

T.C.
MARMARA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
MATEMATİK VE FEN BİLİMLERİ ANABİLİM DALI
MATEMATİK ÖĞRETMENLİĞİ BİLİM DALI

AKADEMİK BAŞARISI YÜKSEK 6. SINIF ÖĞRENCİLERİN
SCRATCH PROGRAMI İLE TAMSAYILAR KONUSUNDA
ALGORİTMA ÜRETME SÜREÇLERİ VE YAPILARININ
İNCELENMESİ

Emre DURASI
(Yüksek Lisans Tezi)

İstanbul, 2021

T.C.
MARMARA ÜNİVERSİTESİ
EĞİTİM BİLİMLERİ ENSTİTÜSÜ
MATEMATİK VE FEN BİLİMLERİ ANABİLİM DALI
MATEMATİK ÖĞRETMENLİĞİ BİLİM DALI

AKADEMİK BAŞARISI YÜKSEK 6. SINIF ÖĞRENCİLERİN
SCRATCH PROGRAMI İLE TAMSAYILAR KONUSUNDA
ALGORİTMA ÜRETME SÜREÇLERİ VE YAPILARININ
İNCELENMESİ

INVESTIGATION OF 6TH GRADE HIGH ACHIEVING STUDENTS'
ALGORITHM PRODUCTION PROCESSES AND STRUCTURES ON
INTEGERS WITH SCRATCH PROGRAM

Emre DURASI
(Yüksek Lisans Tezi)

DANIŞMAN
Dr. Öğr. Üyesi ALAATTİN PUSMAZ

İstanbul, 2021

**Tüm kullanım hakları
M.Ü. Eğitim Bilimleri Enstitüsü'ne aittir.**

© 2021

ETİK BEYANI

Marmara Üniversitesi Eğitim Bilimleri Enstitüsü Müdürlüğü Lisansüstü Tez Yazım Kılavuzuna uygun olarak hazırladığım çalışmamda;

- Sunduğum bilgileri, dokümanları ve verileri akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Çalışmamda yararlandığım eserlerin tamamına atıfta bulunarak kaynak gösterdiğimi,
- Elde ettiğim verilerde ve sonuçlarda herhangi bir değişiklik yapmadığımı bildirir, aksi bir durumda aleyhimde doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

17/12/2021

Emre DURASI

ÖZGEÇMİŞ

2017- ... : Marmara Üniversitesi Eğitim Bilimleri Enstitüsü Matematik Eğitimi

Ana Bilim Dalı Yüksek Lisans Programına Giriş

2016- ... : MEB İstanbul Beyhan Şenyuva İmam Hatip Ortaokulu Matematik Öğretmenliği

2013–2016: MEB Bursa Gündoğdu Ortaokulu Matematik Öğretmenliği

2009–2013: Sakarya Üniversitesi İlköğretim Matematik Öğretmenliği

2005–2009: Şükrü Şankaya Anadolu Lisesi



ÖNSÖZ

Bu araştırmada akademik başarısı yüksek ortaokul öğrencilerinin Scratch programı ile algoritma üretme süreçlerinin incelenmesi amaçlanmaktadır.

Bu süreçte tecrübelerini, emeğini ve tavsiyelerini benimle paylaşan değerli tez danışmanım sayın Dr. Öğr. Üyesi Alaattin PUSMAZ hocama, tez savunma jürimde yer alarak katkı sağlayan değerli Doç. Dr. Eyüp SEVİMLİ ve Dr. Öğr. Üyesi Oğuz KÖKLÜ hocalarıma teşekkür ediyorum.

Yüksek lisans eğitimim boyunca şahsıma büyük katkılar sunan ve tez aşamasında değerli bilgilerini benimle paylaşan Marmara Üniversitesi Atatürk Eğitim Fakültesi Matematik Öğretmenliği Ana Bilim Dalı'nda görev yapan tüm hocalarıma şükranlarımı sunuyorum.

Bu günlere gelmemde büyük pay sahibi olan annem Aynur DURASI'ya, babam Kemal DURASI'ya ve ablam Nurdan DURASI'ya teşekkürlerimi sunuyorum. Ayrıca bu süreçte maddi ve manevi her türlü desteğini benden esirgemeyen sevgili eşim Büşra DURASI'ya en içten sevgilerimi sunuyorum.

Emre DURASI

İstanbul, 2021

ÖZET

Algoritma, belirli bir problemi çözmek veya belirli bir amaca ulaşmak için çözüm yolunun sistematik, sıralı ve açıklayıcı olarak tasarlanmasıdır. Algoritma kavramı günümüzde sıklıkla programlama ya da yazılım geliştirme alanlarında kullanılmasıyla birlikte günlük hayattaki aktiviteler ve matematik gibi derslerde karşılaşılan problemlerin daha kolay çözülmesine de olanak sağlamaktadır. Matematik açısından algoritma, daha önceden bilinen bir formülün uygulanması yerine, eldeki bir problemin tüm yönleriyle ele alınıp özgün bir çözüm yolu oluşturma sürecidir. Bu araştırmanın amacı, 6. sınıf ortaokul öğrencilerinin Scratch programı ile tam sayılar konusunda algoritma üretme süreçlerini ve oluşturdıkları algoritma yapılarını incelemektir.

Araştırmada nitel araştırma yöntemi çeşitlerinden durum çalışması kullanılmıştır. Araştırmanın çalışma grubunu, 2020-2021 eğitim öğretim yılında İstanbul ilinin Sancaktepe ilçesine bağlı bir özel ortaokulunun 6. sınıfında öğrenim görmekte olan akademik başarısı yüksek üç öğrenci oluşturmaktadır. Araştırma, tek bir grup ile üç farklı etkinlik üzerinden yürütülmüştür. Araştırmanın verileri, öğrenci Scratch dosyaları ve öğrencilerle yapılan sesli-düşünme protokol görüşmeleri yoluyla toplanmıştır. Elde edilen algoritma üretme süreçleri “RBC Soyutlama Modeli” ve Moala (2019) tarafından ortaya konulan “Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması” çerçevesi kullanılarak analiz edilmiştir. Oluşturulan algoritmaların yapısı ise Knuth (2014) tarafından iyi bir algoritmada bulunması gereken beş özellik kapsamında geliştirilen “Algoritma Yapısı Değerlendirme Rubriği” ve revize edilmiş “Dr. Scratch Programı” değerlendirme rubriğine göre analiz edilmiştir.

Araştırma sonucunda öğrencilerin üç etkinlik için algoritma üretme süreçlerinde tanıma, kullanma, oluşturma eylemlerini dinamik olarak kullandıkları görülmüştür. Öğrencilerin algoritma üretme süreçlerinde kullandıkları argümanlar incelendiğinde açıkça belirtilen ve doğru olan özelliklerin daha fazla kullanıldığı görülmüştür. Öğrencilerin oluşturdıkları algoritma yapısı, rutin olan problemlerden oluşan iki etkinlik için doğru ve etkin bir yapıda olduğu, rutin olmayan problemde oluşan etkinlikte ise doğru ve etkin bir yapıda oluşmadığı gözlemlenmiştir. Ortaokul seviyesindeki öğrencilerin rutin olmayan problemlere ait algoritma üretme süreçlerine odaklanan araştırmalar yapmak öğrencilerin bu süreçte yaşadıkları zorlukları ve Scratch programının bunları aşmaya nasıl katkı sağlayabileceğini anlamaya yardımcı olacaktır.

Anahtar Kelimeler: Algoritma, Algoritma Üretme Süreci, Scratch Programı, Tam Sayılar

ABSTRACT

An algorithm is a systematic, sequential and explanatory design of a solution path to solve a particular problem or achieve a particular goal. The concept of algorithm is frequently used in programming or software development areas today, it also enables the problems encountered in daily life activities and lessons such as mathematics to be solved more easily. In terms of mathematics, an algorithm is the process of dealing with all aspects of a problem at hand and creating a unique solution rather than applying a previously known formula. The aim of this research is to examine the process of producing algorithms on integers with the Scratch program of 6th grade secondary school students and the algorithm structures they created.

Case study, one of the qualitative research methods, was used in the research. The study group of the research consists of three students with high academic success who are studying in the 6th grade of a private secondary school in the Sancaktepe district of Istanbul in the 2020-2021 academic year. The research was carried out with a single group on three different activities. The data of the research were collected through student Scratch files and think-aloud protocol interviews with students. The algorithm generation processes obtained were analyzed using the "RBC Abstraction Model" and the "Classification of the Arguments Used by Students in the Algorithm Generation Process" framework introduced by Moala (2019). The structure of the created algorithms is the "Algorithm Structure Evaluation Rubric" developed by Knuth (2014) within the scope of the five features that a good algorithm should have, and the revised "Dr. Scratch Program" was analyzed according to the evaluation rubric.

As a result of the research, it was seen that the students used the actions of recognition, use, and creation dynamically in the algorithm production processes for the three activities. When the arguments used by the students in the algorithm production processes were examined, it was seen that the clearly stated and correct features were used more. It was observed that the algorithm structure created by the students was correct and effective for two activities consisting of routine problems, but not in a correct and effective structure in the activity consisting of non-routine problems. Conducting research focusing on the process of producing algorithms for non-routine problems of secondary school students will help to understand the difficulties students experience in this process and how the Scratch program can contribute to overcoming them.

Keywords: Algorithm, Algorithm Generation Process, Scratch Program, Integers.

İÇİNDEKİLER

ETİK BEYANI.....	i
ÖZGEÇMİŞ.....	ii
ÖNSÖZ.....	iii
ÖZET.....	iv
ABSTRACT.....	vi
İÇİNDEKİLER.....	viii
TABLolar LİSTESİ.....	xi
ŞEKİLLER LİSTESİ.....	xii
KISALTMALAR.....	xv
BÖLÜM I: GİRİŞ	1
1.1. Problem Durumu	1
1.2. Araştırmanın Amacı	5
1.3. Araştırmanın Önemi	6
1.4. Araştırmanın Sınırlılıkları.....	7
1.5. Araştırmanın Varsayımları.....	7
1.6. Araştırmanın Tanımları.....	7
BÖLÜM II: İLGİLİ ÇALIŞMALAR.....	9
2.1. 21. Yüzyıl Eğitim Dünyası	9
2.2. 21. Yüzyıl Eğitim Dünyasında Matematik Eğitimi	12
2.2.1. Matematik Eğitiminde Soyutlama Süreci ve RBC Soyutlama Modeli.....	14
2.2.2. Pozitif Tam Sayılar	16
2.2.3. Algoritma.....	18
2.2.3.1. Algoritmik Düşünme.....	24
2.3. Matematiğin Teknoloji ile İlişkisi.....	31
2.3.1. Bilgisayar Destekli Matematik Öğretimi.....	32
2.3.2. Scratch Programı.....	33
2.3.3. Scratch Programının Matematiksel Yönü.....	38
2.3.4. Scratch Programı ile İlgili Çalışmalar.....	39
BÖLÜM III: YÖNTEM.....	43

3.1. Araştırmanın Deseni	43
3.2. Araştırmacının Rolü.....	44
3.3. Çalışma Grubu.....	44
3.4. Veri Toplama Araçları.....	45
3.4.1. Sesli Düşünme Protokolü.....	45
3.4.2. Scratch Programı Etkinlikleri.....	46
3.5. Verilerin Toplanma Süreci.....	48
3.6. Verilerin Analizi.....	52
3.6.1. Algoritma Süreci ile İlgili Değerlendirme Rubriği.....	53
3.6.1.1.RBC Soyutlama Modeli.....	53
3.6.1.2.Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması.....	55
3.6.2. Algoritma Yapısı ile Yapılan Değerlendirme Rubriği...57	
3.6.2.1.Algoritma Yapısı Değerlendirme Rubriği.....	57
3.6.2.2. Dr. Scratch Programı.....	58
3.7. Geçerlik ve Güvenirlik.....	61

BÖLÜM IV: BULGULAR..... 63

4.1. Pozitif Bölenleri Listeleme Etkinliğine Ait Bulgular.....	63
4.1.1. Algoritma Üretme Sürecine Ait Bulgular.....	63
4.1.2. Algoritma Yapısına Ait Bulgular.....	73
4.2. Ardışık Doğal Sayıların Toplamı Etkinliğine Ait Bulgular.....	75
4.2.1. Algoritma Üretme Sürecine Ait Bulgular	76
4.2.2. Algoritma Yapısına Ait Bulgular	80
4.3. Oyun Etkinliğine Ait Bulgular	82
4.3.1. Algoritma Üretme Sürecine Ait Bulgular.....	82
4.3.2. Algoritma Yapısına Ait Bulgular.....	87
4.4. Etkinliklere Ait Genel Değerlendirme.....	88
4.4.1. Etkinliklere Ait Algoritma Üretme Süreci ile İlgili Genel Değerlendirme.....	88
4.4.2. Etkinliklere Ait Algoritma Yapıları ile İlgili Genel Değerlendirme.....	91

BÖLÜM V: SONUÇ, TARTIŞMA VE ÖNERİLER.....93

5.1.Algoritma Üretme Süreci ile İlgili Sonuçlar.....	93
5.1.1. Pozitif Bölenleri Listeleme Etkinliğinin Algoritma Üretme Sürecine Ait Sonuçları.....	93
5.1.2. Ardışık Sayıların Toplamı Etkinliğinin Algoritma Üretme Sürecine Ait Sonuçları.....	95

5.1.3. Oyun Etkinliğinin Algoritma Üretme Sürecine Ait Sonuçları.....	95
5.2.Algoritma Yapısı ile İlgili Sonuçları.....	96
5.2.1. Pozitif Bölenleri Listeleme Etkinliğinin Algoritma Yapısına Ait Sonuçları.....	97
5.2.2. Ardışık Sayıların Etkinliğinin Algoritma Yapısına Ait Sonuçları.....	98
5.2.3. Oyun Etkinliğinin Algoritma Yapısına Ait Sonuçları.....	99
5.3.Tartışma	100
5.4.Öneriler	102
KAYNAKÇA	104
EKLER.....	118
EK-1 Merkezi Sınavlarda Çıkmış Algoritma Soru Örnekleri.....	118
EK-2 Modüler Aritmetik Sınavı.....	120
EK-3 Milli Eğitim Bakanlığı Araştırma İzin Yazısı.....	121
EK-4 Veli İzin Formu	122
EK-5 Katılımcı Formu.....	123

TABLÖLAR LİSTESİ

Tablo 2.1.	Lagrange ve Laval (2019), Algoritmik Çalışma Alanı ve Matematiksel Çalışma Alanı İlişkisi.....	27
Tablo 3.1.	Öğrencilerin Her Etkinlik İçin Harcadıkları Süre	52
Tablo 3.2.	RBC Soyutlama Modeli.....	54
Tablo 3.3.	RBC Soyutlama Modeli Göre Tam Sayılarla İlgili Araştırmada Geçen Örnek Eylemler.....	54
Tablo 3.4.	Algoritma Üretme Sürecinin Çerçevesi (Moala, 2019).....	55
Tablo 3.5.	Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması.....	56
Tablo 3.6.	Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılmasına Örnekler.....	56
Tablo 3.7.	Knuth (2014)'a Göre Algoritmaların Beş Temel Özelliği	57
Tablo 3.8.	Algoritma Yapısı Değerlendirme Rubriği.....	58
Tablo 3.9.	Revize Edilmiş Dr. Scratch Programına Göre Değerlendirme Rubriği.....	60
Tablo 4.1.	Algoritma Yapısı Değerlendirme Rubriğine Göre Pozitif Bölenleri Listeleme Etkinlik Puanı.....	74
Tablo 4.2.	Algoritma Yapısı Değerlendirme Rubriğine Göre Ardışık Sayıların Toplamı Etkinlik Puanı.....	81
Tablo 4.3.	Algoritma Yapısı Değerlendirme Rubriğine Göre Oyun Etkinlik Puanı.....	87

ŞEKİLLER LİSTESİ

Şekil 2.1.	Günlük Yaşam Algoritma Örneği.....	19
Şekil 2.2.	C Programlama Dilinde Yazılmış En Büyük Elemanı Bulma Algoritması (Gündüz, 2016).....	20
Şekil 2.3.	Euclid Algoritmasının Akış Şeması ile Gösterimi.....	22
Şekil 2.4.	Millî Eğitim Bakanlığı'nın Yayımlamış Olduğu Matematik Örnek Sorusu.....	23
Şekil 2.5.	Algoritma Tasarım ve Analiz İşlemi (Levitin,2012).....	28
Şekil 2.6.	Scratch Programı Web Sitesi.....	34
Şekil 2.7.	Scratch 3.19.2 Offline Editor Uygulaması.....	35
Şekil 2.8.	Scratch Programı Kod Seçeneklerinden Bazıları.....	36
Şekil 2.9.	Scratch Programı Örnek Kod Bloğu.....	36
Şekil 2.10.	Hatalı Yapılan Scratch Programı Örnek Kod Bloğu.....	37
Şekil 2.11.	Karakterin Sahnede $(x, y) = (0,0)$ Noktasında Durduğu An.....	38
Şekil 2.12.	Scratch Programı Operatörler Bölümünün Bir Kısımı.....	39
Şekil 3.1.	Kod Blok Kısım Verilmeden Öğrencilere Verilen Scratch Programı Görüntüsü.....	47
Şekil 3.2.	Araştırmacın Üçüncü Etkinlik İçin Oluşturduğu Algoritma.....	48
Şekil 3.3.	Verilerin Toplanma Süreci.....	48
Şekil 3.4.	Öğrencilerin Scratch Programı Üzerinde Tasarladığı Packman Projesi.....	49
Şekil 3.5.	Öğrencilerin Tasarladığı Duvar Oyunu.....	50
Şekil 3.6.	Verilen İki Sayının En Büyük Ortak Bölenini ve En Küçük Ortak Katımı Bulma Çalışması.....	51
Şekil 3.7.	Armstrong Sayıları.....	51
Şekil 3.8.	Veri Analizi Şeması.....	53
Şekil 3.9.	Dr. Scratch Programı Değerlendirme Rubriği	59
Şekil 4.1.	“Sayı Bölü Sayı Büyüktür” Bloğu.....	65
Şekil 4.2.	Mod Bloğu Oluştuktan Sonra Oluşturulan Kod Blokları	66
Şekil 4.3.	“Mod” Bloğu ile “Sayı Bölü Sayı Büyüktür Sayı” Bloğunun Yer Değiştirerek Denenmesi.....	66
Şekil 4.4.	Erdem'in Oluşturduğu Blok.....	67
Şekil 4.5.	Erdem'in Doğru Oluşturduğu Mod Bloğu.....	68

Şekil 4.30.	Etkinliklerde Öğrencilerin Kullandıkları Argümanların Sınıflandırılması.....	90
Şekil 4.31.	Öğrencilerin Ürettikleri Algoritmaların Yapı Bakımından Değerlendirilmesi.....	91
EK Şekil 1.	LGS 2019 Matematik 17. Soru.....	118
EK Şekil 2.	LGS 2020 Matematik 9. Soru.....	118
EK Şekil 3.	LGS 2021 Matematik 6. Soru.....	119
EK Şekil 4.	YKS 2020 Temel Yeterlilik Sınavı Matematik 10.soru.....	119



KISALTMALAR

LGS: Liselere Geçiş Sistemi

MEB: Millî Eğitim Bakanlığı

NCTM: Matematik Öğretmenlerinin Ulusal Konseyi

TÜBİTAK: Türkiye Bilimsel ve Teknik Araştırma Kurumu

vd: ve diğerleri

YKS: Yükseköğretim Kurumları Sınavı



BÖLÜM I: GİRİŞ

Bu bölümde; problem durumu, araştırmanın amacı, araştırmanın önemi, araştırmanın sınırlılıkları, araştırmanın varsayımları ve araştırmanın tanımları üzerinde durulmuştur.

1.1. Problem Durumu

Bilim ve teknolojinin hızla gelişimi, toplumun değişen ihtiyaçları, öğrenme ve öğretme sistemindeki yenilikçi gelişmeler; bireylerden istenen rolleri doğrudan değişime uğratmaya sebep olmuştur. Buna bağlı olarak eğitim sisteminde de değişiklikler yaşanmıştır. 21. yüzyıl eğitim anlayışı; bilgiyi üreten ve farklı çalışma alanları arasında bağlantı kurabilen, dijital teknolojileri aktif kullanabilen, problem çözme becerisine sahip olabilen, elde edilen bilgileri işleyebilen ve kullanabilen bireyler yetiştirmeyi hedeflemektedir (Erdem & Uçak, 2020). Bu kapsamda matematik eğitiminin de değişmesi kaçınılmazdır.

Günümüzde matematik eğitimi; yalnızca matematiği bilen değil, matematik bilgisini üretebilen ve bunu hayatının diğer alanlarında kullanabilen öğrenciler yetiştirmeyi amaçlar (Çiltaş & Yılmaz, 2013). Geleneksel matematik eğitimi anlayışında öğrenciler, bilgiyi dışarıdan hazır alan pasif bir konumdadır. Geleneksel öğretim sürecinde, öğrencilere aktarılan birçok bilgi bir nedene dayandırılmadığı için öğrenciler ezbere dayanan bir öğrenme yoluna başvurmaktadırlar (Kazak, 2012). Günümüz matematik eğitimi anlayışında ise kaliteyi arttırmak için öğrencilerin öğretim sürecinde aktif rol alarak matematiği içselleştirmeleri gerekmektedir. Öğrencilerin verilen bir problemi çözebilmek için gözlem yapma, ilişki arama, soru sorma, muhakeme etme ve sonuç çıkarma süreçlerinde yer almaları gerekmektedir (Akay, 2006). Bu bağlamda matematik öğretiminde en büyük değişim problemin çözümüne yönelik kural ve yönergelerin verilmesi ve bu yönergelerin kullanıldığı alıştırmalar yapılan geleneksel öğretim yaklaşımının yerine kavramsal anlama, akıl yürütme ve problem çözmeye yapılan vurgudur (Stein vd., 2017). Öğrencilerin matematik öğretiminde istenilen hedeflere ulaşabilmesi için problem çözmeyi öğrenmeleri ve bunu yaşantı haline dönüştürmeleri gerekmektedir (Baykul, 2014). Bu kapsamda “algoritma” kavramı son yıllarda matematik eğitiminde gittikçe kullanılmaya başlanan ve günümüzde önemli bir yere sahip olan kavramlardan biridir.

