



**T.C.
AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ORTAOKULLARDA BİLGİSAYARLI VE BİLGİSAYARSIZ
PROGRAMLAMA ÖĞRETİMİNİN AKADEMİK BAŞARIYA VE
PROGRAMLAMA ÖZ YETERLİLİĞİNE ETKİSİ**

YÜKSEK LİSANS TEZİ

GAMZE ORTA

ŞUBAT

GAMZE ORTA

**BİLGİSAYAR VE ÖĞRETİM
TEKNOLOJİLERİ EĞİTİMİ
ANABİLİM DALI**

ŞUBAT 2025

**ORTAOKULLARDA BİLGİSAYARLI VE BİLGİSAYARSIZ
PROGRAMLAMA ÖĞRETİMİNİN AKADEMİK BAŞARIYA VE
PROGRAMLAMA ÖZ YETERLİLİĞİNE ETKİSİ**

Gamze ORTA

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM
DALI**

Danışman

Doç. Dr. Volkan KUKUL

**AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ŞUBAT 2025

Yüksek Lisans Tezi Kabul ve Onay Sayfası

Gamze ORTA tarafından hazırlanan “Ortaokullarda Bilgisayarlı ve Bilgisayarsız Programlama Öğretiminin Akademik Başarıya ve Programlama Öz Yeterliliğine Etkisi” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Amasya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Ve Öğretim Teknolojileri Eğitimi Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Volkan Kukul

Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Amasya Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Başkan : Doç. Dr. Fatih SALTAN

Bilgisayar Ve Öğretim Teknolojileri Eğitimi Anabilim Dalı, Amasya Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Üye: Doç. Dr. Ekmel ÇETİN

Bilgisayar Bilimleri Anabilim Dalı, Kastamonu Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

Tez Savunma Tarihi: 07/02/2025

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr. Ümit YILDIRIM

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Amasya Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu, bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Gamze ORTA

07/02/2025

ORTAOKULLARDA BİLGİSAYARLI VE BİLGİSAYARSIZ PROGRAMLAMA
ÖĞRETİMİNİN AKADEMİK BAŞARIYA VE PROGRAMLAMA ÖZ
YETERLİLİĞİNE ETKİSİ
(Yüksek Lisans Tezi)

Gamze ORTA

AMASYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2025

ÖZET

Bu çalışma, derslerde bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama akademik başarısına ve programlama öz yeterliliğine etkisini incelemek amacıyla yapılmıştır. Araştırma da nicel veri toplama tekniklerinden öntest- son test kontrol gruplu yarı deneysel desen kullanılmıştır. Araştırmanın çalışma grubunu Amasya ilinin Suluova ilçesinde öğrenim görmekte olan deney grubu 24 kontrol grubu 26 olmak üzere toplam 50 öğrenci oluşturmaktadır. Çalışma kapsamında uygulama süreci toplam 6 hafta sürmüştür. Bu süreçte belirlenen kazanımlar deney grubuna Code.org sitesi üzerinden işlenmiş, etkinlikler yapılmıştır. Kontrol grubuna da aynı kazanım ve etkinlikler belirlenmiş olup etkinlikler Code.org sitesi üzerinden çıktı alınarak kâğıt-kalem etkinlikleri olarak öğrencilerle birlikte yapılmıştır. Araştırmada veri toplama aracı olarak “Programlama Öz yeterlilik Ölçeği” ve “Code.org programlama Akademik Başarı Testi” kullanılmıştır. Toplanan veriler SPSS programında analiz edilmiştir. Toplanan programlama akademik başarı testi ve programlama öz yeterlilik ölçeği verileri analizlerinde ve grupların akademik başarı son test ve programlama öz yeterlilik son test puanlarının karşılaştırılmasında Ancova (kovaryans) analizi kullanılmıştır. Araştırma sonucunda, bilgisayarlı ve bilgisayarsız programlama öğretiminin programlama akademik başarısına katkı sağladığı ve gruplar arasında anlamlı bir farkın bulunmadığı sonucuna ulaşılmıştır. Aynı zamanda bilgisayarlı programlama etkinlikleri ile bilgisayarsız programlama etkinliklerinin öğrencilerin programlama öz yeterliliklerini artırdığı sonucuna ulaşılmıştır. Programlama öz yeterliliğindeki artış incelendiğinde deney grubu lehine anlamlı yönde fark tespit edilmiştir.

Sayfa Adedi : 53
Anahtar Kelimeler : Bilgisayarsız etkinlik, Bilgisayarlı etkinlik, Kodlama, Code.Org
Danışman : Doç. Dr. Volkan KUKUL

THE EFFECT OF COMPUTERIZED AND NON-COMPUTERIZED PROGRAMMING
INSTRUCTION ON ACADEMIC ACHIEVEMENT AND PROGRAMMING SELF-
EFFICACY IN SECONDARY SCHOOLS

(Master Thesis)

Gamze ORTA

AMASYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2025

ABSTRACT

This study was conducted to examine the effect of coding activities with and without computers on programming academic achievement and programming self-efficacy. Quasi-experimental design with pretest-posttest control group was used as one of the quantitative data collection techniques. The study group of the research consists of 50 students, 24 in the experimental group and 26 in the control group, who are studying in Suluova district of Amasya province. Within the scope of the study, the implementation process lasted 6 weeks in total. The acquisitions determined in this process were taught to the experimental group through the Code.org website and activities were carried out. The same acquisitions and activities were determined for the control group, and the activities were printed out on the Code.org site and carried out with the students as paper and pencil activities. “Programming Self-Efficacy Scale” and ‘Code.org Programming Academic Achievement Test’ were used as data collection tools in the study. The collected data were analyzed in SPSS program. Ancova (covariance) analysis was used to analyze the collected programming academic achievement test and programming self-efficacy scale data and to compare the groups' academic achievement post-test and programming self-efficacy post-test scores. As a result of the research, it was concluded that computerized and non-computerized programming instruction contributed to programming academic achievement and there was no significant difference between the groups. At the same time, it was concluded that computerized programming activities and non-computerized programming activities increased students' programming self-efficacy. When the increase in programming self-efficacy was examined, a significant difference was found in favor of the experimental group.

Number of pages : 53

Keywords : Computerized activity, Computer-free activity , Coding.

Supervisor : Assoc. Prof. Dr. Volkan KUKUL

ÖN SÖZ VE TEŞEKKÜR

Tezimin tüm aşamalarında desteğini esirgemeyen değerli tez danışmanım Doç. Dr. Volkan KUKUL'a teşekkür ederim. Tez savunmamda yer alarak yapıcı eleştirileri ile tezin son şeklini almasını sağlayan değerli hocalarım Doç. Dr. Fatih SALTAN ve Doç. Dr. Ekmel ÇETİN'e teşekkürü bir borç bilirim. Ayrıca bu süreçte onlara ayırmam gereken zamanlardan fedakârlık eden kızlarım Miray ve Asel'e teşekkürlerimi sunarım.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
ÖN SÖZ VE TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGE DİZİNİ.....	ix
ŞEKİLLER DİZİNİ	x
RESİMLER DİZİNİ.....	xi
SİMGELER VE KISALTMALAR DİZİNİ	xii
1. GİRİŞ... ..	1
2. KAVRAMSAL ÇERÇEVE	5
2.1. Programlama Öğretimi.....	5
2.2. Bilgisayarsız Programlama	6
2.3. Programlama Öz Yeterliliği	7
2.4. Bilgi İşlemsel Düşünme/Hesaplamalı Düşünme.....	7
2.5. Bilgi İşlemsel Düşünme Öz Yeterliliği/Hesaplamalı Düşünme Öz Yeterliliği ..	8
2.6. Blok Tabanlı Programlama	9
2.7. Code.org	10
2.8. Blok Tabanlı Programlama İle İlgili Araştırmalar	15
3.YÖNTEM	22
3.1. Araştırma Modeli	22
3.2. Çalışma Grubu	22
3.3. Veri Toplama Araçları	23
3.3.1. Programlama öz yeterlilik ölçeği.....	23

3.3.2. Programlama akademik başarı testi.....	23
3.4.Uygulama Süreci.....	25
3.4.1. Deney grubu	25
3.4.2. Kontrol Grubu	29
3.5. Verilerin Analizi.....	32
4. BULGULAR VE YORUMLAR.....	33
4.1. Normallik Analizleri	33
4.1.1. Programlama akademik başarı testi ile ilgili yapılan normallik analizleri	33
4.1.2. Programlama öz yeterliliği ile ilgili yapılan normallik analizleri	35
4.2. Uygulanan Ancova Analizi sonuçları	36
4.2.1. Bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama akademik başarısı üzerindeki etkisine yönelik bulgular	36
4.2.2. Bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama öz yeterliliği üzerindeki etkisine yönelik bulgular	37
5. SONUÇ, TARTIŞMA VE ÖNERİLER	39
5.1. Sonuç ve Tartışma.....	39
5.2. Öneriler	42
KAYNAKLAR	43
ÖZGEÇMİŞ	52

ÇİZELGELER DİZİNİ

Çizelge	Sayfa
Çizelge 3.1. Deneysel desen çizelgesi	22
Çizelge 3.2. Çalışma grubu bilgileri	23
Çizelge 3.3. Belirtke çizelgesi ve ilgili test sorusu	24
Çizelge 3.4. Programlama başarı testine ilişkin madde güçlük indeksleri	25
Çizelge 4.1. Kontrol grubu programlama akademik başarı ön test normallik analizi	33
Çizelge 4.2. Deney grubu programlama akademik başarı ön test normallik analizi	34
Çizelge 4.3. Kontrol grubu programlama öz yeterlilik ön test normallik analizi	35
Çizelge 4.4. Deney grubu programlama öz yeterlilik ön test normallik analizi	35
Çizelge 4.5. Deney ve kontrol grubu programlama akademik başarı testi öntest ortalama ve düzeltilmiş ortalama puanları.....	36
Çizelge 4.6. Deney ve kontrol grubu programlama akademik başarı son test analiz sonuçları	37
Çizelge 4.7. Deney ve kontrol grubu programlama öz yeterlilik öntest ortalama ve düzeltilmiş ortalama puanları.....	37
Çizelge 4.8. Deney ve kontrol grubu programlama öz yeterlilik sontest analiz sonuçları ..	38

ŞEKİLLER DİZİNİ**Şekil****Sayfa**

Şekil 4.1. Deney grubu normal q-q başarı öntest grafiği34



RESİMLER DİZİNİ

Resim	Sayfa
Resim 2.1. Code.org kurs katalođu ekranı	11
Resim 2.2. Code.org kontrol panel ekranı	11
Resim 2.3. Code.org seviye ilerleme durum ekranı	12
Resim 2.4. Code.org seviye durum ekranı açıklamaları	12
Resim 2.5. Code.org uygulama ekranı	13
Resim 2.6. Code.org örnek etkinlik ekranı	14
Resim 2.7. Code.org örnek etkinlik ekranı	14

SİMGELER VE KISALTMALAR DİZİNİ

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, yanda açıklamaları verilmek üzere aşağıda listelenmiştir.

Kısaltmalar

Açıklama

BÖTE

Bilgisayar ve öğretim teknolojileri eğitimi

MEB

Milli eğitim bakanlığı

TTKB

Talim ve terbiye kurulu başkanlığı

BTY

Bilişim teknolojileri ve yazılım

1. GİRİŞ

Bilgisayar programlama, mantıksal ve yaratıcı düşünme, problem çözme gibi düşünme becerilerinin gelişmesine yardımcı olan bir yeterlilik olarak algılanmaktadır (Yükseltürk ve Altıok, 2017). Bilgisayar programlama 21. yüzyılda önemli bir beceri haline gelmiştir (Bers, 2019). Programlama becerisi sadece yazılım yapmak için gereken beceri değildir; programlama sürecinde öğrencilerin üst düzey becerileri de gelişir (Türker ve Pala, 2020). Programlama eğitimi insana eleştirel düşünme, problem çözme, yaratıcı düşünme gibi 21.yy becerileri kazandırmaktadır (Gültepe, 2018). Bilgisayar programlama, 21.yüzyıl için önemli bir dijital okuryazarlık becerisi olarak kabul edilmiştir (Wong ve Cheung, 2020). Programlama eğitiminin amacı öğrencilerin problem çözme becerilerini geliştirmek ve öğrencilere çözümlerini ifade etmek için kullanabilecekleri bir programlama dili öğretmektir (Gomes ve Mendes, 2007). Çocuklar için bilgisayar programlama eğitimi, eğitim için gelişen bir alan olmuştur (Strawhacker ve Bers, 2019). 21 Yüzyıl becerilerinin kazanılmasının en önemli yolu eğitimidir ve eğitim, anne karnından başlayarak yaşam boyunca devam eder, devam eden bu sürecin en önemli döneminin erken çocukluk dönemi olduğu söylenebilir (Mercan, 2022). Bu dönem gelişimin hızlı olduğu bir dönem olduğu için 21.yy becerilerinin bu dönemde kazandırılması gerekir (Mercan, 2022). Bu bağlamda erken çocukluk dönemindeki çocukların günlük hayatta karşılaştığı problemlere çözüm bulma, eleştirel ve yaratıcı düşünme becerileri kazandırılmaya çalışılması gerektiği söylenilebilir. Bazı ülkelerde (Estonya, Güney Kore, ABD, İngiltere vb.) kodlama eğitimi okul öncesi dönemde, ülkemizde ise 5. ve 6. sınıf düzeylerinde zorunlu 7. ve 8. sınıf düzeylerinde seçmeli olarak verilmektedir (Soykan ve Kanbul, 2018). Ülkemizde Milli Eğitim Bakanlığı ilk kez 1998 yılında eğitimin her kademesinde bilgisayar okuryazarlığını yaygınlaştırmak amacıyla ilköğretimin 1-2-3-4-5 kademelerine seçmeli bilgisayar dersini getirmiştir (MEB, 1998). 2006 Yılında ise 1998 yılında yürürlüğe giren seçmeli bilgisayar dersi öğretim programı yerini ilköğretim seçmeli bilgisayar (1-8. Sınıflar) dersi öğretim programına bırakmıştır (MEB, 2006). 2012 Yılında ise bilişim teknolojileri ve yazılım dersi çizelgede yer almış, ders yeni bir çerçeve öğretim programı ile uygulanmaya başlanmıştır (BTE Derneği, 2013). Son olarak 2018 yılında yürürlükte olan 2012 tarihli bilişim teknolojileri ve yazılım dersi yürürlükten kaldırılarak 2018 tarihli bilişim teknolojileri ve yazılım dersi uygulanmaya başlanmıştır (MEB, 2018). Zaman içinde seçmeli ile başlayan bilişim teknolojileri ve yazılım dersi günümüzde 5. ve 6. sınıflarda

zorunlu 7 ve 8. sınıflarda seçmeli olarak okutulmaktadır. Ayrıca yapay zekâ, dijital sanatlar, robotik ve kodlama dersleri de seçmeli olarak çizelgeye eklenmiştir (TTKB, 2023). 2018 tarihli öğretim programında 5. sınıflarda “problem çözme ve programlama” ünitesi altında 27 kazanım 36 ders saati 6. sınıflarda 25 kazanım 36 ders saati ile bilişim teknolojileri ve yazılım dersi içinde programlama öğretimine yer verilmiştir (TTKB, 2018). Akkaş Baysal, Ocak ve Ocak (2020), öğretim programlarında yer alan değişimlerle birlikte öğrenciler, arduino, kodlama, robotik gibi kavramlarla tanışmışlar ve bu kavramlarla tanışan öğrenciler eleştirel, algoritmik, yansıtıcı düşünme gibi becerilerin yalnız bu alanlarda değil tüm alanlarda öğrenenlere etkisinin olacağını fark etdiklerini belirtmişlerdir.

Günümüzde programlama öğretimi için farklı yaklaşımlar mevcuttur. Gülbahar ve Karal (2018) programlama öğretim yaklaşımlarını öğrenen odaklı ve araç odaklı olmak üzere incelemişlerdir. Öğrenen odaklı yaklaşım içine eşli programlama, oyun temelli öğrenme, sorgulama temelli öğrenme, proje temelli öğrenme ve programlama eğitiminde disiplinler arası yaklaşım olarak ele almışlardır. Araç odaklı yaklaşımda ise bilgisayarsız, blok tabanlı, metin temelli, fiziksel ve hibrit programlama yaklaşımlarını ele almışlardır. Öğrenen odaklı yaklaşımların temelinde öğrenenlerin olduğu belirtilmiştir. Gülbahar ve Karal (2018) aynı kitabında oyunun, öğretim materyali olarak üç şekilde kullanıldığını belirtmiştir. Bunlar bilgisayarsız etkinlikler, dijital eğitici oyunlar ve oyunlaştırmadır. İlkokul düzeyinde (6-10 yaş) programlama öğretiminde fiziki ortamlarda oyun ve oyunlaştırılmış etkinlikler zengin öğrenme ortamı sunacağı belirtilmiş ve etkinlik olarak dijital ortamlardaki blok tabanlı uygulamalarının fiziki kartlar yapılarak uygulanması verilmiştir. Gülbahar ve Karal (2018), ortaokul düzeyinde (11-13) yaş dönemini profesyonel uygulamaya geçişin ara dönemi olarak tanımlamış bu dönemde ilkokulda başlanan bilgi işlemsel düşünme becerilerini geliştirmeye yönelik etkinliklere devam edilmesi ile birlikte blok tabanlı uygulama araçlarının programlama öğrenmede ağırlık kazanması gerektiğini aynı zamanda da eşli programlama çalışmaları yapılması gerektiğini belirtmişlerdir. Blok tabanlı uygulama araçlarına Code.org, Scratch AppInventor, Google Blockly, Alice, Code Monkey, Pocket Code, Code Combat gibi platformlar örnek verilebilir. Gülbahar ve Karal (2018), lise düzeyini (14-17 yaş) programlama düzeyine başlangıç dönemi olarak ele almışlar ve bu dönemde metin temelli programlama geliştirme çalışmaları ile proje temelli yaklaşımın ön plana çıktığını belirtmişlerdir. Python, Ruby ve Java dilleri lise programlama öğretiminde öncelikli olarak tercih edilebilir (Gülbahar ve Karal, 2018: 121).

