutumuna akademik başarısına ve kalıcılığa etkisi*. Yayınlanmamış Yüksek Lisans Tezi, Adnan Menderes Üniversitesi Sosyal Bilimler Enstitüsü, Aydın.
- Yükseltürk, E. ve Altıok, S. (2016). BT öğretmen adayları tarafından scratch görsel programlama aracı ile geliştirilen eğitsel oyunların incelenmesi. *SDU International Journal of Educational Studies*, 3(1), 59-66. Web: <http://dergipark.org.tr/sduijes/issue/20865/223885>
- Yükseltürk, E. ve Altıok, S. (2016). Bilişim teknolojileri öğretmen adaylarının programlama öğretiminde scratch aracının kullanımına ilişkin algıları. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 12(1), 39-52.
- Yünkül, E., Durak, G., Çankaya, S. ve Mısırlı, Z. A. (2017). The effects of Scratch software on students' computational thinking skills. *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi*, 11(2), 502-517. Web: doi: 10.17522/balikesirnef.373424.
- Zak, D. (2013). *Introduction to programming with C++*. Cengage Learning.

## EKLER

### Ek-1. Çocuklar için Problem Çözme Envanteri Kullanım İzni

**Oguz Serin** <oguzserin@gmail.com>

Alıcı: ben ▾

Merhaba

Öncelikle şahsım ve çalışma arkadaşlarım adına ölçeği kullanma izninizi kabul ediyoruz. Ölçeği kullanmanızda hiçbir sakınca yoktur. Ölçeğe ilişkin bilgilere ve puanlamasına kişisel web sayfasından ulaşabilirsiniz <http://kisi.deu.edu.tr/oguz-serin/>. İyi çalışmalar diliyorum.

Oğuz Serin

yunus alp <yalpmyo@gmail.com>

\*\*\*

Sayın Prof.Dr.Oğuz SERİN,

2010 yılında yayınlamış olduğunuz **Çocuklar için Problem Çözme Envanteri**'ni yürütmekte olduğum "**Blok Tabanlı(Scratch) Programlama Öğretiminin Ortaokul Öğrencilerinin Problem Çözme Becerisine ve Bilgisayara Yönelik Tutumuna Etkisi**" adlı tez çalışmamda kullanmak üzere izninizi talep eder, teşekkürlerimi şimdiden sunarım.

Saygılarımla..

**Yunus ALP**

Yüksek Lisans, İnönü Üniversitesi Eğitim Bilimleri Enstitüsü  
Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı

--

**Prof. Dr. Oğuz SERİN**

Lefke Avrupa Üniversitesi

Dr. Fazıl Küçük Eğitim Fakültesi Dekanı

Lefke/KKTC

## Ek-2. Çocuklar için Problem Çözme Envanteri

Sevgili öğrenciler;

Bu çalışmanın amacı sizin problem çözme beceriniz hakkında bilgi edinmektir. Doğru veya yanlış cevap yoktur. Her cümleyle ilgili görüş belirtirken önce cümleyi dikkatle okuyunuz ve cümlede belirtilen durumun size ne derecede uygun olduğuna karar veriniz. Ardından size en uygun kutucuğa X işareti koyarak cevaplandırınız. Lütfen boş soru bırakmayınız.

“(1)Hiçbir zaman böyle davranmam”

“(4) Sık sık böyle davranırım”

“(2)Ender olarak böyle davranırım”

“(5)Her zaman böyle davranırım ”

“(3) Arada sırada böyle davranırım”

		Hiçbir Zaman	Ender olarak	Arada sırada	Sık sık	Her zaman
1	Sorunlarımdan kaçma yerine sorunumu çözmeye çalışırım.					
2	Ne zaman sorun yaşasam içimde hep bir karamsarlık olur ve kendimi kolay kolay toplayamam.					
3	Karşıma sorunlar çıktığında sakin olmaya çalışırım.					
4	Kafama bir şeyler takıldığında sinirli olurum ve istemediğim sözler söylerim.					
5	Yaşadığım problemlerin herkesin başına gelebileceğine inanırım.					
6	Başıma bir problem geldiğinde çabucak üzülürüm.					
7	Sorun yaşadığımda onu çözmek için bulduğum çözüm yolu işe yarayana kadar vazgeçmem.					
8	Sorun yaşadığımda uzun süre etkisinden kurtulamam.					
9	Sorunlarım olduğunda hep kendi kendime sorular sorarım ve çözüm yolları ararım.					
10	Sorunlarımı çözemediğim zaman her şeyden soğurum.					
11	Karşılaştığım sorunlardan kurtulmak için vazgeçmeden bütün çözüm yollarını denerim.					
12	Sorun yaşadığımda kendimi kolay kolay derse veremem.					
13	Öncelikle sorunlarımdan neden kaynaklandığını bulmaya çalışırım.					
14	Arkadaşlarımla sorun yaşadığımda konuşmak yerine kavga ederim.					
15	Sorunlardan kaçmak yerine işe yarayan bir çözüm yolu bulana kadar uğraşırım.					
16	İş ve sorumluluklarımdan kaçmak için bir çok bahane uydururum.					
17	Sorunlar karşısında oldukça sabırlı ve kararlı davranırım.					
18	Bir sorunum olduğunda ne yaparsam yapayım çözülmeyeceğini düşünürüm.					
19	Sorunlarımı çözemediğimde zamanlarda ailemden ya da arkadaşlarımdan yardım isterim.					
20	Sorunlarımı çözme konusunda genellikle başarılı değilimdir.					
21	Sorunlarım karşısında genellikle yaratıcı ve etkili çözüm yolları bulurum.					
22	Sorunlarım olduğunda küçük çocuk gibi davranmak beni rahatlatır.					
23	Bir sorunla karşılaştığımda tüm çözüm yollarını düşünerek çözeceğime inanırım.					
24	Bir sorunum olduğunda çözüm yolları aramak yerine her şeyi oluruna bırakırım.					