Bir problemi çözmek ve problemi istenen bir sonuca ulaştırmak için tasarlanan yola algoritma denir (Arabacıoğlu vd., 2007). Teknolojik araç ve gereçlerin temelinde algoritma

mantığı yatmaktadır. Bu durum algoritma kavramının daha çok teknoloji ile bağdaştırılmasına sebep olmuştur. Ancak algoritma kavramı ilk olarak 8. yüzyılda yaşamış olan El-Harezmi adlı bir matematik bilgini tarafından kullanılmıştır. El Harezmi'nin "*Cebr ve Mukabele Hesabının Özlü El Kitabı*", birinci ve ikinci dereceden denklemlerin açıklamalı ve sistemli çözüm örneklerini barındırmaktadır. Bu açıklamalı ve sistemli çözüm örnekleri algoritma adını almıştır (Kılan, 2007). Sekizinci yüzyıldan 20. yüzyıla kadar "algoritma" kavramı; problemlerin belli bir sıra ile matematiksel ve mantıksal bir dizi işlem adımı olarak kullanılmıştır. Bilgisayarların ortaya çıkışı da algoritma mantığına dayanmaktadır. Çünkü bilgisayarlar istenen program çıktılarını belli bir sıra ile yaparak ortaya çıkarmaktadır. Teknolojinin -özellikle bilgisayar teknolojisinin- çok hızlı gelişimi, algoritma kavramının bilişim teknolojileri ile özdeşleşmesine sebep olmuştur. Ancak yukarıda belirtilen tarihsel sürece baktığımızda temel olarak algoritma kavramının matematik ile daha yakından ilişkisi olduğu görülmektedir.

Değişen eğitim anlayışı matematik eğitiminde kullanılan "algoritma" kavramının farklılık geçirmesine sebep olmuştur. Algoritmalar, önceki yıllarda problem çözümünde öğrencilerin belli talimatları uygulayarak sonuca ulaşması anlamına geliyordu. Yani geleneksel eğitim anlayışına göre algoritmalar, önceden bulunmuş, ortaya konulmuş çözüm yollarının (algoritmanın) öğrenci tarafından ezberlenmesi anlamına geliyordu. Daha sonra öğrenci, ezberlediği bu çözüm yollarını kullanarak çözüme ulaşıyordu. Günümüzde ise bu anlayış değişmiştir. Son yıllardaki matematik eğitimi araştırmalarında algoritma kavramı problem tarzının değişmesiyle beraber; öğrencinin bir problem karşısında kendisine özgü bir çözüm yolu bulması, bunun doğruluğunu test etmesi ve benzer problemlere uygulayarak genelleştirmesi olarak tanımlanmaktadır. (Thomas, 2014). Bu şekilde öğrencilerinden problem çözümünde hazır yapıları kullanmaktan çok probleme özgü çözümler bulmaları hedeflenmektedir (Nemirovsky & Rasmussen. 2005). Kısacası algoritmalar, günümüz matematik anlayışında öğrenmeyi öğrenme ile ilişkilidir. Algoritmaları anlama, uygulama, değerlendirme ve üretme becerilerinin toplamı "algoritmik düşünme" olarak tanımlanmaktadır (Brown, 2015).

Algoritmik düşünmek, daha verimli, adil ve güvenli çözümler geliştirmek için tümevarım gibi matematiksel kavramları uygulama becerisine sahip olmak anlamına gelmektedir (Aytekin vd., 2018). Bu düşünce, problemin çözümünden daha fazlasını ifade etmektedir (Lockwood vd., 2016). Algoritmik düşünme süreci, bir problemin çözümüne ilişkin geliştirilebilir bir algoritma üretmeyi hedeflemektedir. Üretilen bu algoritmanın doğruluğu mutlaka test edilmelidir. Eğer algoritmanın eksik, fazla veya yanlış tarafları varsa

revize edilmeli ve bu şekilde algoritma son halini alarak problemin çözümünü her durumda garanti etmelidir. (Ferrarello & Mammana, 2018).

Değişen eğitim politikaları, ülkemizde uygulanan merkezi sınavların soru tarzlarında değişiklik yapılmasına yol açmıştır. Özellikle 2018 yılından sonra yapılan merkezi sınavlarda bu değişiklik net bir şekilde görülmektedir. Bu değişikliğin temel sebebi yukarıda bahsettiğimiz günümüz öğrencilerinden beklenen özelliklerin olmasıdır. Artık sorularda matematiksel formüller direkt verilmektedir. Öğrencilerden çoğu zaman soruda verilen formül veya çözüme ulaşması beklenmektedir. Bu merkezi sınavlardaki sorular incelendiğinde matematik dersindeki bazı soruların, öğrencilerin algoritma oluşturmaları gereken tarzda sorular olduğu görülmektedir. Merkezi sınavlarda çıkmış algoritma soru örnekleri ve algoritmik olarak açıklamaları EK-1’de verilmiştir.

Akıl yürütme ile ilgili soruların artması, matematik öğretiminin de bu yönde ilerlemesi gerektiğinin bir göstergesidir. Bu da algoritmik düşüncenin gitgide öneminin artacağına bir işaretidir. Moala (2019) iki farklı öğrenci grubunun oluşturdukları algoritmaları ve algoritma üretme süreçlerini inceleyen bir araştırmasında, iki grubun da verilen problemi doğru çözdüğünü belirtmiştir. Ancak oluşturulan algoritmalar benzer başka problemlere genelleştirmesi istenildiğinde ise bir grubun daha başarılı olduğunu ortaya koyan bir sonuca ulaşmıştır. Araştırmanın sonucunda ise öğrencilerin algoritma çözüm mekanizmalarının daha fazla araştırmaya konu olması gerektiği Moala tarafından vurgulanmıştır. Son yıllardaki matematik ve bilişim teknolojileri araştırmalarına bakıldığında algoritmik düşünmenin öneminin arttığı görülmektedir. (Hart & Sandefur 2018; Weintrop et al. 2016). Bunun en büyük nedenlerinden biri de modern teknoloji dünyasını anlamaktır (Moala, 2019). Yukarıda belirtildiği üzere algoritmalar teknoloji dünyasının temelini oluşturmaktadır. Teknolojik makinelerin (telefon, bilgisayar vb.) oluşum süreçlerine baktığımızda iki başlık altında toplayabiliriz. Bunlardan birincisi donanım, ikincisi yazılımdır. Yazılım kısmının oluşum sürecine bakıldığında matematiksel terimlerin, teoremlerin, algoritmaların kullanıldığı görülmektedir (Stewart, 2012). Bu nedenlerden dolayı teknolojiyi anlamak ve anlamlandırmak için öğrencilerin algoritmik düşünme sistemini anlamaya ve arttırmaya yönelik çalışmalar hız kazanmıştır (Dourish, 2016; Modeste, 2016).

Teknoloji ilk çıktığı andan itibaren eğitim ile sıkı bir ilişki içinde olmuştur. Günümüzde de bu ilişki yaygınlığını devam ettirmekle birlikte, teknolojik makinelerin –özellikle bilgisayar- bir öğrenme aracı olarak ayrı bir potansiyele sahip olduğu eğitimciler tarafından fark edilmeye başlamıştır. Bu potansiyelden ilk defa Papert söz etmiştir. Papert'a göre

bilgisayarların doğru epistemolojik yaklaşımlarla kullanılması, öğrenme süreçlerini ve geleneksel eğitim ortamlarını değiştirecektir (Papert, 1980).

Teknolojinin hızla ilerlemesi ve eğitim-öğretim programlarındaki değişiklikler teknolojinin sınıf ortamında kullanılmasının önünü açmıştır (Polat & Özcan, 2014). Matematik dersinin diğer derslere göre daha soyut kavramlar içermesi ve matematik dersinin öğrenciler tarafından sıkıcı ve zor bir ders olarak algılanması matematik eğitimi araştırmalarını çeşitlendirmiştir (Boaler, 2016; Frenkel, 2013). Bu gibi sebeplerden dolayı teknolojinin matematik eğitiminde kullanılması araştırmalara konu olmaktadır (Gökçek, 2004; Saka & Çelik, 2018).

İlköğretim öğrencileri soyut kavramları anlamakta zorluk çekmektedirler. Bu yüzden ilköğretim öğrencilerine uygun teknolojik araçlar ile dersler işlenirse, öğrencilerin matematik gelişimleri hızlanabilir ve ileri seviyedeki matematiksel kavramları öğrenme motivasyonları artabilir (Köse & Yavuzsoy, 2008). Bu teknolojik araçlardan bir tanesi de Scratch programıdır. Aslında bir kodlama programı olan Scratch programı matematik öğretiminde de kullanılmaktadır. Scratch programının matematik öğretimindeki etkisini saptamak için birçok araştırma yapılmıştır. (Gülbahar & Kalelioğlu, 2014; Shin & Park, 2013; Wang, Huang & Hwang, 2014).

Bu araştırmada, son yıllarda matematik eğitimi araştırmalarına sıkça konu olan ve algoritma kavramı ile yakından ilişkili olan Scratch programı seçilmiştir. Scratch Programı, 2003 ile 2007 yılları arasında MIT (Massachusetts Teknoloji Enstitüsü) Laboratuvarı tarafından geliştirilen bir kodlama programıdır (Çatlak, Tekdal & Baz, 2015). Scratch programını geliştirilenler, programın ara yüzünü öğrencilerin rahatlıkla kullanabileceği bir şekilde oluşturmuştur. Scratch programı üzerindeki projeler, öğrencilerin çok basitten başlayarak gittikçe daha da zor ve karmaşık etkinlikler üretmeleri için fırsatlar veren bir programdır. Ayrıca program blok temelli oluşturularak metinsel programdan ziyade görsel program oluşturmayı hedeflemiştir. Scratch programı ile öğrenciler etkinlikler oluştururken farklı araçları organize ederek ve sorumluluğu üstlenerek kendi kontrollerinde bir interaktif tasarım süreci içerisine girmektedirler (Genç & Karakuş, 2011). Bunun yanı sıra öğrenciler programı 15 dakika gibi kısa bir sürede öğrenebileceklerdir fakat program, proje üretme çeşitliliği ve derinliği bakımından yıllarca öğrencileri meşgul edebilecek kapasiteye sahiptir (Resnick vd. 2009).

Matematik eğitiminde Scratch programı kullanımına odaklanan araştırmalarda, ilköğretim öğrencilerinin birçok becerisinin geliştiği görülmüştür. Bunlar; akıcı olma, yaratıcı düşünme, algoritmik-sistemik düşünme ve problem çözme becerileridir (Kobsiripat,

2014). Problem çözme becerisi matematik öğretiminde önemli becerilerden biridir ve Scratch programı bu beceriyi geliştirmeye etkisi olan bir programdır (Shin & Park, 2014). Scratch programı üzerine yapılan araştırmalarda daha çok matematiksel becerilere odaklanılmıştır. Bunun haricinde algoritmalar ile ilgili olan araştırmalar daha çok bilişim teknolojileri alanı ile ilişkili olduğu görülmektedir (Çatlak vd., 2015).

Teknoloji, matematiğin birçok konusunda kullanılmaktadır. Bunlardan bir tanesi de matematikte önemli bir yere sahip olan tam sayılar konusudur (Altıparmak & Özdoğan, 2010; Hativa & Cohen, 1995). Tam sayılar konusunun alt öğrenme alanı olan pozitif tam sayılar konusu, öğrenciler tarafından negatif tam sayılar göre daha basit olarak algılanmaktadır (Ünüvar & Barut, 2020). Ancak öğrenciler bu konuda basit hatalar yapabilmektedirler. Matematikte basit gibi görünen hatalar, öğrencilerin sonradan öğrenecekleri konularda da hata yapmalarına sebep olmaktadır (Baki, 1988). Bu araştırmada teknolojik imkanlar kullanarak öğrencilerin pozitif tam sayılar konusundaki algoritma yapıları incelenecektir.

1.2. Araştırmanın Amacı

Bu araştırmanın amacı, akademik başarısı yüksek ortaokul öğrencilerinin pozitif tam sayılar konusunda Scratch Programı ile algoritma üretme süreçlerini incelemektir. Bu amaç doğrultusunda şu sorulara cevap aranmıştır;

- 1) Öğrencilerin pozitif tam sayılar konusunda Scratch programı ile algoritma üretme süreçleri nasıldır?
- 2) Öğrencilerin pozitif tam sayılar konusunda Scratch programı ile oluşturdukları algoritmaların yapısı nasıldır?

Bu sorulara verilecek olası cevaplarla öğrencilerin Scratch programı üzerinden matematiksel algoritmaları oluştururken doğru ya da yanlış sonuca nasıl ulaştıkları ve araştırma sorusuna konu olan süreç üzerinde çalışılmıştır. Araştırma soruları oluşturulurken ilgili literatürdeki boşluklar belirlenmiş ve bazı araştırmalardan bu konuda destek alınmıştır.

1.3. Araştırmanın Önemi

Son yıllarda müfredatta yapılan değişiklikler ve gelişen teknoloji ile eğitim sistemi farklılaşmaktadır. Artık eğitim-öğretim sistemi ve iş dünyası; bilgiyi hazır olarak kullanan değil, bilgiyi kendi üreten ve yapılandıran insan modeline ihtiyaç duymaktadır (Erdem &

Uçak, 2020; Fındıkçı, 1998). Bu kapsamda matematik eğitimi de değişmektedir. Matematik eğitimi formüllerin ezberlendiği bir ders olmaktan çıkıp sorularda formüllerin hazır verildiği ve öğrencinin verilen formülleri kullanıp soruyu çözebileceği bir derse dönüşmüştür.

Günümüzde eğitimin en önemli hedeflerinden biri bireylere günlük hayatta karşılaştıkları problemleri çözme yeterliliği kazandırmaktır (MEB, 2018). Bu yeterlilik matematik dersinde matematiksel düşünme ile geliştirilebilmektedir (Şimşek, 1997). Problem çözme ise algoritma ve algoritmik düşünce ile sıkı bir ilişki içindedir (Demir & Cevahir, 2020). Bu nedenden dolayı öğrencilerin problem çözme becerilerini arttırmak için çeşitli yöntemlere başvurmak gerekmektedir. Bu yöntemlerden biri algoritmik düşünme becerilerini arttırmaktır. Öğrencilerin matematik dersini öğrenmelerini destekleme olanakları, doğru pedagojik ve didaktik ön koşullar altında dijital manipülatif eserlerle mevcuttur (Kaup, 2020). Gelişen teknoloji ile alakalı olarak kodlama dersi (Scratch kullanımı) algoritmalar ile ilgili olduğundan dolayı problem çözme becerisini artırmak için kullanılabilecek yöntemlerden biri olarak değerlendirilebilir (Yünkül vd., 2017).

Teknolojiyi çoğu öğrenci oyun olarak görmektedir. Bu yüzden bilgisayar çalışmaları matematik öğretiminde kullanılabilir. Oyunlar öğrenciye özgüven, matematik öğrenebileceğine inanma, öğrenmeye istekli olma, zevk alma gibi duyuşsal beceriler kazandırır (MEB, 2013). Öğrenciler Scratch programını genel olarak bir oyun olarak görmektedirler. Bu da öğrencilerin güdülenmesini arttırarak olumlu bir sınıf ortamı oluşturmaktadır (Theodorou & Kordaki, 2010).

Bu araştırmada öğrencilerin Scratch programını kullanırken ürettikleri matematiksel algoritma yapıları ve algoritma üretirlerken ortaya çıkan algoritmik düşünme süreçleri incelenmiştir. Bu kapsamda matematiksel çıkarım ve yorumlama becerileri analiz edilmiştir. Araştırmanın, matematiği bilişim dünyası ile ilişkilerini geliştirerek eğitimde kaliteyi artıracığı, öğrencilerin problem çözme sürecindeki başarılarını geliştireceği, algoritma üretme süreçlerini ayrıntılı bir şekilde irdeleneceğinden dolayı matematiksel düşünme ve akademik başarılarına etkisi bakımından önemli olacağı düşünülmektedir.

Birçok öğrenme alanından biri olan tam sayılar konusu öğrenciler tarafından soyut bulunmaktadır. Bu yüzden konu somut hale getirilerek anlatılmalıdır. Tam sayılar, ilerleyen süreçlerde de diğer konuları öğrenebilmek için önemlidir (Hayes ve Stacey, 2003; Kilhamn, 2008). Öğrenci tamsayılar konusunu öğrenemediği takdirde; üslü sayılar, denklemler, köklü sayılar, rasyonel sayılar gibi konularında bir kısmını öğrenip bir kısmını öğrenemeyecektir. Bu da öğrencinin duyuşsal olarak matematikten soğumasına sebep olabilmektedir. Matematik dersinin sarmal olarak ilerlediği göz önüne alındığında, aradaki bir bilginin eksik

ya da hatalı öğrenilmesi olumsuz sonuçlar doğuracaktır. Bu yüzden öğrencilerde oluşabilecek eksiklikleri önlemek ve oluşabilecek hataları tespit etmek açısından araştırmamızın yol gösterici bir araştırma olması hedeflenmiştir.

1.4. Araştırmanın Sınırlılıkları

- Bu araştırma boyunca toplanmış veriler; veri toplama, uygulama ve değerlendirme araçları ile sınırlıdır.
- Bu araştırma, İstanbul ili Sancaktepe ilçesinde 2020-2021 eğitim-öğretim yılında ortaokul öğrenime devam eden üç öğrenci ile sınırlıdır.

1.5. Araştırmanın Varsayımları

Bu araştırmada;

- Öğrencilerin belirlenen Scratch programını kullanmaları sırasında oluşabilecek durumlarda araştırmacı ile sürekli iletişim halinde olacağı varsayılmıştır.
- Araştırmada öğrencilerin veri toplama araçlarına verecekleri cevabın gerçeği yansıtacağı düşünülmektedir.

1.6. Araştırmanın Tanımları

Algoritma: Bir problemin nasıl çözüleceğini, yani verilen verilerden başlayarak tanımlanmış bir hedefin nasıl elde edileceğini açıklayan, organize edilmiş talimatların sonlu bir dizisidir. Adımlar, algoritmanın alıcısı tarafından temel olarak kabul edilmeli ve algoritma belirsiz olmamalıdır. (Modeste vd., 2019).

Algoritmik Düşünme Becerisi: Karmaşık bir hedefi, mevcut araçları kullanarak bir dizi (düzenli) adıma bölmek için kullanılan mantıklı, organize bir düşünme yoludur (Wood vd., 2016). Matematiksel bir alan olarak algoritmik düşünme algoritmaların sonuçlarıyla değil daha çok algoritmaların nasıl oluşturulduğu ve nasıl performans gösterdiği ile ilgilenmektedir (Lagrange, 2014).

Scratch Programı: 2003 yılında Massachusetts Teknoloji Enstitüsü (MIT) tarafından, özellikle 8-16 yaş grubu bireyler için tasarlanan görsel bir programlama ortamıdır. Kullanıcılar kodlama yapmak için kod bloğunu bir yerden bir yere sürükleyerek etkinlikler

oluřturabilmektedir. Özel bir kod yazımı gerekmemektedir. Özellikle algoritma öğreniminde güçlük çekenler ya da programlamaya yeni başlayan kullanıcılar için tasarlanmış bir programdır (Çatlak vd., 2015; Scratch About, 2015).



BÖLÜM II: İLGİLİ ARAŞTIRMALAR

Bu bölümde; 21. yüzyıl eğitim dünyası, 21. yüzyıl eğitim dünyasında matematik eğitimi, algoritmalar, matematik öğretiminde teknoloji kullanımı, Scratch programı ve pozitif tam sayılar başlıkları üzerinde durulmuştur.

2.1. 21. Yüzyıl Eğitim Dünyası

Günümüzde içinde bulunduğumuz toplum, bilgi toplumu olarak adlandırılmaktadır. Bilgi toplumunun ortaya çıkışı 1960'lı yıllar olarak kabul edilmektedir. İnsanlar, 1960 yılından 300 yıl önceye kadar sanayi toplumu daha öncesinde ise tarım toplumu olarak adlandırılmışlardır (Çelik & Karaaslan, 2004). Sanayi toplumundan bilgi toplumuna geçişte en büyük etkenin elektronik devrim olduğu ifade edilmektedir (Kalyoncu, 2012). Elektronik devrim ile bireyler değişmeye başlamıştır. Bilgisayarlar bu dönemde çıkmış ve yoğun bir şekilde insan yaşamını etkilemiştir. İletişimin hızla artması ve dünyanın her yerinden iletişimin sağlanması yine bu dönemdedir. Yine aynı şekilde bireyler, bilgiye artık çok daha rahat bir şekilde teknolojik makineler ile ulaşmaya başlamışlardır. Bireylerin bu değişimi, toplumun ve eğitiminde önemli bir değişim yaşanmasına neden olmuştur (Şan, 2006).

20. yüzyıldaki eğitimin amacı, bireylere öğretim programı kapsamındaki konularla ilgili bilgi aktarmak, onlara sanayide çalışmak için birkaç beceri kazandırmak ve onları ileride yaşayacağı koşullara hazırlamaktır. 21. yüzyıldaki eğitimin amacı ise bu ilkelere ek olarak bireylere ihtiyaç duyacağı bilgi ve beceriyi nasıl kazanabileceğini öğretmek, sürekli değişen hayat durumlarına uyum sağlayabilecek karşılaştığı çeşitli probleme farklı çözümler üretebilecek bireyler yetiştirmektir. Kısacası 21. yüzyıl eğitimi amacı bireylere öğrenmeyi öğretmektir (Köroğlu ve Yeşildere, 2004). 21. yüzyıl eğitimi; öğretim programı daha esnek olan, öğrencilerin bir şeyler üretmelerini bekleyen, ders işleme süreci oldukça karmaşık olan, yeni imkanları sunduğu kadar bir o kadar da fazla problemlere sahip ve hızla değişen bir dünyayı hedef alan bir sistem olarak tanımlanmaktadır. Millî Eğitim Bakanlığı'nın yeniden düzenleyip sunduğu eğitim-öğretim programlarında, 21. yüzyıl bireylerini ifade ederken “bilgiyi üreten, hayatta işlevsel olarak kullanabilen, problem çözebilen, eleştirel düşünen, girişimci, kararlı, iletişim becerilerine sahip, empati yapabilen, topluma ve kültüre katkı sağlayan vb. niteliklere” (MEB, 2018) sahip olması gerektiğini vurgulamıştır.

Eğitim anlayışının değişmesiyle birlikte sınıftaki ders işleme yolları, öğretmen ve öğrencinin sınıftaki rolleri de değişmiştir. Geleneksel eğitim sisteminde öğretmen aktif, öğrenci pasif konumdayken 21. yüzyılda eğitim sisteminde bu durum değişmiştir. Günümüz eğitim sisteminde öğrenci aktif, öğretmen rehber konumuna gelmiştir. Bu öğretim yönteminde dersin işlenişi şu şekildedir:

1. Öğretmen dersin başında öğrencilere konu hakkında kısa bir tanıtım yaptıktan sonra öğrencilere konu ile ilgili ilk örneği verir.
2. Öğrenciler örneği açıklarlar.
3. Öğretmen, konu ile ilgili ilk örneğe benzer ek örnek verir.
4. Öğrenciler ek örneği açıklarlar.
5. Öğretmen ek ve zıt örnek verir.
6. Öğretmen, öğrencilerden zıt örnek vermelerini bekler.
7. Öğretmen konuyu toparlar.
8. Öğrenciler ilke ve genellemeleri yapar.
9. Bu yöntemde konu, öğrencinin ek ve zıt örnek vermesiyle tamamlanır (Dilaver, 2016).