Oyunlar; öğrenme için giderek daha popülerlerdir ve bilgisayar bilimine yeni başlayanlar için bilgisayar programlama becerileri kazandırmaya yardımcı olabilirler (Yang ve Kopcha, 2022). Oyun tabanlı öğrenme ve blok tabanlı programlamanın birleşmesi eğitimin verimini artırır, ilgiyi canlandırır (Glushkova, 2016). Oyun tabanlı öğrenme ortamları öğrencilerin aktif tutarak faaliyetlerini sürdürebilecekleri araç olmalarının yanında, öğrencilere yaparak-yaşayarak öğrenmelerine fırsat veren ortamlar sunar (Yıldırım, 2016). Blok tabanlı programlama, kullanıcıya komutların nerede ve nasıl kullanacağına ilişkin görsel ipuçları vererek yapboz parçası kadar ilkel bir metaforu kullanır (Weintrop ve Wilensky, 2017). Programlama, komut bloklarını komut bloklarının bulunduğu alandan çalışma alanına sürükleyerek ve blok yığınları oluşturmak için alt alta eklenerek, yapboz parçaları gibi birleştirilerek yapılır (Maloney, Peppler, Kafai, Resnick ve Rusk, 2008). Blok tabanlı programlama geleneksel programlama ortamlarının aksine kullanıcının istediği kod bloklarını sürükle-bırak mantığıyla oyunlar, hikâyeler, animasyonlar yapmaya imkân sağlayarak özellikle küçük yaştaki çocukların kodlamayı öğrenmelerine kolaylık sağlar (MEB, 2017). Çocuklar görsel programlama dillerini kullanarak temel programlama becerilerini kazanırlar ve bu programlama dillerinin küçük çocuklar tarafından kullanılmasını kolaylaştırır (Kraleva, Kraleve ve Kostadinova, 2019). Günümüzde blok tabanlı programların ve yeni arayüzlerin geliştirilmesi kodlama ve hesaplamalı düşünme becerilerini geliştirmektedir (Bers, 2019).

Literatür incelendiğinde bilgisayarlı ve bilgisayarsız programlama etkinliklerinin programlama öz yeterliliği ve başarısı üzerindeki etkisine yer veren çalışmaların fazla olmadığı görülmüştür. Bundan dolayı bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama öz yeterliliği ve programlama başarısı üzerindeki etkisini ortaya koymaya yönelik bu çalışmanın ortaokul 5. sınıf öğrencileri ile Code.org üzerinden yapılmasına karar verilmiştir. Derslerde bilgisayarlı ve bilgisayarsız kodlama etkinliklerin Code.org üzerinden işlenmesi sonucunda kodlama etkinliklerinin etkililiğinin ortaya konması açısından alanyazına katkı sağlayacağı düşünülmektedir. Araştırmanın problemi “Ortaokullarda bilgisayarlı ve bilgisayarsız programlama öğretiminin akademik başarıya ve programlama öz yeterliliğine etkisi var mıdır?” şeklinde belirlenmiştir. Bu bağlamda araştırma sorusuna cevap bulmak için alt problemler belirlenmiştir. Belirlenen alt problemler şu şekildedir.

- Deney ve kontrol gruplarının kodlama etkinlikleri arasında programlama başarısı üzerinde anlamlı bir fark var mıdır?
- Deney ve kontrol gruplarının kodlama etkinlikleri arasında programlama öz yeterliliği üzerinde anlamlı bir fark var mıdır?



2. KAVRAMSAL ÇERÇEVE

Çalışmanın bu bölümünde programlama öğretimi, bilgisayarsız programlama, programlama öz yeterliliği, bilgi işlemsel düşünme/hesaplamalı düşünme, bilgi işlemsel düşünme öz yeterliliği/hesaplamalı düşünme öz yeterliliği, blok tabanlı programlama, Code.org ve blok tabanlı programlama ile ilgili araştırmalara yer verilmiştir.

2.1. Programlama Öğretimi

Çağımızdaki bilim ve teknoloji alanındaki gelişmeler eğitimi de etkilemiştir (Yaman ve Hamedoğlu, 2014). Bilgisayar programlama, kodlama olarak da anılır ve çok önemli bir yüzyıl becerisi olarak ortaya çıkmıştır (Tsarava, Moeller, Pinkwart, Butz, Trautwein ve Ninaus, 2017). Kodlama eğitimi, problem çözme ve ürün geliştirebilme gibi özelliklere sahip teknolojiyi doğru ve aktif kullanan bireylerin eğitiminde önemli bir yere sahiptir (Şahin Çakır, Çakır, Damar, Erdemir Yılmaz, 2021). Kodlama, teknolojik gelişmeler ile şekillenmiş günlük hayatta ve iş dünyasının her alanında çalışanlar için önemli bir özellik olarak görüldüğünden günümüzde mantıksal akıl yürütmenin bir parçası olarak kabul edilmiş, aynı zamanda 21.yy becerilerinden biri haline gelmiştir (MEB, 2017). Kodlama, arduino, STEM gibi bilim ve teknolojiyi bir araya getirerek bunların uyumlu bir şekilde çalışmasını sağlayabilen çalışmalar öğrencilerin düşünme becerilerini farklı açılardan harekete geçirmektedir (Akkaş Baysal ve ark., 2020). Programlama, problemleri çözmenin, fikirleri hayata geçirmenin ve dijital dünyada yeni bir şeyler üretmenin bir yoludur (Nouri, Zhang, Mannila ve Norén, 2020). Programlama bilgisayara komutlar vererek istenilen iş ve işlevlerin yaptırılması olarak tanımlanabilir (Akkaş Baysal ve ark., 2020). Programlamaya ilginin hızla artması programlama giriş derslerini popüler haline getirmiştir ancak programlamayı öğrenmek zordur (Robins, Rountree ve Rountree, 2003). Programlama öğrenmek üst düzeyde problem çözme becerisi gerektirdiğinden dolayı çocukların programlama başarıları üzerindeki etkenler literatürde tartışılan konu olmuştur (Yağcı, 2016). Java, Javascript, C, C#, Python, php, Visual Basic, Pascal gibi geleneksel kodlama dil yapılarının kullanılması karmaşık ve zor bir yapıya sahiptir (Aytekin, Sönmez, Çakır, Yücel ve Kulaöz, 2018). Metin tabanlı programlama araçları oldukça pasif ve öğrenme başarılarını tatmin edememektedir (Threekunprapa ve Yasri, 2020). Bu yüzden kodlama yapmayı kolaylaştıran Scratch, Google Blockly ve Code.org gibi bilişim teknoloji araçları ile kullanılabilen platformlar çocuklar ve kodlamaya yeni başlayanlar için uygundur

(Aytekin ve ark., 2018). Scratch, Alice ve Code.org gibi araçların öncülüğünde, birçok programlama ortamı ve giriřimi blok tabanlı modaliteyi kullanmaktadır (Brown, Mönig, Bau, ve Weintrop, 2016).

2.2. Bilgisayarsız Programlama

Bilgisayarsız kodlama, biliřim teknolojileri araçları olmadan fiziksel aktivite ve kart oyunları ile yapılan etkinliklerdir. Robot Turtles ve Tospaa.org bilgisayar gerektirmeyen fiziksel oyunlara bilgisayarlı kodlama denir (Yılmaz ve İzmirli, 2023). Bilgisayarsız etkinlikler genellikle biliřim teknoloji araçlarının olmadığı yerlerde veya biliřim teknoloji araçlarının taşınması, kurulması zor olan yerlerde sıklıkla kullanılmaktadır (Jaguřt, Krzic, Gledec, Grgic ve Bojic, 2018). Bilgisayarsız kodlama etkinliklerine Tospaa.org, Csunplugged.org, Code.org gibi platformlarda yer alan etkinlikler örnek verilebilir. Birçok kodlama öğrenme etkinlięi bir kalem ve biraz kâğıtla fiřsiz bir senaryoda gerçekleştirilebilir (Klopfenstein, Fedosyeyev ve Bogliolo, 2017). Bilgisayarsız kodlama etkinlikleri aynı zamanda bağlantısı kesik ve fiři çekilmiş etkinlikler olarak da ifade edilmektedir. Bağlantısı kesik bilgisayar bilimi etkinliklerinin, öğrencilerin bilgisayar bilimine olan ilgisini artırmaktadır (Rodriguez, Rader ve Camp, 2016). Genel olarak bağlantısız etkinlikler hedefe ulaşmak için bilgisayar bilimlerinin kavramları ve problem çözme ile ilgilenir (Bell, Alexander, Freeman ve Grimley, 2009). Bilgisayarsız bilgisayar bilimi etkinliklerinde bilgisayara erişim sorun olarak görülmemektedir; oyunlar öğretme ve öğrenme sürecine dâhil edilebilir (Mumcu, Kidiman ve Özdiñ, 2023). Artık birçok ülkede öğrenciler bilgisayara rahat erişim sağlayabilir ve fiřsiz etkinlik yaklaşımı sadece erişim için deęil bir materyal olarak da kullanılabilir hale gelmiştir (Munasinghe, Bell ve Robins, 2023). Bu durumda öğrenciler ve öğretmenler daha çok ve düzenli zaman geçirerek kavramları daha iyi geliřtirebilir (Munasinghe ve ark., 2023). Canbeldek (2020)'e göre bilgisayarlı kodlama, çocukların kodlama etkinlikleri yapması için teknolojik araçlara ihtiyaç duymadan algoritma, döngüler, koşullar gibi kodlama kavramlarının da öğretilbildięi etkinlikler olarak ifade edilebilir. Bu bağlamda bilgisayarlı kodlama etkinlikleri, çocuklarla kodlama yaparken teknolojik araçları kullanmaya gerek duymadan kodlama etkinliklerinin yapılabilmesini sağladığını söyleyebiliriz. Bilgisayarsız etkinlikler, programlama öğretiminde temel kavramların anlaşılması, programlama becerilerinin geliřmesi ve sürecin sonunda kaliteli ürünlerin çıkması açısından önem taşımaktadır (Gülbahar ve Karal, 2018: 191). Bilgisayarsız kodlama etkinliklerinin yararları arasında

programlamayı öğrenmede öz yeterliliği teşvik etme yeteneği ve aktif öğrenme yer alır (Threekunprapa ve Yasri, 2020).

2.3. Programlama Öz Yeterliliği

Çocuklar için programlama eğitiminde karşılaşılan önemli öğelerden biri programlama öz yeterliliğidir (Durak, Yılmaz ve Yılmaz, 2019). Bilgisayar programlama öz yeterliliğine sahip olmak, bir öğrencinin programlama görevlerini yerine getirmek için gerekli yetenek ve becerilere sahip olduğu inancını destekler (Ma, Zhao, Wang, Wan, Cavanaugh, ve Liu, 2021). Programlama gibi bir beceriyi öğrenmek için de hedefler belirlenmesi öğrenmenin değeri öz yeterlilik becerisi oluşturulabilmesi için dikkate alınması gereklidir (Abdunabi, Hbaci ve Ku, 2019). Öğrencilerin programlamada başarılı olmaları ve hedeflenen kazanımlara ulaşabilmeleri, problem çözme becerilerine ve programlamaya yönelik öz yeterlilik ve tutumlarına bağlıdır (Yağcı, 2016). Öz yeterlilik programlamayı öğrenme çabasında başarının anahtarıdır ve programlama öğrenmek öz güven ve öz saygıyı geliştirir (Kovari ve Katona, 2023). Bu bağlamda programlama başarısını etkileyen faktörlerden birisinin de öz yeterlilik olduğunu söyleyebiliriz. Yüksek öz yeterliliğe sahip öğrenciler görevleri tamamlamaktan ziyade görevleri çözülmesi gereken bir zorluk olarak görürler (Wei-Ying ve Tzu-Chuen, 2024). Programlama becerisi bilgisayar bilimlerinde bireysel farklılıkların önemli olduğu ve bireyler tarafından zor algılanılan bir beceridir (Mazman ve Altun, 2014). Bireylerin programlamayı baştan zor olarak kabul etmelerinden dolayı programlama derslerinde başarısız olmaları olasıdır (Altun ve Mazman, 2012). Ramalingam, ve ark. (2004) çalışmalarında programlama öz yeterliliğinin önceki deneyimlerinden etkilendiğini ve deneyimler arttıkça programlama öz yeterlilik becerilerinin artacağını belirtmişlerdir. Aydoğdu (2020) programlamayı yeni öğrenen öğrencilerin veya programlama yapmaktan korkan öğrencilerin öz yeterliliklerinin artırılması için eğitim sürecinde blok tabanlı programlama etkinliklerinden yararlanabileceğini belirtmiştir.

2.4. Bilgi İşlemsel Düşünme/Hesaplamalı Düşünme

Bilgi işlem araçlarını etkili bir biçimde kullanarak onlarla nasıl iletişim kuracağımızı anlamak yani başarılı iletişim kurmak bilgi işlemsel düşünme olarak tanımlanabilir (Shute, Sun ve Asbell Clarke, 2017). Hesaplamalı düşünme zekâyı, zihni, insan davranışlarını ve

hesaplanabilirliği bilimsel düşünceyle paylaşır, bu yüzden hesaplamalı düşünmeyi bir analitik düşünme türü olarak düşünebiliriz (Wing, 2008). Hesaplamalı düşünme, bilgi işlem olanaklarını bulmak için ortak bir dil ve dijital yerliler arasında azalan merak duygusunu ele almanın yoludur (Hu, 2011). Hu (2011) aynı makalesinde bazılarının, bilgisayarla ilgilenmeyen birçok kişinin bilmeden hesaplamalı düşünme yaptığına inandığından bahsetmiştir. Hesaplamalı düşünme, öğrencilerin algoritmik düşünme, problemlere çözüm üretme ve kodlama ile ilgili gelişmelere artan ilgiyi tanımlamak için kullanılan terimdir (Angeli ve Giannakos, 2020). Yine aynı makalesinde Angeli ve Giannakos (2020) hesaplamalı düşünmenin çocukların, programlama ve algoritma uygulayarak geliştirdikleri becerilere odaklandığından; örüntü tanıma, problem çözme, soyut düşünme ve mantıksal akıl yürütme gibi özelliklerin gelişmesine katkıda bulunduğundan bahsetmişlerdir. Bilgisayarlı düşünme 21.yüzyılda önemli bir beceri olarak kabul görmüş problem çözme araçlarından biridir (Chongo, Osman ve Nayan, 2021). Gülbahar, Kert ve Kalelioğlu (2019) çalışmalarında bilgi işlemsel düşünmenin her yaşta bireyin kazanabileceği bir beceri olduğuna değinerek dijital çağın sınıflarındaki öğrencilerin 21.yy becerilerini kazanmış problem çözen, üretken, uyumlu, doğru bilgiye ulaşabilen bireyler olması gerektiğini belirtmişlerdir. Kuleli ve Kısı (2024) mevcut çağda yaşamın farklı alanlarındaki dönüşümün 21.yy becerilerine sahip olmak için gereksinimlerin çıkmasına neden olduğuna değinmiş ve bu becerileri karar verme, yaratıcı düşünme, problem çözme şeklinde sıralamışlardır. Moon, Cheon ve Kwon (2022) ise gelişen bir dünyada hesaplamalı düşünme öğrencilerin zorluklara hazırlanmaları ve potansiyellerini artırmaları için gerekli olan problem çözme becerisi olarak ortaya çıktığını belirtmişlerdir. Günümüz dünyasında, bireylerin hangi alana yönelecekleri fark etmeksizin, içinde bulunduğumuz çağın bir gerekliliği olan hesaplamalı düşünme becerileriyle okul öncesinden itibaren yetiştirilmesi gereklidir (Yıldız ve Seferoğlu, 2021). Hesaplamalı düşünme, zorunlu eğitim müfredatına entegre edilerek geliştirilebilir (Arıstawati, Budiyanto ve Yuana, 2018).

2.5. Bilgi İşlemsel Düşünme Öz Yeterliliği/Hesaplamalı Düşünme Öz Yeterliliği

Bilgi işlemsel düşünme becerisi okul öncesinden itibaren her yaşta kazanılabilecek bir beceri olmakla birlikte bilgi işlemsel düşünme becerileri 21. yüzyıl becerileri kapsamında tanımlanmıştır (Gülbahar, Kert ve Kalelioğlu, 2019). Bireyin bir performansı başarılı bir şekilde yerine getirebilmesi için kendine inanması ya da kendi yargısına öz yeterlilik;

öğrenen bireyin okul hayatındaki akademik başarısının belirleyici olmasının yanında okul dışındaki hayatında da ömür boyu kullanacağı beceriye ise kendine olan öz yeterlilik inancı denir (Arseven, 2016). Öz yeterlilik, başarı, öğrenme ve teknolojiye uyum sağlama ile ilişkilidir (Kovari ve Katona, 2023). Hesaplamalı düşünme öz yeterliliği, öğrencilerin hesaplamalı düşünme öz yeterliliğine ilişkin inançları ve algıları olarak tanımlanabilir (Gümüş, Kukul ve Korkmaz, 2024). Kendine güven ve öz yeterlilik insanların bilgisayar kullanımı konusunda girişimde bulunmalarını veya ısrar etmelerini etkileyebilir (Galpin, Sanders, Turner ve Venter, 2003). Öz yeterlilik, özellikle başlangıç seviyesindeki derslerde en çok ele alınan faktörlerden birisidir (Mazman, Altun, 2014). Öz yeterlilik, kullanım kolaylığı üzerinde doğrudan ve dolaylı olarak etkisi bulunmaktadır (Igbaria ve Iivari, 1995). Ramalingam, LaBelle ve Wiedenbeck (2004), programlama öz yeterliliğinin deneyimlerinden etkilendiğini ve seviye ilerledikçe öz yeterliliklerinin arttığını belirtmişlerdir.