### Ek-3. Bilgisayara Yönelik Tutum Ölçeği

Sevgili öğrenciler;

Aşağıda yer alan ölçek sizin bilgisayara yönelik tutumunuzu belirlemek amacıyla hazırlanmıştır. Ölçekte bilgisayara yönelik tutum cümleleri ile her cümlenin karşısında **Tamamen Katılıyorum**, **Çoğunlukla Katılıyorum**, **Orta Düzeyde Katılıyorum**, **Az Katılıyorum**, **Katılmıyorum** seçenekleri yer almaktadır. Her cümleyi dikkatlice okuduktan sonra kendiniz için en uygun seçeneği işaretleyiniz. Lütfen boş soru bırakmayınız.

	Katılmıyorum	Az Katılıyorum	Orta Düzeyde Katılıyorum	Çoğunlukla Katılıyorum	Tamamen Katılıyorum
1.Günümüzde birçok iş bilgisayar kullanmayı gerektiriyor.					
2.En kısa zamanda bilgisayar kullanmayı isterim.					
3.Bilgisayar yolu ile öğrenmeyi seviyorum.					
4.Param olsa hemen bir bilgisayar alırım					
5.Sınıfta bir bilgisayarın olması benim için eğlenceli olur.					
6.Bilgisayar işlerinden hoşlanmam.					
7.Eğitim ve öğretimde bilgisayardan yararlanılmalıdır.					
8.Bilgisayar eğitimin kalitesini artırır.					
9.Bilgisayarların yaygınlaştırılması insanların zararındadır.					
10.Gerektiği gibi kullanılırsa bilgisayar iş verimini artırır.					
11.Bilgisayar birçok işi çok çabuk sonuçlandırdığı için zaman ve enerji kazandırır.					
12.Bilgisayarlar beni sinirlendirir.					
13.Bilgisayar kullanmayı öğrenmek benim için sıkıcı olur.					
14.Bilgisayar kullanmayı gerektiren işlerde çalışmak istemem.					
15.Bilgisayarın başına geçtiğimde zamanın nasıl geçtiğini anlamam.					
16.İnsanlar bilgisayarlardan nasıl hoşlanıyorlar anlamıyorum.					
17.Bilgisayar ile ilgili çalışmaktan zevk alırım.					
18.Bilgisayar toplumu robotlaştırıyor.					
19.Bilgisayarlar hayatı daha eğlenceli hale getiriyor.					
20.Saatlerce bilgisayarın başında oturmak beni çok sıkır.					
21.Bilgisayar yoluyla öğrenmek öğrenmeyi kolaylaştırır.					
22.Bence bilgisayarlar yaratıcılığı köreltiyor.					
23.Bilgisayarlar yüzünden insanlar tembelleşeceklerdir.					
24.Bilgisayarların hayatımızdaki rolü önemlidir.					