Bu ders işlenişine göre öğretmen öğrencilerine konu ile ilgili örnekler eşliğinde kısa örnekler vermektedir. Öğrenciler bu örnekler ile zihninde yapılandığı bilgileri kendine özgü verdiği örnekler ile açıklaması gerekmektedir. Bu şekilde öğrencilerin kendi öğrenme sorumluluklarını üstlenerek aktif bir şekilde bilgiyi yapılandırmaları beklenmektedir

21. yüzyılda teknolojik makinelerin hayata dahil olması ile tüm kurumlar ve sistemler değişim yaşamıştır. Bu kapsamda eğitim-öğretim faaliyetlerinin de teknoloji ile değişmesi kaçınılmaz olmuştur. Teknoloji diğer tüm alanlarda olduğu gibi eğitimde de kullanılmaya başlanmıştır. Gelişen süreçle birlikte olumlu ya da olumsuz çeşitli durumlar ortaya çıkmıştır. Bu yüzden teknolojinin eğitimde kullanılması belirli çerçevelere oturtulmaya çalışılmıştır. Eğitimde teknoloji kullanımı ile ilgili Uluslararası Eğitimde Teknoloji Derneği (ISTE, 2016), 21. yüzyılda eğitim teknolojisi çerçevesini altı madde olarak belirlemiştir.

- Eğitimde yenilik ve bilgiyi yapılandırma,
- Teknolojik araçların, öğrencilerin iletişim ve iş birliğine olan katkısı,
- Dijital teknolojinin bilgiye ulaşma, değerlendirme ve bilgiyi günlük hayatta kullanmaya dönük etkisi,
- Teknolojik araçların problem çözme, eleştirel düşünme ve karar verme süreçlerine katkı sağlaması,

- Dijital vatandaşlık kapsamında, öğrencilerin teknolojiye yönelik kişisel, kültürel veya toplumsal problemleri, yasal ve etik davranışlar çerçevesinde çözebilmesini sağlaması,
- Teknolojik faaliyetlerin kapsamlı sistem oluşumlarının anlaşılmasına yönelik sağladığı katkısıdır.

21. yüzyıl öğrenme odağı; öğretmenin saf bilgi aktarımı yerine, öğrencilerin güncel yaşamda karşılaşılabilecekleri problemlerin çözümü için gereken becerileri kazanmasıdır. Öğrencilere; problemleri çözme, hayat boyu öğrenme, bir konu alanında yetkinleşebilme gibi beceriler kazandırılması hedeflenmektedir. Bu sayede öğrenci karşısına çıkabilecek ve belki de henüz var olmayan durumların üstesinden gelebilecek yetiyi kazanmış olacaktır. Çünkü 2015 Dünya Ekonomik Forumu'nda ilköğretime başlayan öğrencilerin %65'inin var olmayan işlerde çalışacağı belirtilmiştir. Bu kapsamda eğitim faaliyetleri de araştırma ve sorgulama temelli, problem çözme odaklı, proje ve tasarım temelli bir öğrenme anlayışı haline gelmiştir. Bu yüzyılda, farklı bilgi kaynaklardan veriler toplama, önemli olanların farkına varabilme, bu verileri değerlendirip yeni fikirler geliştirme, derinlemesine bir kavrayış kazanma ve problem çözme becerilerinin kazandırılması hedeflenmektedir. Elde edilen bilginin beceriye dönüştürülerek öğrencinin neyi, nerede, nasıl bulacağı ve kullanacağını bilmesi; bunu elde ettikten sonra problemlere çözüm üretmesi istenmektedir. Eğitim öğrencilere bilgi aktarımından ziyade, günlük yaşam problemleri ve gelecek ile ilgili oluşabilecek problemlerin çözümü için gereken yetkinliği kazandırılması üzerinde durmaktadır (Trilling & Fadel, 2009). Yukarıdaki açıklamalar kapsamında Küçüköğlu (2018), 21. yüzyıl becerilerini; öğrenci iş birliği, dijital okur yazarlık, eleştirel düşünme ve problem çözme gibi yetkinlikler olarak belirtmiştir.

Millî Eğitim Bakanlığı 2018 yılında yayınladığı “Matematik Dersi Öğretim Programı'nda” Türkiye Yeterlilikler Çerçevesi adı altında sekiz adet yeterlilik belirlemiştir. Bunlardan üç tanesi şunlardır;

- Matematiksel yetkinlik ve bilim/teknolojide temel yetkinlikler: Matematiksel yetkinlik, öğrencinin günlük hayatta karşılaşılabileceği problemleri çözmek için matematiksel düşünme yeterliliği olarak ifade edilmiştir. Bilim ve teknolojiye temel yetkinlik ise gelişen ve değişen koşullara bireyin uyumu, vatandaşlık sorumluluklarını anlama ve yerine getirebilme gücü olarak ifade edilmiştir.
- Dijital yetkinlik: Gereken bilgisayar teknolojilerinin iş hayatında ve günlük hayatta kullanılması olarak ifade edilmiştir.

- Öğrenmeyi öğrenme: Bireyin kendi öğrenme biçimini tanıyıp, zamanını etkili kullanabilmesi ve bilgi yönetimi ile bazı stratejiler kullanarak düzenleyebilmesi olarak ifade edilmiştir (MEB, 2018).

Millî Eğitim Bakanlığı'nın yayınlamış olduğu yetkinlikler ile dünyada kabul görmüş 21. yüzyıl yetkinlikleri birbirleri ile uyum içinde olduğu görülmektedir.

Bu bölümde 21. yüzyıl eğitim dünyasının geleneksel eğitimden ayrıştığı yönler, bu çağda uygulanması gereken ders işleniş yönteminden ve 21.yüzyıl becerilerinden bahsedilmiştir. Bir sonraki bölümde 21. yüzyıl eğitim dünyası bağlamında matematik eğitiminden bahsedilecektir.

2.2. 21. Yüzyıl Eğitim Dünyasında Matematik Eğitimi

21. yüzyıl eğitim anlayışında saf bilgi aktarımı yerine öğrencinin bilgiyi yapılandırarak öğrenmesi hedeflenmektedir. Bu kapsamda matematik eğitim anlayışı da değişmektedir. Geleneksel matematik eğitimi anlayışında kurallar ve formüller öğrencilere öğretmen tarafından hazır olarak verilerek öğrencilerin problemleri çözmesi anlamına geliyordu. Bu problemler, genellikle bir gündelik olay örgüsüne bağlanmayan sadece cebirsel ifadelerden oluşuyordu. Bu eğitim anlayışında hedeflenen; öğrencilerin hesaplama becerisine sahip olmasıydı (Pusmaz, 2008). Teknolojinin gelişmesi ile bu hesaplamaları yapan makinelerin icat edilmesi matematik eğitim faaliyetlerinin değişmesinde önemli bir rol oynamıştır. Matematik eğitiminde problem çözme, analiz ve muhakeme etme gibi 21. yüzyıl becerileri ön plana çıkmıştır. Öğrencilerin akıl yürütme ve muhakeme becerilerinin gelişimini arttırmak için matematik eğitimi almış olması ve bu eğitimler sonucu anlamlı çıkarımlar yapabilmeyi öğrenmesi yaşamına pozitif katkılar sağlamaktadır. Öğrencilerin öğrenmeyi öğrenebilmesi, bilgiyi arayıp bulabilmesi ve geçmiş bilgilerini kullanarak yeni bilgiyi üretebilmesi için matematik programları bu detayları göz önünde bulundurarak oluşturulmaktadır. Çocuklar yapıları gereği olaylar ve meseleler üzerinde düşünmekten hoşlanırlar, hoşlandıkları için yaparlar (Skemp, 1986). Bu yüzden bilgiyi kendi kendilerine ürettiklerinde de bu durumdan haz duyarlar. Bilgi kendilerine direkt olarak verildiğinde bu durum hoşlarına gitmez. Günümüzde matematiğe bunun için daha fazla ihtiyaç duyulmaktadır. Çünkü matematik önceki dönemlerde sadece toplumun gündelik ihtiyaçlarının karşılanmasında yararlanılan bir alandı. Oysa bugün matematik, öğrencilerin bilgiyi kendilerinin oluşturmasına fırsat veren, işbirlikçi etkinlikler içeren, olguların mantığını anlama arzusunun ötesine geçerek her mesleğin ve gündelik yaşamın bir parçası

haline gelen bir ders olmuştur (Şengül & Zengin, 2005). Çocuklar, her mesleğin içerisinde bulunan matematikten öncelikle keyif almalıdır. Bunun için öğretmenin değişik etkinliklerle dersi desteklemesi gerekmektedir. İyi bir matematik eğitimi alan birey, gündelik yaşantısının büyük bir bölümünde matematiği kullanarak hayatını kolaylaştıracaktır.

Millî Eğitim Bakanlığı 2018 yılında yayınladığı “Matematik Dersi Öğretim Programı’nda” matematik dersinin özel amaçlarını belirlemiştir. Bunlardan üç tanesi şunlardır;

- Matematiksel kavramları anlayabilen ve bu kavramları günlük hayatta kullanabilen,
- Üstbilişsel bilgi ve becerilerini geliştirmek için istekli, bu bilgi ve becerileri öğrenirken süreci bilinçli biçimde yönetebilen,
- Araştırma yapan, bilgiyi üretmekte istekli ve öğrendiği bilgileri kullanma becerilerini geliştirebilecek potansiyele sahip bireyler yetiştirmek. (MEB, 2018).

Bu açıklamalar kapsamında günümüz matematik programının genel amacını; öğrencilerin, problem içindeki bilgiler arasında bağlantı kurma potansiyelini geliştirerek bu verilerin değerlendirilmesi ve bunların anlamlı bir şekilde kullanılarak bir sonuca varmasını sağlamak olarak ifade edebiliriz (Çekici & Yıldırım, 2011).

Öğrencilerin 21. yüzyıl matematik amaçlarına ulaşmaları için geleneksel matematiksel ders işleme yollarının değişmesi gerekmektedir ve değişimler yaşanmaktadır. Bu kapsamda öğrenmeyi öğrenme ile ilgili olarak algoritmik yaklaşım benimsenmektedir. Ayrıca her alanda olduğu gibi teknoloji, matematik eğitimini de etkileyerek matematik dersinde kendine yer bulmuştur. Bundan sonraki bölümlerde pozitif tam sayılar konusu, algoritma, algoritmik düşünme ve matematiğin teknoloji ile ilişkisi detaylı bir şekilde ele alınacaktır.

21. yüzyıl matematik eğitimi kapsamında öğrencilerin matematiği ne düzeyde öğrendiklerini araştıran çalışmaların yerini nasıl öğrendikleri ile ilgili araştırmaların aldığı görülmektedir (Altun & Memnun, 2012). Matematik dersinin soyut olmasından kaynaklı olarak (Baki, 2006) öğrencilerin matematiği zihinlerinde nasıl soyutladıklarını anlamak günümüz araştırmalarında önemli bir konu olarak karşımıza çıkmaktadır. Bu kapsamda bir sonraki bölümde matematik eğitiminde soyutlama süreci ve özel olarak RBC soyutlama modeli ilgili bilgiler ve araştırmalar sunulmuştur.

2.2.1. Matematik Eğitiminde Soyutlama Süreci ve RBC Soyutlama Modeli

Matematik dersi doğası gereği diğer derslere göre soyut olan bir derstir (Frenkel, 2013). Bu nedenle matematik eğitiminde çeşitli zorluklar ortaya çıkmaktadır. Bu zorluklardan bir tanesi öğrencinin matematik bilgilerini nasıl öğrendiğinin incelenmesidir. Matematiğin bir soyutlama bilimi olması ve matematik kavramlarının büyük çoğunluğunun soyutlama sonucu elde edilmesi, matematik eğitiminde soyutlamayı içeren bilgi oluşturma sürecini anlamayı ayrıca önemli kılmaktadır. Öğrencinin bilgiyi nasıl yapılandığı, yapılandırma sürecinde nelerin etkili olduğu, ne tür koşulların öğrenmenin niteliğini arttırdığı, bilgi oluşturma, soyutlama, soyutlama süreci gibi hususlar, 21. yüzyıl matematik eğitimi anlayışında önemli araştırma konuları haline gelmiştir (Altun & Memnun, 2012). Hershkowitz vd., (2001)'e göre soyutlama süreci, nispeten yeni bir matematiksel yapıya ulaştırmak için bilinen eski yapıların yeniden düzenlenmesi, bu yapılar arasında bağlantı kurulması ve bir matematiksel düşünce süreci içinde birleştirilmesi olarak tanımlanmaktadır. Soyutlama süreci doğrudan gözlem olanağı olmayan bir durumdur. Bu nedenle soyutlama süreci ile ilgili araştırmacılar tek bir anlam üzerinde anlaşamamışlardır. Bununla birlikte soyutlama sürecinde iki farklı görüş hakimdir. Bunlardan birincisi bilişsel soyutlama görüşü iken ikincisi sosyokültürel soyutlama görüşüdür (Yeşildere, 2006). Bilişsel soyutlama görüşü, soyutlamanın sadece insan zihninde gerçekleşen bir süreç olduğu ve bu süreçte oluşan yapıların ortak özelliklerine göre sınıflandırılması ile daha üst bir yapı oluşturulması olarak ifade edilmektedir (Monaghan & Özmantar, 2007). Sosyokültürel soyutlama görüşü ise soyutlamanın sadece bilişsel etkenlere bağlı olmadığı bilişsel etmenlerin yanında çevre, araç-gereç kullanımı, iletişim becerileri gibi sosyokültürel etmenlere de bağlı olduğu belirtilmiştir (Yeşildere, 2006).

Bu araştırma öğrencilerin tam sayılar konusunda algoritma üretme süreçlerini ve süreç sonunda oluşan algoritma yapılarını incelemeyi amaçlamaktadır. Bu kapsamda sadece bilişsel etmenlerin önemli olmadığı bunların yanında öğrenme ortamı, çevre, iletişim becerileri ve diğer etkenlerin önemli olduğu düşünülmüştür. Bundan dolayı sosyokültürel soyutlama sürecinin temel alınmasının uygun olduğu düşünülmüştür. Bu araştırmaların verileri sosyokültürel görüşe sahip olan Hershkowitz, Schwarz ve Dreyfus (2001) tarafından ortaya koyulan RBC soyutlama modeli ile analiz edilmiştir. RBC modeli, öğrencilerin sosyal hayatın matematikleştirilmesi ile ilgili öğretim uygulamaları sırasındaki bilgi yapılandırılmasının yani soyutlamayı bilişsel eylemler üzerinden derinlemesine incelenmesine fırsat veren bir modeldir.

RBC modeli, Hershkowitz, Schwarz ve Dreyfus (2001) tarafından geliştirilmiştir. RBC, Recognizing- Building with- Constructing olarak ifade edilen üç bilişsel eylemin baş harflerinin birleştirilmesi ile kısaltılmıştır. Bu model birçok araştırmacı tarafından benimsenen bir modeldir (Ahsbahs & Bikner, 2004; Altun & Memnun, 2012; Dreyfus, Hershkowitz & Schwarz, 2001; Monaghan & Özmantar, 2007). Bu modelin kullanılma amacı ise soyutlamanın karmaşık süreçlerini detaylı olarak analiz yapmaya olanaklar sunmasıdır (Dreyfus, 2007).

RBC modelinde öğrencilerin fikirleri yine öğrencilerin düşünce ve davranışlarına dayandırılır. Bu fikirler üç farklı gözlenebilir bilişsel eylem [tanıma- recognizing, kullanma- building with, oluşturma- constructing] üzerinden açıklanmaktadır (Altun & Memnun, 2012). Bilişsel eylemlerden ilki olan tanıma, bireyin önceden öğrenmiş olduğu bilgilerle eldeki probleme uygun anlam yüklemesi demektir. Kullanma eylemi, öğrencinin bir hedefe ulaşabilmesi için problemle karşı karşıya kaldığında eyleme geçmesidir. Oluşturma eylemi, yeniden düzenleme ve yeniden yapılanma sürecini ifade eder. Bu süreçte öğrenci sahip olduğu bilgileri yeni öğrendiği bilgilerle tamamlamaya veya birleştirmeye çalışır (Hershkowitz vd., 2001). Bu eylemler bazen sıralı olarak görülebileceği gibi bazen de dinamik olarak birbiri içine geçmiş olabilmektedirler (Dreyfus, 2007).

Tanıma, kullanma ve oluşturma eylemlerine aşağıdaki gibi bir örnek verilebilir (Hisar, 2020). Tanıma: Öğretmen tahtaya altıgen çizer. Öğrenciye bu şeklin kaç kenar olduğunu ve bu şekli nasıl tanıdığını sorar sorar. Öğrenci cevap verir:

- Altıgendir (Tanıma). Çünkü; altı kenarı, altı köşesi vardır ve kapalıdır (Özellik ifade etme- Tanıma).

Kullanma: Öğretmen tahtaya düzgün altıgen çizer. Öğrenciden bu şeklin bir iç açısını ölçmesini ister. Öğrenci açı ölçer ile düzgün altıgenin bir iç açısını ölçer ve cevap verir:

- Bu şeklin bir tane iç açısı 120 derecedir (Açı ölçer kullanma-Kullanma).

Oluşturma: Öğretmen tahtaya dörtgen, beşgen ve altıgen çizer. Öğrenciden bu şekillere ait kaç köşegen olduğunu bulmasını ister. Öğrenci her şeklin köşegenlerini bulduktan sonra öğretmen, her şekil için bir köşeden çizilen köşegen sayısını da bulmasını ister ve bunu matematiksel bir dille ifade etmesini ister. Öğrenci:

- Dörtgenin bir köşesinden çizilebilen 1, beşgenin 2, altıgenin 3 tane köşegen vardır. Yani bir köşeden çizilebilen köşegen sayısı, kenar sayısının 3 eksiği kadardır. Bu durumu $(n-3)$ ile ifade edebiliriz (Matematiksel dil geliştirme- Oluşturma)

Şeklinde cevap verir.

2.2.2. Pozitif Tam Sayılar

Tam sayılar, doğal sayıların zamanla insan ihtiyacını karşılayamamasından dolayı binlerce yıl öncesinden ortaya çıkmıştır. Bilimde ve matematikte hatta günlük yaşamın her alanında tam sayılar karşımıza çıkmaktadır. Tam sayılar tarihi üçe ayrılmaktadır. Bunlar; pozitif tam sayıların, sıfırın ve negatif tam sayıların tarihidir. Pozitif tam sayıların tarihi tam olarak bilinmemekle birlikte negatif tam sayılardan daha öncesinde insanlar tarafından kullanıldığı belirtilmektedir (Körükçü, 2008).

Tam sayılar birçok öğrenme alanının temelini oluşturmaktadır, fakat bu sayılar öğrencilerin zihinlerinde soyut kalmakta ve anlam bulamamaktadır. 6. Sınıfta ilk kez karşılaşılan tam sayılar konusuna “M.6.1.4.1. Tam sayıları tanıy ve sayı doğrusunda gösterir.” Kazanımı ile başlanmaktadır. Bu kazanımda Millî Eğitim Bakanlığı öneri olarak öğrencilerin tam sayılara olan ihtiyacının fark ettirilmesini önermektedir. Aynı zamanda pozitif ve negatif tam sayılara günlük hayattan örnekler verilmesi istenmiştir. Konunun daha da soyut kalmaması için kazanıma, müfredatta olduğu gibi günlük hayatta karşılaşılan tam sayı örnekleriyle başlanmalıdır. Öğrencilere günlük hayatta tam sayılarla hava sıcaklığında, tarih dersinde ya da asansör düğmelerinde karşılaşıldığı örnekleri verilebilir.

Tam sayılar konusu ilk andan itibaren öğrencilerin sonraki birçok öğrenme alanlarını etkileyeceğinden oldukça önemlidir (Hayes ve Stacey, 2003; Kilhamn, 2008). Çünkü tam sayılar konusunun işlendiği sınıf seviyesinden sonra ortaokul müfredatı; üslü sayılar, köklü ifadeler, rasyonel sayılar, koordinat sistemi, cebirsel ifadeler ve denklemler gibi tam sayılar sayıları içeren konularla devam etmektedir. Tam sayıları anlamlandıramayan, negatif ve pozitif sayıları birbirinden ayırmakta zorlanan öğrenciler, haliyle bu sayılar ile ilgili yapılabilecek işlemlerde de zorlanacaktır. Aynı eksiklik ve hatayla devam eden öğrenciler, hayatlarının sonraki dönemlerinde de büyük engel ve zorluklarla karşılaşacak, hatta bu durum matematikten soğumaya kadar gidecektir.

Avcu ve Durmaz (2011), yaptıkları araştırmada ortaokul 6 ve 7. Sınıf öğrencilerinin tam sayıları öğrenirken ne gibi hatalar yaptıklarını ne gibi zorluklar yaşadıklarını tespit etmeyi amaçlamışlardır. Araştırmanın sonucunda 6 ve 7. sınıf düzeyinde öğrenim görmekte olan öğrencilerin tam sayılar konusunda istenilen düzeye ulaşamadıkları görülmüştür. Öğrenciler; negatif ve pozitif tam sayıları ayırt edebilmişler fakat hangi sayının daha küçük, hangi sayının daha büyük olduğunu belirlemede güçlük yaşamışlardır. 0 sayısının önüne +

ya da – koyarak hata yaptıkları görülmüştür. Ayrıca öğrencilerin çarpma ve bölme konusunda çok fazla hata yaptıkları, bunun temel sebebinin ise işareti yok saydıkları için gerçekleştiği gözlemlenmiştir.

Şengül ve Cantimer (2018), 7. sınıf öğrencilerinin tam sayılara yönelik kavram imajlarını değerlendirdikleri çalışmalarında; öğrencilerin tam sayılarda kavram imajlarının yeterli seviyede oluşmadığı sonucuna ulaşmışlardır. Bu çalışmada özellikle tam sayı kavramı öğrenciler tarafından tam olarak açıklanamamış ve sıfır sayısı göz ardı edilmiştir.

Yenilmez ve Bağdat (2013), yaptıkları araştırmada 7. sınıf öğrencilerinin tam sayılar konusunda yaptıkları işlemlerde ne gibi zorluklarla karşılaştıklarını belirlemeyi amaçlamışlardır. 2012-2013 eğitim-öğretim yılında 7. Sınıfta öğrenim görmekte olan 10 öğrenciye 11 açık uçlu soru sorulmuş ve ardından görüşme yapılmıştır. Araştırmanın sonucunda öğrencilerin en çok tam sayılar ile çıkarma işlemi yaparken zorlandıkları saptanmıştır. Bunun sebebinin ise negatif tam sayının işareti ile çıkarma işlemi işaretinin birbirine karışmasından dolayı olduğu belirlenmiştir. Ayrıca öğrencilerin Avcu ve Durmaz'ın (2011) yaptığı araştırma sonucunda olduğu gibi bu araştırmada da çarpma ve bölme işlemlerinde öğrencilerin işaretleri görmezden geldikleri belirtilmiştir.