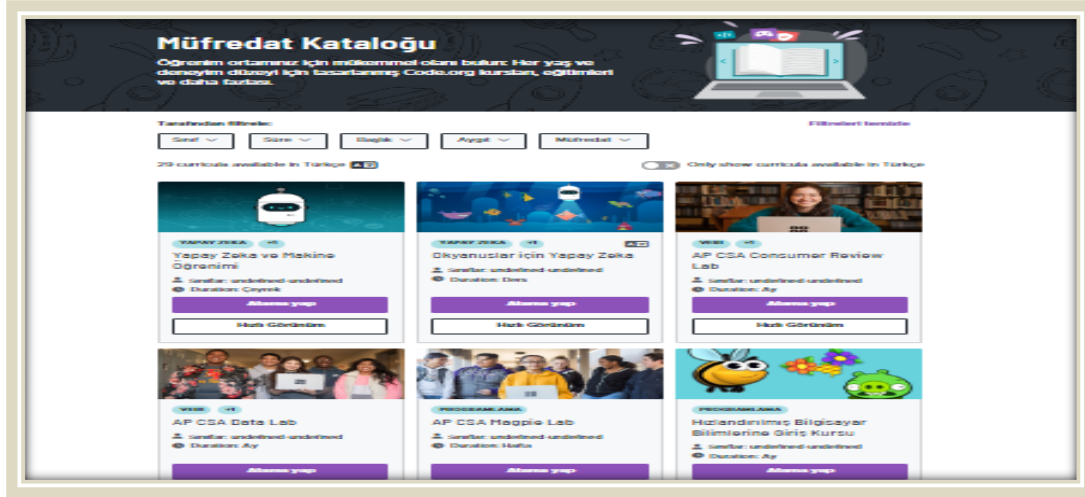
2.6. Blok Tabanlı Programlama

Yazılımlar kullanılarak problemlerin modellenmesi ve bilgisayar ortamına aktarılma süreci programlama olarak tanımlanabilir (Benzer ve Erütmit, 2017). Programlama öğrenmek öğrencilerin problem çözme, yaratıcı düşünme, işbirlikli öğrenme gibi becerileri edinmesinde rol oynar (Çatlak, Tekdal ve Baz, 2015). Blok tabanlı yazılım geliştirme, öğrencilerin programlama ve bilgisayar bilimiyle tanışması için bir kapı haline gelmiştir (Brown ve ark., 2016). Blok tabanlı programlamaya ilginin artmasıyla bilgisayar bilimlerine giriş için ortaya çıkan programlama ortamları umut vermiştir. (Emerson, Rodríguez, Mott, Smith, Min, Boyer ve Lester, 2019). Programlama eğitiminin anlaşılır, eğlenceli, motive edici ve kolay olması için blok tabanlı programlama araçları geliştirilmiştir (Çilengir ve İzmirli, 2023). Blok tabanlı programlama uygulamalarına Scratch, Code. Org, Tynker Mblock, MakeCode App Inventor örnek verilebilir. Blok tabanlı programlamada kod blokları kullanılarak karmaşıklık azaltılmış ve kodlamaya yeni başlayanlar için programlama daha çekici hale getirilmiştir (Singla ve Theodoropoulos, 2022). Blok tabanlı yazılımlarda genellikle sürükle-bırak mantığı ile kodlama yapılmaktadır. Programlama komut bloklarını sürükleyerek yapboz parçaları gibi alt alta ekleyerek blok yığınlarının oluşturulmasıyla yapılır (Maloney ve ark., 2008). Bu bağlamda blok tabanlı yazılımların çocuklar ve kodlamaya yeni başlayanlar için uygun olduğu söylenebilir (Aytekin ve ark., 2018). Bloklar sözdizimi ezberlemeyi ortadan kaldırır ve

kodların blok halinde olması insan konuşmasına benzeyen doğal bir dil kullanmasına imkân sağlar (Moors, Luxton-Reilly ve Denny, 2018). Blok tabanlı programlama gerçek durumların modellerini oluşturmalarına, oyunlar geliştirmelerine, müfredatın içerikleriyle kolay bir şekilde bağlantı kurmalarına olanak sağlar (Perin, Silva ve Valentim, 2023). Mors ve ark. (2018) blok tabanlı uygulamaların kavramları birbirinden ayırmak ve blokların nereye yerleşeceğini belirtmek için farklı şekil ve renkler kullandığını ayrıca blok tabanlı programlama ortamlarının söz dizimi olmayışı, blok yapıları, basitleştirilmiş dil, şekil gibi ortak temel özelliklerinin olduğunu belirtmişlerdir. Blok tabanlı öğretimin sınırlamalarını dikkate almak ve öğrenenlerin gelecekteki kodlama çalışmaları için çeşitli programlama dilleri ve araçlarıyla tanışmaları önemlidir (Taş, Bolat ve Savaş Başkara, 2023).

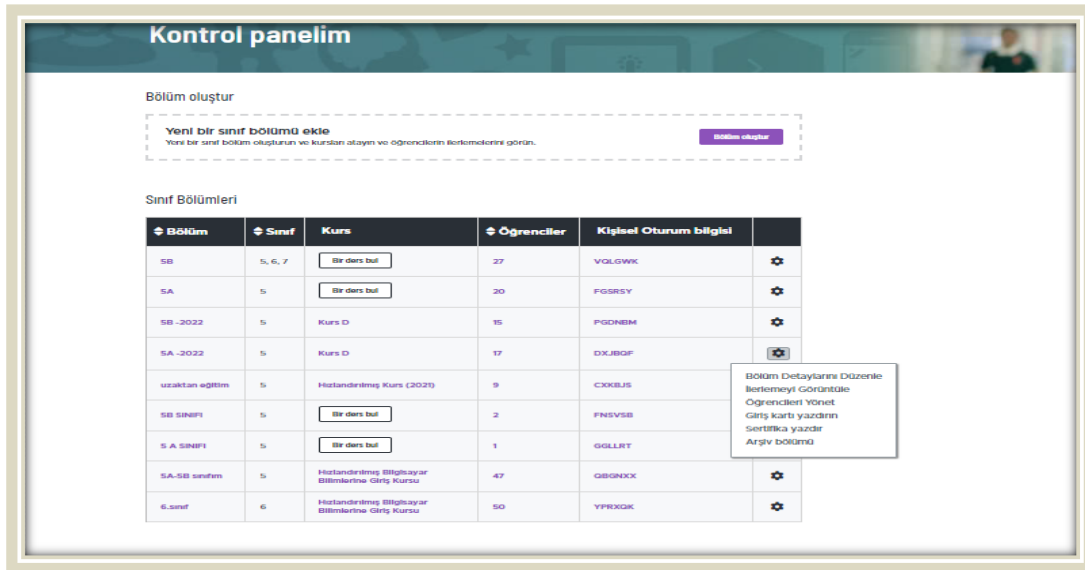
2.7. Code.org

2013 Yılında ikiz kardeşler Hadi ve Ali Partovi tarafından kurulmuştur (Code.org, 2024). Vizyonu, okullardaki her öğrencinin bilgisayar bilimi öğrenmesine olanak sağlamaktır (Code.org, 2024). Code.org sitesi dünyada yaygın olarak kullanılan programlama dillerinden olan Javascript'i, bazı derslerinde de HTML ve CSS'de kullanır (Code.org, 2024). Bu bağlamda Code.org görsel programlamayı temel alarak programlama yapmayı hedef alan bir web sitesidir (Code.org, 2024). Bilgisayar programlamayı öğretmek için Code.Org görsel bir yaklaşım kullanır (Du, Wimmer ve Rada, 2016). Code.org platformunda çocuklara sadece kod yazma öğretilmiyor aynı zamanda döngü kurma öğretilmekte ve problem çözme becerilerini geliştirerek kod bloklarına eklentiler yapmaları istenmektedir (Aytekin ve ark, 2018). Code.org sitesi üzerinde öğretmenler üye olarak öğrencilerine sınıf oluşturabilir öğrenciler üye olmadan bu sınıflar üzerinden gizli kelimeleri ile Code.org'a giriş yapabilirler. Öğretmenler öğrencilerini kurslara tanımlayabilir, etkinlik gönderebilir ve gönderdiği etkinliklerin durumunu istediği zaman takip edebilir. Kurs bitiminde öğrenciler için sertifika yazdırabilir. Yeni kurslara öğrencileri tanımlayabilir. Code.org platformuna <https://code.org/> adresinden erişim sağlanır.



Resim 2.1. Code.org kurs kataloğu ekranı

Resim 2.1’de code.org platformunun kurs kataloğu ekranı gösterilmektedir. Kurs kataloğunda her yaş ve öğrenim düzeyi için tasarlanmış Code.org eğitimleri, kursları bulunmaktadır. Kursların içinde kurslara tanımlanmış etkinlikler bulunmaktadır. Kurs kataloğu alanında seçtiğimiz kurslara atama yap butonunu tıklayarak öğrencilerimizi kurslara ekleyebilir ve öğrencilerimizin ilerleme düzeylerini Code.org sitesi üzerindeki kontrol paneli ekranından takip edebiliriz. Öğrenciler Code.org sitesine kayıt olmadan verilen sınıf şifreleri ile etkinliklerini yapabilirler.



Resim 2.2. Code.org kontrol panel ekranı

Resim 2.2’de Code.org platformunun kontrol panel ekranı gösterilmektedir. Kontrol panel ekranından tanımlanan sınıflar/gruplar, sınıflara atanılan kurslar, öğrenci sayıları, öğrencilerin bilgileri görüntülenmektedir. Kontrol panelinden öğretmen; öğrencileri yönetebilir, öğrencilerin kurslarda ilerlemelerini takip edebilir. Yine bu bölümden öğrencilerin siteye giriş yapabilmeleri için gerekli bilgilerin yer aldığı giriş kartları yazdırılabilir. Ayrıca kursu bitiren öğrenciler için sertifika çıktısı da alınabilir.



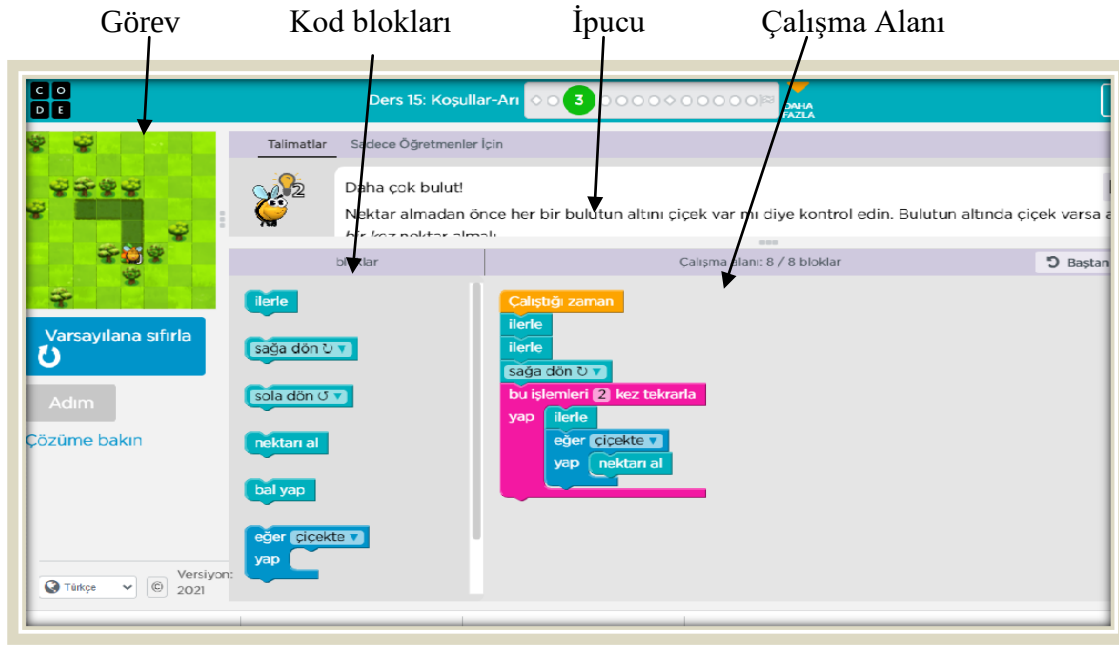
Resim 2.3. Code.org seviye ilerleme durum ekranı

Resim 2.3’te Code.org platformunun seviye durum ekranı gösterilmektedir. Bu alandan sınıflara/gruplara atanmış olan öğrencilerin kurs ilerleme seviyelerini takip edebiliriz. Seviye durumu renklendirme ve şekillerine göre öğrencilerin takıldıkları etkinlikleri görebiliriz.

Seviye türü	Seviye Durumu					
	İlerleme Durumu Mevcut değil	Başlamamış	Devam ediyor	Tamamlandı (çok fazla blok)	Tamamlandı (mükemmel)	Değerlendirmeler / Anketler
Kavram	—	◇	◇	Mevcut değil	◆	Mevcut değil
Etkinlik	—	○	○	●	●	●

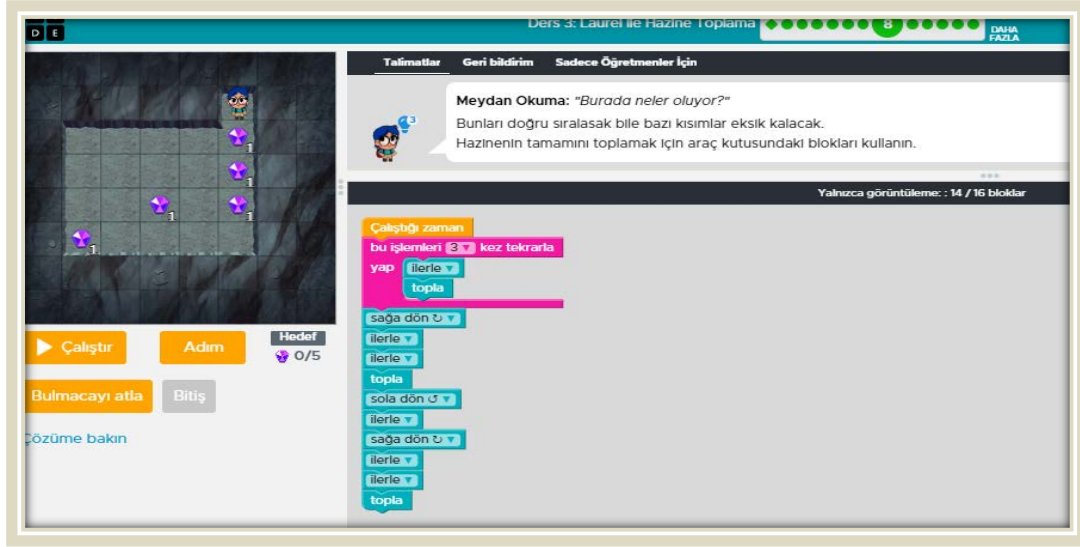
Resim 2.4. Code.org seviye durum ekranı açıklamaları

Resim 2.4 Code.org seviye durum ekranı açıklamaları görselinde daire renklendirmelerine göre seviye ilerlemelerin açıklamaları belirtilmektedir. Şekilde dairenin içinin tam dolu koyu yeşil olması gerekli blokları kullanarak etkinliği tamamladığını, boş renksiz daire etkinliğe hiç başlamadığını/yapmadığını, çerçevesinin koyu yeşil içi açık yeşil daire çok fazla blok kullanılarak etkinliğin tamamlandığını, çerçeve yeşil içi boş daireler etkinliğe yeni başlanmış olduğunu etkinliğe devam ettiğini göstermektedir.



Resim 2.5. Code.org uygulama ekranı

Resim 2.5 Code.org uygulama ekranı görselinde Code.org platformunun uygulama arayüzü gösterilmektedir. Bu görsel de istenilen görev, görevleri yerine getirirken başvurulabilecek ipucu, kod blokları ve kod bloklarını sürükleyip bırakabildiğimiz çalışma alanı yer almaktadır. Öğrenciler kod bloklarını sürükleyerek çalışma alanına bırakıp kod bloklarını oluştururlar. Daha sonra çalıştır düğmesi tıklanarak kod bloğu çalıştırır. Etkinlik gerekli bloklar kullanarak gerçekleştirilememişse uyarı verir ve öğrenci tekrardan bloklarını düzenleyebilir. Etkinlik başarılı bir şekilde tamamlandığında ise bir sonraki etkinliğe geçilir. Geçilen etkinlik daireleri yeşil renk ile boyanır.



Resim 2.6. Code.org örnek etkinlik ekranı

Resim 2.6 Code.org örnek etkinlik ekranı görselinde Code.org platformunun örnek bir etkinliği gösterilmektedir. Etkinlik “Ders 3: Laurel ile hazine toplama” etkinliğidir. Laurel ile hazine toplama etkinliği toplam 13 alt etkinliğe sahiptir. Etkinliğin talimatlar kısmında öğrenci için ipuçları veya uyarılar yer alabilir. Blokların hemen üzerinde toplam kaç bloktan oluştuğu ve öğrencinin kaç tane blok kullandığı hakkında bilgi verilmektedir. Ayrıca öğretmenler için çözümleri de bulunmaktadır. Çözümüne bakın linkini tıklayarak çözüm görülebilir.



Resim 2.7. Code.org örnek etkinlik ekranı

Resim 2.7’de Code.org platformunun bir başka etkinliđi gösterilmektedir. Etkinlik “Ders 11: Koşullar-Arı” etkinliđidir. Koşullar Arı etkinliđi toplam 13 alt etkinliđe sahiptir. Etkinliđin talimatlar kısmında öğrenci için ipuçları veya uyarılar yer alabilir. Blokların hemen üzerinde toplam kaç bloktan oluştuđu ve öğrencinin kaç tane blok kullandığı hakkında bilgi verilmektedir. Ayrıca öğretmenler için çözümleri de bulunmaktadır. Çözüme bakın linkini tıklayarak çözüm görülebilir. Çözümü görmeyi bırak seçeneđi tıklanarak çözümü görme durdurulabilir.