### Ek-4. Kontrol Grubu Ders İeriđi

Hafta	İerik	Kazanım
Hafta 1	1. Etik, İnternet Etiđi 2. Siber Zorbalık, Dijital Zorbalık	<ul style="list-style-type: none"> <li>✓ İnternet etiđinin önemini ifade eder.</li> <li>✓ Etik ilkelerin ihlali sonucunda karřılařılabilecek durumlara örnekler verir.</li> <li>✓ Sanal ortamlarda da dođru ve dürüst olunması gerektiđi vurgulanır.</li> <li>✓ Siber zorbalık kavramını açıklayarak korunma amacıyla alınabilecek önlemleri tartıřır.</li> <li>✓ Sanal ortamda karřılařılabilecek olumsuz davranıřlara karřı duyarlı davranılması üzerinde durulur.</li> </ul>
Hafta 2	1. Telif Hakları, Korsan Ürün 2. Lisans, Lisans Türleri, Beta Sürüm, Freeware, Shareware, Demo 3. Açık Eriřim Felsefesi, Yaratıcı Birliktelik (Creative Commons)	<ul style="list-style-type: none"> <li>✓ Telif hakkı kavramını ve önemini arařtırır.</li> <li>✓ Kullanım haklarını düzenleyen lisans türlerini açıklar.</li> <li>✓ Açık eriřim felsefesi ve etik kullanım üzerinde durulur.</li> </ul>
Hafta 3	1. Biliřim Suları, İnternet Bilgi İhbar Merkezi 2. Siber Su, Siber Dünya	<ul style="list-style-type: none"> <li>✓ Biliřim sularının neler olduđunu tanımlayarak ilgili kanunları özetler.</li> <li>✓ Biliřim sularına karřı alınabilecek önlemler ve stratejiler geliřtirir.</li> <li>✓ İnternet Bilgi İhbar Merkezi hakkında bilgi verilir.</li> </ul>
Hafta 4	1. Dijital Yurttařlık, Dijital Ayak İzi, Kültürel Yozlařma 2. Oyun, Oyun Türleri, Oyun Platformları	<ul style="list-style-type: none"> <li>✓ Biliřim teknolojilerinin sosyal ve kültürel hayata katkıları ve riskleri örnekler üzerinden tartıřır.</li> <li>✓ Biliřim teknolojileri kullanılarak kültürler arası etkileřim olabileceđi gibi kültürel bozulmaların da olabileceđi ifade edilir.</li> <li>✓ Dijital Vatandařlık</li> <li>✓ Dijital paylařımların kendisi ve bařkaları üzerindeki etkilerini fark eder.</li> <li>✓ Dijital ayak izinin, kendisinden geride izler bıraktıđı vurgulanır.</li> <li>✓ Öđrenci, biliřsel ve ahlaki geliřimine uygun olan dijital oyun ve ierikleri ayırt eder.</li> <li>✓ Öđrencinin bilinli bir kullanıcı olması için öz kontrol becerisini geliřtirmesi sađlanır.</li> </ul>
Hafta 5	1. EBA, Bilgi Gizliliđi, Güvenlik, 2. Bilgi Koruma, Eriřilebilirlik, Veri Bütünlüđü, Gizlilik 3. Őifreleme, Veri Güvenliđi	<ul style="list-style-type: none"> <li>✓ Biliřim teknolojilerinin kullanımında gizlilik ve güvenlik boyutlarının önemini tartıřır.</li> <li>✓ Gizlilik, bütünlük, eriřilebilirlik gibi kavramlara deđinilir.</li> <li>✓ Güvenlik açıklarının oluřumu konusunda yorum yapar.</li> <li>✓ Bilgi koruma yöntemlerini ifade eder.</li> <li>✓ Bilgi paylařımı sürecinde olası riskleri deđerlendirerek alınabilecek önlemleri tartıřır.</li> </ul>
Hafta 6	1. Virüs, Zararlı Yazılım, Yayılma, Bulařma, Zarar Verme 2. Virüsten Korunma, Güvenlik Duvarı, Antivirüs, Kare Kod, Zararlı Yazılım, Koruyucu Yazılım	<ul style="list-style-type: none"> <li>✓ Zararlı yazılımları kavrar.</li> <li>✓ Virüs, spam, truva atı vb. zararlı yazılımlardan bahsedilir.</li> <li>✓ Güvenlik yazılımlarının kullanım amalarını açıklar.</li> </ul>
Hafta 7	1. Bilgisayar Ađları, Ađ Bileřenleri, Donanım ve Yazılım, Ađ Kartı, Ađ Kablosu	<ul style="list-style-type: none"> <li>✓ Ađ kurmak için gerekli bileřenleri ve bileřenlerin özelliklerini açıklar.</li> <li>✓ Donanım ve yazılım bileřenlerine vurgu yapılır.</li> <li>✓ Bir ađdan dosya ve yazıcı paylařımı yapar.</li> <li>✓ Bilgisayar ađlarının boyutlarına ve bileřenlerine iliřkin farklılıkların nedenlerini tartıřır.</li> </ul>
Hafta 8	1. Arama Motorları 2. Bilgi Yönetimi, Bilgi Kirliliđi 3. EBA	<ul style="list-style-type: none"> <li>✓ Arama motorlarını kullanarak ileri düzeyde arařtırma yapar.</li> <li>✓ Bilgiye ulařırken zararlı ve gereksiz ierikleri ayırt eder.</li> <li>✓ Ulařılan bilgilerin dođruluđu konusunda Őüpheli yaklařımlarla farklı kaynaklardan sorgulama yapmaları sađlanır.</li> <li>✓ Bilgi yönetimi kavramını ve önemini ifade eder.</li> <li>✓ Bilgi kirliliđi konusunda duyarlı davranmanın gerekliliđi vurgulanır.</li> <li>✓ EBA üzerinden farklı ieriklere eriřim sađlar.</li> </ul>