Atayev (2015), yaptığı araştırmada 6. sınıf öğrencilerinin tam sayıları kavraması ve tam sayılarla sıralama yapabilmesi konularında başarı düzeylerini incelemeyi, bu konuda yaptıkları hataları belirlemeyi ve bunun sonucunda yaptıkları hataların nedenlerini araştırmayı amaçlamıştır. 262 kişiden oluşan bu araştırmada 8 tane açık uçlu soru kullanılmıştır. Sonuçlara göre öğrencilerin tam sayılarla sıralama sorularında orta seviyede oldukları, kavrama sorularında ise başarılarının daha yüksek seviyede olduğu görülmüştür. Kavrama sorularında yaptıkları hatalar genellikle yanlış sembol (+,-) kullanımı ve yanlış hizalamadan kaynaklı olmuştur. Sıralama sorularında yapılan hatalar ise genellikle sayıları ters sıralama, küçüktür ve büyüktür işaretini karıştırmaktan dolayı kaynaklanmıştır. Bu hataların birçoğu soruyu hatalı okumaktan, sayı doğrusunun mantığını tam olarak anlayamamaktan ve tam sayıları da doğal sayı gibi kabul etmekten dolayı gerçekleşmiştir.

Tam sayılar ile ilgili yapılan araştırmalara göre öğrencilerin daha çok negatif tam sayıları anlamlandırmada sorun yaşadığı gözlemlenmiştir. Bunun nedeni pozitif sayılarla günlük hayatlarında daha çok karşılaşmaları gösterilebilir. Bu sebeple tam sayılar konusundaki araştırmalar daha çok negatif tam sayılar ağırlıklı olarak yapılmaktadır.

Özdeş (2013), 9. sınıfların doğal sayılar konusundaki kavram yanlışları üzerine çalıştığı araştırmasında bir sayının pozitif tam bölenlerinin belirlenmesi sorusunda öğrencilerin %50'sinin doğru yaptığını belirlemiştir. Araştırmada öğrencilerin en çok yaptıkları hatalar;

pozitif bölenlerinin kaç tane olduğunu bulma, sıfırın pozitif bölen olduğunu düşünme ve bir sayının bölenleri yerine katlarını bulma olarak gösterilmiştir. Aynı çalışmada, öğrencilere asal sayılar ile ilgili sorular verilmiş ve öğrencilerin ortalama %77'si sorulara doğru cevabı verebilmişlerdir. Öğrencilerin asal sayı sorularında en sık yaptıkları hatalar asal sayıların negatif işaretlilerinin de asal sayı olduğunu düşünme, sıfırın ve 1'in asal olduğunu düşünme olarak belirlenmiştir.

Matematiğin birçok konusunda olduğu gibi tam sayılar konusunun öğretiminde de teknolojiden yararlanılmaktadır. Yapılan araştırmalara göre tam sayıların öğretiminde teknolojiden yararlanılması öğrencilerin öğrenmelerini ve motivasyonlarını arttırmaktadır (Tengah vd., 2017; Yuliwijayanti vd.; 2021). Mercan (2019), Scratch programı destekli tam sayılar konusu ile ilgili yaptığı araştırmasında, bu eğitimi alan öğrencilerin tam sayılar konusundaki başarılarının arttığını gözlemlemiştir.

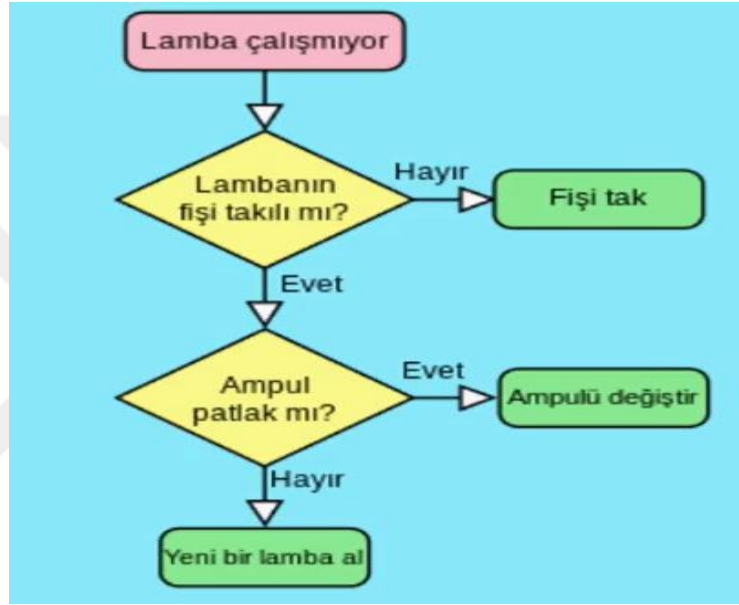
Pozitif tam sayılar konusunun öğrenciler tarafından daha kolay anlaşılmasından dolayı literatürde daha az yer kaplamaktadır. Ancak hayatımızın neredeyse her yerinde karşımıza çıkmasından dolayı ve matematiğin temelini oluşturmasından dolayı bu çalışmada pozitif tam sayılar konusu seçilmiştir.

2.2.3. Algoritma

Algoritma; bir problemin çözümüne veya istenilen amaca nasıl ulaşılabileceğinin belirlendiği yoldur. Algoritma kelimesi, MS 800'lü yıllarda El-Harezmi adlı bir matematik bilgini tarafından ortaya atılmıştır. El-Harezmi "*Cebr ve Mukabele Hesabının Özlü El Kitabı*" adlı yapıtında o günün koşullarında karşılaşılan sorunlar ile ilgili olarak matematiksel çözümler önermektedir. El-Harezmi, bu çözümleri sistemli ve açıklamalı olarak ifade etmektedir (Kılan, 2007). Bu çözüm şekli daha sonra "algoritma" adını alarak günümüze de sistemli-açıklamalı çözüm yöntemi olarak ulaşmıştır. Sözcük anlamı olarak algoritma; bir hedef için oluşturulan yöntemlerin, işe yarar sonuç elde edilinceye kadar devam ettirilmesidir. Başka bir ifadeyle belirli bir kurala bağlı olarak yapılan her türlü hesaplama verilen isimdir. Matematikte kullanılan algoritma ise problemin çözümünü belli bir aşamada veren sistematik yöntemdir. Algoritma tarif, süreç yöntem, teknik, prosedür gibi kelimelerle ifade edilmektedir (Knuth, 2014).

Algoritmalar günlük yaşantımızın birçok alanında karşımıza çıkmaktadır. Evden çıkıp okula giderken hangi yoldan en kısa şekilde gideceğimizi bulmak bir algoritmadır. Bu algoritmayı bulabilmek için zihnimizde çeşitli kombinasyonlar kurabiliriz veya bilmediğimiz yerlere

giderken teknolojiden yararlanabiliriz. Örneğin; Google Maps gibi programlar iki adres arasındaki tüm muhtemel yolları göz önünde bulundurarak bizi en kısa yoldan gideceğimiz yere kadar götürebilir. Ülkemizde çeşitli belediyeler tarafından da kullanılan algoritmalar, kişinin hangi otobüse binip hangi durakta ineceğini bilgilendirmede kullanılır. Ayrıca geleceğe yönelik olasılık ve istatistik tahminleri yaparken, fotoğraf makinesinin yüz algılama özelliğini kullanırken, internette arama motorları ile araştırma yaparken altyapısal olarak algoritma sisteminden yararlanırız. Örneğin; evimizdeki lamba çalışmadığında izleyeceğimiz yol da bir algoritma örneğidir (Bkz Şekil 2.1).



Şekil 2.1. Günlük Yaşam Algoritma Örneği

Algoritma tarif, süreç yöntem, teknik, prosedür, rutin ve tekerlemeye benzer bir anlamı da vardır (Knuth, 2014). “Algoritma” sözcüğü öğrenciler için yabancı ve karışık olarak algılansa da aslında son derece basit bir kavramdır. Günlük yaşantılarında öğrenciler bu kavramı zihinlerinde rahatlıkla eşleştirebilirler. Örneğin; bir yemeğin yapım aşaması ve kullanılan malzemelerin sırası algoritmaya güzel bir örnektir. Günlük yaşamın her alanında var olan algoritmalar; özellikle teknolojik araçların yazılım kısmında kullanılmaktadır. Teknolojik olarak algoritma “*Problemin/uygulamanın beklenen sonuç ya da sonuçlarını verecek- insan makine etkileşimini sağlayacak- yürütecek/yerine getirilecek işlem adımlarını, işleme sırası koşulları ile belirleme yöntemi*” olarak tanımlanmaktadır (Kılan, 2007). Özellikle bilgisayarların yazılım kısmında algoritmaların kullanılması algoritmaların

bilgisayar ile özdeşleşmesini sağlamıştır. Teknolojik aletler ile yapılan her etkileşim aslında birer algoritma ile ilgilidir. Örneğin; ülkemizde üniversite sınavlarına her yıl yaklaşık 2 milyon kişi girmektedir. Bu kişiler arasından birinci olanı belirlemek için aşağıdaki gibi bir algoritma yapılabilir (Bkz Şekil 2.2.). Bu algoritma sayesinde 2 milyon kişiden birinciye belirlemek çok kısa bir sürede tamamlanabilir.

```
#include <stdio.h>

// En büyük değeri bulan fonksiyon
int maksimum(int A[], int N)
{
    int i;
    int max;

    max = A[0];

    for (i = 1; i < N; i++)
    {
        if (max < A[i])
            max = A[i];
    }

    return max;
}

// Ana fonksiyon
int main()
{
    int a;
    int A[] = { -32, 13, -13, -6, 0, 8 };

    a = maksimum(A, 6);

    printf("Dizinin en büyük elemanı: %d\n", a);
}
```

Şekil 2.2. C Programlama Dilinde Yazılmış En Büyük Elemanı Bulma Algoritması (Gündüz, 2016).

Teknolojik aletler, bir problemin nasıl çözüleceği konusunda insanlara doğrudan yardımcı olamazlar. Kişilerin çözüm için oluşturduğu algoritmaları kabiliyetleri noktasında hatasız olarak uygularlar. Bunun için oluşturulan çözüm yolunda hiçbir belirsizlik olmaması gerekmektedir. Bu sebeple oluşturulacak algoritmada her türlü durumlar önceden hesap edilmeli ve karşılaşılabilecek farklı durumlarda teknolojik aletlerin çözüme ne şekilde devam edeceği belirlenmiş olmalıdır. Bu kapsamda bir algoritma genellik, sonluluk ve kesinlik özelliklerine sahip olmalıdır.

Algoritmalar bir problemin çözüm prosedürünü sunmaktadır. Bu prosedür sadece sonuçla ilgili değil süreçle de alakalıdır. Oluşturulan algoritmaların sade ve hızlı olması önemlidir. Algoritmanın, kesin olarak tanımlanmış bir dizi talimattan oluşmasının yanı sıra, her zaman bir sonu olmalı ve her durumda çalıştığı kanıtlanabilmelidir (Aytekin vd., 2018). Yılmazel (2016), bir algoritmanın sahip olması gereken temel özellikleri şöyle ifade etmiştir:

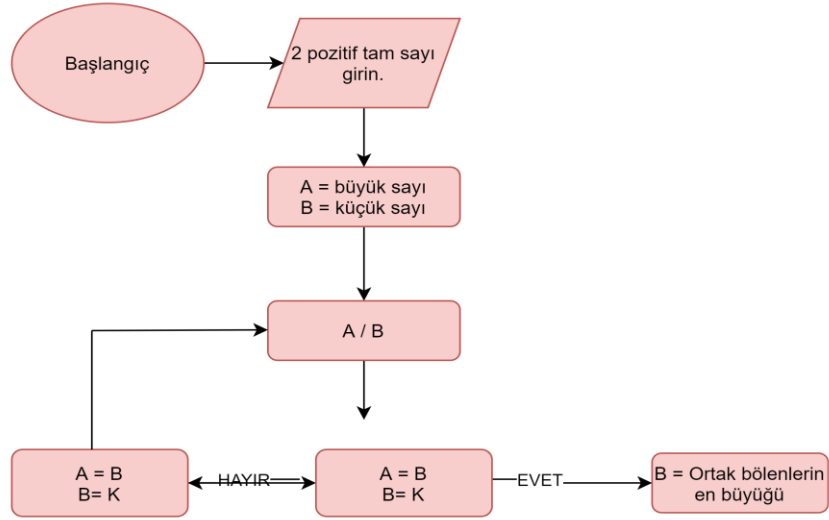
- *Girdi ve Çıktı Bilgisi:* Algoritmalar girdi ve çıktı bilgilerini içermelidir. Algoritmaya girdiler dışarıdan kullanıcı tarafından verilirken çıktıları ise algoritmaları kendi içerisinde üretir.

- *Açıklık*: Algoritmayı oluşturan adımlar kesin bir dille ve doğru şekilde açıklanmalıdır.
- *Doğruluk*: Farklı girdiler olsa bile algoritmalar doğru çıktıları üretmelidir.
- *Sonluluk*: Algoritmaların her zaman bir sonu olmalıdır.
- *Verimlilik*: Algoritmayı oluşturan adımlar ekonomik olmalı, gereksiz adım ve işlemlerden kaçınılmalıdır.
- *Genellik*: Bir algoritma, benzer problemler uygulandığında geçerli ve doğru sonucu vermelidir.

Algoritmaların önemini kavramak ve bu algoritmaların neler olduğunu bilmek çağımıza ayak uydurmak için gereklidir. Evde, işte ve hayatımızın diğer alanlarında karşılaştığımız problemleri etkin bir şekilde analiz ederek, çözüme en kısa yoldan en sade şekilde ulaşabilmek algoritmaları anlamaktan geçmektedir. Ayrıca özgün problemleri hızlıca çözüme ulaştırabilmek adına algoritmalar geliştirebilmek, algoritmaları benimsemeye bağlıdır. Yukarıda bahsedildiği gibi “algoritma” ifadesi ilk olarak matematiksel bir terim olarak ortaya çıkmıştır.

Matematiksel algoritmaların birçoğu tam sayılar konusuna dayanmaktadır. Bunlardan bir tanesi Euclid algoritmasıdır. Euclid’in iki sayının en büyük ortak bölenini belirlemeye yönelik açık ve kesin bir dille oluşturduğu çözüm yöntemi Euclid Algoritması adını almıştır. Bu algoritma ifade edilirken farklı şekillerde ifade edilebilir. Bunlar günlük konuşma dili ve akış şeması ile ifade edilmektedir. Akış şeması, algoritmaların ifade edilmesinde kullanılan oldukça yararlı bir yöntemdir. Bir akış şemasında algoritma adımlarını ifade eden kutucuklar, adımlar arası geçişleri gösteren oklar, karar verme mekanizmaları olarak kullanılan şekiller bulunabilir.

Akış şeması, bir algoritmanın görsel halini ifade eder. Görsellik, algoritmaların daha kolay anlaşılabilmesini sağlar. İki pozitif tamsayının ortak bölenlerinin en büyüğünü bulmak için kullanılan Euclid algoritmasının akış şeması ile gösterimi Şekil 2.3.’te verilmiştir.



Şekil 2.3. Euclid Algoritmasının Akış Şeması ile Gösterimi

Günümüz matematiğinde problemler standart bir algoritma ile çözülen (rutin) ve standart bir algoritma ile çözülemeyen (rutin olmayan) şeklinde bir sınıflama yapılmaktadır (Cai, 2000). Standart algoritma ile çözülen problemler daha çok işlemsel bilginin kullanıldığı problemlerde kullanılmaktadır. Standart olmayan algoritma ile çözülen problemler ise daha çok kavramsal bilginin kullanılması gereken problemlerde kullanılmaktadır. Matematikte daha çok işlemsel bilgi ile problemlerin çözülmesi standart algoritmanın daha çok kullanılmasına neden olmuştur (Sabella, Redish 1995). “Problem Çözme” kavramının araştırmalara sıkça konu olması ile birlikte algoritmalar, matematik dünyasında ezber bilgi olarak algılandı. Bu şekilde teknoloji dünyasında önemi artan algoritmalar, matematik dünyasında geri planda kalmaya başladı. Son yıllarda ise bu durum değişmeye başlamıştır. 21. yüzyıl matematik eğitim anlayışında algoritmalar, problemlere özgün çözümler üretilmesi olarak ifade edilmeye başlanmıştır (Erümit vd., 2018). Yani öğretmen öğrenciye problemi verecektir. Daha sonra öğrenci problemde ki verilere ve formüllere bakarak kendine has bir çözüm yöntemi bulacaktır. Bu durum algoritma ile ilgilidir. Örneğin; Millî Eğitim Bakanlığı Liselere Geçiş Sistemi sınavına hazırlanan öğrenciler için her ay örnek sorular yayımlamaktadır. Son olarak yayımladığı Ekim ayı örnek sorularından bir tanesi Şekil 2.4.’te verilmiştir.

4. $a \neq 0$, $b \neq 0$ ve k, m, n birer tam sayı olmak üzere $a^n \cdot a^m = a^{n+m}$, $(a^n)^m = a^{n \cdot m}$ ve $a^k \cdot b^k = (a \cdot b)^k$ dir.

Aşağıdaki sepetlerde üzerlerinde birer tam sayının yazılı olduğu dörder tane top bulunmaktadır.



Yunus ve Gamze bu sepetlerin her birinden birer tane top alıyorlar. Her biri kendi toplarında yazan sayılardan en büyüğünü taban, en küçükünü üs, diğerini de oluşan üslü ifadenin üssü olarak yazıp birbirine denk iki üslü ifade oluşturuyorlar.

Yunus'un toplarında yazan sayılar 4, -3, -1 olduğuna göre Gamze'nin toplarında yazan sayıların toplamı kaçtır?

A) -8

B) -7

C) -3

D) 0

4

Şekil 2.4. Millî Eğitim Bakanlığı'nın Yayımlamış Olduğu Matematik Örnek Sorusu

Soruda öncelikle üslü sayılarla ilgili kurallar verilmiştir. Daha sonra bu kurallara bağlı olarak öğrencinin tüm olasılıkları denemesi beklenmiştir. Bu sorunun çözümünde öğrenci dördün küpünü 64 bularak Yunus'un bulduğu sonuca ulaşacaktır. Öğrenci bu duruma göre soruya özgü şöyle bir algoritma oluşturabilir:

1. Soruya göre Yunus'un oluşturduğu sayı $(4^{-3})^{-1} = 64$ olur.
2. 64 sayısı başka hangi sayılarla oluşturulabilir? 8^2 , $(-8)^2$, 4^3 , $(-2)^6$ veya 2^6 olabilir.
3. Şimdi yukarıda ki sayılarımız sepetlerde var mı diye bakalım.
4. Sepetlerde 8 olmadığı için 8^2 , $(-8)^2$ durumlarını eylelim. 4'ü Yunus seçtiği ve $-2 < 2$ olduğu için Gamze'ye son seçenek olarak 2 kalıyor.
5. Sepetlerdeki sayılara bakarak 2^6 sayısını oluşturalım.
6. Bunun için 2, -3, -2 sayılarını seçersek ve $(2^{-3})^{-2} = 64$ sayısını oluşturabiliriz.
7. Bu sayıları toplarsak -3 cevabına ulaşabiliriz.

Algoritma kavramı sadece Millî Eğitim Bakanlığı'nın müfredatında değil Yükseköğretim Kurulu tarafından 2021 yılında yayımlanan İlköğretim Matematik Öğretmenliği Lisans Programı'nda da yer almaktadır. 4. yarıyıl da alan eğitimi altında "Algoritma ve Programlama" dersi yer almaktadır. Dersin açıklamasında ise şu ifadeler yer verilmektedir;

Algoritma tasarımı; akış diyagramları, girdi-çıkış kavramları, döngüler, karar yapıları, karar verme ve döngüsel problemlere uygun algoritmaların geliştirilmesi; algoritma ve akış şemalarının görselleştirilerek kullanıldığı (scratch, code.org gibi) programların uygulamaları; fonksiyon kullanarak uygun çözüm algoritmalarının oluşturulması; tek ve

çift boyutlu diziler kullanarak uygun çözüm algoritmalarının geliştirilmesi; oluşturulan algoritmaların Bilgisayar Cebir Sistemlerinde kodlanması ve uygulamaları.

Yukarıda yazılanlardan da anlaşılacağı gibi algoritmalar günlük hayatımızın birçok alanında ve matematiğin içinde bulunmaktadır. Bu yüzden algoritmalar ile ilgili araştırmalar, matematik eğitimi alanında yapılan konferanslara da konu olmaktadır (Modeste vd., 2019; Kaup, 2020). Çalışmanın cevap aradığı problemlerden biri olan algoritmik düşünme ile ilgili literatür bir sonraki başlık altında verilmiştir.