2.8. Blok Tabanlı Programlama İle İlgili Araştırmalar

Clements ve Gullo (1984), çalışmalarında bilgisayar programlama öğrenmenin bilişsel gelişimi üzerindeki etkilerini araştırmışlardır. Çalışmanın katılımcıları birinci sınıfa giden 18 öğrencidir. Araştırma sonucunda programlama grubunun önemli ölçüde iyi puan aldığını belirtilmiştir.

Du ve ark. (2016), araştırmalarında katılımcıların programlamaya yönelik tutumları ve programlama yeteneklerindeki değişimi incelemişlerdir. Araştırmanın katılımcılarını iki üniversiteden lisans öğrencileri oluşturmaktadır. Katılımcılar Hour of Code kursunu tamamlamışlardır. Araştırmanın sonucunda kodlama saati eğitiminin kodlamaya yönelik tutumları üzerinde olumlu etkisi olduğu öğrencilerin programlama becerileri üzerinde önemli bir etkisi olmadığı sonucuna varılmıştır.

Yiğit (2016), araştırmasında görsel programlama ile öğretimin öğrencilerin programlamaya karşı tutumlarına ve programlamayı öğrenmelerine etkisini incelemiştir. Araştırmasında yarı deneysel desen kullanmıştır. Araştırmanın katılımcılarını Ondokuz Mayıs Üniversitesinde öğrenim gören 42 BÖTE öğrencisi oluşturmaktadır. Çalışmasında araştırmacı görsel programlama aracı olarak Blockly kullanmıştır. Kontrol grubu öğrencilerine ise geleneksel programlama öğretim dili olan Python kullanmıştır. Araştırmanın sonucunda, Blockly ile öğrenim gören öğrencilerin Python ile öğrenim gören öğrencilere göre programlamayı öğrenmede daha başarılı oldukları belirtilmiştir. Ayrıca Blockly ile eğitim gören öğrencilerin Python dili ile eğitim gören öğrencilere göre programlamaya karşı tutumları yüksek olduğu ve bu yüksekliğin istatistiksel olarak anlamlı olmadığı belirtilmiştir.

Jagušt ve ark. (2018), farklı bilgisayarsız etkinlikler uygulamışlar ve sonucunda bilgisayarsız etkinliklerin hem öğrenciler hem de veliler yönünden kabul gördüğünü, hesaplamalı ve algoritmik düşünmeye katkısının olduğunu belirtmişlerdir.

Çavdar (2018), Code.org platformunu, eğitim programı ve yazılımsal olarak incelemiştir. Çalışmasında karma yöntem modelini kullanmıştır. Araştırmanın katılımcılarını 22 ortaokul öğrencisi oluşturmaktadır. Araştırmanın sonucunda başarı testi yönünden son test lehine anlamlı bir artış olduğuna değinmiştir. Ayrıca belirlenen kazanımlara ulaşma konusunda platformun kısmen yeterli olduğundan ancak bu durumun öğrenme yaşantısı eksiklerinden veya başarı testinin puanlamasının dereceli puanlama anahtarına göre yapılmasından kaynaklanabileceğine değinmiştir.

Karaduman (2019), çalışmasında ortaokul 6. sınıf öğrencilerinin programlama öğrenimi üzerinde Code.org platformunun etkisini incelemiştir. Araştırma deneysel desen kullanmıştır. Araştırma sonucunda Code.org platformunun programlama öğrenmelerine ve programlama kalıcılığına normal öğretim yöntemlerine göre olumlu etkisi olduğunu belirtmiştir. Ayrıca bilgisayar destekli kodlama eğitimi tutumlarına yönelik de tutumlarını olumlu yönde artırdığını belirtmiştir.

Alp (2019), araştırmasında blok tabanlı programlama öğretiminin öğrencilerin problem çözme becerileri ve bilgisayara yönelik tutumları üzerindeki etkisini incelemiştir. Araştırmada yarı deneysel desen kullanılmıştır. Programlama öğretim aracı olan Scratch yazılımı kullanılmıştır. Araştırmanın katılımcılarını 6. sınıf öğrencileri oluşturmuştur. Araştırmanın sonucuna göre her iki grupta bulunan öğrencilerin problem çözme becerileri ve bilgisayara yönelik tutumları uygulama öncesine göre anlamlı derecede arttığı belirtilmiştir. Ancak araştırmanın sonucunda deney ve kontrol grubu arasında öğrencilerin bilgisayara yönelik tutumları açısından farklılaşma olmazken problem çözme becerileri yönünden deney grubu lehine anlamlı pozitif yönde farklılaşma görüldüğü belirtilmiştir.

Çilengir (2019), araştırmasında oyunlaştırma ile blok tabanlı programlamanın motivasyona ve başarıya etkisini incelemiştir. Araştırmada yarı deneysel desen kullanılmıştır. Araştırmanın katılımcılarını 6. sınıf 47 öğrenci oluşturmuştur. Araştırmada Scratch blok tabanlı yazılım kullanılmıştır. Araştırmanın sonucunda, oyunlaştırma yaklaşımı ile eğitim gören grubun başarısı geleneksel yöntemle eğitim gören gruba göre yüksek çıkmıştır. Bu

yüksekliğin anlamlı derecede olduğu belirtilmiştir. Kontrol ve deney grupları motivasyon bakımından incelendiğinde anlamlı derece farklılık ortaya çıkmamıştır.

Bakırcı (2019), araştırmasında programlama öğretiminde Scratch programını kullanımının öğrencilerin derse yönelik güdülenmeleri, algoritma geliştirmeye yönelik öz yeterlilik algıları ve programlama başarısına etkisini incelemek istemiştir. Araştırmada yarı deneysel desen kullanılmıştır. Araştırmanın katılımcılarını 6. sınıf 64 öğrenci oluşturmuştur. Araştırmanın sonucunda öğrencilerin öz yeterlilik algıları ve akademik başarı puanlarında istatistiksel olarak anlamlı bir artış görülmüştür. Ayrıca Scratch blok tabanlı uygulama yazılımı ile ders işlenen sınıfta öğrencilerin güdülenme düzeyinde artış görülürken geleneksel yöntemle dersin işlendiği grupta öğrencilerin güdülenme düzeylerinde azalma görülmüştür. Ancak gruplar arasında akademik başarı puanı, güdülenme, öz yeterlilik algısı yönünden anlamlı bir farklılık bulunmadığı belirtilmiştir.

Tonbuloğlu ve Tonbuloğlu (2019), çalışmalarında bilgisayarsız etkinliklerin hesaplamalı düşünme becerilerine etkisini incelemişlerdir. Çalışmaya 114 ortaokul öğrencisi katılmıştır. Araştırmanın sonucunda bilgisayarsız kodlama etkinliklerinin öğrencilerin hesaplamalı düşünme becerileri üzerinde olumlu katkısı olduğunu belirtmişlerdir.

Busuttil ve Formosa (2020), çalışmalarında 9. sınıf öğrencilerinden oluşan bir sınıfa bilgisayar öğretmek için pedagojik strateji bağlantısız bilişimi araştırmışlardır. Pedagojiyi bilgisayar kavramlarını öğretirken kullanmanın uygulanabilirliğini belirlemek için öğrencilerin bağlantısız etkinliklere maruz kaldıklarındaki tepkileri analiz etmişlerdir. Çalışmanın sonucunda bağlantısız bilişimin avantajları dezavantajlarına göre daha ağır bastığını karmaşık konuları tanıtmak için kullanıldığında faydasının olduğu belirtilmiştir.

Tiryaki (2020), araştırmasında kodlama eğitiminin, öğrencilerin yaratıcı düşünme becerileri ve programlama öz yeterlilik düzeylerine etkisini incelemiştir. Araştırmaya ortaöğretimde öğrenim gören 71 öğrenci katılmıştır. Araştırmada nicel yöntemlerden kontrol gruplu yarı deneysel desen kullanılmıştır. Araştırmanın sonucunda, uygulama öğrencilerin programlama öz yeterlilik düzeyini ve yaratıcı düşünme becerilerini artırdığını belirtmiştir.

Polat (2020), araştırmasında 6. sınıf öğrencilerinin programlama öğretiminde kullanılan bilgisayarlı ve bilgisayarsız etkinliklerin akademik başarı ve bilgisayarca düşünme becerilerine etkisini incelemiştir. Araştırmada karma desen kullanılmış, çalışmaya 84 öğrenci katılmıştır. Araştırmada blok tabanlı uygulama programı olarak Code.org, Scratch, Compute.it, LightBot ve bazı bilgisayar etkileşimi sağlayacak olan uygulamalar Kahoot.it, draw.io, Learningapp.org gibi uygulamalar kullanılmıştır. Bilgisayarsız etkinlik grubu içinde bilişim teknoloji ve yazılım dersi öğretim programı uygulanmıştır. Araştırmanın sonucunda, bilgisayarsız etkinlikler grubu lehine akademik başarı yönünden anlamlı fark bulunmuştur. Bilgisayarca düşünme becerileri yönünden ise gruplar arasında anlamlı bir fark bulunmamıştır. Araştırmanın nitel boyutunda ise bilgisayarlı etkinlikler grubundaki öğrenciler süreçteki etkinlikleri oyun gibi eğlenceli bulduklarını belirtmişlerdir. Bilgisayarsız etkinlik grubundaki öğrenciler ise konuları öğrendiklerini düşündükleri için sıkılmadıklarını ve kaygı duymadıklarını belirtmişlerdir. Sonuç olarak araştırmada, bilgisayarsız programlama etkinliklerinin ortaokul temel bilgisayar programlama becerisi kazandırmada etkili olduğu belirtilmiştir. Ayrıca bilgisayarca düşünme becerisi edinmelerinde de bilgisayarsız etkinliklerin bilgisayarlı etkinlikler kadar etkili olabildiği belirtilmiştir. Bilgisayarsız etkinliklerin bilgisayarlı etkinliğin yapılamayacak zamanlarda işlevsel olduğuna da değinilmiştir.

Ünsal (2020), araştırmasında öğrencilerin blok tabanlı etkinliklere katılmalarının etkinlik algılarına ve bilgi işlem becerilerine etkisini incelemiştir. Araştırmada yarı deneysel desen kullanılmıştır. Çalışmanın katılımcılarını 2. sınıfa giden 40 öğrenci oluşturmuştur. Araştırmada deney grubuna blok tabanlı uygulama olarak Code.org kullanılmış kontrol grubuna ise herhangi bir uygulama kullanılmamıştır. Grupların son testlerinde deney grubu lehine anlamlı bir fark bulunmuştur. Sonuç olarak ilkökul düzeyinde blok tabanlı uygulama ile ders işlemenin bilgi işlemsel düşünme becerilerini artırdığını belirtmiştir.

Arslan (2020), araştırmasında grup ve bireysel olarak çalışmanın öğrencilerin blok tabanlı programlama öz yeterlilik algıları ve robotik programlama tutumları üzerindeki etkisini incelemiştir. Çalışmanın katılımcılarını 7. sınıfa giden 32 öğrenci oluşturmuştur. Araştırma yarı deneysel desen ile desenlenmiştir. Araştırmada blok tabanlı programlama Scratch yazılımı kullanılmıştır. Araştırmanın sonucuna göre öğrencilerin grupla veya bireysel olarak çalışmaları robotik programlama tutumlarını ve blok tabanlı programlama öz yeterlilik algılarını anlamlı olarak etkilemediği belirtilmiştir. Ek olarak grupla veya

bireysel çalışmak öğrencilerin bulundukları grup içindeki blok tabanlı programlama öz yeterlilik algılarında anlamlı fark olduğu belirtilmiştir.

Threekunprap ve Yasri (2020), çalışmalarında oyun tabanlı öğrenme ortamında akış bloklarıyla fişsiz kodlama etkinliği gerçekleştirmişlerdir. Etkinlikte bilgisayar yardımı olmadan programlama ve hesaplamalı düşünme kavramlarını geliştirmeyi amaçlamışlardır. Çalışmanın katılımcılarını programlama bilgisi olmayan 160 ortaokul öğrencisi oluşturmuştur. Araştırmada nicel yöntem kullanmışlardır. Araştırmanın sonucunda öğrencilerin istatistiksel olarak gelişmiş öğrenme başarısı ve istatistiksel olarak önemli ölçüde artan öz yeterlilik sergilediğini belirtmişlerdir.

Küçükkara ve Aksüt (2021), okul öncesi öğrencileri için bilgisayarsız kodlamayı temel alan algoritma etkinliklerinin uygulama sürecini incelemiş sonuç olarak araştırmada kullanılan etkinlikler okul öncesi dönemindeki öğrencilerin problem durumlarına çözüm önerileri geliştirebildiklerini ve algoritma kavramını kazandıklarını belirtmişlerdir.

Arslan Namlı (2021), araştırmasında bilgisayarsız bilgisayar bilimi ve blok tabanlı programlama etkinliklerinin öğrencilerin öz yeterliliklerine, bilgi işlemsel düşünme becerilerine ve akademik başarılarına etkisini incelemiştir. Çalışmanın katılımcılarını üç farklı ortaokulda 5. sınıfta okuyan 84 öğrenci oluşturmuştur. Araştırmada nicel yöntemler kullanılmıştır. Araştırmanın deney1 grubunda bilgisayarsız bilgisayar bilimi etkinlikleri, deney2 grubunda blok tabanlı uygulama programı Scratch, kontrol grubunda ise geleneksel yöntemle ders anlatılmıştır. Araştırmanın sonucunda blok tabanlı programlama etkinliklerinin öğrencilerin bilgi işlemsel düşünme becerilerini olumlu yönde etkilediğini belirtmiştir. Ayrıca gruplar arasında bilgi işlemsel düşünme becerilerinin öz yeterliliklerinin gruplara göre farklılaşmadığı belirtilmiştir.

Yüksel (2022), araştırmasında öğrencilerin çoklu zekâ alanlarıyla blok tabanlı programlama dili Scratch'a yatkınlıkları arasında ilişkinin anlamlı olup olmadığını incelemiştir. Araştırmanın çalışma grubunu 45 ortaokul öğrencisi oluşturmuştur. Araştırma nicel yöntemlere göre sürdürülmüştür. Araştırmanın sonucunda, öğrencilerin çoklu zekâ alanlarının bazıları ile blok tabanlı programlama öz yeterlilik algıları arasında pozitif yönde anlamlı ilişki tespit edildiğini belirtmiştir.

Varçın (2022), araştırmasında öğrencilerin blok tabanlı programlamaya yönelik tutumlarını, üstbilişsel farkındalıklarını belirlemek ve üst bilişsel farkındalıkları ile programlamaya yönelik tutumlarını farklı değişkenlerle ilişkisini incelemiştir. Araştırmanın katılımcılarını 6. sınıf 340 öğrenci oluşturmuştur. Araştırmada blok tabanlı programlama aracı olarak Scratch yazılımı kullanılmıştır. Araştırma nicel yöntemlerden ilişkisel araştırma ve tarama araştırması ile yapılmıştır. Araştırmanın sonucunda, öğrencilerin blok tabanlı programlamaya tutumlarının genel olarak olumlu yönde olduğunu belirtmiştir. Ayrıca Öğrencilerin üst bilişsel farkındalıkları ile programlama yönelik tutumları arasında zayıf pozitif yönde ilişki olduğu belirtilmiştir.

Çanakçı (2022), araştırmasında temel programlama becerilerini, blok tabanlı programlama araçlarıyla eğitim alan öğrencilerle metin tabanlı programlama araçlarıyla eğitim alan öğrencilerin ileri düzey uygulama sonucunda elde edecekleri becerilerin arasındaki farkını ortaya çıkarmayı amaçlamıştır. Araştırmada karma araştırma deseni kullanılmıştır. Araştırmada blok tabanlı programlama aracı olarak Scratch, metin tabanlı yazılım olarak Python dili kullanılmıştır. Çalışmanın katılımcılarını 24 ortaokul öğrencisi oluşturmuştur. Araştırmanın sonucunda, blok tabanlı eğitim almadan metin tabanlı programlama etkinlikleri ile öğrenim gören grubun programlama da daha başarılı olabileceği, temel programlama becerilerini metin tabanlı uygulamalarda hayata geçirebilme konusunda blok tabanlı eğitimi ile başlayan gruba göre daha başarılı olduklarını belirtmiştir.

Zurnacı (2023), çalışmasında okul öncesi dönemde bilgisayarsız kodlama ve eğitsel robot kodlama etkinliklerini karşılaştırarak öğrencilerin yürütücü işlev ve bilgi işlemsel düşünme becerileri üzerindeki etkisini incelemiştir. Araştırmada yarı deneysel desen kullanılmıştır. Araştırmanın katılımcılarını 48 okul öncesi öğrencisi oluşturmuştur. Araştırmanın sonucunda, iki grupta bulunan öğrencilerin bilgi işlemsel düşünme becerileri ön test- son test puanları arasında son test lehine anlamlı fark bulunmuştur. Bilgisayarsız kodlama ve eğitsel robot ile eğitim yapan öğrencilerden bilgi işlemsel düşünme son test puanları arasında eğitsel robot grubu lehine anlamlı fark bulunmuştur. Ayrıca yürütücü işlev becerilerini desteklemek için yapılan çalışmalarda gruplar arasında anlamlı fark bulunmamıştır.

Hakkoz (2023), araştırmasında öğrencilerin blok tabanlı programlamaya ilişkin öz yeterlilik algısı ve eleştirel düşünme eğilimleri arasındaki ilişkiyi belirlemeye çalışmıştır.