### Ek-5. Deney Grubu Ders İçeriği

Hafta	İçerik	Kazanım
<b>Hafta 1</b>	<ol style="list-style-type: none"> <li>1. Problem Çözme Ve Algoritma</li> <li>2. Akış Şeması, Tekrarlı Yapı, Karar Yapı, Doğrusal Yapı</li> <li>3. Yönergeleri Takip Etme, İşlem Basamakları</li> </ol>	<ul style="list-style-type: none"> <li>✓ Günlük hayatta karşılaştığı problemlere çözüm önerileri getirir.</li> <li>✓ Verilen bir problemi uygun adımları kullanarak çözer.</li> <li>✓ Bir problemi çözmek için farklı algoritmalar tasarlanabileceği vurgulanır.</li> <li>✓ Problem çözme sürecinde takip edilmesi gereken adımları fark eder.</li> <li>✓ Verilen bir problemi analiz eder.</li> <li>✓ Problemi çözmek için gerekli değişken, sabit ve işlemleri açıklar.</li> <li>✓ Akış şeması bileşenlerini ve işlevlerini açıklar.</li> <li>✓ Bir algoritma için akış şeması çizer.</li> </ul>
<b>Hafta 2</b>	<ol style="list-style-type: none"> <li>1. Scratch Arayüz ve Yapısı: Proje Ekranı (Sahne), Blok Paketi (Diziler), Kodlama Alanı, Menü Çubuğu, Kuklalar (Karakterler), Kılıklar (Kostümler), Sesler</li> <li>2. Karaktere Komutlar Verme</li> </ol>	<ul style="list-style-type: none"> <li>✓ Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanıır.</li> <li>✓ Açık kaynak kodlu veya ücretsiz erişilebilen programlama platformları kullanılabilir.</li> <li>✓ Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.</li> </ul>
<b>Hafta 3</b>	<ol style="list-style-type: none"> <li>1. Sahne çizimi ve Karakter seçimi</li> <li>2. Olaylar Blokları <ol style="list-style-type: none"> <li>2.1. ... Tuşa Basılınca</li> </ol> </li> <li>3. Kontrol Blokları <ol style="list-style-type: none"> <li>3.1. Sürekli Tekrarla</li> </ol> </li> <li>4. Hareket Blokları <ol style="list-style-type: none"> <li>4.1. ... Adım git</li> <li>4.2. ... Yönüne dön</li> </ol> </li> </ol>	<ul style="list-style-type: none"> <li>✓ Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.</li> <li>✓ Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar.</li> <li>✓ Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.</li> </ul>
<b>Hafta 4</b>	<ol style="list-style-type: none"> <li>1. Sıralama ve Döngü</li> <li>2. Dekor, Kukla ve Kılık Kütüphanesi</li> <li>3. Olaylar Blokları <ol style="list-style-type: none"> <li>3.1. ... Tuşa Basılınca</li> </ol> </li> <li>4. Kontrol Blokları <ol style="list-style-type: none"> <li>4.1. Sürekli Tekrarla</li> <li>4.2. ... Saniye bekle</li> </ol> </li> <li>5. Hareket Blokları <ol style="list-style-type: none"> <li>5.1. Kenara geldiysen sek</li> <li>5.2. ... Adım git</li> <li>5.3. Kuklanın şekli sağa-sola dönebilsin</li> </ol> </li> <li>6. Görünüm Blokları <ol style="list-style-type: none"> <li>6.1. Sonraki Kılık</li> </ol> </li> </ol>	<ul style="list-style-type: none"> <li>✓ Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar.</li> <li>✓ Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.</li> <li>✓ Doğrusal mantık yapısını içeren programlar oluşturur.</li> <li>✓ Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.</li> <li>✓ Döngü yapısını içeren programlar oluşturur.</li> <li>✓ Döngü yapısını içeren programları test ederek hatalarını ayıklar.</li> </ul>
<b>Hafta 5</b>	<ol style="list-style-type: none"> <li>1. Olaylar Blokları <ol style="list-style-type: none"> <li>1.1. Yeşil bayrak tıklanınca</li> <li>1.2. ... Haberini sal</li> <li>1.3. ... Haberini sal ve bekle</li> <li>1.4. ... Haberi gelince</li> </ol> </li> <li>2. Görünüm Blokları <ol style="list-style-type: none"> <li>2.1. .... de .... saniye</li> <li>2.2. Kılığına geç</li> </ol> </li> <li>3. Ses Blokları <ol style="list-style-type: none"> <li>3.1. ... Sesini bitene kadar çal</li> </ol> </li> </ol>	<ul style="list-style-type: none"> <li>✓ Blok tabanlı programlama ortamında sunulan hedeflere ulaşmak için doğru algoritmayı oluşturur.</li> <li>✓ Doğrusal mantık yapısını uygular.</li> <li>✓ Döngü mantık yapısını uygular.</li> </ul>
<b>Hafta 6</b>	<ol style="list-style-type: none"> <li>1. Değişken Oluşturma, Koşul kullanma, Girdi-Çıktı Kontrolü, Prosedür (Procedure)</li> <li>2. Olaylar Blokları <ol style="list-style-type: none"> <li>2.1.1. Yeşil bayrak tıklanınca</li> <li>2.1.2. Bu kukla tıklanınca</li> </ol> </li> </ol>	<ul style="list-style-type: none"> <li>✓ Doğrusal mantık yapısını içeren programlar oluşturur.</li> <li>✓ Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.</li> <li>✓ Karar yapısını içeren programlar oluşturur.</li> <li>✓ Karar yapısını içeren programları test ederek</li> </ul>