2.2.3.1. Algoritmik Düşünme

Algoritmik düşünme; bir problemin çözümüne ulaşmak için oluşturulan sistematik ve ayrıntılı işlemler bütünü olarak açıklanmaktadır (Çatlak vd., 2015). Algoritmik düşünme becerisinin özellikle bilişim dersi için geçerli olduğu düşünülse de matematik ve fen bilimleri gibi dersler için de stratejik bir öneme sahiptir. Örneğin, öğrenciler çarpma işlemini yaparken algoritmik süreçleri kullanmaktadırlar. Böylece; öğrenciler problem çözümünde gerekli olan tanımlama, analiz ve değerlendirme aşamalarını uygulamış olmaktadır. Asıl amaç ise bu algoritmik yaklaşımı öğrencilerin bilinçli olarak farklı koşullarda da uygulayabilme becerisine sahip olmasıdır (Tynker, 2021). Algoritmik düşünme, algoritmaları istenen şekilde belirlemeyi, algoritmanın çalışması için gerekli olan süreci analiz etmeyi ve algoritmanın doğru cevabı vereceğini garanti eden işlemleri içermektedir. Algoritma, kısaca bir problem çözme prosedürüdür. Algoritma çözümlerinde önceden hazırlanmış kalıplar kullanılmasına karşın her zaman daha kısa ve daha az maliyetli algoritmalar üretilmeye çalışılır. Algoritma ile ilgili düşünceler, davranışlar, beceriler “Algoritmik Düşünme” adı altında toplanmaktadır (Yıldız, Çiftçi & Karal, 2017). Sabella ve Redish’e göre (1995), algoritmik düşünme sistemi, öğrencilerin belli talimatları uygulayarak problemleri çözmesi olarak anlaşıyordu. Yani öğrenci problem çözüm yolunu (algoritmayı) öğretmenden öğrenip daha sonra karşısına çıkan problemleri bu çözüm yolunu kullanarak çözmesi anlamına geliyordu. Son yıllarda ki çalışmalar algoritmik düşünmeyi, “öğrencinin bir problem karşısında kendisinin bir çözüm yolu bulması, bunun doğruluğunu test etmesi ve benzer problemlere uygulayarak genelleştirmesi” olarak tanımlamaktadır (Thomas, 2014). Bu şekilde öğrencilerin yeni durumlara uygulanan hazır yapıları kullanmalarından ziyade özgün çözümler üretmeleri önem kazanmıştır (Nemirovsky & Rasmussen, 2005). Algoritmik düşünme; algoritmaları anlama, uygulama, değerlendirme ve üretme becerisi olarak da tanımlanabilir (Brown, 2015). Örneğin; Carl Friedrich Gauss

(1777-1855) 10 yaşında iken 1'den n'ye kadar olan sayıların toplamını formüle dökerek aslında bir algoritma oluşturmuştur. Gauss'un ilkökul öğretmeni tarafından sorulduğu 1'den 100'e kadar olan sayıların toplamını algoritmik düşünme sistemi ile sistemli bir hale getirmeye çalışan Gauss, bu sorunun cevabını direkt bularak herkesi şaşırtmıştır. Bu kadar hızlı yapmasındaki sebep, sayı listesinin iki zıt ucundan birer sayı alıp topladığında hep aynı sonucun çıktığını fark etmesiydi. ($1 + 100 = 101$, $2 + 99 = 101$, $3 + 98 = 101$, ...) Böylece 1'den 100'e kadar olan sayıların toplamı $50 \times 101 = 5050$ oluyordu. Bunu formülleştirerek 1'den istenen bir n sayısına kadar olan sayıların toplamını $n*(n+1)/2$ ile temsil etmiştir. Bu çözüm şekli algoritmik düşünme ile ilgili bir örneği bize sunmaktadır. Gauss yöntemi günümüzde ardışık sayıların toplamı olarak da ifade edilmektedir. Ancak öğrencilerin bu konudaki öğrenme bilgilerini ölçen araştırmalar sınırlıdır. Ekmekçi ve Güler (2016), yaptıkları araştırmada matematik öğretmeni adaylarının ardışık tek sayıların toplamı konusundaki ispat becerilerini incelemişlerdir. Araştırma sonunda öğretmen adaylarının ispat becerilerinin oldukça zayıf olduğu belirlenmiştir.

Algoritmik düşünme, modern yazılımları tasarlamamıza ve anlamamıza yardımcı olan resmi ve yapılandırılmış bir düşünme türüdür. Bunu öğrenmek kolay değildir. Çok fazla pratik ve dayanıklılık gerektirir. Algoritmik düşünme, bir problemi nasıl çözebileceğimize dair bir prosedürdür. Algoritmik yaklaşımda aslında öğrenciden beklenen, ortaya konulan problem durumuna ilişkin çözüm yolunu öğrenme sorumluluğunu üstlenmesidir (Rasmussen vd. 2005). Bunu yaparken önceden öğrendiği matematiksel terim, teorem ve işlemleri kullanabilir. Burada öğrenciden istenen, bilgiyi yapılandırıp çözüme ulaşabilecek bir forma getirmesidir. Ayrıca öğrenci, algoritmayı test edip doğruluğunu kanıtlamalıdır. Öğrencilerin istenilen algoritmaları kolaylıkla oluşturdukları ama her zaman uyguladığında ya da benzer problemlere uygulamaya çalıştıklarında doğru sonucu veremediği görülmüştür (Moala, 2019).

20. yüzyılın ikinci yarısından itibaren algoritma ile ilgili araştırmalarda teknolojinin gelişmesi ile birlikte artış yaşandığı görülmektedir (Knuth, 1980). Son yıllarda ki araştırmalara bakıldığında ise öğrencilerin algoritmaları nasıl algıladıkları, algoritma üretirken nasıl bir yol izledikleri üzerinde çalışmalar artmıştır (Moala, 2019; Thomas, 2014). Bu araştırmaların amacı algoritma oluştururken izlenen yolları, düşünceleri ortaya çıkarmaktır. Bu şekilde günümüzde teknolojinin yaygınlaşması ve matematik öğretiminin değişmesiyle önem kazanan algoritmalar daha detaylı anlaşılacak ve öğrencilere daha doğru bir şekilde aktarılacaktır.

Teknoloji, 1950’li yıllarda Turing’in günümüz bilgisayarına benzer makine icat etmesiyle günümüze kadar hızlı bir gelişim göstermiştir. Bu yeni teknolojinin altyapısında algoritmalar vardır (Knuth, 1980; Lagrange. 2014). Modeste (2019), bu algoritmaların matematik açısından çerçevesini problemlerle karşı karşıya kalan öğrenciler tarafından geliştirilen stratejileri tanımlamaya ve analiz etmeye yönelik olarak çizmiştir. Bu çerçevede kullanıcılar bilgisayarlar üzerinde kodlama yaparken matematiksel tasarımı, analizi ve problem çözümlerini kullanmaktadırlar. Knuth (1980), algoritmik düşünmenin çerçevesini formül işlemleri, gerçekliğin temsili, daha basit probleme indirgeme, soyut akıl yürütme, bilgi yapıları ve algoritma ile ilişkilendirmiştir. Algoritma oluşturulurken öğrenci yapacağı işlemle ilgili olarak basit formülleri bilmesi gerekmektedir. Daha sonra istenen problemi analiz ederek temsili olarak bunu ifade etmelidir. Bu kısımda problem daha kolay bir şekilde düşünülerek gerekli işlem ve kavramlar anlaşılmalı ve eldeki veriler ile akıl yürüterek algoritma oluşturulmalıdır.

Günümüz matematik öğretiminin amacı, kişiye günlük yaşamın gerektirdiği matematik bilgi ve becerileri kazandırmak, ona problem çözmeyi öğretmek ve olayları problem çözme yaklaşımı içinde ele alan düşünme biçimi kazandırmaktır (Altun, 2002). Buradaki problem çözmeyi öğretme kısmı; oluşan farklı ve sürekli değişen durumlara rağmen öğrencilerin o duruma özgü çözümleri oluşturması ve benzer durumlarda uyguladığında doğru sonucu vermesi istenmektedir. Bu da algoritma ile ilgilidir (Moala, 2019). Algoritmik düşünme ile yapılan bir araştırmada, iki grup öğrenciye bir problem durumu verilip iki gruptan da algoritma oluşturması istenmiştir. İki grupta doğru algoritmayı oluşturmuş ancak yalnızca bir grubun algoritmasının benzer diğer problemlere de uygulanabileceği belirtilmiştir. Bunun nedeni olarak oluşturulan algoritmanın daha açık ve farklı olası durumları da içeren bir şekilde oluşturulduğundan kaynaklı olduğu belirtilmiştir (Moala,2019). Yoon vd. (2018), öğrencilere günlük hayat problemi vererek bununla ilgili algoritma oluşturmalarını istemişlerdir. Birçok öğrencinin algoritmaları sistematik ve genellenebilir bir çözüm yolu değil, bir kalıp, formül, denklem olarak gördüğü ortaya çıkmıştır. Bu araştırma sonucunda Yoon vd. öğrencilerin algoritma oluşturmaya ve revize etmeye yönelik parça parça yaklaşımı hakkındaki bulguları, öğrencilerin hazır algoritmaları uygulamak yerine kendi algoritmalarını oluşturmak için daha fazla fırsata sahip olmaları gerektiğini vurgulamışlardır.

Lagrange ve Laval (2019), matematik eğitiminde bilgisayar programlamayı içeren Algoritmik Çalışma Alanı ve Matematiksel Çalışma Alanı adı altında iki farklı çalışma alanı belirleyerek bir araştırma yapmışlardır. Ortaöğretim fonksiyon konusu üzerinde kâğıt-kalem

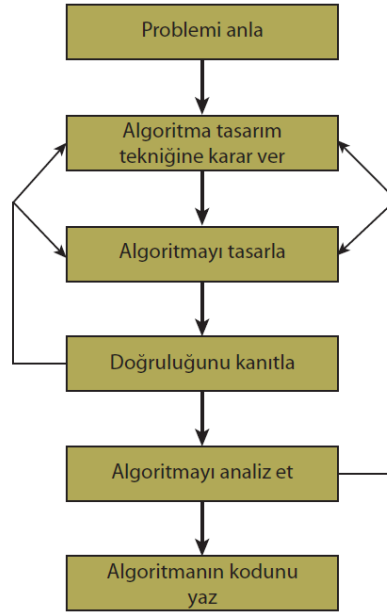
ve bilgisayar programı üzerinde çözme olarak çalışma yürütmüşlerdi. Çalışma sonunda Algoritmik Çalışma Alanı ve Matematiksel Çalışma Alanı ile ilgili bir tablo yapılmıştır (Bkz: Tablo 2.1.).

Tablo 2.1. Lagrange ve Laval (2019), Algoritmik Çalışma Alanı ve Matematiksel Çalışma Alanı İlişkisi

Matematiksel Çalışma Alanı	Algoritmik Çalışma Alanı
• Söylemsel boyut: – Matematiksel bir çözüme odaklanın, – Fonksiyonların özellikleri. • Göstergibilimsel boyut: – Olağan matematiksel biçimler, – Sonsuz dizinin biçimleri, – Grafikler (ikonik). • Enstrümantal boyut: – Kâğıt kalem hesaplamaları, Grafikler.	• Söylemsel boyut: – Algoritmik Yaklaşım sürecine odaklanma: sonlandırma, etkinlik, verimlilik. • Göstergibilimsel boyut: – Çözümlerin özellikli belirteçleri, – Değişkenleri, – Matematiksel ifadeler. • Enstrümantal boyut: – Otomatik cihaz tarafından yürütme.

Tablo 2.1. incelendiğinde algoritmik çalışma alanının matematik çalışma alanını kapsadığı görülmektedir. Algoritmalar oluşturulurken öğrencinin hem matematiksel çözümü bilmesi hem de bunu kodlama ile oluşturması gerekmektedir. Bunun yanında bilgisayar üzerinde algoritma çalışmanın da avantajları vardır. Bunlardan bir tanesi çözümden hemen sonra algoritma otomatik olarak cihaz tarafından denenir. Eğer yanlışsa, yanlış olduğu dönütüyle beraber direkt düzeltme imkânına sahip olunabilir. Eğer doğruysa, bundan sonraki özelliklerle büyük sayıların kullanıldığı işlemlerde zaman ve işlem sayısı bakımından oldukça tasarruf sağlanabilir.

Algoritmik düşünme ile problem çözmenin algoritması Levitin (2012) tarafından aşağıdaki gibi ifade edilmiştir.



Şekil 2.5. Algoritma Tasarım ve Analiz İşlemi (Levitin, 2012)

Diyagrama baktığımızda öncelikle bir problemi tüm detaylarıyla anladıktan sonra yapacağımız iş, algoritmayı hangi ayarlara ve özelliklere sahip bilgisayarda çalıştırmak istediğimizdir. Bu özellikler arasında işlemci hızı, bilgisayarın hafızası gibi işlevler önemlidir. Önce problemimizi küçük parçalara ayırıp her bir parçayı çözdükten sonra sonuçları bilgisayar üzerinde birleştirerek gerçek sonucumuza ulaşabiliriz. Bu yöntem, bize zamandan tasarruf etmemizi sağlayacaktır.

İkinci aşamada ise algoritma tasarlama süreci bulunmaktadır. Algoritmaları tasarlarken konuşma dilinden, akış şemasından ve sözel kodlardan yararlanabiliriz. Konuşma dilini kullanacaksak çok dikkatli olunması gerekir. Bu dil herkesin anlayabileceği bir dil olmalıdır. Bu yüzden algoritma tasarımında genellikle konuşma dili kullanılmaz. Bilgilerin yapısını seçerken mutlaka olumlu ve olumsuz yönlerini dikkate almalıyız. Ayrıca teknoloji için kullanacağımız algoritmanın programlama dili de algoritma tasarımındaki önemli hususlardan biridir. Nesne tabanlı veya blok tabanlı bir dil kullanacaksak, tasarımlarımız da bu dile uygun olmalıdır.

Algoritma tasarımı bittikten sonra, algoritmanın tüm girişler için doğru sonucu vereceğinin garanti edilmesi gerekmektedir. Bir algoritmanın hatalı çalıştığını gösterebilmek için bir tane yanlış sonuç yeterli olmaktadır. Fakat doğru çalıştığını ispatlamak için tüm girişlerde doğru sonucu verdiğini teyit etmek gerekmektedir. Örneğin, bölme işlemi yapan bir algoritmanın, sayı bölü sıfır işleminde tanımsız sonucuna ulaşması gerekmektedir. Algoritmanın kodunu

yazmaya başlamadan önceki son işlem algoritmayı analiz etmektir. Bu analiz yapılırken algoritmanın sonucu verme süresi, karmaşıklık düzeyi ve hafızası test edilmelidir. Matematiksel açıdan algoritma burada bitmektedir. Teknolojik olarak algoritmada son işlem seçilen programlama dilinde algoritmanın kodunu yazmaktır. Kodlama işi bittikten sonra algoritmayı belirli sınamalardan geçirmek, programın güvenilirliğini arttıracaktır (Gündüz, 2016).

Algoritma ile problem çözme sürecinde şu adımlar izlenebilir;

a. Problemin Tanımlanması: İlk olarak verilen problem tam olarak anlaşılmalıdır. Eksik veya yanlış anlaşılan bir problemin çözümü otomatik olarak doğru cevabı vermeyecektir.

Burada yapılacak en ufak bir yanlış ileriki adımlarda ortaya çıkarsa çözüme en baştan başlamak gerekebilir. Problemin tanımında eldeki veriler ve birbirleri ile ilişkileri tüm yönleri ile ele alınmalıdır. Sonrasında istenenler belirlenmeli ve bunların eldeki veriler ile ilişkileri ortaya çıkarılmalıdır. En son olarak çözüme ulaşmak için yapılacak işlemler seçilmelidir.

b. Algoritma Geliştirme: Problem tanımından sonra çözüme ulaştıracak yol belirlenmeye çalışılmalıdır. Problemlerin birden fazla çözüm yolu olabilir. Bunlar arasından en uygunu seçilmeye çalışılmalıdır. Problem karışık ise problem alt bölümlere ayrılmalıdır. Bundan sonra her alt bölümün çözümü ayrı olarak yapılmalıdır. Buradaki önemli husus parçalar arasındaki ilişkinin korunmasıdır.

c. Girdi ve Çıktı Biçimi Belirleme: Algoritmanın başka kişilere aktarımı en ideal biçimde yapılmalıdır. Program çıktısı istenen çözüm formatında olmalıdır. Bu format kullanıcılar tarafından anlaşılabilir ve kullanılabilir olmalıdır.

d. Akış Şemasını Çizme: Akış şeması, algoritmanın belli bir sıra halinde verilmesidir. Algoritma oluşturulduktan sonra daha iyi anlaşılabilir olması için akış şeması çizilebilir. Akış şeması yapılması gerekli bir şart değildir. Ancak algoritmanın daha iyi anlaşılması için sıkça kullanılan bir yöntemdir. Akış şeması ile problemin çözüm basamakları ve bunların birbirleri ile ilişkileri daha rahat görülebilir ve varsa yanlışlıklar düzeltilebilir.

e. Kodlama: Yapılan algoritmanın teknoloji ortamına aktarılması durumudur. Bu adımda eldeki algoritmaya uygun programlama dili seçilmelidir. Genellikle birden fazla programlama dilleri ile yazılabilen kodlamalar arasından en uygunu belirlenmelidir.

f. Programı Sınama: Algoritma teknolojik ortama aktarılın veya aktarılmasının sonuçları daha önceden belirlenen girdileri programa girerek eldeki çıktılarla sonuçlar karşılaştırılmalıdır. Bu şekilde algoritmanın doğru çalışıp çalışmadığı test edilmelidir. Daha sonra küçük ve

büyük örnekler için algoritma test edilmelidir. Çünkü bir algoritma, problemin tüm girdileri için doğru çözümü üretmesi gerekmektedir (Orhan, 2010).

Algoritmik düşünmeyi geliştirmek için Millî Eğitim Bakanlığı ve TÜBİTAK ayrı ayrı projeler geliştirmişlerdir. 2023 Eğitim Vizyonu kapsamında Millî Eğitim Bakanlığı'na bağlı Yenilik ve Eğitim Teknolojileri Genel Müdürlüğü “Bilgisayarsız Algoritmik Düşünme Eğitimi” düzenlemiştir. Bu kapsamda algoritmik düşünmenin sadece bilgisayar programlama sürecinde değil tüm hayatımızda kullandığımız bir düşünme türü olduğunu belirtmişlerdir.

Bu çalışmada algoritmik düşünme bir problem çözme yolu olarak tanıtılmıştır. Bu süreçte algoritmik düşünme şu adımlardan oluşmaktadır:

- Problemin farkına varmak,
- Problemde neyin önemli olduğuna bakarak sorunu ayıklamak,
- Problemdeki temel unsurları soyutlamak,
- Problemi çözebilmek için olası çözümleri bulmak,
- Problemi gerçekten çözecek çözümü bulmaktır.

Bu kapsamda algoritmik düşünme; sonuca giderken kullanılan bütün aşamaları değerlendirerek daha verimli sonuca ulaşma yöntemi olarak tanımlanmaktadır. Algoritma üretme sürecinde sadece sonuç odaklı değil süreç odaklı olarak da düşünmek gerektiği vurgulanmıştır. Bu projeye göre teknoloji kullanılmadan yapılan algoritmik düşünmenin geleneksel problem çözme yönteminden farkı, çözümün açık, net ve anlaşılır sonlu sayıdaki adımlar ile ifade edilmesidir (MEB). TÜBİTAK ise “Etkinlik Temelli Algoritmik Düşünme Eğitimi” projesini 2021 yılının Ağustos ayında gerçekleştirmiştir. Proje okul öncesi öğretmenleri ve sınıf öğretmenlerine yönelik olarak gerçekleştirilmiştir. Projede, günlük hayatta karşılaşılan problemlerin çözümünden, en zor matematik problemlerinin çözümüne, insan ilişkilerinde başarıdan, bilimsel alanda başarıya kadar her türlü alanda algoritmik düşünme becerilerinin kullanıldığı ifade edilmiştir. Bu kapsamda öğrencilerin 21. yüzyılın gerektirdiği teknolojik altyapıya sahip olabilmesi, aynı zamanda bir grubun parçası olabilmesi, problem çözebilmesi, günlük hayatta etkili kararlar alıp uygulayabilmesi için algoritmik düşünme becerisini kazanmış olarak mezun olması gerektiği belirtilmiştir. Eğitim-öğretim faaliyetlerinin temel amaçlarından biri de bireye algoritmik düşünme becerisini kazandırmak olmalıdır. Bu amaç ile öğretmenlere “Etkinlik Temelli Algoritmik Düşünme Eğitimi” eğitimleri verilmiştir (TÜBİTAK, 2021).

2.3. Matematiğin Teknoloji ile İlişkisi

Teknolojinin yaygınlaşması ile beraber matematik eğitiminde de farklı sorun ve ihtiyaçlar ortaya çıkmaya başlamıştır. Öğrencilerin istenen hedef ve davranışlara ulaşmaları için teknolojinin öğretimde nasıl kullanılacağı önemli bir soru haline dönüşmüştür (NCTM, 2000). Eğitim-öğretim faaliyetlerinde kullanılan bilişim teknolojileri, öğrencileri pasif durumdan kurtarmış ve öğrencilerin derslere olan ilgisini arttırmıştır (Yalın, 2006). Bu sebeple matematik öğretiminde teknolojiden faydalanmak, öğrencilerin matematiği daha kolay öğrenmesini ve olumlu yönde güdülenmelerini sağlayacaktır.

Teknolojinin matematik eğitiminde kullanılması öğrencileri ve öğretmenleri olumlu yönde etkilemiştir. Öğrenciler için dersler daha ilgi çekici olmuş ve konuları kavramlarına yardımcı olmaya başlamıştır. Öğretmenler için ise konuyu anlatmak için farklı içerikler kullanmasının önünü açmıştır. Örneğin 7.sınıf kazanımlarından olan “Cisimlerin farklı yönlerden görünümelerini çizer” kazanımını dinamik geometri programı olan GeoGebra ile işlenmesi öğrenciler için fayda sağlayan bir araç olmuştur. 21. yüzyıl becerilerinin kazandırılmasında teknoloji destekli öğretimin rolü git gide artmaktadır. Öğrencilerin teknoloji ile içi içe büyümesi, teknoloji ile ders işlemesini gerekli kılmaktadır. Çünkü öğrenciler geleneksel ders materyallerinden hemen sıkılabilmektedirler. Örneğin Artırılmış gerçeklik ile öğrenciler sanal ortamda görselleri üç boyutlu olarak bir gözlükle görebilirler. Bu araç nesnelerin arasında öğrencinin dolaşmasına bile fırsat tanımaktadır. Öğrenciler sanal nesneleri istedikleri şekilde gözlemlemekte, onları hareket ettirmekte ve döndürebilmektedirler. Bu sayede öğrenci derse karşı olumlu duygular beslemekte ve derse aktif katılım göstermektedir.

Teknoloji, öğrencilerin matematik dersini öğrenme süreçlerinde kullanılabilecek en etkili yollardan biridir (Saka & Çelik, 2018). Öğrencilerin matematik dersine olan kaygı düzeylerini azaltabilmek için teknolojinin matematikle bir etkileşim halinde olması öğrenciye iyi bir fırsat sunmaktadır. Teknoloji ile matematik öğretiminde daha çok alıştırmaya, uygulama ve eğitsel oyun programları kullanılmaktadır (Alakoç, 2003). Akıllı tahtaların sınıflara gelmesiyle birlikte daha çok etkileşimli eğitici oyunların oynatılmaya başlanması matematik algısını olumlu yönde değiştirmeye başlamıştır.

Yukarıdan da anlaşılabileceği üzere eğitimde teknolojinin etkin bir biçimde kullanılması, öğrencilerin matematik dersine karşı ilgilerinin ve motivasyonlarının arttığı yönündedir. Öğretimi zenginleştirmek için derslerde mutlaka teknolojiden yararlanılmalıdır. Teknolojik alet olan bilgisayarlar ile matematik öğretimi bir sonraki başlık altında toplanmıştır.