Araştırmasında nicel yöntemlerden ilişkisel ve betimsel tarama yöntemlerini kullanmıştır. Araştırmanın katılımcılarını Tokat ilinde 5 farklı ortaokulda öğrenim gören toplam 629 öğrenci oluşturmuştur. Blok tabanlı programlama aracı olarak Scratch yazılımı kullanılmıştır. Araştırmanın sonucunda öğrencilerin blok tabanlı programlama öz yeterlilik algıları konusunda kendilerine biraz, eleştirel düşünme eğiliminde kendilerine oldukça güvendiklerini belirtmişlerdir. Sonuç olarak öğrencilerin eleştirel düşünme eğilimleri ve blok tabanlı programla öz yeterlilik algıları arasındaki ilişki arasında pozitif yönde orta düzeyde anlamlı bir ilişki bulunduğu belirtilmiştir.

Yılmaz (2023), araştırmasında bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin öğrencilerin bilgi işlemsel düşünce becerilerine etkisini incelemiştir. Çalışmasında deneysel desen kullanmıştır. Çalışmanın katılımcılarını 5. sınıf 36 öğrenci oluşturmuştur. Çalışmasında Scratch blok tabanlı uygulama programı, bilgisayarsız kodlama grubunda da Tospaa etkinlikleri kullanılmıştır. Kodlama etkinliklerinin öğrencilerin bilgi işlemsel düşünme becerilerini artırdığı belirtilmiştir. Ayrıca bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin grupların bilgi işlemsel düşünme becerilerine yönelik algısı arasında anlamlı bir ilişki bulunmamıştır. Aynı şekilde grupların bilgi işlemsel düşünme becerileri performansları arasında da anlamlı bir farklılık görülmemiştir.

Piedade ve Dorotea (2023), Scratch etkinliklerinin 4. sınıf öğrencilerinin geliştirdiği bilgisayarlı düşünme becerilerini ve elde ettikleri kavramlar üzerindeki etkisini analiz etmeyi hedeflemişlerdir. Çalışmaya 189 öğrenci katılmıştır. Çalışmanın sonucunda gruplar arasında anlamlı farklılıklar bulduklarını belirtmişlerdir. Scratch ile etkinlik yapanlar kontrol grubundaki öğrencilerden (Scratch) kullanmayanlardan yüksek puan almışlardır.

3.YÖNTEM

Çalışmanın bu bölümünde araştırma modeli, çalışma grubu, veri toplama araçları, uygulama süreci ve verilerin analizinden bahsedilmiştir.

3.1. Araştırma Modeli

Bu araştırmada nicel veri toplama tekniklerinden öntest-sontest kontrol gruplu yarı deneysel desen kullanılmıştır. Yarı deneysel desen, eşleştirilmiş grupların seçkisiz bir şekilde deney grupları olarak belirlendiği çalışmalardır (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz ve Demirel, 2020: 206). Bu bağlamda daha önceden var olan iki sınıf deney ve kontrol grubu olarak belirlenmiştir. Gruplara katılımcılar grup eşleştirme yöntemi ile seçilmiştir. Önceden oluşturulmuş gruplar üzerinden çalışmaların yapılmak zorunda kalındığında tercih edilebilir (Büyüköztürk ve ark., 2020: 216).

Çizelge 3.1. Deneysel desen çizelgesi

Gruplar	Ön Test	Deneysel Manipülasyon	Son Test
Deney Grubu	Programlama Akademik Başarı testi Programlama Öz Yeterlilik Ölçeği	Bilişim teknoloji araçları ile Code.org sitesi üzerinden ders akışı	Programlama Akademik Başarı testi Programlama Öz Yeterlilik Ölçeği
Kontrol Grubu	Programlama Akademik Başarı Testi Programlama Öz Yeterlilik Ölçeği	Kâğıt kalem-çıktı ile Code.org etkinlik akışı	Programlama Akademik Başarı Testi Programlama Öz Yeterlilik Ölçeği

3.2. Çalışma Grubu

Araştırmada seçkisiz olmayan örneklemenin bir türü olan uygun örnekleme kullanılmıştır. Uygun örnekleme, araştırmacının kolayca ulaşabileceği örneklemden verilerini toplaması olarak ifade edilir (Büyüköztürk ve ark., 2020: 95). Buna göre çalışma grubunu Amasya

ilinin Suluova ilçesindeki bir devlet okulunda eğitim gören 5. sınıf öğrencileri oluşturmaktadır. Araştırmada 5. sınıf öğrencileri seçilme sebebi daha önce bu dersi almamış olmalarıdır. Araştırma iki şubede okuyan 50 öğrenci üzerinde yapılmıştır.

Çizelge 3.2. Çalışma grubu bilgileri

Grup türü	Öğrenci Sayısı				
	Kız	%	Erkek	%	Toplam
Deney	7	29,16	17	70,83	24
Kontrol	12	46,15	14	53,84	26
Toplam	19	%38	31	%62	50

3.3. Veri Toplama Araçları

3.3.1. Programlama öz yeterlilik ölçeği

Ortaokul öğrencilerinin programlama öz yeterliliğini ölçmek amacı ile Kukul, Gökçe Arslan ve Günbatar (2017) tarafından geliştirilen programlama öz yeterlilik ölçeği kullanılmıştır. Programlama öz yeterlilik ölçeği 31 maddelik tek faktörden oluşmaktadır. Programlama öz yeterlilik ölçeğinde açımlayıcı faktör analizi yapılmıştır. Maddelerin faktör yük değerleri 0,473 ile 0,718 arasında değiştiği belirtilmiştir. Ölçeğin açıkladığı toplam varyans %41,15'tir. Ölçekte yer alan tüm maddeler orta ve yüksek düzeyde ilişkiye sahip olduğu görülmektedir (Kukul ve Ark., 2017). Madde testi korelasyon değerleri 0,492 ile 0,712 arasındadır. Madde geçerliliği için korelasyon değerleri ve ölçeğin homojenliği, ölçek maddelerinin geçerli olduğunu ve aynı şeyi ölçtüğünü kanıtlamaktadır (Kukul ve ark., 2017). Ölçeğin iç tutarlılığını belirlemek için Cronbach alfa katsayısı hesaplanmış cronbach alfa katsayısı 0,95, iki yarı metodu sonuçları ise 0,96 çıktığı belirtilmiştir. Bu sonuç ölçme aracının güvenilir olduğunu göstermiştir (Kukul ve ark., 2017).

3.3.2. Programlama akademik başarı testi

Akademik başarı testi ortaokul öğrencilerinin programlamaya dönük başarılarını ölçmek amacı ile geliştirilmiştir. Başarı testi geliştirilirken Milli Eğitim Bakanlığı Talim ve Terbiye Kurulu Başkanlığı tarafından yayımlanan bilişim teknolojileri ve yazılım dersi

çerçeve programı kazanımları dikkate alınarak belirtke çizelgesi oluşturulmuştur (TTKB, 2018). Belirtke çizelgesi kazanımları altı haftalık bir süreci kapsamaktadır. Başarı testi belirtke çizelgesindeki kazanımları kapsayacak şekilde çoktan seçmeli 30 soru olarak hazırlanmış (Bilgi düzeyinde 6, kavrama düzeyinde 3, uygulama düzeyinde 20 analiz düzeyinde 1) ve uzman görüşü alınarak bu dersi almış 101 öğrenciye uygulanarak pilot çalışma yapılmıştır.

Çizelge 3.3. Belirtke çizelgesi ve ilgili test sorusu

Kazanımlar	Bilgi	Kavrama	Uygulama	Analiz	Sentez	Değ.
BT.5.5.2.1. Prog. İle ilgili temel kav. Açıklar	2					
BT.5.5.2.2. Blok tabanlı programlama aracının arayüz ve özelliklerini tanıır.	1					
BT.5.5.2.3. Blok tabanlı programı ortamında sunulan hedeflere ulaşmak için doğru algoritmayı oluşturur		16	15,11			
BT.5.5.2.4. Doğrusal mantık yapısını açıklar.	7, 24					
BT.5.5.2.5. Doğrusal mantık yapısını kullanan algoritmalar geliştirir.			9,10,			
BT.5.5.2.6. Karar yapısını ve işlevlerini açıklar.	3,21					
BT.5.5.2.7. Karar yapıları içeren algoritmalar geliştirir.			6,19, 22,25			
BT.5.5.2.8. Döngü yapısını ve işlevlerini açıklar.	13	8				
BT.5.5.2.9. Döngü yapısı içeren algoritmalar oluşturur		27	12,14,17, 18, 5, 26, 28, 29, 30,4			
BT.5.5.2.10. Farklı yapılar için oluşturduğu algoritmaların sonucunu yordayarak hatalarını ayıklar.			20	23		

Pilot uygulama sonucunda madde ayırt edicilik gücü 0,30'un altında kalan maddeler testten çıkarılmıştır. Yapılan pilot uygulama sonucunda programlama akademik başarı testinin Kr-20 değeri 0,78, ortalama güçlük indeksi 0,58, ortalama madde ayırt ediciliği 0,41 olarak hesaplanmış ve maddelerin her birinin ayırt edicilik değerleri aşağıda çizelge 3.4 programlama başarı testine ilişkin madde güçlük indeksleri çizelgesinde gösterilmiştir.

Çizelge 3.4. Programlama başarı testine ilişkin madde güçlük indeksleri

Madde No	Ayırt Edicilik	Madde No	Ayırt Edicilik
M1	0,33	M12	0,66
M2	0,37	M13	0,51
M3	0,40	M14	0,66
M4	0,48	M15	0,51
M5	0,44	M16	0,66
M6	0,51	M17	0,55
M7	0,40	M18	0,48
M8	0,48	M19	0,33
M9	0,40	M20	0,40
M10	0,51	M21	0,40
M11	0,66		

Çizelge 3.4 programlama başarı testine ilişkin madde güçlük indekslerindeki verilere göre, programlama akademik başarı testinin programlama kazanımlarına ilişkin akademik başarılarını ölçebileceği söylenebilir.

3.4.Uygulama Süreci

Uygulama Amasya ilinin Suluova ilçesinde bir ortaokul'da altı hafta boyunca uygulanmıştır. Ön test ve son test uygulama süreleri bu süreye dâhil edilmemiştir. Uygulama sürecinde kazanımlar için ayrılan süre uygulama süreci kısmında ayrıntılı olarak ele alınmıştır. Deney ve kontrol gruplarında aynı etkinlik ve kazanım işlenmiştir. Konular deney grubunda Code.org sitesi üzerinden işlenmiştir. Kontrol grubunda ise Code.org sitesinden alınan çıktılar üzerinden işlenmiştir. Hafta hafta deney ve kontrol gruplarının konu-kazanım etkinlikleri detaylı olarak alt başlıklarda ele alınmıştır.

3.4.1. Deney grubu

Belirtilen konu ve kazanım ile deney grubunda haftada iki ders saati işlenmiştir. Belirlenen kazanımlar ve her haftaya uygun etkinlik Code.org sitesinden altı hafta boyunca yapılmıştır. Öğrencilerin haftalık etkinlikleri Code.org sitesinde sınıf oluşturularak takip edilmiştir.

Birinci hafta ders içeriği

Deney grubunda birinci hafta ilk ders bilişim teknolojileri ve yazılım dersinin “programlama çocuk oyunu” konusu işlenmiştir. Konunun kazanımları “programlama ile ilgili temel kavramları açıklar” şeklindedir. Dersin girişinde programlama kavramı ve programlama üzerine merak uyandırmak ve ön bilgileri hatırlatmak amacıyla öğrencilere aşağıdaki iki soru sorulmuştur.

1.Bilgisayarda programlama yapmak size ne ifade ediyor?

2. Bilgisayarda programlanarak yapılmış neler vardır?

Öğrencilerden cevapları aldıktan sonra günlük yaşamdan öğrenciye örnekler verilmiştir. Daha sonra programlama, programlama dili ve programcı kavramları öğrencilere açıklanmıştır. Blok tabanlı programlama sunusu öğrencilerle paylaşılmış bunun üzerine konuşulmuştur.

Birinci hafta 2. ders saati deney grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “programlama çocuk oyunu” konusu işlenmiştir. Konunun kazanımı “blok tabanlı programlama aracının arayüz ve özelliklerini tanıır.” şeklindedir. Bu dersin etkinliği Code.org sitesi incelenmesi ve öğrencilere program arayüzünün tanıtılmasıdır. Öğrenciler Code.org sitesinde bir sınıf oluşturarak o sınıfa tanımlanmıştır. Öğrencilere Code.org sitesi sınıf şifreleri verilerek uygulamaya giriş yapmaları sağlanmıştır. Öğrencilere siteyi incelemeleri için biraz süre verildikten sonra etkileşimli tahta üzerinden code.org programlama sitesi arayüzü tanıtılmış kod bloklarından bahsedilmiştir.

İkinci hafta ders içeriği

İkinci hafta deney grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “programlama çocuk oyunu” konusu işlenmiştir. Konunun kazanımı “blok tabanlı programlama ortamında sunulan hedeflere ulaşmak için doğru algoritmayı oluşturur” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde öğrencilere tanımlanmış “angry birds ile labirentte programlama” dersi altında 13 tane etkinlik bulunmaktadır. Labirentte programlama dersinde angry birds oyununun karakteri kullanarak öğrenciler, kuşu

labirentin diğerk ucundaki domuza götürmek için kod bloklarını kullanarak sıralı algoritmalar geliştirmeleri gerekmektedir. Labirentte programlama dersi altında bulunan 13 etkinliğı öğrenciler ders süresince yapmıştır. Öğrenci etkinlikleri Code.org sitesi üzerinde açılan sınıf üzerinden takip edilmiştir.

Üçüncü hafta ders içeriğı

Üçüncü hafta deney grubu öğrencilerine bilişim teknolojileri ve yazılım dersinin “labirentte kayboldum” konusu seçilmiştir. Bu konu Code.org sitesinden işleneceğı için koleksiyoncu programlama dersi ele alınmıştır. Konunun kazanımları “doğrusal mantık yapısını açıklar” ve “doğrusal mantık yapısını kullanan algoritmalar geliştirir.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde öğrencilere tanımlanmış “laurel ile hazine toplama” dersi altında 13 tane etkinlik bulunmaktadır. Bu haftanın diğerk bir etkinliğı “sanatçı-şununla say döngüleri” dersi altında 8 tane etkinlik bulunmaktadır. Bu derste öğrenciler artan karmaşık kod bloklarıyla karşılaşacaklar ve sağa dön, sola dön, 100 px ilerle vb. kodları tanıyacaklar; karşımıza çıkan yeni bir karakter laurel ile öğrenciler, laurel’in yolda yürürken hazine toplamasını sağlayacak sıralı kod blokları hazırlayacaklardır. Öğrenci ilerlemeleri Code.org sitesi üzerinde açılan sınıf üzerinden takip edilmiştir.

Dördüncü hafta ders içeriğı

Dördüncü hafta 1. ders için deney grubu öğrencilerine bilişim teknolojileri dersinin “kaplumbağaya kadar hızlıyım” konusu seçilmiştir. Konu Code.org sitesi üzerinden ele alınacağı için koşullar arı dersi üzerinden gidilmiştir. Konunun kazanımı “doğrusal karar yapısını ve işlevlerini açıklar.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde öğrencilere tanımlanmış “arı ile eğer/başka” dersi altında 10 alt etkinlik bulunmaktadır. Bu derste öğrenciler, koşullarla kodlamaya başlayacak ve programda karşılaştığı koşullara göre farklı bir biçimde kod blokları oluşturabilecekler. Ders süresince koşullar arı etkinlik altında bulunan 10 alt etkinlik yapılmıştır. Öğrenci ilerlemeleri Code.org sitesi üzerinde açılan sınıf üzerinden takip edilmiştir.

Dördüncü hafta 2. ders için deney grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “Code.org ile çözebildiğimiz problemler” konusu işlenmiştir. Konunun kazanımı

“karar yapıları içeren algoritmalar geliştirir.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde öğrencilere tanımlanmış “çiftçi döngüleri” dersi altında 13 alt etkinlik bulunmaktadır. Çiftçi döngüleri etkinliğinde öğrenciler farmer’deki çukurları doldurmak ve toprak kazmak için çalışırlar. Öğrenciler bu deliklerin boyutunu ve toprak öbeklerinin yükseklerini bilmedikleri için yeni bir döndü kurmaya çalışırlar. Ders süresince bu etkinlik altında bulunan 13 etkinlik yapılmıştır. Öğrencilerin ilerlemeleri Code.org sitesinde tanımlanan sınıf üzerinden takip edilmiştir.

Beşinci hafta ders içeriği

Beşinci hafta deney grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “Code.org ile çözebildiğimiz problemler” konusu işlenmiştir. Konunun kazanımı “döngü yapısını ve işlevlerini açıklar.” şeklindedir. Bu haftanın ilk ders etkinliğinde Code.org sitesinde öğrencilere tanımlanmış “reys ve bb-8 ile döngüler” dersi altında 14 tane etkinlik bulunmaktadır. Reys ve bb-8 ile döngüler etkinliğinde star wars karakteri öğrencilerden bb-8 labirentini geçmesine yardımcı olunması için döngüler kullanılmasını istiyor. Bu haftanın 2. ders etkinliğinde ise Code.org sitesinde öğrencilere tanımlanmış “labirentte iç içe döngüler” dersi altında 13 tane etkinlik bulunmaktadır. Labirentte iç içe döngüler etkinliğinde öğrenciler döngü anlayışlarını yeni bir seviyeye taşımaya fırsat bulabilecekler. Etkinlikte öğrenciler; bitkiler, arılar ile oynayarak bir döngünün içine başka bir döngünün nasıl programlanabileceğini öğrenecekler. Bu etkinliklerin altındaki 27 etkinlik yapılarak ders pekiştirilmiştir. Öğrenci ilerlemeleri Code.org sitesinde tanımlanan sınıf üzerinden takip edilmiştir.