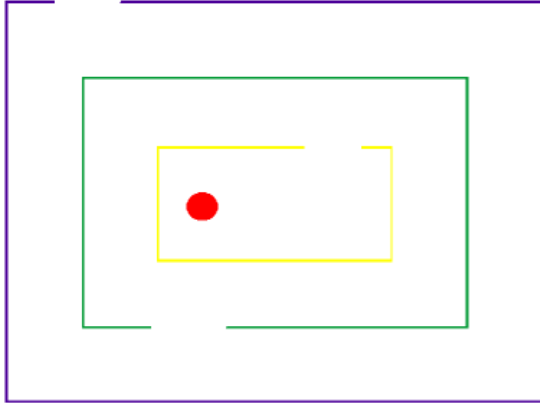
	<p>3.Kontrol Blokları</p> <p>3.1. ... tuşuna basılınca</p> <p>3.2. .. defa tekrarla</p> <p>3.3. Eğer ... ise</p> <p>3.4. ... saniye bekle</p> <p>3.5. ... olana kadar bekle</p> <p>4.Hareket Blokları</p> <p>4.1. y'yi ... arttır</p> <p>4.2. ... sn.de x: ... y: ... a süzül</p> <p>5.Görünüm Blokları</p> <p>5.1. .... de .... saniye</p> <p>5.2. Kılığına geç</p> <p>5.3. Dekoruna geç</p> <p>5.4. Gizlen</p> <p>6.Veri Blokları</p> <p>6.1. Bir değişken oluştur</p> <p>6.2. Değişken, ... olsun</p> <p>7.Algılama Blokları</p> <p>7.1. ... diye sor ve bekle</p> <p>7.2. Yanıt</p> <p>7.3. ... tuşu basılı mı?</p> <p>8.İşlemler Blokları</p> <p>8.1. ... ile ... birleştir</p> <p>8.2. ... = ...</p> <p>9.Özel taşlar Blokları</p> <p>9.1. Bir taş oluştur</p>	<p>hatalarını ayıklar.</p> <p>✓ Çoklu karar yapıları içeren programlar oluşturur.</p> <p>✓ Çoklu karar yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Döngü yapısını içeren programlar oluşturur.</p> <p>✓ Döngü yapısını içeren programları test ederek hatalarını ayıklar.</p>
<b>Hafta 7</b>	<p>1.Fare Olayları (Mouse Events)</p> <p>2.Sahne koordinat düzlemi, x ve y Konumları</p> <p>3.Olaylar Blokları</p> <p>3.1. Yeşil bayrak tıklanınca</p> <p>3.2. Bu kukla tıklanınca</p> <p>3.3. ... haberini sal</p> <p>3.4. ... haberi gelince</p> <p>4.Kontrol Blokları</p> <p>4.1. Sürekli tekrarla</p> <p>4.2. Eğer ... ise ... değilse</p> <p>4.3. ... defa tekrarla</p> <p>4.4. ... saniye bekle</p> <p>5.Kalem Blokları</p> <p>5.1. Klemi bastır</p> <p>5.2. Klemi kaldır</p> <p>5.3. Kalem kalınlığını ... yap</p> <p>5.4. Temizle</p> <p>5.5. Kalem rengini ... yap</p> <p>6.Hareket Blokları</p> <p>6.1. ... adım git</p> <p>6.2. ... derece dön</p> <p>6.3. Fare okuna git</p> <p>7.Algılama Blokları</p> <p>7.1. Fareye basılı mı?</p> <p>7.2. Farenin y'si</p> <p>8.Görünüm Blokları</p> <p>8.1. Kılığına geç</p> <p>9.İşlemler Blokları</p> <p>9.1. ... ve ...</p>	<p>✓ Doğrusal mantık yapısını içeren programlar oluşturur.</p> <p>✓ Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Karar yapısını içeren programlar oluşturur.</p> <p>✓ Karar yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Çoklu karar yapıları içeren programlar oluşturur.</p> <p>✓ Çoklu karar yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Döngü yapısını içeren programlar oluşturur.</p> <p>✓ Döngü yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Bir algoritmayı uyarlamak için en uygun karar yapılarını seçer.</p> <p>✓ Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.</p>
<b>Hafta 8</b>	<p>1.Nesne Kontrolü, Grafik Kontrol, Olay Kontrolü, Çoklu Ortam</p> <p>2.Operatörler, Karar Verme ve Kontrol İşlemleri</p>	<p>✓ Doğrusal mantık yapısını içeren programlar oluşturur.</p> <p>✓ Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.</p>

<p>3. Olaylar</p> <p>3.1. Yeşil bayrak tıklanınca</p> <p>3.2. ... haberini sal</p> <p>3.3. ... haberi gelince</p> <p>4. Kontrol Blokları</p> <p>4.1. Sürekli tekrarlar</p> <p>4.2. Eğer ... ise</p> <p>4.3. ... defa tekrarlar</p> <p>4.4. ... saniye bekle</p> <p>5. Hareket Blokları</p> <p>5.1. Rastgele konumu'na git</p> <p>5.2. y ... olsun</p> <p>5.3. y'yi arttır</p> <p>5.4. y konumu</p> <p>5.5. x'i arttır</p> <p>5.6. ... yönüne dön</p> <p>5.7. ... adım git</p> <p>5.8. ... derece dön</p> <p>5.9. Kenara geldiysen sek</p> <p>5.10. x: ... y: ... noktasına git</p> <p>5.11. Fare oku'na doğru dön</p> <p>6. Algılama Blokları</p> <p>6.1. ... tuşa basılı mı?</p> <p>6.2. ... a değdi mi?</p> <p>6.3. ... rengi ... rengine değdi mi?</p> <p>6.4. Fare oku'na mesafe</p> <p>7. İşlemler Blokları</p> <p>7.1. ... &lt; ...</p> <p>7.2. ... &gt; ...</p> <p>7.3. ... ile ... arasında bir sayı tut</p> <p>8. Ses Blokları</p> <p>8.1. ... sesini çal</p> <p>9. Veri Blokları</p> <p>9.1. Bir değışken oluşturun</p> <p>9.2. Değışken 'i ... arttır</p> <p>9.3. Değışken, ... olsun</p> <p>10. Görünüm Blokları</p> <p>10.1. Görün</p> <p>10.2. Gizlen</p> <p>10.3. ... kılığına geç</p>	<p>✓ Karar yapısını içeren programlar oluşturur.</p> <p>✓ Karar yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Çoklu karar yapıları içeren programlar oluşturur.</p> <p>✓ Çoklu karar yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Döngü yapısını içeren programlar oluşturur.</p> <p>✓ Döngü yapısını içeren programları test ederek hatalarını ayıklar.</p> <p>✓ Bir algoritmayı uyarlamak için en uygun karar yapılarını seçer.</p> <p>✓ Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.</p>
---	---