2.3.1. Bilgisayar Destekli Matematik Öğretimi

Öğretim teknolojilerinden biriside bilgisayar destekli öğretimdir. Bu öğretimde bilişim teknoloji araçları olarak bilgisayar gibi aletler kullanılmaktadır. Bu aletler hızlı hesaplama yapabilmek ve üç boyutlu nesneleri daha kolay tanıyabilmek için kullanılabilecek en etkili araçlardır. Bu kapsamda matematik öğretimde kullanılabilecek birçok bilgisayar programı vardır. GeoGebra, Cabri ve Grafik Analiz gibi yazılımlar matematik için özel olarak tasarlanmış yazılımlardan birkaçıdır. Bu yazılımlar dinamik olarak öğrenciyi sürece katarak matematiği daha iyi kavramasına olanak sağlamaktadır. Bu programların kullanımı aşamasında çeşitli kolaylıklar sunması ve zaman tasarrufu sağlamasından dolayı matematik öğretimde tercih edilmektedir. Bu yazılımların dışında algoritmik işlemlerin ön planda olduğu Scratch programı gibi görsel programlamalar da matematik eğitiminde kullanılmaktadır.

Öğrencilerin problem çözme, matematiksel muhakeme, algoritmik düşünme gibi günümüzde matematik eğitiminde ön planda olan becerileri kazanmaları için birçok yöntemler vardır (Shin & Park, 2014). Bu yöntemlerden bir tanesi de bilgisayar destekli öğretimdir. Bu konuda yapılan araştırmalarda, bilgisayar programlarının kullanılması problem çözme, mantıksal ve analitik muhakeme becerileri gibi matematik ile ilgili becerilerin arttırdığı gözlemlenmiştir (Akpınar & Altun, 2014; Annamalai & Salam, 2017). Matematik dersi öğretim programında öğrencinin konuyu kavraması sürecinde bilişim teknolojilerinin derslerde kullanılması istenmiştir. Bazı geometri ve veri işleme kazanımlarında üç boyutlu dinamik geometri yazılımlarından ve uygun iletişim teknolojilerinden etkileşimli olarak yararlanılabileceği önerilmiştir. Çünkü bilgisayar destekli öğretimin öğrenciye birçok avantaj sağlayacağı düşünülmüştür.

Bilgisayar destekli öğretimin avantajları şunlardır:

- ✓ Geleneksel öğretim yöntemlerinin yerine geçmekten ziyade onları desteklemesi.
- ✓ Öğrenmeyi kolaylaştırarak kalıcılığı sağlaması.
- ✓ Etkileşimli yapısı sayesinde öğrencilerin benzer durumlara genellemelerini sağlayacak etkinliklere kolay ulaşılmalarına olanak vermesi.
- ✓ Anlık geri bildirim alarak bilgiyi anlık olarak yapılandırabilmesini sağlaması.
- ✓ Öğretimde niteliğin artmasını sağlaması.
- ✓ Öğrencinin öğrenme eksikliklerinin tespitinde ve giderilmesinde kullanılması.
- ✓ Öğrencinin uzamsal zekasını geliştirmeye katkı sağlaması.
- ✓ Zaman tasarrufu sağlaması ile daha çok alıştırmaya yapmaya olanak sağlaması.

Bilgisayar destekli öğretimin sınırlılıkları şunlardır:

- ✓ Öğrencilerin bilgisayar ile fazla etkileşime girmesi sonucu diğer öğrencilerle daha az iletişim kurması sonucu sosyo-psikolojik olarak olumsuz etkilenmeleri.
- ✓ Maliyetli olması.
- ✓ Özel donanım ve beceri gerektirmesi.
- ✓ Hedef kazanıma uygun olmayan programların kullanılma olasılığının olması.

Bilişim teknolojilerinin klasik somut ders materyallerinden temel farkı öğrencilerin program üzerinden yapılan her bir değişikliği anlık olarak görmesidir. Örneğin; GeoGebra programında bir tablodaki verilerin grafik üzerinde gösterildiğini düşünelim. Veriler değiştikçe aritmetik ortalama standart sapmadan uzaklaştıkça grafikteki eğilimde farklılaşacaktır. Bu sayede öğrenciler bilgileri yapılandırırken matematiksel kavramların değişimlerini inceleyerek hipotezler oluşturabilirler. Bu hipotezleri direkt olarak test ederek aktif bir öğrenme süreci içerisine girerler.

Şahin ve Namlı'nın 2017 yılında yaptıkları araştırmada; öğrenciler iki gruba ayrılmıştır. Bir gruba bilgisayar destekli eğitim ile algoritma öğretimi diğer gruba ise drama yöntemi ile algoritma öğretimi uygulayarak öğrencilerin problem çözme becerilerini analiz etmişlerdir. Araştırmanın sonunda iki grubunda problem çözme becerilerinin arttığını fakat bilgisayar destekli eğitim alan öğrencilerin daha fazla problem çözme becerilerini arttığı gözlemlenmiştir. Bunun sebebi olarak da bilgisayar destekli eğitim alan öğrencilerin algoritma öğreniminde kullanılan bilgisayar programlarının neden-sonuç ilişkisi kurma olanağı verdiği belirtilmiştir.

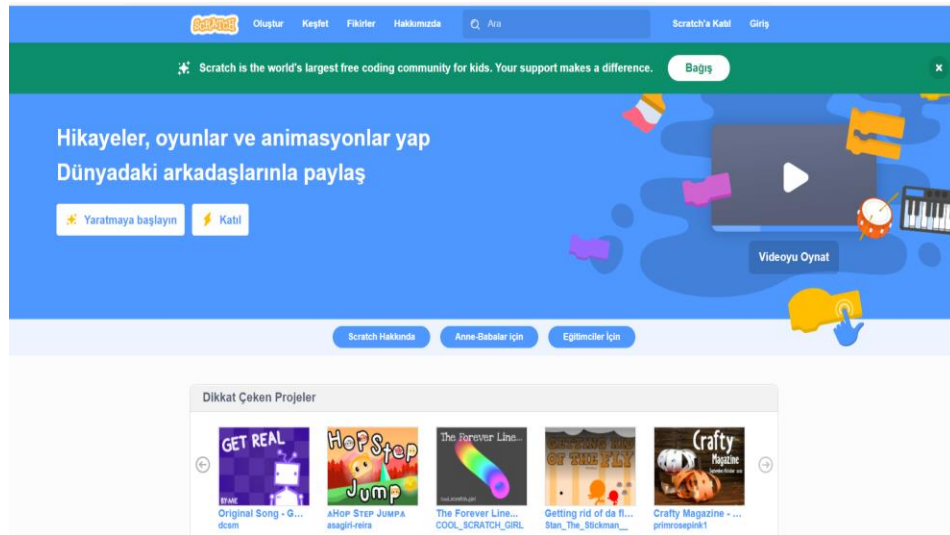
2.3.2. Scratch Programı

Bilgisayar destekli matematik öğretim araçlarından bir tanesi de Scratch programıdır. Scratch programı aslında bilişim teknolojileri dersi için bir kodlama programı olarak ortaya çıkmıştır. Ancak algoritma düzeninde olması ve içeriğinin matematiğe uygun olması sebebiyle matematik eğitiminde de kullanılan bir programdır. Ayrıca akademik araştırmalara da konu olmaktadır. Scratch programı basit kodlama programı olmanın dışında müzik, resim, oyun, video, animasyon gibi projeler oluşturması bakımından bilişim teknolojileri dersi dışında diğer derslerde de rahatlıkla yer alabilir. Algoritma ve programlama öğretiminin haricinde, matematik, fen bilimleri, yabancı dil, sosyal bilgiler gibi derslerde Scratch programı kullanılmaktadır. Bu sayede konular eğlenceli ve oyun oynatarak öğretilir (Oluk vd., 2018).

Millî Eğitim Bakanlığı, 2013-2014 eğitim-öğretim yılında Bilişim Teknolojileri ve Yazılım dersini 5 ve 6. sınıflar için haftalık iki saatlik zorunlu ders kapsamına almıştır. Hazırlanan Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı'nın özel amaçları arasında “Problem çözme ve bilgi-işlemsel düşünme becerileri edinmelerini ve geliştirmelerini, algoritma tasarımına ilişkin anlayış geliştirerek sözel ve görsel olarak ifade edebilmelerini, problemleri çözmek için uygun programlama yaklaşımını seçerek uygulayabilmelerini amaçlamaktadır” maddeleri bulunmaktadır (MEB, 2018).

Ücretsiz etkileşimli grafik programlama aracı olması, geniş bir kullanıcı tarafından kullanılması, çok sayıda dil desteği -özellikle Türkçe- ve anlaşılır arayüzü ile farklı sınıf düzeylerine hitap ediyor olması gibi özelliklerinden dolayı Millî Eğitim Bakanlığı 5 ve 6. sınıflarda bilişim dersinde programlama için Scratch programının kullanılmasını önermiştir. (<http://scratch.mit.edu/> ;<http://scratch.eba.gov.tr/>).

Scratch programı, Massachusetts Institute of Technology (MIT) tarafından üretilmiş ve geliştirilmiştir. Scratch programının sloganı hayal et-üret-paylaş olarak belirlenmiştir. Kişiler oluşturdukları projeleri kolaylıkla paylaşabilir veya diğer kişilerin paylaştığı projeleri kolaylıkla inceleyebilirler. Bu durum kişileri işbirliğine özendirilmektedir. Kodlama programları genellikle metin tabanlı olmasına karşın Scratch, görsel araçları sürükle ve bırak yöntemi ile çalışan bir programdır. Scratch programına “scratch.mit.edu” web sitesinden ulaşılabilir.



Şekil 2.6. Scratch Programı Web Sitesi

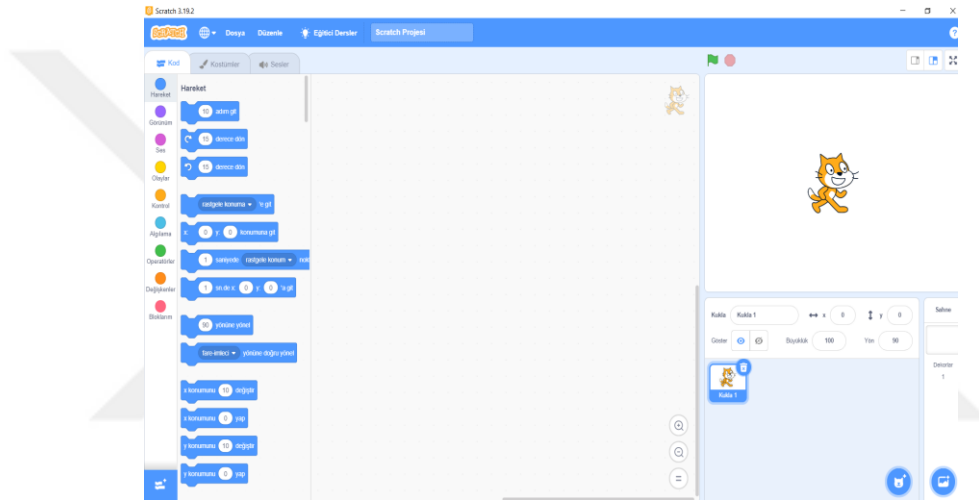
Scratch programının ara yüzünün kolay anlaşılabilir olması, kişilerin ilgilerini çekmesi ve ücretsiz olması birçok kullanıcı tarafından kullanılmasına sebep olmaktadır. Scratch

programı 150’den fazla ülkede ve 60’tan fazla dilde kullanılabilmektedir (Scratch About, 2021).

Scratch programı daha çok yaşları 8 ile 16 yaş arasında değişen çocuk ve gençler için hazırlanmıştır. Ancak zamanla her yaştan kişilerin kullandığı bir program olmuştur.

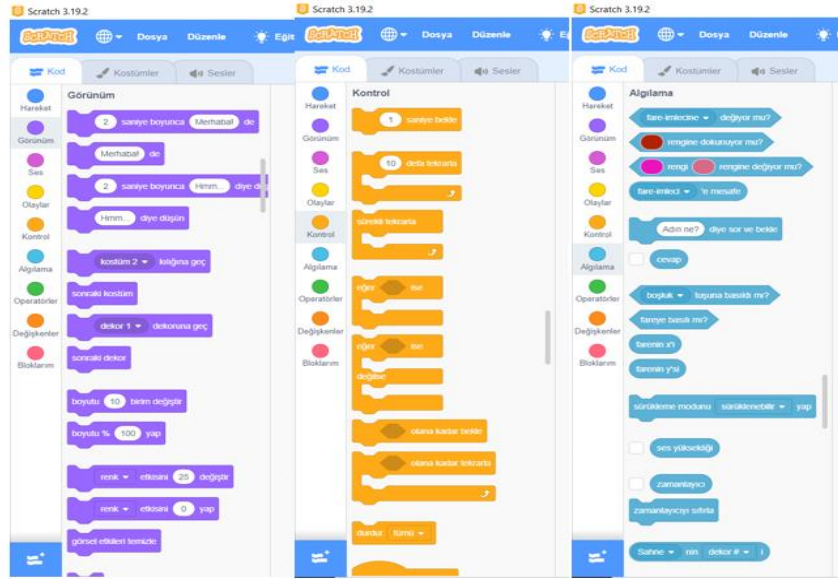
Scratch programının ara yüzü, diğer programlama dillerinden oldukça farklı olarak görselliği ön plana çıkarmıştır. Bu sayede eğlenceli projelerde çalışılması tanınırlığını artırmıştır.

Bu çalışmada “Scratch 3.19.2 Offline Editor” uygulaması kullanılmıştır. Bu uygulama kendi sitesinden kolaylıkla indirilebilir ve çevrimdışı çalışılabilir bir programdır. Programa ilişkin görsel Şekil 2.7.’de verilmiştir.



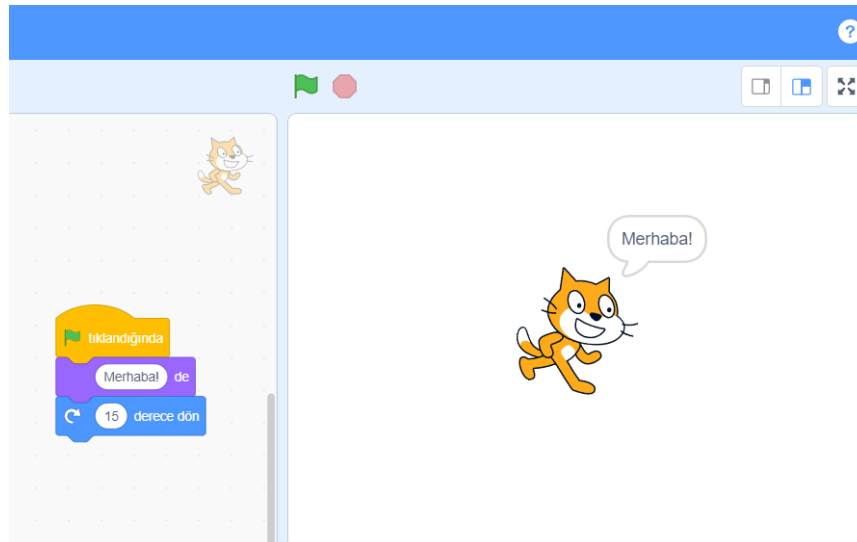
Şekil 2.7. Scratch 3.19.2 Offline Editor Uygulaması

Scratch programı, sürükle bırak özelliği ile birçok kullanıcının rahatlıkla kodlama yapabilmesine fırsat tanımaktadır (Resnick vd., 2009). Programın içeriğinde karakterler, karakterlerin canlandırılabilceği sahne, kod blokları ve kod blokların sürüklenebileceği alan vardır. Programı ilk açtığımızda karşımıza Şekil 7’deki arayüz karşımıza çıkmaktadır. Kod bölümünde ise hareket, görünüm, ses, olaylar, kontrol, algılama, operatörler, değişkenler ve bloklarım menüleri vardır. Her menü farklı renkten oluşmaktadır. Bu renkler hangi kodu hangi bölümden aldığımızı belli etmek için tasarlanmıştır. Scratch programı kod seçeneklerinden bazıları Şekil 2.8.’de gösterilmiştir.



Şekil 2.8. Scratch Programı Kod Seçeneklerinden Bazıları

Şekil 2.8.’deki görünüm menüsü karakterin görüntüsünü değiştirmek için kodlar içeren bölümdür. Hareket menüsüne tıkladığımızda karakterleri çeşitli yönlerle hareket ettirmemizi sağlayan kod blokları vardır. Ses menüsünde karaktere ses efekti verebileceğimiz bloklar yer almaktadır. Bu blokları birleştirebilmek için fare ile blokları ortada ki boş kısma sürüklemeliyiz. Bloklar tıpkı bir yapbozun parçaları gibi alt alta birleştirilmelidir. Bu durumun örneği Şekil 2.9.’da verilmiştir.

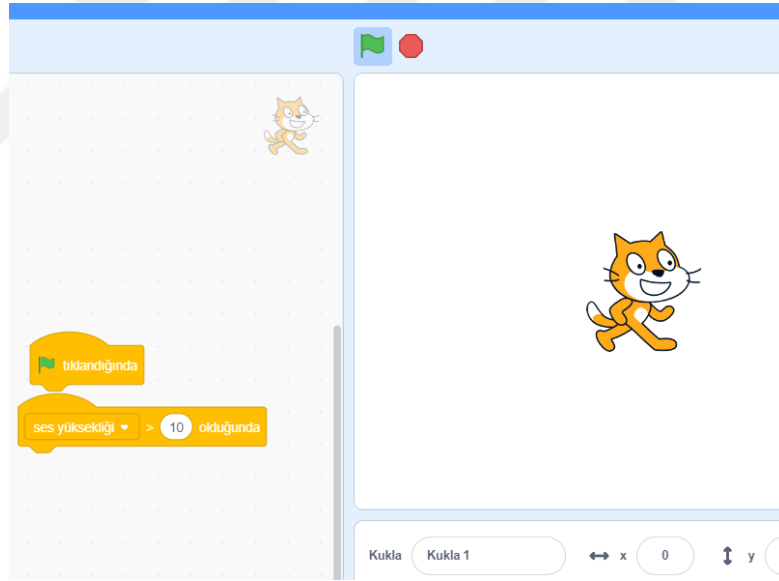


Şekil 2.9. Scratch Programı Örnek Kod Bloğu

Programa yazılan kodlar sahnede ki karakter tarafından canlandırılmaktadır. Kullanıcılar böylece kodların mantığını canlandırmalar sayesinde daha iyi anlayacaktır. Şekil 2.9.'da gösterilen proje ekranı bölümünde ki sağ üstte gösterilen yeşil bayrak yapılan kodlama sonucu oluşan projenin başlatılmasını sağlamaktadır. Kırmızı daire ise projeyi durdurmakla görevli butondur.

Şekil 2.9.'daki kod bloğunu çalıştırmak için yeşil bayrağa tıkladığımızda karakter bir konuşma balonu içerisinde önce “Merhaba” demiş ve kendini saat yönünde 15 derece döndürmüştür. Bu örnekten anlaşılacağı üzere karakterimize hareket bloklarıyla görevler verebilir, onu konuşturabilir ve hatta matematiksel işlemleri bile yaptırabiliriz.

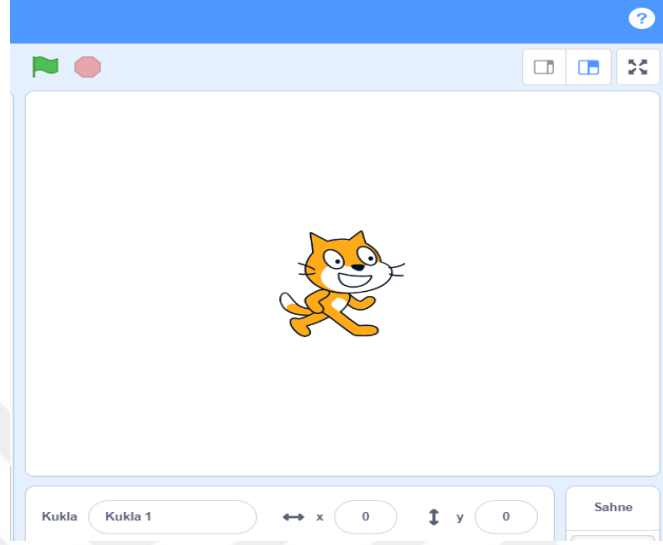
Karakterin yapacağı işlemler sıralı şekilde olmalıdır. Yapılan yanlış ve mantıksız birleştirmelere program izin vermemektedir. Burada programın bize izin vermediği blokları birleştiremeyeceğimizi, blokların yapboz gibi birleşmemesinden anlayabiliriz. Bu durum, kullanıcının daha az hata yapıp programı kolay öğrenmesine neden olmaktadır. Yapılan hatalı kodlama Şekil 2.10.'da verilmiştir.



Şekil 2.10. Hatalı Yapılan Scratch Programı Örnek Kod Bloğu (Bloklar Birbirlerine Eklenmiyor.)

2.3.3. Scratch Programının Matematiksel Yönü

Scratch programı ekranında karakterin olduğu sahne aslında bir koordinat düzleminden oluşmaktadır. İlk başta karakterimiz $(x,y)=(0,0)$ noktasında durmaktadır (Külekci vd., 2013). Karakterimizin $(0,0)$ noktasında olduğu durum Şekil 2.11.'de verilmiştir.

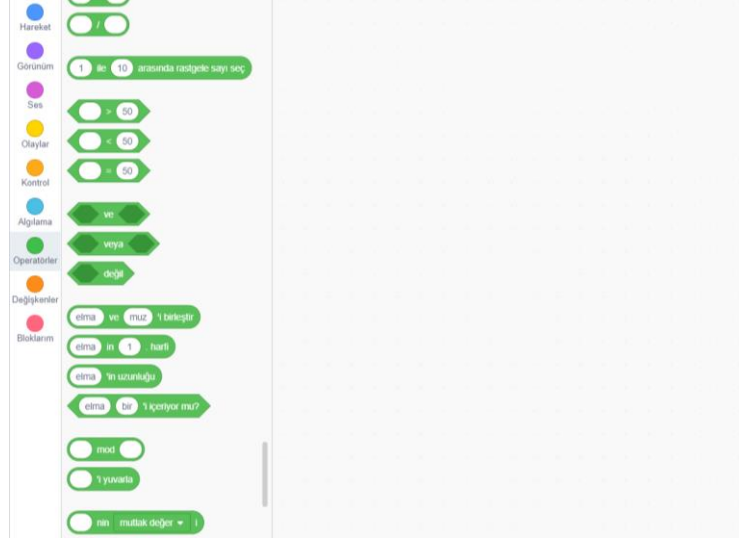


Şekil 2.11. Karakterin Sahnede $(x, y) = (0,0)$ Noktasında Durduğu An

Öğrencinin Şekil 11'deki karakteri istediği konuma getirebilmek için x ve y değişkenlerini değiştirmesi yeterlidir. Örneğin y değerine negatif bir tam sayı yazdığında karakter aşağı doğru hareket ediyorken pozitif değer yazdığında ise yukarı doğru hareket edecektir. Burada öğrencinin tam sayıları sayı doğrusunda gösterme ve koordinat düzlemi konularını basitçe bilmesi yeterlidir.