Altıncı hafta ders içeriği

Altıncı hafta deney grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “Code.org ile çözebildiğimiz problemler” konusu seçilmiştir. Konunun kazanımı “Farklı yapılar için oluşturduğu algoritmaların sonucunu yordayarak hatalarını ayıklar” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde öğrencilere tanımlanmış “labirentte hata ayıklama” dersi altında 10 tane etkinlik bulunmaktadır. Labirentte hata ayıklama dersinde öğrenciler yanlış çözülmüş bulmacalar ile karşılaşırlar ve bu kodları düzenlerler. Labirentte hata ayıklama etkinliği altında bulunan 10 etkinlik yapılmıştır. Öğrenci ilerlemeleri Code.org sitesinde tanımlanan sınıf üzerinden takip edilmiştir.

3.4.2. Kontrol Grubu

Belirtilen konu ve kazanım ile kontrol grubunda haftada iki ders saati işlenmiştir. Belirlenen kazanımlar altı hafta boyunca işlenmiştir. Haftalara uygun etkinlikler Code.org sitesinden çıktı alınarak gerekli ön hazırlıklar yapılmış ve sınıfta öğrencilerle etkinlikler yapılmıştır. Öğrencilerin haftalık etkinlik takipleri bir etkinlik takip çizelgesi oluşturularak takip edilmiştir.

Birinci hafta ders içeriği

Kontrol grubunda birinci hafta ilk ders bilişim teknolojileri ve yazılım dersinin “programlama çocuk oyunu” konusu işlenmiştir. Konunun kazanımı “programlama ile ilgili temel kavramları açıklar” şeklindedir. Dersin girişinde programlama kavramı ve programlama üzerine merak uyandırmak ve ön bilgileri hatırlatmak amacıyla öğrencilere aşağıdaki iki soru sorulmuştur.

1. Bilgisayarda programlama yapmak size ne ifade ediyor?
2. Bilgisayarsız programlama yapmak size ne ifade ediyor?
3. Bilgisayarda programlanarak yapılmış neler vardır?

Öğrencilerden cevapları aldıktan sonra günlük yaşamdan öğrenciye örnekler verilmiştir. Daha sonra programlama, programlama dili ve programcı kavramları öğrencilere açıklanmıştır. Blok tabanlı programlama sunusu öğrencilerle paylaşılmış bunun üzerine konuşulmuştur.

Birinci hafta ikinci ders kontrol grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “programlama çocuk oyunu” konusu işlenmiştir. Konunun kazanımı “Blok tabanlı programlama aracının arayüz ve özelliklerini tanıır.” şeklindedir. Bu haftanın etkinliği Code.org sitesi incelenmesi ve öğrencilere program arayüzünün tanıtılmasıdır. Öğrencilere Code.org sitesi arayüz ve bloklarının bulunduğu çıktılar dağıtılarak açıklamalarda bulunulmuştur.

İkinci hafta ders içeriği

İkinci hafta kontrol grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “programlama çocuk oyunu konusu işlenmiştir. Konunun kazanımı “blok tabanlı programlama ortamında sunulan hedeflere ulaşmak için doğru algoritmayı oluşturur” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde “angry birds ile programlama” dersi altında 13 tane etkinlik bulunmaktadır. Labirentte programlama dersinde angrybirds oyununun karakteri kullanarak öğrenciler, kuşu labirentin diğer ucundaki domuza götürmek için kod bloklarını kullanarak sıralı algoritmalar geliştirmeleri gerekmektedir. Labirentte programlama dersi için bu alt 13 etkinlik çıktı alınarak öğrencilere dağıtılmıştır. Öğrenciler, etkinlikleri ders süresince yapmıştır. Öğrencilerin, etkinlik ilerlemeleri oluşturulan etkinlik çizelgesi ile takip edilmiştir.

Üçüncü hafta ders içeriği

Üçüncü hafta kontrol grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “labirentte kayboldum” konusu seçilmiştir. Bu konu Code.org sitesinden işleneceği için koleksiyoncu programlama dersi ele alınmıştır. Konunun kazanımı “doğrusal mantık yapısını açıklar” ve “doğrusal mantık yapısını kullanan algoritmalar geliştirir.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde “laurel ile hazine toplama” dersi altında 13 tane etkinlik bulunmaktadır. Bu haftanın diğer bir etkinliği “sanatçı-şununla say döngüleri” dersi altında 8 tane etkinlik bulunmaktadır. Bu derste öğrenciler artan karmaşık kod bloklarıyla karşılaşacaklar ve sağa dön, sola dön, 100 px ilerle vb kodları tanıyacaklar; karşımıza çıkan yeni bir karakter laurel ile öğrenciler, laurel’in yolda yürürken hazine toplamasını sağlayacak sıralı kod blokları hazırlayacaklardır. Öğrencilere etkinlik çıktısı üzerinden açıklama yapılarak etkinlik çıktıları öğrencilere dağıtılmıştır. Öğrenciler, etkinlikleri ders süresince yapmıştır. Öğrencilerin, etkinlik ilerlemeleri oluşturulan etkinlik çizelgesi ile takip edilmiştir.

Dördüncü hafta ders içeriği

Dördüncü hafta kontrol grubu öğrencilerine teknolojileri dersinin “kaplumbağa kadar hızlıyım” konusu seçilmiştir. Konu Code.org sitesi üzerinden ele alınacağı için arı ile eğer/başka dersi üzerinden gidilmiştir. Konunun kazanımı “doğrusal karar yapısını ve işlevlerini açıklar.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde “arı ile eğer/başka” dersi altında 13 tane etkinlik bulunmaktadır. Bu derste öğrenciler, koşullarla kodlamaya başlayacak ve programda karşılaştığı koşullara göre farklı bir biçimde kod blokları oluşturabilecekler. Öğrencilere etkinliklere başlamadan önce açıklamalarda bulunulmuş ve etkinlik çıktıları öğrencilere dağıtılmıştır. Öğrencilerin, etkinlik ilerlemeleri oluşturulan etkinlik çizelgesi ile takip edilmiştir.

Dördüncü hafta 2. ders kontrol grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “Code.org ile çözebildiğimiz problemler” konusu işlenmiştir. Konunun kazanımı “karar yapıları içeren algoritmalar geliştirir.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde “farmer'da süresince döngüleri” dersi altında 13 tane etkinlik bulunmaktadır. Çiftçi döngüleri etkinliğinde öğrenciler farmer'deki çukurları doldurmak ve toprak kazmak için çalışırlar. Öğrenciler bu deliklerin boyutunu ve toprak öbeklerinin yükseklerini bilmedikleri için yeni bir döndü kurmaya çalışırlar. Öğrencilere etkinliklere başlamadan önce açıklamalarda bulunulmuş ve etkinlik çıktıları öğrencilere dağıtılmıştır. Öğrencilerin, etkinlik ilerlemeleri oluşturulan etkinlik çizelgesi ile takip edilmiştir.

Beşinci hafta ders içeriği

Altıncı hafta kontrol grubu öğrencileri ile bilişim teknolojileri ve yazılım dersinin “Code.org ile çözebildiğimiz problemler” konusu işlenmiştir. Konunun kazanımı “döngü yapısını ve işlevlerini açıklar.” ve “döngü yapısı içeren algoritmalar oluşturur.” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde “reys ve bb-8 ile döngüler” dersi altında 14 tane etkinlik bulunmaktadır. Reys ve bb-8 ile döngüler etkinliğinde star wars karakteri öğrencilerden bb-8 labirentini geçmesine yardımcı olunması için döngüler kullanılmasını istiyor. Bu haftanın 2. ders etkinliğinde de Code.org sitesinde “labirentte iç içe döngüler” dersi altında 13 tane etkinlik bulunmaktadır. Labirentte iç içe döngüler etkinliğinde öğrenciler döngü anlayışlarını yeni bir seviyeye taşımaya fırsat bulabilecekler. Etkinlikte öğrenciler; bitkiler, arılar ile oynayarak bir döngünün içine başka bir döngünün

nasıl programlanabileceğini öğrenecekler. Bu etkinlik altında bulunan 13 etkinlik yapılarak ders pekiştirilecektir.

Altıncı hafta ders içeriği

Altıncı hafta kontrol grubu öğrencileri ile bilişim teknolojileri dersinin “Code.org ile çözebildiğimiz problemler” konusu seçilmiştir. Konunun kazanımı “farklı yapılar için oluşturduğu algoritmaların sonucunu yordayarak hatalarını ayıklar” şeklindedir. Bu haftanın etkinliğinde Code.org sitesinde “labirentte hata ayıklama” dersi altında 10 tane etkinlik bulunmaktadır. Labirentte hata ayıklama dersinde öğrenciler yanlış çözülmüş bulmacalar ile karşılaşır ve bu kodları düzenlerler. Labirentte hata ayıklama etkinliği altında bulunan 10 etkinlik çıktıları öğrencilere dağıtılmıştır. Etkinlikleri öğrenciler verilen süreler içinde gerçekleştirmişlerdir. Öğrencilerin, etkinlik ilerlemeleri oluşturulan etkinlik çizelgesi ile takip edilmiştir.

3.5. Verilerin Analizi

Araştırmada toplanan veriler SPSS istatistik programında analiz edilmiştir. Toplanan veriler üzerinde ilk olarak normal dağılım gösterip göstermediğine bakılmıştır. Bilgisayarlı ve bilgisayarsız programlama öğretimi akademik başarı testi ve programlama öz yeterlilik ölçeğinden edinilen puanların analizinde; deney ve kontrol gruplarının son test karşılaştırılmasında Ancova (kovaryans) analizi kullanılmıştır. Ancova (kovaryans) analizinin amacı araştırmada etkisi incelenen faktörlerin dışında, bağımlı değişken ile ilişkisi bulunan değişkenlerin istatistiksel olarak kontrol altına alınmasını sağlamaktır (Büyüköztürk, 2020: 121). Bu araştırmada da deney ve kontrol grubu ön test puanlarını kontrol altına almak için kovaryans analizi kullanılmıştır. Bu bağlamda deney ve kontrol gruplarının ön testleri kontrol altına alınarak son testleri karşılaştırılmış bulunan sonuçlar bulgular bölümünde detaylı olarak açıklanmıştır.

4. BULGULAR VE YORUMLAR

Çalışmanın bu bölümünde programlama akademik başarı testi ile ilgili yapılan normallik analizleri, programlama öz yeterlilik ile ilgili normallik analizleri ve uygulanan ancova (kovaryans) analizi sonuçlarına yer verilmiştir.

4.1. Normallik Analizleri

Toplanan veriler üzerinde ilk olarak deney ve kontrol grupların programlama akademik başarı ön test puanına ve programlama öz yeterlilik ön test toplam puanlarına normallik analizi yapılmıştır. Grup büyüklüğü 50'den az olduğu durumlarda Shapiro-Wilk, grup büyüklüğü 50'den fazla olan durumlarda Kolmogorov-Smirnov testi puanların normalliğe uygunluğunu incelemeye kullanılan testlerdir (Büyüköztürk, 2020: 42). Çalışma grubundaki öğrenci sayısı 50 olduğundan dolayı (n=50) normallik testinde Shapiro-Wilk sonuçlarına bakılmıştır. Shapiro-Wilk test sonuçları .05 anlamlık üzerinde verilerin normal dağılım gösterdiği söylenebilir (Büyüköztürk, 2020: 42).

4.1.1. Programlama akademik başarı testi ile ilgili yapılan normallik analizleri

Analizimizin ilk kısmında kontrol ve deney gruplarının programlama akademik başarı ön test normallik analizlerine bakılmıştır.

Çizelge 4.1. Kontrol grubu programlama akademik başarı ön test normallik analizi

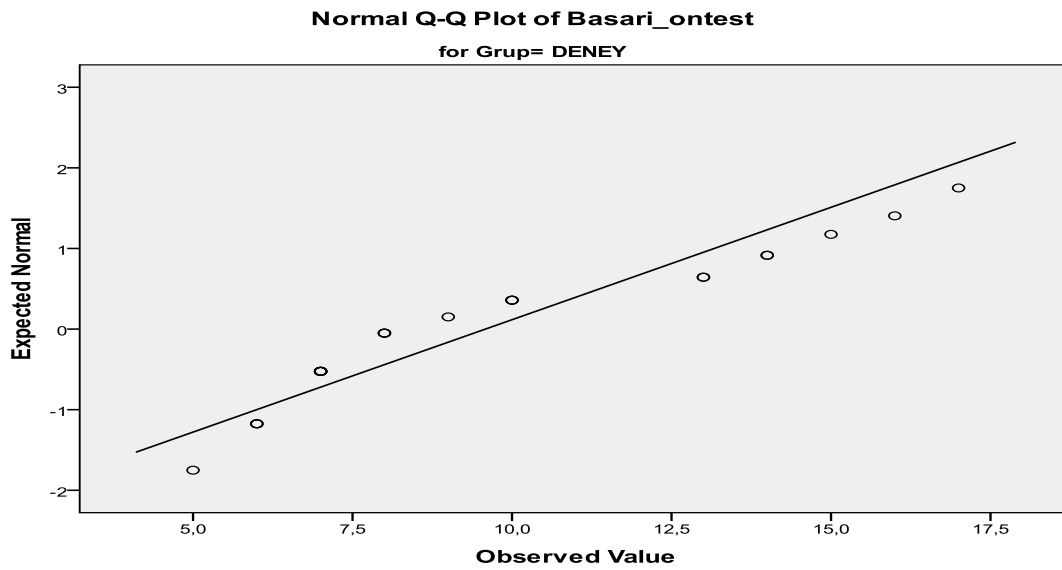
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
KONTROL	,167	26	,060	,941	26	,143

Çizelge 4.1 kontrol grubu programlama akademik başarı ön test normallik analizi incelendiğinde kontrol grubunun Shapiro- Wilk testinde anlamlılık değeri (p).143 çıkmıştır. Bu değer. 05 değerinden büyük olduğu için kontrol grubunun akademik başarı ön testinin normal dağılım gösterdiğini söyleyebiliriz.

Çizelge 4.2. Deney grubu programlama akademik başarı ön test normallik analizi

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
DENEY	,212	24	,007	,882	24	,009

Çizelge 4.2 deney grubu programlama akademik başarı ön test normallik analizi incelendiğinde deney grubunun Shapiro- Wilk testinde anlamlılık değeri (p).009 çıkmıştır. Bu değer .05 değerinden küçük olduğu için Skewness değeri incelenmiştir. Skewness değeri .754 olduğundan deney grubu akademik başarı ön testinin normal dağılım gösterdiğini söyleyebiliriz. Skewness (çarpıklık katsayısının) ∓ 1 sınırları içinde ise puanların normal dağılımdan önemli bir sapma göstermediği söylenilebilir (Büyüköztürk, 2020: 40). Ayrıca dağılımın normalliği incelenmesinde başvuru olan diğer bir özellikte grafik incelemesidir (Büyüköztürk, 2020: 40).



Şekil 4.1. Deney grubu normal q-q başarı öntest grafiği

Grafik köşegenler odağında bir doğru etrafında toplanıyorsa normal dağılım varsayımı sağlanıyor demektir (Terzi, 2014). Bu bağlamda Resim 4.1 deney grubu normal q-q başarı öntest grafiği incelendiğinde deney grubu programlama akademik başarı ön testinin normal dağılım gösterdiğini söyleyebiliriz.

4.1.2. Programlama öz yeterliliği ile ilgili yapılan normallik analizleri

Analizimizin ilk kısmında kontrol ve deney gruplarının programlama öz yeterliliği ile ilgili ön test normallik analizlerine bakılmıştır.

Çizelge 4.3. Kontrol grubu programlama öz yeterlilik ön test normallik analizi

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
KONTROL	,136	26	,200*	,972	26	,670

Çizelge 4.3 kontrol grubu programlama öz yeterlilik ön test sonuçları incelendiğinde kontrol grubunun Shapiro- Wilk testinde anlamlılık değeri (p).670 çıkmıştır. Bu değer. 05 değerinden büyük olduğu için kontrol grubu programlama öz yeterlilik ön testinin normal dağılım gösterdiğini söyleyebiliriz.

Çizelge 4.4. Deney grubu programlama öz yeterlilik ön test normallik analizi

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	p	Statistic	df	p
DENEY	,132	24	,200*	,928	24	,090

Çizelge 4.4 deney grubu programlama öz yeterlilik ön test normallik analizi incelendiğinde kontrol grubunun Shapiro- Wilk testinde anlamlılık değeri (p).090 çıkmıştır. Bu değer. 05 değerinden büyük olduğu için deney grubu programlama öz yeterlilik ön testinin normal dağılım gösterdiğini söyleyebiliriz.

4.2. Uygulanan Ancova Analizi sonuçları

Analizimizin bu bölümünde uygulanan ancova (kovaryans) analiz sonuçları yer almaktadır.

4.2.1. Bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama akademik başarısı üzerindeki etkisine yönelik bulgular

Analizimizin bu kısmında bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama akademik başarısı üzerindeki etkilere yönelik bulguları yer almaktadır. Yapılan analizlerde deney ve kontrol grubu programlama akademik başarı testi öntest ortalama ve düzeltilmiş ortalamaları ile deney ve kontrol grubu programlama akademik başarı son test analizlerine bakılmıştır.

Çizelge 4.5. Deney ve kontrol grubu programlama akademik başarı testi ön test ortalama ve düzeltilmiş ortalama puanları

Grup	N	$\bar{x}$	Düzeltilmiş Ortalama
Deney	24	13,71	13,385
Kontrol	26	14,15	14,452

Çizelge 4.5 deney ve kontrol grubu programlama akademik başarı testi ön test ortalama ve düzeltilmiş ortalama puanları incelendiğinde grupların başarı ön test puanlarına göre ortalamaları deney grubu için 13,71, kontrol grubu için 14,15 olarak hesaplanmıştır. Bu puanlara bakarak farkın olduğu ve kontrol grubu puanının yüksek olduğunu söyleyebiliriz. Grupların ön test puanları kontrol altına alındığında düzeltilmiş ön test ortalama puanlarında değişimler olduğu görülmektedir. Düzeltilmiş ortalama puanlarına göre başarı ön test deney grubu için 13,38, kontrol grubu için 14,45 olarak hesaplanmıştır.