## Ek-6. Scratch Etkinlikleri

### Hafta 2



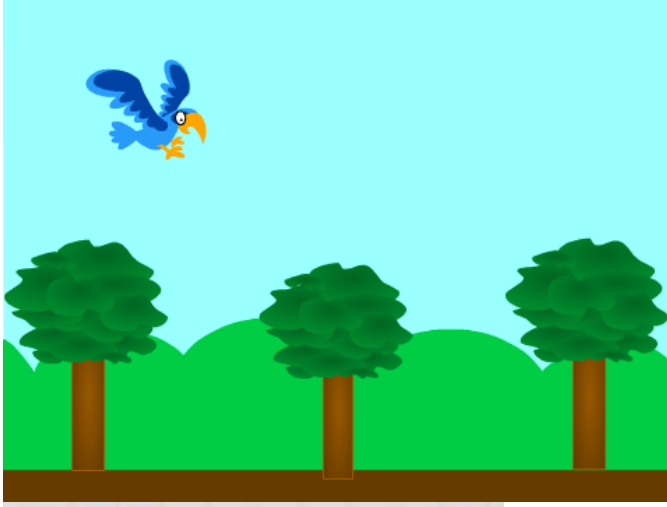
**Hafta 3**

The code consists of four event-driven blocks, each triggered by a key press:

- yukarı ok tuşu basılınca** (up arrow key pressed):
  - 0° yönüne dön (turn to 0 degrees)
  - 10 adım git (move 10 steps)
- sağ ok tuşu basılınca** (right arrow key pressed):
  - 90° yönüne dön (turn to 90 degrees)
  - 10 adım git (move 10 steps)
- aşağı ok tuşu basılınca** (down arrow key pressed):
  - 180° yönüne dön (turn to 180 degrees)
  - 10 adım git (move 10 steps)
- sol ok tuşu basılınca** (left arrow key pressed):
  - 90° yönüne dön (turn to -90 degrees)
  - 10 adım git (move 10 steps)

Coordinates: x: 151, y: 159

## Hafta 4



boşluk tuşu basılınca

sürekli tekrarla

sonraki kılık

10 adım git

kenara geldiysen sek

0.25 saniye bekle

kuklanın şekli sağa-sola dönebilsin

x: -111  
y: 93



## Hafta 5




  
 tıklanınca
   
 Selam! de 2 saniye
   
 Selam! haberini sal ve bekle
   
 Penguenlerin en sevdiği yiyecek nedir biliyor musun? de 2 saniye
   
 Soru 1 haberini sal ve bekle
   
 Hayır de 2 saniye
   
 Cevap 1 haberini sal ve bekle
   
 Buzburger de 2 saniye
   
 Cevap 2 haberini sal
   
 x: -95
   
 y: -75


  
 Selam! haberi gelince
   
 gobo-b kılığına geç
   
 Selam! de 2 saniye
   
 x: 55
   
 y: -59

Soru 1 haberi gelince
   
 Balık? de 2 saniye

Cevap 1 haberi gelince
   
 Ne peki? de 2 saniye

Cevap 2 haberi gelince
   
 gobo-c kılığına geç
   
 laugh-female sesini bitene kadar çal

## Hafta 6



```

when this puppet is clicked
  space → dekoruna geç
  nano-a → kılıfına geç
  Adın ne? diye sor ve bekle
  isim , yanıt olsun
  Merhaba ile isim i birleştir de 2 saniye
  Okula gidiyor musun ile isim i birleştir diye sor ve bekle
  eğer yanıt = Evet ise
    Harika de 2 saniye
  değilse
    Himm de 2 saniye
  Ben şimdi aya gidiyorum benimle gelmek ister misin? diye sor ve bekle
  eğer yanıt = Evet ise
    moon → dekoruna geç
    nano-b → kılıfına geç
    zıpla
  tanımla zıpla
  5 defa tekrarla
    y'yi 10 arttır
    0.1 saniye bekle
    y'yi -10 arttır
    0.1 saniye bekle
  