Scratch 3.19.2 programına giriş yapıp sağ tarafa baktığımızda ise operatörler kısmında matematiksel işlemleri içeren yuvarlama, mutlak değer, mod, rastgele sayı seçme ve mantık gibi birçok bloğa yer verilmiştir. Programın bu bölümü Şekil 2.12.'de gösterilmiştir.

Öğrenciler operatörler kısmını kurcalayarak bile koordinat düzlemini, değişkenleri ya da rasgele sayılar gibi önemli matematiksel kavramları öğrenebilirler (Quinn, 2012). Öğrenci bu bölümü kullanarak iki sayıyı karşılaştırabilir. Scratch programı üzerinden bir hesap makinesi oluşturup dört işlem gerektiren problemleri de öğrenci çok rahatlıkla çözebilir. Bunun için yine operatörler kısmındaki $+$, $-$, $/$, $\times$ işaretlerinden yararlanabilir.



Şekil 2.12. Scratch Programı Operatörler Bölümünün Bir Kısım

Scratch programı, dünya genelinde matematik öğretmenleri tarafından kullanılan bir programdır. Bunun için Scratch programının matematik eğitiminde kullanılmasına yönelik çalışmalar yapan bir teknik ekip bulunmaktadır (Scratch About, 2015).

2.3.4. Scratch Programı ile İlgili Çalışmalar

Bu bölümde Scratch programının araştırmanın konusu ile doğrudan ya da yakından ilişkili olduğu araştırmalara yer verilmiştir.

Calao ve diğerleri (2015), 6. Sınıfta öğrenim görmekte olan kontrol grubuna geleneksel matematik öğretimiyle, deney grubuna ise Scratch programıyla sayısal becerilerini kullanmalarını sağlayarak 3 ay süren bir çalışma yapmışlardır. Bu çalışma sonucunda deney grubunun modelleme, muhakeme, problem çözme ve uygulama gibi temel matematik düşünme becerilerinin önemli ölçüde arttığını gözlemlemişlerdir. Scratch programı; öğrencilere problem çözme, özgün projeler üretme ve kendini ifade etme gibi üst düzey beceriler kazandırmaktadır (Sarioğlu, & Kartal, 2017).

Scratch Programı bilgisayarca düşünme ile yakından ilgilidir. Wing, (2006) günümüzde bilgisayarca düşünmenin önemine vurgu yaparak bilgisayarca düşünme; okuma, yazma ve matematik gibi temel yetenekler arasında olacağını ve birbirlerini etkileyeceklerini ifade etmiştir. Barr, Harrison ve Conery (2011) bilgisayarca düşünme becerilerinin genel özelliklerini şöyle ifade etmektedirler: problemi, bilgisayar ve diğer araçların yardımcı olabileceği şekilde formüleleştirme, verileri mantıklı bir şekilde düzenleme ve analiz etme,

soyutlama yoluyla verileri sunma, algoritmik düşünme yardımıyla çözümleri otomatikleştirme, belirleme, analiz etme, amaca ulaşırken en etkili, en verimli aşamalar ve kaynaklar yardımıyla olası çözümleri uygulama, problem çözme sürecini problem çeşitliliğine dönüştürme ve yaygınlaştırma.

Maloney ve diğerleri (2010) Scratch programını 9-17 yaşları arasındaki öğrenciler için uygun olduğunu belirtmişlerdir. Bunun en büyük sebebini ise öğrencilerin proje üretirlerken sanki oyun veya animasyon ile uğraştıkları hissinde olduklarına bağlamışlardır. Scratch programı ara yüzü ve dil desteği sayesinde ülkemiz Bilişim Teknolojileri ve Yazılım derslerinde kullanılmaktadır (Karabak & Güneş, 2013.).

Scratch programı, belirli bir matematiksel fikir kümesini kolaylaştırmak için özel olarak tasarlanmamıştır, ancak çocuklar etkinlik oluştururken önemli matematiksel fikirleri bu program sayesinde öğrenebilirler. Aynı zamanda kod bloklarını oluşturmada kullanılan matematiksel kavramları anlayabilirler. Lewin, Somekh ve Steadman (2008) yaptıkları çalışmada öğrencileri Scratch programı gibi yaratıcı bir çevreye yerleştirdiklerinde oldukça karmaşık matematiksel bir programlamayı keşfedebileceklerini ve kullanabileceklerini ortaya çıkarmışlardır. Öğretmenler, bu çalışma için Scratch programının çocuklara problem çözme ve bazı matematiksel fikirleri algoritmik bir formatta özgürce keşfedebilme fırsatları sunduğunu belirtmişlerdir. Yine bu çalışmada “Düşük” matematiksel bir anlayışa sahip olduğu tespit edilen çocukların karmaşık bilgisayarca düşünme becerilerini geliştirebildiği ve karmaşık matematiksel fikirleri kullanabildiği görülmüştür. Araştırmanın bir başka sonucunda Scratch programı çocukların problem çözme stratejilerini geliştirmelerine, yaratıcı düşünmelerine, sistematik bir şekilde düşünmelerine ve iş birliğine dayalı çalışmalarına yardımcı olmuş gibi görünmektedir. Ayrıca bu çalışmayı yürüten öğretmenler, öğrencilerde olumlu sosyal değişimler olduğunu ifade etmişlerdir.

Foerster (2016), 6.sınıfta bilgisayar programlarının -özellikle Scratch programının- kullanılmasıyla öğrencilerin üçgen, kare ve altıgen oluşturarak geometrik becerilerinin geliştirdiğini ortaya koymuştur. Marques ve Marques (2012), günümüzde problem çözme ve eleştirel düşünmenin önemine vurgu yapmışlar ve bu becerilerin Scratch kullanımı ile geliştiğini öne sürmüşlerdir. Wilson ve Moffat (2012), Permatasari, Yuana ve Maryono (2018) ise 8-9 yaşındaki öğrenciler üzerinde yaptıkları araştırmalarda Scratch kullanımının öğrenciler için eğlenceli bir deneyim olduğunu ve öğrencilerin Scratch üzerinde proje oluşturmak için hevesli olduklarını belirtmişlerdir. Förster ve diğerleri (2018), Alman öğrenciler üzerinde yaptıkları araştırmada 6.sınıf öğrencilerine Scratch programı üzerinde

geometrik şekiller yardımıyla projeler üretmelerini istemişlerdir. Araştırma sonucunda öğrencilerin geometrik becerilerinin ve sezgisel becerilerinin arttığını belirlemişlerdir.

Marimuthu ve Govender (2018), Güney Afrika’da 10 ve 11. sınıflara yönelik yaptıkları araştırmada Scratch programı kullanımının öğrenciler tarafından mantıklı düşünme ve problem çözme gibi matematiksel beceri gerektiren becerileri geliştirdiğini belirtmişlerdir. Ayrıca araştırmacılar Scratch programı kullanımına, ara yüzünün kolay olmasından dolayı daha erken yaşlarda başlanılmasının sakıncalı olmadığını ifade etmişlerdir.

İngiltere 2014 yılında öğretim müfredatında değişikliğe gitmiş ve kodlama konusuna derslerde daha fazla yer vermiştir. Bu kapsamda yapılan araştırmada, 55 okul ile iletişime geçilmiş, öğretmenlere kodlamanın önemi anlatılmış ve Scratch programı üzerinden kodlama ile ilgili eğitici hizmetiçi çalışmalar yapılmıştır. Buradaki amaç öğretmenlerin, pedagojik fikirlerinin gelişmesi olarak açıklanmıştır (Wilson vd., 2019). Daha sonra öğrenciler ile çalışma yapıp geri dönüş vermeleri istenmiştir. Buradaki amaç, öğrencilere (9-11 yaş) matematiksel kavramları ve matematiksel akıl yürütebilmeleri için temel hesaplama fikirlerini Scratch programı ile öğretmek olmuştur (TheScratchMaths (SM) Projects). Araştırma sonuçları ise beklendiği gibi olmamıştır. 55 okuldan 28’i cevap vermiş ve kalan 27 okuldan net bir veri akışı sağlanamamıştır. Bunun en önemli nedenleri; sürekli yaşanan öğretmen değişiklikleri ve ulusal düzeyde yapılan sınavlardan dolayı öğrencilerin daha çok sınavlara yönelik çalışma yapmaları olarak belirlenmiştir. Gelen kısıtlı veriye bakıldığında ise, proje ilerledikçe derslerin daha akıcı ilerlediği ve öğrencilerin matematiksel keşif özelliklerinin arttığı gözlemlenmiştir.

Bu araştırmada son yıllarda birçok araştırmaya konu olan Scratch programı kullanılmıştır. Literatür incelendiğinde Scratch programı ile yapılan araştırmalarda genellikle matematik öğretimini destekleyici bir öğretim aracı olarak kullanıldığı görülmektedir. Bu araştırmaların çalışma grubunu ise genellikle bir veya birden fazla sınıftan oluşan öğrenciler oluşturmuştur. Scratch programı ile yapılan araştırmalar incelendiğinde, matematiğin farklı kavramlarına odaklanan ve daha küçük gruplar ile yapılan araştırmalara çok az rastlanmıştır (Özkan, 2020). Scratch programında pozitif tam sayılar ile algoritmalar üretilmesini amaçlayan araştırmalara ise rastlanmamıştır. Bu araştırmada Scratch programında üç kişilik bir grup ile pozitif tam sayılar konusunda matematiksel algoritma üretilmesi hedeflenmektedir. Bu haliyle bu araştırmanın, ilgili çalışmalardan ayrıştığı ve literatüre katkı sağlayacağı düşünülmüştür.

Gündelik yaşamımızda gelişen, değişen durumlara ayak uydurabilmek için bireylerin; kendi öğrenme sorumluluğunu üstlenerek öğrenmeyi öğrenen, eleştirel, yaratıcı ve esnek

düşünebilen, teknolojiyi gerektiği yerde doğru kullanabilen, duygu ve fikirlerini açıkça ve duru bir şekilde ifade edebilen, grup çalışmaları yapabilen, problem çözebilen ve bilgiyi üretebilen özelliklere sahip olmaları gerekir (Şen & Erişen, 2002). Bu süreci inceleyebilmek için öğrencilerin somuttan soyut döneme geçiş döneminde, kodlamaya giriş derslerinde sıklıkla kullanılan Scratch Programı kullanılmıştır. Teknolojik araçlarda kodlama ve programlama algoritma ile yakından ilgilidir (Demir & Cevahir, 2020). Scratch programı algoritma mantığı ile kurgulanmış bir programdır. Ayrıca Scratch programı üzerinde yapılan çalışmalarda kodlamalar, yazılan yanlış bir adım da çalışmamaktadır. Yapılan çalışmada sadece sonuç değil süreç ve yazılan her bir blokta önemlidir. Bu gibi özellikler algoritmalar oluşturulurken de kullanılmaktadır (Aytekin vd. 2018). Bu da Scratch programı ile algoritmalar arasında sıkı bir ilişki olduğunu göstermektedir.

İlgili araştırmalar incelendiğinde son yıllarda algoritma ile ilgili çalışmaların arttığı görülmektedir (Lagrange & Laval, 2019; Moala, 2019). Algoritma alanında yapılan çalışmalar oldukça geniş yelpazede yapıldığını söylemek mümkündür. Bu çalışmalarda farklı seviyelerde öğrencilerle çalışıldığı görülmektedir. Buna karşın algoritma üretme sürecinde kullanılan Scratch programı ile yapılan araştırma sayısı sınırlıdır (Çatlak vd., 2015). Scratch ile yapılan araştırmalar genellikle bütün sınıf ve okuldaki öğrencilere uygulanmış, özel gruplara etkisi incelenmemiştir. Küçük bir grubun algoritma geliştirme süreçlerinin teknoloji desteğiyle nasıl ortaya çıktığını incelemenin onların nasıl algoritmalar geliştirdiklerini ve bu süreçte teknolojinin nasıl kullanıldığını anlamak mümkün olacaktır. Yukarıdaki ilgili literatür incelendiğinde Scratch programı ile ilgili çalışmaların 2010 yılından sonra bir artış gösterdiği görülmüştür. Bu araştırmalar sonucunda Scratch programının öğrencilere iyi bir öğrenme ortamı sunduğu ve öğrencilerin matematiksel becerilerini geliştirdiği anlaşılmıştır.

Bu araştırmanın amacı akademik başarısı yüksek ortaokul öğrencilerinin pozitif tam sayılar konusunda Scratch programı ile algoritma üretme süreçlerini ve oluşan algoritmaların yapısını incelemektir. Bu amaç doğrultusunda şu sorulara cevap aranmıştır;

- 1) Öğrencilerin pozitif tam sayılar konusunda Scratch programı ile algoritma üretme süreçleri nasıldır?
- 2) Öğrencilerin pozitif tam sayılar konusunda Scratch programı ile oluşturdukları algoritmaların yapısı nasıldır?

BÖLÜM III: YÖNTEM

Bu bölümde araştırmanın deseni, çalışma grubu, veri toplama araçları, verilerin toplanması ve verilerin analizi başlıklarına yer verilmiştir.

3.1. Araştırmanın Deseni

Bu araştırmada tam sayılar konusunda Scratch programı kullanılarak, öğrencilerin algoritma üretme süreçleri ve süreç sonunda ortaya çıkan algoritma yapılarının incelenmesi amaçlanmıştır. Algoritma üretme sürecinin ve bu algoritma yapılarının detaylı bir analizi yapılması gerektiğinden dolayı bu çalışmada, nitel araştırma yaklaşımı benimsenmiştir. Nitel araştırma yaklaşımı; gözlem, görüşme ve doküman analizi gibi nitel veri toplama yöntemlerinin kullanıldığı, algıların ve olayların doğal ortamda gerçekçi ve bütüncül bir biçimde ortaya konmasına yönelik nitel bir sürecin izlendiği araştırma yaklaşımı olarak tanımlanmaktadır (Şimşek & Yıldırım 2013).

Bu araştırmada 6. sınıf öğrencilerinin algoritma üretme süreçlerinin ve oluşturulan algoritma yapılarının detaylı incelenmesi amaçlandığından “durum çalışması” yaklaşımı kullanılmıştır. Durum çalışması, sınırlı bir durumun nasıl işlediği ve nasıl çalıştığı hakkında düzenli ve sıralı bilgi toplamak için çoklu veri toplama kullanılarak o sistemin derinlemesine incelenmesini içeren yöntembilimsel bir yaklaşımdır (Chmiliar, 2010). Durum çalışması, sonuçtan çok sürece ağırlık vermekte ve olayı derinlemesine inceleme fırsatı sunmaktadır. Durum çalışması, eğitimin çeşitli konularını anlamada, özellikle ne, nasıl ve niçin soruları yöneltildiğinde kullanılabilen bir yöntemdir. Bu araştırmada durum çalışmasının seçilme sebebi, nasıl ve niçin sorularını temel alan, araştırmacının kontrol edemediği bir olgu ya da olayı en ince ayrıntısına kadar incelemesine olanak sağlayan araştırma yöntemi olmasıdır (Yıldırım & Şimşek, 2011). Bu kapsamda bu araştırmada ele alınan durum, 6. sınıf öğrencilerinin Scratch programı ile algoritma oluşturma süreçleri ve ürettikleri algoritmaların yapısıdır. Durum çalışmasında, katılımcıların sorulara kendi cümleleriyle yanıt vermelerine olanak tanımakla beraber verilen yanıtlar sonucunda “nasıl ve niçin” gibi sorular da anında sorulabilmektedir. Bu sebeplerden dolayı araştırmada durum çalışması modeline göre verilerin toplanması, çözümlenmesi ve analiz edilmesi söz konusudur.

3.2. Araştırmacının Rolü

Bu çalışmada, araştırmacı etkinlikler sırasında gözlemci rolünü üstlenmiştir. Algoritma üretme süreçlerinin ve oluşturulan algoritmaların yapısının incelenmesinde ise değerlendirici rolündedir. Araştırmacı, etkinlikler sırasında öğrencilerin algoritma üretme çalışmalarına herhangi bir müdahalede bulunmayıp sadece onların yaptıkları eylemlerin sebeplerini öğrenmeye çalışmıştır. Bu nedenle öğrencilere uygulama esnasında çalışma ile ilgili olduğunu düşündüğü yerlerde daha fazla veri için sorular yöneltmiştir. Araştırmacı tarafından öğrencilere sınırlı katkı sağlanmıştır.

3.3. Çalışma Grubu

Bu araştırmanın çalışma grubunu, araştırma problemlerine uygun olarak 2020-2021 eğitim-öğretim yılında İstanbul ilinin Sancaktepe ilçesine bağlı bir özel okulda öğrenim görmekte olan üç öğrenci oluşturmaktadır. 6. sınıfa devam eden bu öğrenciler, İstanbul İl Milli Eğitim Müdürlüğü'nün düzenlediği “Temel Fikirlerden Temel Bilimlere Bilim Olimpiyatları Projesi” kapsamında matematik olimpiyatları eğitimi alan öğrenci grubundan seçilmiştir. Katılımcılar, amaçlı örneklem yöntemi türlerinden kolay ulaşılabilir örneklem yöntemi esas alınarak belirlenmiştir. Kolay ulaşılabilir örneklem; araştırmacıya göre ulaşılması hızlı, kolay ve maliyet bakımından ekonomik olan katılımcıları seçmesine dayanmaktadır (Patton, 2018). Genellikle araştırmacıların, kullanacakları çalışma grubu büyüklüğünün genelleme yapmaya olanak sağlamayacağı, sadece ulaşılması kolay ve çalışılması pahalı olmayan durumları seçtikleri bilinmektedir (Vogt vd., 2012). Kolay ulaşılabilir örneklem bu sebeplerden dolayı eleştirilmektedir. Fakat ulaşılması kolay ve maliyetin düşük olması her zaman elde edilecek verinin yetersiz olacağı anlamına gelmemektedir. Nitekim bu çalışmada seçilen katılımcılar araştırmacının kolay ulaşabileceği öğrencilerden seçilmiştir. Ancak araştırmanın amacına en uygun çalışma grubu, akademik başarısı yüksek olan ve kendini iyi ifade edebilen öğrenciler olması gerektiği belirlenmiştir. Bu öğrencilerin akademik başarıları okul karne not ortalamalarının yüksek olması ile belirlenmiştir. Ayrıca öğrenciler İstanbul İl Milli Eğitim Müdürlüğü'nün düzenlemiş olduğu İstanbul Bilim Olimpiyatları sınavında Sancaktepe ilçesine bağlı bulunan tüm 6. sınıflar arasından ilk 10'a girmişlerdir. Bu durum öğrencilerin matematik dersinde akademik başarılarının yüksek olduğunun bir diğer göstergesidir. Öğrencilerin iletişim becerilerinin iyi olup olmadığı araştırmacı tarafından gözlemlenmiş ve aynı zamanda sınıf öğretmenlerinden görüş alınmıştır. Ayrıca

öğrencilerin akademik başarılarının yüksek ve iletişim becerilerinin iyi olmasının öğrencilerden zengin veri elde etmeye imkân sağlayacağı düşünülmüştür. Çalışma grubu belirlendikten sonra yapılacak uygulamalar kapsamında öğrencilere bilgi verilmiş ve gönüllülük esasına göre öğrenciler çalışmaya katılmışlardır.

3.4. Veri Toplama Araçları

Nitel araştırmalarda veriler toplanırken birden fazla veri toplama yönteminin kullanılması verilerin güvenilirliğinin ve geçerliliğinin sağlanmasında etkili olmaktadır (Yıldırım ve Şimşek, 2013). Bu araştırmanın veri kaynağı öğrencilerle yapılan sesli düşünme protokolü görüşmeleri ve öğrencilerin oluşturduğu Scratch programındaki algoritmalarıdır. Öğrencilere süreç içerisinde kullanabilecekleri mod konusu anlatılmıştır. Bu konu hakkında daha sonra bir uygulama yapılmıştır (EK 3). Öğrencilerin Scratch programı üzerinden yaptıkları etkinlikler süresince işlemler, ekran kaydı ile kayıt altına alınmıştır. Bu şekilde oluşturulan algoritma yapıları ve algoritma oluşturma süreçleri ayrıntılı olarak incelenmiştir. Kullanılan veri toplama araçları aşağıda ayrıntılı olarak açıklanmıştır.

3.4.1. Sesli-Düşünme Protokolü

Bu araştırmada verileri derinlemesine incelemek için sesli düşünme protokolü görüşmeleri uygulanmıştır. Sesli düşünme protokolleri (think aloud protocol) çevrimiçi izleme yöntemlerindendir. Bu yöntemle, bilişsel süreçlere dayalı bilgiler uygulama esnasında toplanmaktadır. Sesli düşünme protokollerinde, bireylerin bir etkinlik üzerinde çalışırken fikirlerini sözlü olarak ifade etmeleri istenmektedir. Sözlü olarak belirtilen fikirler sistematik olarak kaydedilir ve değerlendirilir. Bu şekilde sesli düşünme protokolleri ile bireylerin çalışma esnasındaki duygu ve düşüncelerinin öğrenilmesi amaçlanmaktadır. Sesli düşünme protokollerinin temel stratejisi, öğrencilerin etkinlikleri yaparken düşündükleri ve yaptıkları her adımı sesli bir şekilde söylemeleri olarak ifade edilmektedir (Patton, 2018).

Sesli düşünme protokolü düşünme süreçlerinin dışa vurma tekniğidir. Düşünen birey, bir çalışma esnasında ortaya çıkan tüm düşüncelerini veya duygularını sesli olarak ifade eder. Araştırmacı ise etkinliği yapan bireyin söylediklerine aktif olarak katılır, doğruluğu inceler, hataları işaret eder ve düşünen kişinin sesli olarak düşüncelerini ifade etmesini sağlar.

Bu arařtırmada, uygulama srecindeki verileri toplayabilmek amacıyla sesli dřnme protokolleri yntemi kullanılmıřtır. Bu amacı gerekleřtirebilmek iin arařtırmacı, ğrencilere;

- **Uygulama bařlamadan nce:** *“Etkinliėin her bir adımımda duygu ve dřncelerinizi sesli olarak ifade edecek řekilde bařlayabilirsiniz.”*
- **Uygulama sırasında:** *“Yaptıėınız son adımın aıklamasını yapınız.”*
- **Uygulama sonunda:** *“Matematik aısından sizce bu etkinliklerin nasıl faydası oldu?”*

İfadelerini ynelterek ğrencilerin algoritma retme srelerini uygulama sırasında sergilemelerini saėlamaya alıřmıřtır. Uygulama esnasında da iki ğrenci her yaptıkları adımı sesli bir řekilde ifade etmiřlerdir. Diėer ğrenci ise etkinliklerde daha ok sessiz kalmıřtır. Arařtırmacı bu kiřiye sorular sorarak etkinlikte sesli dřnmesini saėlamaya alıřmıřtır.