Çizelge 4.6. Deney ve kontrol grubu programlama akademik başarı son test analiz sonuçları

Gruplar	Kareler Toplamı	sd	Kareler ortalaması	F	P
Ön test	509,680	1	509,680	63,598	
Grup	14,066	1	14,066	1,755	.192
Hata	376,663	47	8,014		
Toplam	10605,000	50			
Düzeltilmiş Toplam	888,820	49			

Çizelge 4.6 deney ve kontrol grubu programlama akademik başarı son test analiz sonuçları incelendiğinde grupların düzeltilmiş ön test puanlarına göre deney grubu ve kontrol grubu son test sonuçları arasında yapılan ancova (kovaryans) analizi sonuçlarına göre anlamlı bir fark bulunmamaktadır ($F(1,47)=1,75$, $p>0.5$).

4.2.2. Bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama öz yeterliliği üzerindeki etkisine yönelik bulgular

Analizimizin bu kısmında bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlama öz yeterliliği üzerindeki etkilere yönelik bulguları yer almaktadır. Yapılan analizlerde deney ve kontrol grubu programlama öz yeterlilik ön test ortalama ve düzeltilmiş ortalama puanları ile deney ve kontrol grubu programlama öz yeterlilik son test analiz sonuçlarına bakılmıştır.

Çizelge 4.7. Deney ve kontrol grubu programlama öz yeterlilik ön test ortalama ve düzeltilmiş ortalama puanları

Grup	N	$\bar{x}$	Düzeltilmiş Ortalama
Deney	24	118,4167	117,553
Kontrol	26	102,6538	103,451

Çizelge 4.7 deney ve kontrol grubu programlama öz yeterlilik ortalama ve düzeltilmiş ortalama puanları incelendiğinde grupların öz yeterlilik toplam ön test ortalama puanları

deney grubu için 118,41, kontrol grubu için 102,65 olarak hesaplanmıştır. Bu puanlara bakarak farkın olduğu ve deney grubu ortalama puanının yüksek olduğunu söyleyebiliriz. Grupların programlama öz yeterlilik ön test toplam puanları kontrol altına alındığında düzeltilmiş öz yeterlilik ön test puanlarında değişimler olduğu görülmektedir. Düzeltilmiş ortalama puanlarına göre programlama öz yeterlilik ön test deney grubu için 117,553 kontrol grubu için 103,451 olarak hesaplanmıştır.

Çizelge 4.8. Deney ve kontrol grubu programlama öz yeterlilik son test analiz sonuçları

Gruplar	Kareler Toplamı	sd	Kareler Ortalaması	F	p
Ön test	296,887	1	296,887	.651	.424
Grup	2223,937	1	2223,937	4,880	.032
Hata	21420,831	47	455,762		
Toplam	632241,000	50			
Düzeltilmiş Toplam	24818,580	49			

Çizelge 4.8 deney ve kontrol grubu programlama öz yeterlilik son test analiz sonuçları incelendiğinde grupların düzeltilmiş ön test puanları dikkate alındığında deney ve kontrol grubu son test toplam puan sonuçları arasında yapılan ancova (kovaryans) analizi sonuçlarına göre deney grubu lehine anlamlı bir fark olduğu söyleyebiliriz. ($F(1,47)=1.75$, $p<0.5$).

5. SONUÇ, TARTIŞMA VE ÖNERİLER

Çalışmanın bu bölümünde araştırmadan elde edilen analizlere göre sonuçlar, literatürle bu sonuçların karşılaştırılması, tartışma ve ileride yapılacak araştırmalara yönelik öneriler yer almaktadır.

5.1. Sonuç ve Tartışma

Bu çalışmanın amacı ortaokullarda bilgisayarlı ve bilgisayarsız programlama öğretiminin programlama akademik başarısına ve programlama öz yeterliliğine etkisini incelemektir. Araştırmanın sonucunda grupların kendi içlerinde programlama akademik başarı ön test ve son test puanları arasında anlamlı fark bulunmuştur. Bilgisayarsız ve bilgisayarlı programlama etkinliklerin her iki grup içinde farklılığın programlama akademik başarı son test lehine olması her iki gruptaki öğrencilerin programlama başarılarını artırdığını göstermektedir. Asıl problemimiz olan gruplar arası farklılık incelendiğinde bilgisayarlı ve bilgisayarsız programlama öğretiminde gruplar arasında programlama akademik başarı üzerinde anlamlı farklılık bulunmamıştır. Bilgisayarlı ve bilgisayarsız programlama öğretiminde gruplar arasında farklılık çıkmaması iki grubunda öğrenme sürecindeki etkinliklerde aktif rol almalarına bağlanabilir. Bu bulgunun literatürlede benzerlik gösterdiğini söyleyebiliriz. Karaduman (2019) araştırması sonucunda Code.org platformunun programlama öğrenmelerine katkısı olduğu sonucuna ulaşmıştır. Yılmaz ve İzmirli (2023) bilgisayarlı etkinlik grubunda Scratch, bilgisayarsız grupta Tospaa etkinlikleri kullanmışlar ve grupların bilgi işlemsel düşünme becerileri algısı arasında anlamlı bir farklılık olmadığını belirtmişlerdir. Kinaneva, Hristov, Zahariev, ve Raychev (2019) yaptıkları çalışmalarında blok tabanlı yazılımlarla drone kontrolüne yönelik programlama kodlarını geliştiren öğrencilerin başarılarını incelemişler ve sonucunda öğrencilerin mantıksal düşünmelerini artırdığını ve teknik becerilerini geliştirmelerine yardımcı olduğunu belirtmişlerdir. Vihavainen, Airaksinen ve Watson (2014) öğretim müdahalelerinin programlama başarısını neredeyse aynı oranda artırdığını aralarında anlamlı bir fark olmadığını ve öğretim müdahalelerinin etkililiğinde marjinal farklılıklar olduğunu belirtmişlerdir. Deniz ve Korucu (2023)' da çalışmalarında kodlama eğitiminde Scratch uygulamasının kullanılması öğrencilerin programlamaya dönük becerilerinde kodlama eğitime olumlu katkı sağladığı ve anlamlı fark yarattığını belirtmişlerdir. Weintrop (2021), öğrencilerin metin tabanlı bir dil öğrenen öğrencilere kıyasla blok tabanlı

programlama araçlarıyla programlamayı öğrenen öğrencilerin daha iyi puanlar aldıklarını belirtmiştir. Saygıner ve Tüzün (2023), araştırmasında blok tabanlı programlama aracı Scratch'ın öğrenci başarısına etkisini araştırmış sonucunda programlama öğretiminde blok tabanlı programlama öğretimi öğrencilerin programlama becerilerine olumlu katkı sağladığını belirtmişlerdir.

Bu araştırmanın alt problemlerinden biri de ortaokullarda bilgisayarlı ve bilgisayarsız programlama öğretiminin programlama öz yeterliliğine etkisini incelemektir. Araştırmanın sonucunda grupların kendi içlerinde programlama öz yeterlilik ön test ve son test puanları arasında anlamlı fark bulunmuştur. Bilgisayarsız ve bilgisayarlı programlama etkinliklerin her iki grup içinde farklılığın programlama öz yeterlilik son test lehine olması her iki gruptaki öğrencilerin programlama öz yeterliliklerini artırdığını göstermektedir. Tiryaki (2020) araştırmasında uygulama öğrencilerin programlama öz yeterlilik düzeyini artırdığı sonucuna ulaşmıştır. Asıl alt problemimiz olan gruplar arası farklılık incelendiğinde bilgisayarlı ve bilgisayarsız programlama öğretiminde gruplar arasında programlama öz yeterliliği üzerinde deney grubu lehine anlamlı farklılık bulunmuştur. Bilgisayarlı ve bilgisayarsız programlama öğretiminde bilgisayarlı grup lehine farklılık olması teknolojik araç gereçlerin daha çok dikkat çekmesi, blok tabanlı kodlama uygulamasının öğrencilere oyun gibi gelmesi ayrıca etkinlikleri tekrar edebilme ve etkinliklerde yapılan hatalarda anında dönüt-düzeltilmelerin olması öğrencilerin öz yeterlilik algılarını artırdığı yorumu yapılabilir. Bilgisayarsız etkinlik grubundaki öğrenciler kâğıt kalem etkinliklerini tekrar etmekten sıkıldıklarından sadece verilen etkinlikleri başarıyla sonuçlandırmaya odaklandıklarından dolayı öz yeterlilik algılarını bilgisayarlı etkinlik grubundaki öğrenciler kadar geliştiremedikleri yorumu yapılabilir. Weintrop ve Wilensky (2017) araştırmasında blok tabanlı kodlama grubundaki öğrencilerin daha fazla öğrenme ve bilgisayar derslerine yüksek ilgi gösterdiğini; metin tabanlı programlama grubundaki öğrencilerde ise programlama yeteneklerini geliştirmede daha etkili gördüklerini belirtmişlerdir. Namlı ve Aybek (2022) çalışmalarında programlama öğretiminin programlama öz yeterliliklerine olumlu katkı sağladığını belirtmişlerdir. Kasalak ve Altun (2020), ortaokul düzeyinde robotik kodlama etkinliklerinin blok tabanlı programlamaya ilişkin öz yeterliliklerini araştırmış araştırması sonucunda öğrencilerin programlamaya ilişkin öz yeterliliklerinin olumlu yönde arttığını ifade etmişlerdir. Soykan ve Kanbul (2018) çalışmalarının sonucunda öğrencilerden kodlama eğitimi alanların kodlama eğitimi almayanlara göre öz yeterliliklerinin daha yüksek olduğunu belirtmişlerdir. Ramalingam ve

ark. (2004) çalışmalarında programlama öz yeterliliğinin önceki deneyimlerinden etkilendiğini ve seviye ilerledikçe programlama öz yeterliliklerinin arttığını belirtmişlerdir. Deniz ve Korucu (2023) da çalışmalarında kodlama eğitiminde Scratch uygulamasının kullanılması öğrencilerin öz yeterlilik algılarına olumlu yönde katkısı olduğunu ifade etmişlerdir. Namli ve Aybek (2022) ise yaptıkları çalışmalarında programlama dersinin; programlama etkinliklerinin soyutlama, ayırttırma, genelleme ve akıl yürütme öz yeterliliklerine olumlu katkı sağladığını belirtmişlerdir. Yılmaz ve İzmirli (2023) bilgisayarlı kodlama yöntemlerinden olan blok tabanlı kodlamanın ve bilgisayarsız kodlamanın çocuklara programlamayı öğretmede etkili olduğunu belirtmişlerdir. Brackmann, Román-González, Robles, Moreno-León, Casali ve Barone (2017), çalışmalarında fişsiz etkinliklerin faydasının ilkökul öğrencileri arasında da etkili olduğunu belirtmişler ayrıca fişsiz etkinliklere katılan öğrencilerin geleneksel öğrenme yöntemine maruz kalan öğrencilere kıyasla hesaplamalı düşünme becerilerinde de daha yüksek düzeyde gelişme gösterdiğini belirtmişlerdir. Threekunprap ve Yasri (2020), fişsiz kodlama etkinliği çalışmaları sonucunda fişsiz etkinliklerin istatistiksel olarak gelişmiş öğrenme başarısı ve öz yeterlilik sergilediklerini belirtmişlerdir. Tonbuluğlu ve Tonbuluğlu (2019), bilgisayarsız kodlama etkinliklerinin öğrencilerin hesaplamalı düşünme becerileri üzerinde olumlu katkısı olduğunu belirtmişlerdir. Dolayısıyla bilgisayarlı ve bilgisayarsız kodlama etkinliklerinin programlamaya dönük başarıyı ve programlama öz yeterliliklerini artırdığını söyleyebiliriz. Bilişim teknoloji araç gereçlerine ulaşımın zor olduğu anlarda bilgisayarsız kodlama etkinliklerinin kullanılması programlama başarılarına ve programlama öz yeterliliklerine pozitif yansıyacaktır.

Sonuç olarak derslerde programlama öğretiminde bilgisayarlı ve bilgisayarsız etkinliklerin kullanılması öğrencilerin akademik başarılarına pozitif yönde katkı sağlamaktadır. Akademik başarılarındaki artış gruplar açısından incelendiğinde bilgisayarlı ve bilgisayarsız programlama öğretiminde gruplar arasında akademik başarı üzerinde anlamlı farklılık bulunmamıştır. Derslerde bilgisayarlı ve bilgisayarsız etkinliklerin kullanılması programlama öz yeterlilikleri yönünden incelendiğinde ise öğrencilerin programlama öz yeterliliğini artırdığını söyleyebiliriz. Programlama öz yeterliliklerindeki artış gruplar açısından incelendiğinde deney grubu lehine pozitif yönde anlamlı farklılık olduğu söylenilebilir.

5.2. Öneriler

Çalışmanın bu kısmında ileride yapılacak araştırmalara yönelik öneriler yer almaktadır.

1. Bilgisayarlı ve bilgisayarsız programlama etkinliklerinin akademik başarıya etkisi göz önüne alındığında programlama öğretiminde kullanılan diğer uygulamaların araştırılıp derslerde kullanılabileceği önerilmektedir.
2. Araştırma 5. Sınıf öğrencileriyle yapılmış olup diğer sınıf düzeylerinde de çalışmaların yapılabileceği önerilmektedir.
3. Nicel araştırma yöntemlerinin kullanıldığı çalışmalar, nitel yöntemlerle de desteklenebilir.

KAYNAKLAR

- Abdunabi, R., Hbaci, I. and Ku, H. Y. (2019). Towards enhancing programming self-efficacy perceptions among undergraduate information systems students. *Journal of Information Technology Education*, 18, 185–206.
- Angeli, C. and Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185.
- Arıstawatı, F. A., Budıyanto, C. and Yuana, R. A. (2018). Adoptinge educational robotics to enhance undergraduate students' self-efficacy levels of computational thinking. *Journal of Turkish Science Education*, 15 (STEM Special Issue), 42-50.
- Arseven, A. (2016). Öz yeterlilik: Bir kavram analizi. *Electronic Turkish Studies*, 11(19), 63-80.
- Arslan, E. (2020). *Bireysel ve grupta çalışmanın ortaokul öğrencilerinin blok tabanlı programlama öz-yeterlik algılarına ve robotik programlama tutumlarına etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü, Sakarya.
- Arslan Namlı, N. (2021). *Blok tabanlı programlama ve bilgisayarsız bilgisayar bilimi öğretim etkinliklerinin 5. sınıf öğrencilerinin bilgi işlemsel düşünme becerileri, öz yeterlilikleri ve akademik başarıları üzerindeki etkisi*, Yayınlanmamış Doktora Tezi, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü, Adana.
- Altun, A. ve Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formunun güvenirlik ve geçerlik çalışması. *Journal of Measurement and Evaluation in Education and Psychology*, 3(2), 297-308.
- Alp, Y. (2019). *Blok tabanlı programlama öğretiminin ortaokul öğrencilerinin problem çözme becerisine ve bilgisayara yönelik tutumuna etkisi*, Yayınlanmamış Yüksek Lisans Tezi, İnönü Üniversitesi Eğitim Bilimleri Enstitüsü, Malatya.
- Aytekin, A., Sönmez Çakır, F., Yücel, Y. B. ve Kulaöz, İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi*, 5(5), 24-41.
- Aydoğdu, Ş. (2020). Blok tabanlı programlama etkinliklerinin öğretmen adaylarının programlamaya ilişkin öz yeterlilik algılarına ve hesaplamalı düşünme becerilerine etkisi. *Eğitim Teknolojisi Kuram ve Uygulama*, 10(1), 303-320. <https://doi.org/10.17943/Etku.649585>.
- Akkaş Baysal, E., Ocak, G. ve Ocak, İ. (2020). Kodlama ve arduino eğitimleri ile ilgili lise öğrencilerinin görüşleri. *Elektronik Sosyal Bilimler Dergisi*, 19(74), 777-796. <https://doi.org/10.17755/esosder.625496>.

- Bakırcı, F. (2019). Blok tabanlı programlama aracının 6. sınıf öğrencilerinin programlama başarısı, algoritma geliştirme öz-yeterlikleri ve güdülenmelerine etkisi, Yayınlanmamış Yüksek Lisans Tezi, Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü, Sakarya.
- Bell, T., Alexander, J., Freeman, I. and Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20-29.
- Benzer, A. İ. and Erümit, A. K. (2017). The analysis of the graduate theses related to programming instruction. *Journal of Instructional Technologies and Teacher Education*, 6 (3), 99-110.
- Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, 6(4), 499-528.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A. and Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. *In Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65-72).
- Busuttil, L. and Formosa, M. (2020). Teaching computing without computers: Unplugged computing as a pedagogical strategy. *Informatics in Education*, 19(4), 569–587.
- Brown, N. C., Mönig, J., Bau, A. and Weintrop, D. (2016, February). Panel: Future directions of block-based programming. *In Proceedings of the 47th ACM technical symposium on computing science education* (pp. 315-316).
- Büyüköztürk, Ş. (2020). *Sosyal bilimler için veri analizi el kitabı* (27. Baskı). Ankara: Pegem Akademi A Yayıncılık.
- Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö.E., Karadeniz, Ş. ve Demirel, F. (2020). *Bilimsel araştırma yöntemleri* (28. Baskı). Ankara: Pegem Akademi.
- Canbeldek, M. (2020). *Erken çocukluk eğitiminde üreten çocuklar kodlama ve robotik eğitim programının etkilerinin incelenmesi*, Yayınlanmamış Doktora Tezi, Pamukkale Üniversitesi Eğitim Bilimleri Enstitüsü, Denizli.
- Clements, D. H. and Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational psychology*, 76(6), 1051-1058.
- Chongo, S., Osman, K. And Nayan, N. A. (2021). Impact of the plugged-in and unplugged chemistry computational thinking modules on achievement in chemistry. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(4), 1-21.
- Çanakçı, Y. E. (2022). *Programlama eğitime başlamada blok temelli programlama ile metin temelli programlamanın karşılaştırılması*, Yayınlanmamış Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Lisansüstü Eğitim Enstitüsü, Samsun.