```



**tıklanınca**

Merhaba arkadaşlar! de 2 saniye

Biliyorsunuz yarın 23 Nisan, de 3 saniye

Okulumuzu hep beraber süsleyelim mi? de 2 saniye

1 sn.de x: -200 y: -120 a süzül

Önce "1" ve "2" tuşlarına basarak bayraklarımızı asalım, de 2 saniye

2 tuşu basılı (mı?) olana kadar bekle

1 saniye bekle

Şimdi "3"e basarak afişimizi asalım, de 2 saniye

3 tuşu basılı (mı?) olana kadar bekle

1 saniye bekle

Son olarak "4"e basarak balonlarımızı asalım, de 2 saniye

4 tuşu basılı (mı?) olana kadar bekle

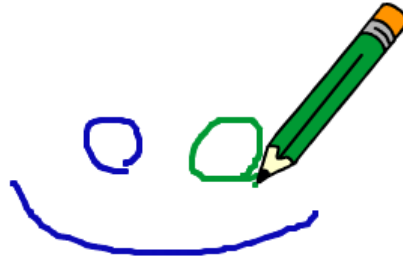
1 saniye bekle

Teşekkür ederim arkadaşlar, de 2 saniye

x: -200  
y: -120

<p><b>tıklanınca</b></p> <p>gizlen</p> <p>x: -214 y: 67</p> <p>1 tuşu basılınca</p> <p>görün</p>	<p><b>tıklanınca</b></p> <p>gizlen</p> <p>x: 213 y: 67</p> <p>2 tuşu basılınca</p> <p>görün</p>
<p><b>tıklanınca</b></p> <p>gizlen</p> <p>x: -1 y: 100</p> <p>3 tuşu basılınca</p> <p>görün</p>	<p><b>tıklanınca</b></p> <p>gizlen</p> <p>x: 88 y: -30</p> <p>4 tuşu basılınca</p> <p>görün</p>

## Hafta 7



Scratch script for drawing a smiley face:

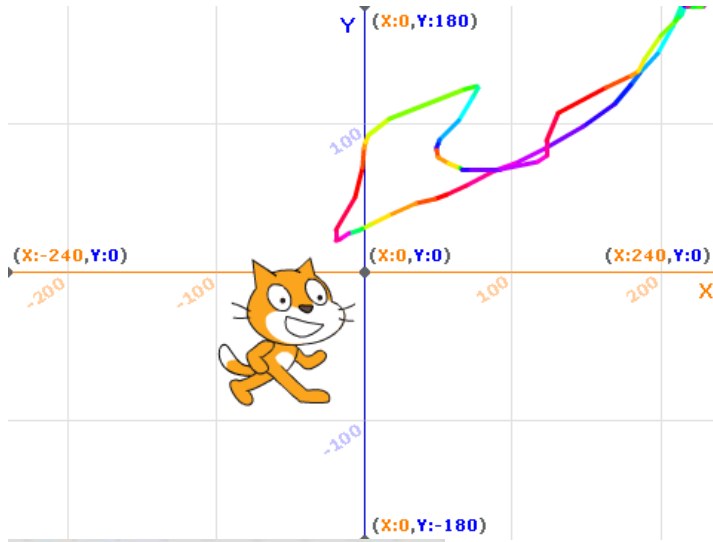
- When clicked:**
  - temizle
  - mavikalem kılıfına geç
  - kalem rengini yap
  - sürekli tekrarla
    - kalem kalınlığını Kalınlık yap
    - fare oku 'na git
    - eğer fareye basılı mı? ve farenin y'si > -125 ise
      - kalem bastır
    - değilse
      - kalem kaldır
- When green flag clicked:**
  - Yeşil haberi gelince
    - yeşilkalem kılıfına geç
    - kalem rengini yap
  - Mavi haberi gelince
    - mavikalem kılıfına geç
    - kalem rengini yap
  - Silgi haberi gelince
    - silgi kılıfına geç
    - kalem rengini yap

Coordinates: x: 224, y: 165

Scratch script for drawing a smiley face:

- When green flag clicked:**
  - bu kukla tıklanınca
    - Yeşil haberini sal
  - bu kukla tıklanınca
    - Silgi haberini sal
  - bu kukla tıklanınca
    - Mavi haberini sal
  - bu kukla tıklanınca
    - temizle

Coordinates: x: -181, y: -158; x: 139, y: -167; x: -122, y: -158; x: 204, y: -148

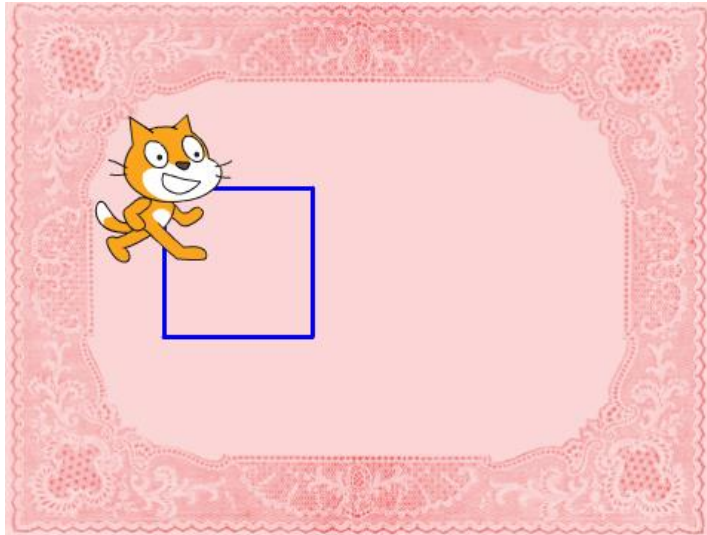


```

boşluk tuşu basınca
temizle
kalem kalınlığını 3 yap
kalemi bastır
sürekli tekrarla
  fare oku 'na git
  kalem rengini 10 arttır

```

x: -52  
y: -39



```

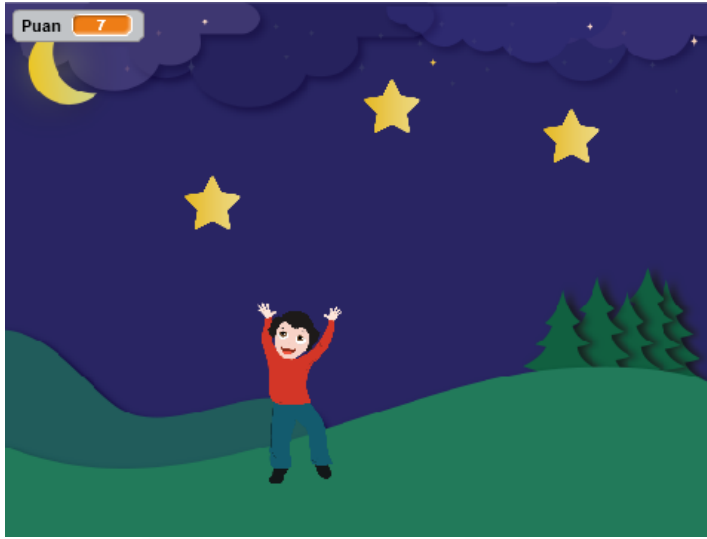
tıklanınca
temizle
kalemi bastır
kalem kalınlığını 3 yap
4 defa tekrarla
  100 adım git
  90 derece dön
  1 saniye bekle

```

x: -133  
y: 55



## Hafta 8



**tıklanınca**

rastgele konum 'na git

y, 180 olsun

**sürekli tekrarla**

y'yi -5 arttır

eğer y konumu < -170 ise

rastgele konum 'na git

y, 180 olsun

**tıklanınca**

Puan , 0 olsun

**sürekli tekrarla**

eğer Çocuk a değdi (mı?) ise

Puan 'i 1 arttır

pop sesini çal

rastgele konum 'na git

y, 180 olsun

x: 21  
y: 110

**tıklanınca**

**sürekli tekrarla**

eğer sağ ok tuşu basılı (mı?) ise

x'i 10 arttır

eğer sol ok tuşu basılı (mı?) ise

x'i -10 arttır

x: -43



```

tıklanınca
görün
90 yönüne dön
sürekli tekrarla
2 adım git
-20 ile 20 arasında bir sayı (tut) derece dön
kenara geldiysen sek
eğer [red] rengi [brown] rengine değdi (mı?) ise
yakalandım haberini sal
gizlen
3 saniye bekle
x: -200 y: -200 ile 200 arasında bir sayı (tut) noktasına git
görün

```

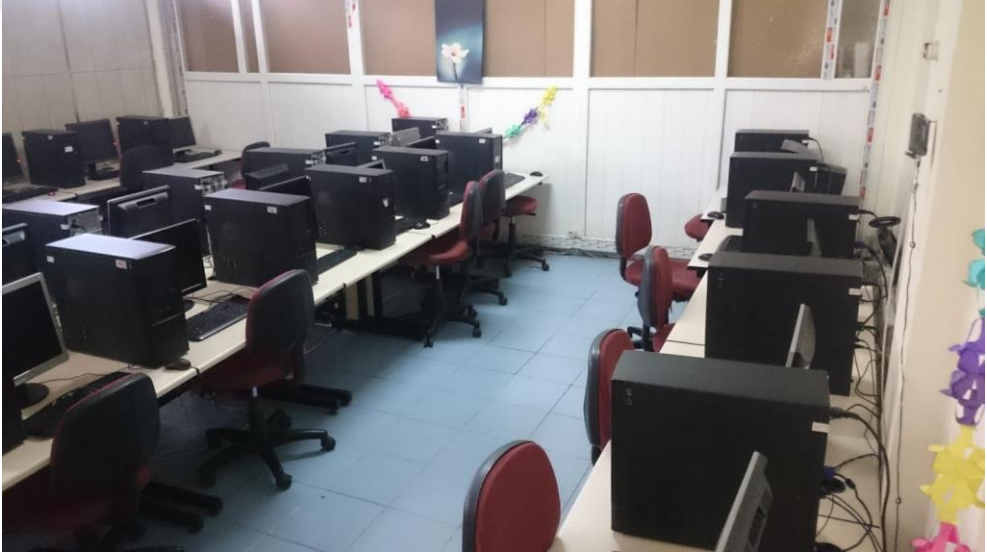
x: -104  
y: -62

```

yakalandım haberini gelince
chomp sesini çal
2 defa tekrarla
balık-kapalı kılıfına geç
0,3 saniye bekle
balık-acık kılıfına geç
Puan 1 arttır
tıklanınca
balık-acık kılıfına geç
Puan , 0 olsun
sürekli tekrarla
eğer fare oku a mesafe > 10 ise
fare oku 'ye doğru dön
5 adım git

```

x: 214  
y: 172

**Ek-7. Bilişim Teknolojileri Sınıfı**



**Ek-8. Çalışma Fotoğrafı**