- Çavdar, L. (2018). *Kodlama öğretiminde kullanılan çevrimiçi platformların değerlendirilmesi: Code.org örneği*, Yayınlanmamış Yüksek Lisans Tezi, Tokat Gaziosmanpaşa Üniversitesi Eğitim Bilimleri Enstitüsü, Tokat.
- Çatlak, Ş., Tekdal, M. ve Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies and Teacher Education*, 4(3), 13-25.
- Çilengir, M. D. (2019). *Oyunlaştırma yaklaşımı ile blok tabanlı programlama öğretiminin başarı ve motivasyona etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Çanakkale 18 Mart Üniversitesi Eğitim Bilimleri Enstitüsü, Çanakkale.
- Çilengir, M. D. ve İzmirli, S. (2023). Blok tabanlı programlama öğretiminde oyunlaştırma yaklaşımı kullanımının başarı ve motivasyona etkisi. *International Journal of Computers in Education*, 6(2), 79-103.
- Deniz, T. ve Korucu, A. T. (2023). Farklı görsel programlarla tasarlanan kodlama eğitiminin ortaokul öğrencilerinin akademik başarılarına, tutum ve öz-yeterliklerine etkisi. *Öğretmen Eğitimi ve Hayat Boyu Öğrenme Dergisi*, 5(1), 307-323.
- Du, J., Wimmer, H. and Rada, R. (2016). "Hour of code": Can it change students' attitudes toward programming? *Journal of Information Technology Education: Innovations in Practice*, 15, 53-73.
- Durak, H. Y., Yilmaz, F. G. K. and Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, 10(2), 173-197.
- Emerson, A., Rodríguez, F. J., Mott, B., Smith, A., Min, W., Boyer, K. E. and Lester, J. (2019). Predicting early and often: predictive student modeling for block-based programming environments. *International Educational Data Mining Society* (pp. 39–48).
- Galpin, V., Sanders, I., Turner, H. and Venter, B. (2003, April). Gender and educational background and their effect on computer self-efficacy and perceptions. *School of Computer Science, University of the Witwatersrand*.
- Gültepe, A. (2018). Kodlama öğretimi yapan bilişim teknolojileri öğretmenleri gözüyle öğrenciler kodluyor. *Uluslararası Liderlik Eğitimi Dergisi*, 2(2), 50-60.
- Gümüş, M. M., Kukul, V. ve Korkmaz, Ö. (2024). Relationships between middle school students' digital literacy skills, computer programming self-efficacy, and computational thinking self-efficacy. *Informatics in Education*, 23(3), 571-592.
- Gülbahar, Y., Kert, S. B. ve Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 10(1), 1-29.

- Gülbahar, Y. ve Karal, H. (2018). *Kuramdan uygulamaya programlama öğretimi* (1. Baskı). Ankara: Pegem Akademi.
- Gomes, A. ve Mendes, A. J. (2007, June). An environment to improve programming education. *In Proceedings of the 2007 international conference on Computer systems and technologies* (pp. 1-6).
- Glushkova, T. (2016). Application of block programming and game-based learning to enhance interest in computer science. *Journal of innovations and sustainability*, 2(1), 21-32.
- Hakkoz, A. (2023). *Ortaokul öğrencilerinin blok temelli programlamaya ilişkin öz-yeterlik algısı ve eleştirel düşünme eğilimleri arasındaki ilişkinin belirlenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü, Tokat.
- Hu, C. (2011, June). Computational thinking: What it might mean and what we might do about it. *In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 223-227).
- Igbaria, M. and Iivari, J. (1995). The effects of self-efficacy on computer usage. *Omega*, 23(6), 587-605.
- İnternet 1: BTE Derneği (2013). Türkiye’de ilk ve ortaokullarda (ilköğretim) okutulan bilişim teknolojileri derslerinin tarihi. URL: https://bte.org.tr/wp-content/uploads/2020/12/btderslerinin_tarihi_ilk-ortaokul_ilkogretim.pdf. Son Erişim Tarihi: 08.10.2024.
- İnternet 2: Code.org (2024) .Code.org içerik bilgileri. URL: <https://code.org/about>. Son Erişim Tarihi: 08.10.2024.
- İnternet 3: Milli Eğitim Bakanlığı (2017). Eba’da düşün, tasarla, kodla. URL: <http://www.meb.gov.tr/eba8217da-8220dusun-tasarlakodla8221/haber/14309/tr#%20sayfas%C4%B1ndan%2003.01.%202018%20tarihinde%20eri%C5%9Ffilmi%C5%9Ftir>. Son Erişim Tarihi: 29.11.2022.
- İnternet 4: Milli Eğitim Bakanlığı (1998). İlköğretim okulları seçmeli bilgisayar dersi (1-2-3-4-5) öğretim programı. URL: 1998 (meb.gov.tr). Son Erişim Tarihi: 08.10.2024.
- İnternet 5: Milli Eğitim Bakanlığı (2006). İlköğretim seçmeli bilgisayar dersi (1-8.sınıflar) öğretim programı. URL: [2006 \(meb.gov.tr\)](http://2006.meb.gov.tr). Son Erişim Tarihi: 08.10.2024.
- İnternet 6: Milli Eğitim Bakanlığı (2018). Bilişim teknolojileri ve yazılım dersi. URL: [2731_Agustos_2018.pdf \(meb.gov.tr\)](http://2731_Agustos_2018.pdf). Son Erişim Tarihi: 08.10.2024.
- İnternet 7: TTKB (2023). Ortaöğretim kurumları haftalık ders çizelgeleri. URL: <https://ttkb.meb.gov.tr/www/haftalik-ders-cizelgeleri/kategori/7>. Son Erişim Tarihi: 10.10.2024.
- Jagušt, T., Krzic, A. S., Gledec, G., Grgić, M. and Bojic, I. (2018, October). Exploring different unplugged game-like activities for teaching computational thinking. *In 2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.

- Karaduman, Ü. (2019). *Code.org platformunun 6. sınıf öğrencilerinin programlama öğrenimine etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü, İzmir.
- Kasalak, İ. and Altun, A. (2020). Effects of robotic coding activities on the effectiveness of secondary school students' self-efficacy for coding. *Ilkogretim Online*, 19(4), 2171-2182.
- Krалеva, R., Krалеv, V. and Kostadinova, D. (2019). A methodology for the analysis of block-based programming languages appropriate for children. *Journal of Computing Science and Engineering*, 13(1), 1-10.
- Klopfenstein, L., Fedosyeyev, A. and Bogliolo, A. (2017). Bringing an unplugged coding card game to augmented reality. In *INTED2017 Proceedings* (pp. 9800-9805).
- Kinaneva, D., Hristov, G., Zahariev, P. and Raychev, J. (2019). Play to learn: Using drone-aircrafts and block-based programming for improving learning success rates. *Proceedings Of University Of Ruse*, 58, 45-49.
- Kovari, A. and Katona, J. (2023). Effect of software development course on programming self-efficacy. *Education and Information Technologies*, 28(9), 10937-10963.
- Kukul, V., Gökçe Arslan, Ş. ve Günbatar, M. S. (2017). Ortaokul öğrencileri için programlama öz yeterlik ölçeği: geliştirme, geçerlik ve güvenilirlik. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(1), 158-179.
- Kuleli, S. ve Kışla, T. (2024). An examination of the self-efficacy perceptions of eighth-grade students regarding computational thinking competencies. *Turkish Online Journal of Educational Technology-TOJET*, 23(4), 21-38.
- Küçükçkara, M. F. ve Aksüt, P. (2021). Okul öncesi dönemde bilgisayarsız kodlama eğitimine bir örnek: Problem çözme becerileri için etkinlik temelli algoritma. *Journal of Inquiry Based Activities*, 11(2), 81-91.
- Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M. and Rusk, N. (2008, March). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education* (pp. 367-371).
- Ma, H., Zhao, M., Wang, H., Wan, X., Cavanaugh, T. W. and Liu, J. (2021). Promoting pupils' computational thinking skills and self-efficacy: A problem-solving instructional approach. *Educational Technology Research and Development*, 69(3), 1599-1616.
- Mazman, S. G. ve Altun, A. (2014). Programlama-I dersinin BÖTE Bölümü öğrencilerinin programlamaya ilişkin öz yeterlilik algıları üzerine etkisi. *Journal of Instructional Technologies and Teacher Education*, 2(3), 24-29.
- Mercan, Z. (2022). Erken çocukluk döneminde 21. yüzyıl becerileri. *Muallim Rifat Eğitim Fakültesi Dergisi*, 4(2), 87-105.

- Moors, L., Luxton-Reilly, A. and Denny, P. (2018, April). Transitioning from block-based to text-based programming languages. In *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, (pp. 57-64). IEEE.
- Moon, H., Cheon, J. and Kwon, K. (2022). Difficult concepts and practices of computational thinking using block-based programming. *International Journal of Computer Science Education in Schools*, 5(3), 1-15.
- Mumcu, F., Kidiman, E. and Özdiñ, F. (2023). Integrating computational thinking into mathematics education through an unplugged computer science activity. *Journal of Pedagogical Research*, 7(2), 72-92.
- Munasinghe, B., Bell, T. and Robins, A. (2023). Unplugged activities as a catalyst when teaching introductory programming. *Journal of Pedagogical Research*, 7(2), 56-71.
- Namlı, N. A. and Aybek, B. (2022). An investigation of the effect of block-based programming and unplugged coding activities on fifth graders' computational thinking skills, self-efficacy and academic performance. *Contemporary Educational Technology*, 14(1): 1–16.
- Nouri, J., Zhang, L., Mannila, L. and Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1-17.
- Perin, A. P. J., dos S Silva, D. E. and Valentim, N. (2023). Investigating block programming tools in high school to support Education 4.0: A Systematic Mapping Study. *Informatics in Education*, 22(3), 463-498.
- Piedade, J. and Dorotea, N. (2023). Effects of Scratch-based activities on 4th-grade students' computational thinking skills. *Informatics in Education*, 22(3), 499-523.
- Polat, E. (2020). *Ortaokulda temel programlama öğretiminde kullanılan bilgisayarsız ve bilgisayarlı etkinliklerin başarıya ve bilgisayarca düşünmeye etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü, Erzurum.
- Ramalingam, V., LaBelle, D. and Wiedenbeck, S. (2004, June). Self-efficacy and mental models in learning to program. In *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 171-175).
- Rodriguez, B., Rader, C. and Camp, T. (2016, July). Using student performance to assess CS unplugged activities in a classroom environment. In *proceedings of the 2016 ACM conference on innovation and technology in computer science education* (pp. 95-100).
- Robins, A., Rountree, J. and Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.
- Saygıner, Ş. and Tüzün, H. (2023). The effects of block-based visual and text-based programming training on students' achievement, logical thinking skills, and motivation. *Journal of Computer Assisted Learning*, 39(2), 644-658.

- Singla, A. and Theodoropoulos, N. (2022). From {Solution Synthesis} to {Student Attempt Synthesis} for block-based visual programming tasks. *ArXiv preprint arXiv:2205.01265*.
- Strawhacker, A. and Bers, M. U. (2019). What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, 67(3), 541-575.
- Shute, V. J., Sun, C. and Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22, 142-158.
- Soykan, F. and Kanbul, S. (2018). Analysing K12 students' self-efficacy regarding coding education. *TEM Journal*, 7(1), 182-187.
- Şahin Çakır, Ç., Çakır, M., Damar, A. ve Erdemir Yılmaz, D. (2021). Özel yetenekli öğrencilerin kodlama hakkındaki görüşlerinin belirlenmesi. *Eskişehir Osmangazi Üniversitesi Türk Dünyası Uygulama ve Araştırma Merkezi Eğitim Dergisi*, 6(2), 151-179.
- Taş, N., İslam Bolat, Y. İ. ve Savaş Başkara, Ö. (2023). Blok tabanlı oyunlaştırılmış öğretimin bilgi-işlemsel düşünme ve kodlamaya yönelik tutuma etkisi. *Ekev Akademi Dergisi* (95), 238-255. <https://doi.org/10.17753/sosekev.1268523>.
- Terzi, Y. (2014). SPSS 20.0. Statistical Packages for the Social Sciences.
- Threekunprapa, A. and Yasri, P. (2020). Patterns of computational thinking development while solving unplugged coding activities coupled with the 3s approach for self-directed. *Learning European Journal of Educational Research*, 9(3), 1025-1045.
- Threekunprapa, A. and Yasri, P. (2020). Unplugged coding using flowblocks for promoting computational thinking and programming among secondary school students. *International Journal of Instruction*, 13(3), 207-222.
- Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U. and Ninaus, M. (2017, October). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. *Academic Conferences International Limited*. In *European conference on games based learning* (pp. 687-695).
- Tonbuloğlu, B. and Tonbuloğlu, I. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students. *Informatics in Education*, 18(2), 403-426.
- Tiryaki, A. (2020). *Robotik kodlama eğitiminin ortaöğretim öğrencilerinin programlama öz yeterlik düzeylerine ve yaratıcı düşünme becerilerine etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Hatay Mustafa Kemal Üniversitesi Fen Bilimleri Enstitüsü, Hatay.

- Türker, P. M. and Pala, F. K. (2020). The effect of algorithm education on students' computer programming self-efficacy perceptions and computational thinking skills. *International Journal of Computer Science Education in Schools*, 3(3), 19-32.
- İnternet: TTKB (2018). Bilişim teknolojileri ve yazılım dersi öğretim programı (ortaokul 5 ve 6. sınıflar). URL: <http://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=374>. Son Erişim Tarihi: 10.10.2024.
- Ünsal, İ. (2020). *Blok tabanlı programlama etkinliklerinin ilkökul 2.sınıf öğrencilerinin bilgi işlemsel düşünme becerisi ve etkinlik algısı üzerine etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü, Sakarya.
- Weintrop, D. and Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1-25.
- Weintrop, D. (2021). The role of block-based programming in computer science education. *Understanding computing education*, 1, 71-78.
- Wei-Ying, L. and Tzu-Chuen, L. (2024). Effect of an owlspace programming course on the computational thinking of elementary school students. *Informatics in Education*, 23(1), 145-178.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- Wong, G. K. W. and Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 28(4), 438-450.
- Varçın, E. (2022). *6. Sınıf öğrencilerinin blok tabanlı programlamaya yönelik tutumlarının ve üst bilişsel farkındalıklarının incelenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.
- Vihavainen, A., Airaksinen, J. and Watson, C. (2014, July). A systematic review of approaches for teaching introductory programming and their influence on success. In *Proceedings of the tenth annual conference on International computing education research* (pp. 19-26).
- Yağcı, M. (2016). Effect of attitudes of information technologies (IT) preservice teachers and computer programming (CP) students toward programming on their perception regarding their self-sufficiency for programming. *Journal of Human Sciences*, 13(1), 1418-1432.
- Yaman, E. ve Hamedoğlu, M. A. (2014). Bilgisayarlı öğretim. *Sakarya Üniversitesi Eğitim Fakültesi Dergisi*, (3), 165-173.
- Yang, D. and Kopcha, T. J. (2022). Designing a board game for beginning block-based programmers. *International Journal of Designs for Learning*, 13(1), 35-45.

- Yılmaz, T. (2023). *Bilgisayarsız ve bilgisayarlı kodlama etkinliklerinin ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerine etkisi*, Yayınlanmamış Yüksek Lisans Tezi, Çanakkale Onsekiz Mart Üniversitesi Lisansüstü Eğitim Enstitüsü, Çanakkale.
- Yılmaz, T. and İzmirli, S. (2023). Effect of unplugged and plugged coding activities on secondary school students' computational thinking skills. *Journal of Educational Technology and Online Learning*, 6(4), 1180-1193.
- Yıldırım, E. (2016). Dijital oyun tasarım programlarının eğitimde önemi. *Mesleki Bilimler Dergisi (MBD)*, 5(2), 12-19.
- Yıldız, T. and Seferoğlu, S. S. (2021). The effect of robotic programming on coding attitude and computational thinking skills toward self-efficacy perception. *Journal of Learning and Teaching in Digital Age*, 6(2), 101-116.
- Yiğit, M. F. (2016). *Görsel programlama ortamı ile öğretimin öğrencilerin bilgisayar programlamayı öğrenmesine ve programlamaya karşı tutumlarına etkisinin incelenmesi*, Yayınlanmamış Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Eğitim Bilimleri Enstitüsü, Samsun.
- Yukselturk, E. and Altıok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789-801.
- Yüksel, M. A. (2022). *Ortaokul öğrencilerinin çoklu zeka alanları ve blok tabanlı programlamaya yönelik öz- yeterlilik algılarının incelenmesi*. Yayınlanmamış Yüksek Lisans Tezi, Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Zurnacı, B. (2023). *Robotik ve bilgisayarsız kodlamanın okul öncesi çocuklarının bilgi işlemsel düşünme ve yürütücü işlevlerine etkilerinin karşılaştırılması*, Yayınlanmamış Yüksek Lisans Tezi, Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü, Erzurum.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı-Soyadı: Gamze ORTA

EĞİTİM DURUMU

Lisans Öğrenimi: Kahramanmaraş Sütçü İmam Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi

Yüksek Lisans: Amasya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi

İŞ DENEYİMİ

Çalıştığı Kurumlar ve Yıl: MEB 2016 -