3.4.2. Scratch Programı Etkinlikleri

Arařtırmada kullanılan pozitif tam sayılar konusu ile ilgili Scratch programı etkinliklerinin amacı, ğrencilerin bu konudaki algoritma retme srelerinin algoritma yapılarının incelenmesidir.

Pozitif tamsayılar konusu ile ilgili etkinlikler arařtırmacı ve matematik ğretimi alanında uzman bir ğretim yesi tarafından hazırlanmıřtır. Etkinlikler oluřturulurken 6. sınıf matematik konuları ve kazanımları dikkate alınmıřtır. Etkinlikler hazırlandıktan sonra biliřim teknolojileri alanında uzman bir ğretmenden etkinliklerin Scratch programı ile uyumlu olup olmadıėı grřlmřtr. Etkinlikler arařtırmacı, uzman ğretim grevlisi, uzman bilgisayar ve ğretim teknolojileri ğretmeni tarafından ğrencilerin seviyesine uygun grlmřtr.

Bu kapsamda hazırlanan  etkinlik řu řekildedir;

- 1) ğrencilerden, Scratch programına girilen sayının pozitif tam blenlerinin belirlenmesi ve sayının asal olup olmadıėının belirlenmesi ynelik bir algoritma geliřtirilmesi istenmiřtir.
- 2) Arařtırmacı tarafından oluřturulan bir algoritma ğrencilere kod bloklarının olduėu kısım gsterilmeden sadece cevap kısmının olduėu sahne gsterilmiřtir (řekil 3.1.). řekil 3.1.’deki gibi sadece 10 sayısının cevabı verilmemiř, ayrıca 12 ve 15

sayılarının cevaplarının da sırasıyla 78 ve 120 olduğu gösterilmiştir. Öğrencilerden buradaki matematiksel mantığı anlamaları ve buna göre bir algoritma geliştirmeleri istenmiştir.

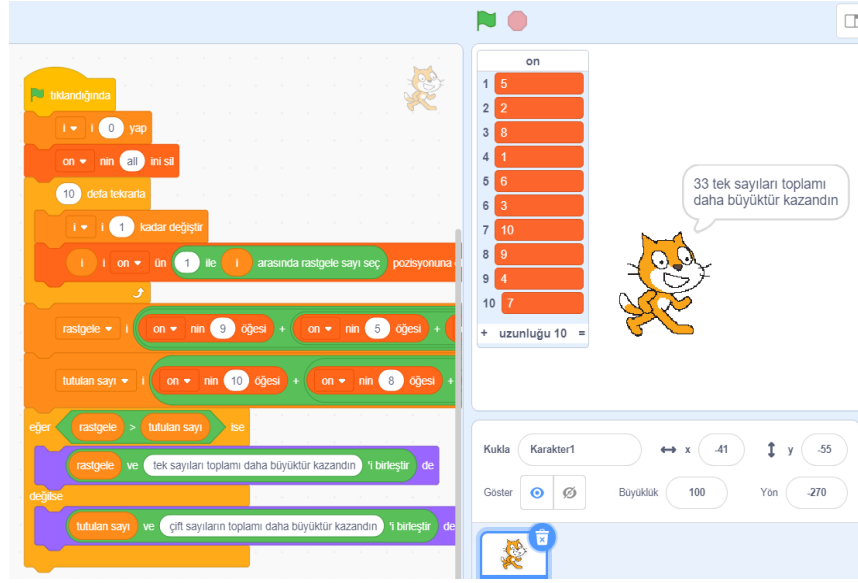


Şekil 3.1. Kod Blok Kısmı Verilmeden Öğrencilere Verilen Scratch Programı Görüntüsü

3) Öğrencilere sözel olarak bir matematiksel oyun anlatılmıştır. Öğrencilerden bu oyunun kesin kazanma stratejisinin belirlenmesi beklenmiştir. Matematiksel oyun kuralları şu şekildedir;

- Oyun iki kişi ile oynanacaktır.
- 1'den 10'a kadar olan sayılar rastgele ve karışık olarak dizilecektir.
- Sırası gelen oyuncu sıranın ya en başından ya da en sonundan bir sayı seçecektir. Aralardan sayı seçemeyecektir.
- İki kişide toplam 5 sayı seçtikten sonra eldeki sayılar toplanacak ve sayılarının toplamı fazla olan kişi oyunu kazanacaktır.

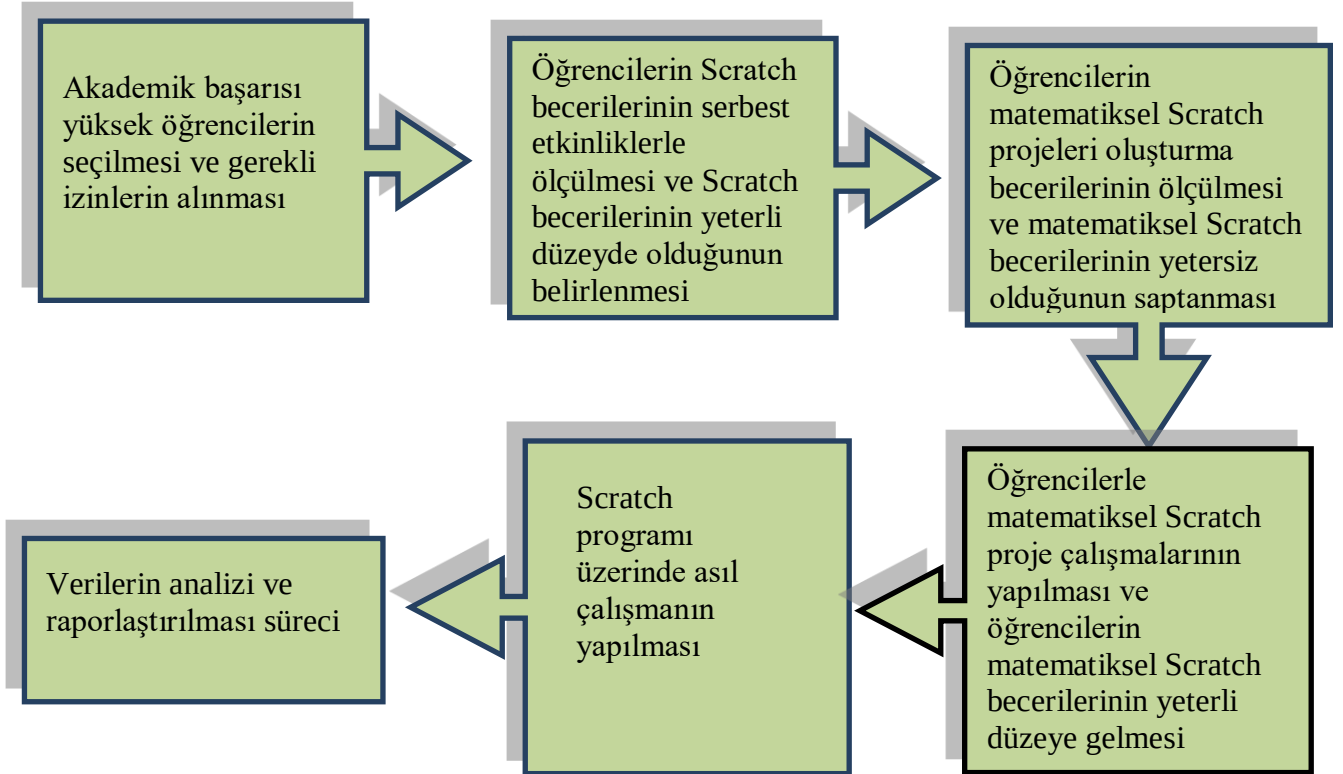
Araştırmacı tarafından belirlenen kesin kazanma stratejisi şu şekildedir; sayılar rastgele ve karışık dizildiğinde tek sıradaki sayıların toplamı ile çift sıradaki sayıların toplamını karşılaştırmalıdır. Hangisi büyükse oradaki sayıları seçmeye çalışmalıdır. Araştırmacının oluşturduğu algoritma Şekil 3.2'de verilmiştir.



Şekil 3.2. Araştırmacın Üçüncü Etkinlik İçin Oluşturduğu Algoritma

Öğrencilerin etkinlik sırasındaki düşüncelerini anlamak için sesli düşünme protokolü görüşme veri aracı olarak kullanılmıştır.

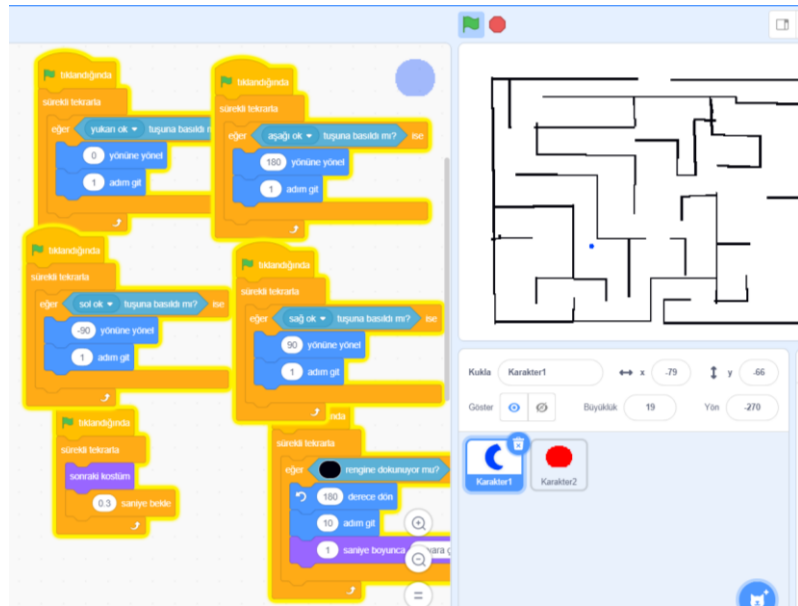
3.5. Verilerin Toplanma Süreci



Şekil 3.3. Verilerin Toplanma Süreci

Araştırma için verileri toplamadan önce Marmara Üniversitesi Eğitim Bilimleri Enstitüsü Araştırma ve Yayın Etik Kurulu Kararına göre araştırma için gerekli izin alınmıştır. Veri toplama sürecinde, araştırmanın çalışma grubunu Millî Eğitim Bakanlığına bağlı bir özel ortaokul öğrencileri oluşturduğundan dolayı ilk olarak gerekli izinler alınmıştır. Marmara Üniversitesi Eğitim Bilimleri Enstitüsü aracılığı ile İstanbul İl Milli Eğitim Müdürlüğü arasında yapılan yazışma sonucunda araştırma ile ilgili izin yazısı EK-3'te verilmiştir. Daha sonra araştırmaya katılması istenen çalışma grubuna araştırma süreci anlatılmıştır. Öğrencilerin velilerine bilgilendirme ve onay almak amaçlı “Veli İzin Formu” (EK-5) ulaştırılmıştır. Ayrıca katılımcı formu (EK-6) öğrencilere verilmiştir. Öğrencilerin araştırmaya katılmaları hususunda onayları alındıktan sonra araştırmanın uygulama aşamasına geçilmiştir.

Araştırma 2020-2021 eğitim-öğretim yılının sömestr tatilinde gerçekleşmiştir. Bunun nedeni öğrencilerin çalışmaya daha verimli vakit ayırabileceklerinin düşünülmesidir. Çalışmaya öğrencilerin Scratch programı hakkındaki bilgilerini tespit etmek amaçlı bir oturum yapılarak başlanmıştır. Bu oturumda öğrencilerden serbest olarak bir Scratch programı çalışması yapmaları istenmiştir. Öğrenciler Şekil 3.4. ve Şekil 3.5.’teki gibi projeler geliştirmişlerdir. Yapılan projeler araştırmacı ile bir uzman tarafından incelenmiş ve öğrencilerin Scratch programı kullanımına yönelik bilgilerinin yeterli düzeyde olduğu sonucuna varılmıştır.



Şekil 3.4. Öğrencilerin Scratch Programı Üzerinde Tasarladığı Packman Projesi



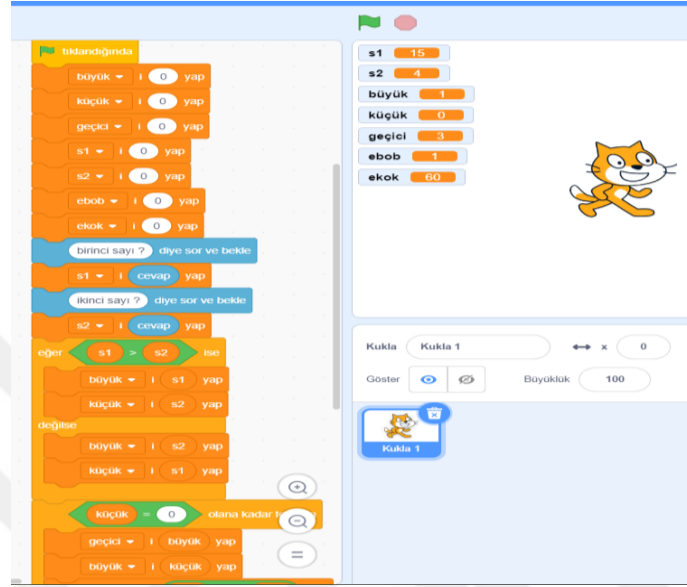
Şekil 3.5. Öğrencilerin Tasarladığı Duvar Oyunu

Daha sonra öğrencilerin matematiksel bir etkinliği Scratch programı üzerinde yapabilmeye becerilerini görebilmek için bir oturum yapılmıştır. Öğrencilerden bu oturumda program üzerinden bir sayının katlarını bulan Scratch algoritması hazırlamaları istenmiştir. Bu çalışmada öğrenciler herhangi bir algoritma tasarlayamamışlardır. Öğrencilerin Scratch programı üzerinde oyun kurma gibi projeler yapabildikleri fakat matematik ile ilgili bir proje üretmedikleri görülmüştür. Bu kapsamda öğrencilere uygulama öncesinde Scratch programı ile matematiksel etkinlik tasarımları için pilot uygulamalar yapılması kararlaştırılmıştır. Pilot çalışma için verilen etkinlikler aşağıda belirtilmiştir.

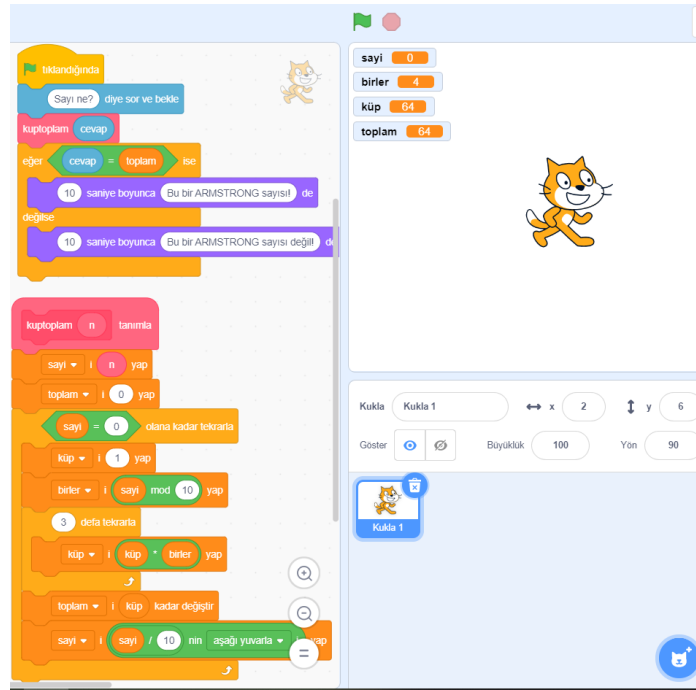
- Verilen iki sayının en büyük ortak bölenini ve en küçük ortak katını bulma
- Bir sayının Armstrong sayısı olup olmadığını belirleme (Sayıyı oluşturan rakamların küpleri toplamı sayıyı veriyorsa bu bir Armstrong sayısıdır.)

Etkinliklerin uygulanması sırasında araştırmacı bilgisayarından ilk adımı oluşturmuş ve bu adımı öğrencilere açıklamıştır. Araştırmacı her adımı oluşturduktan sonra öğrencilerin de adımları oluşturması beklenmiştir. Öğrenciler ilk adımı bitirdiklerinde diğer adıma geçilmiştir. Bu şekilde öğrencilerin yukarıdaki matematik projelerini yaparken matematiksel proje üretme sürecini öğrenmesi hedeflenmiştir. Öğrenciler ile yapılan algoritmalar Şekil 3.6. ve Şekil 3.7.'de verilmiştir. 1. etkinlik 30, 2. etkinlik 37 dakika sürmüştür. Etkinlikler sırasında “modüler aritmetik” konusu kullanılması gereken bir yer ile karşılaşılmış ve öğrencilere bu konu anlatılmıştır. Öğrencilerin anlayıp anlamadıklarını tespit etmek için

küçük bir “modüler aritmetik sınavı” yapılmıştır (EK-2). Sınavın sonuçlarına göre öğrencilerin mod kavramını anladıkları sonucuna ulaşılmıştır. Bu etkinlikler sonucunda öğrencilerin matematiksel proje oluşturma konusunda yeterli bilgiye ulaştıkları araştırmacı, matematik öğretimi alanında uzman bir öğretim görevlisi ve bilişim teknolojileri alanında uzman bir öğretmen ile belirlenmiştir.



Şekil 3.6. İki Sayının En Büyük Ortak Bölüneni ve En Küçük Ortak Katımı Bulma



Şekil 3.7. Armstrong Sayıları

Pilot çalışma sonrası asıl çalışma Covid-19 salgını sebebiyle uzaktan Zoom programı yardımıyla yapılmıştır. Öğrenciler toplantıya katıldıktan sonra her etkinlik için birer toplantı yöneticisi seçilmiştir. Toplantı yöneticisinin görevi kendi bilgisayarındaki Scratch programını açarak araştırmacının verdiği etkinlikleri, grupça çalışarak yapmaktır.

- 1. Etkinliğin yöneticisi Erdem,
- 2. Etkinliğin yöneticisi Yalçın,
- 3. Etkinliğin yöneticisi Cem seçilmiştir.

Öğrencilere etkinlikler araştırmacı tarafından sözlü ve görsel olarak verilerek çalışma süreci başlamıştır. Öğrencilerin etkinlik için harcadıkları süreler aşağıdaki tablo 3.1.'de verilmiştir. Öğrenciler etkinlikleri oluştururken araştırmacı tarafından yaptıkları adımlara göre onlara sorular yöneltilmiş ve algoritma oluşturma süreci bu şekilde ortaya çıkarılmaya çalışılmıştır. Öğrencilere düşünceleri için fırsat verilmiş ve süre sınırı olmadığı söylenmiştir. Etkinlik oluşturma sürecinde ara verilebilecekleri veya etkinliği tamamlayamadıkları durumda bırakabilecekleri belirtilmiştir. İlk iki etkinliği tamamlayan öğrenciler, üçüncü etkinliği 30. dakikada bırakmışlardır.

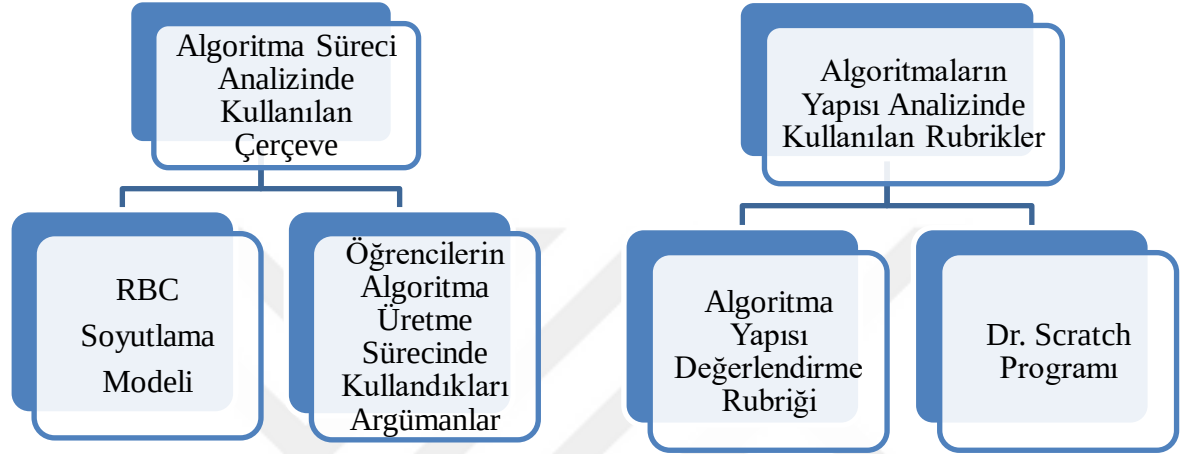
Tablo 3.1. Öğrencilerin Her Etkinlik İçin Harcadıkları Süre

Etkinlik Adı	Etkinlikler İçin Harcadıkları Süre
1. Etkinlik	34 dakika
2. Etkinlik	18 dakika
3. Etkinlik	30 dakika

3.6. Verilerin Analizi

Bu araştırmanın verilerini, sesli düşünme protokolünün ile gerçekleştirilen ekran kayıtları ve öğrencilerin üç etkinlik için oluşturdukları Scratch programı projeleri oluşturmaktadır. Verilerin analiz edilmesinde *içerik* ve *betimsel analiz* yöntemleri kullanılmıştır. İlk olarak sesli düşünme protokolü ile elde edilen veriler araştırmacı tarafından her bir etkinlik için transkript edilmiş daha sonra içerik analizi sürecinde her etkinlik bazında sınıflandırmalar ve kodlamalar yapılmıştır. Bu sınıflama ve kodlamalar veri analizi bölümünde verilmiştir Patton (2018), betimleme sürecinde bulguların yorumlanacağını ve doğrudan alıntılar yapılarak verilerin anlamlı olmasının sağlanacağını belirtmiştir. Elde edilen verilerden

öğrencilerin pozitif tam sayılar konusundaki algoritma üretme süreçleri incelenmiştir. Ayrıca etkinlikler sonucunda oluşturulan algoritmaların yapısı incelenmiştir. Algoritma üretme sürecinin analiz edilmesinde iki değerlendirme çerçevesi kullanılmıştır. Bunlar “RBC Soyutlama Modeli” ve “Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması” kullanılmıştır. Algoritmaların yapısının analizinde ise “Algoritma Yapısı Değerlendirme Rubriği” ve “Dr. Scratch Programı” kullanılmıştır.



Şekil 3.8. Veri Analizi Şeması

Veri analizinde kullanılan değerlendirme çerçeveleri ve rubrikleri ayrı başlıklar altında aşağıda sunulmuştur.

3.6.1. Algoritma Süreci ile Yapılan Değerlendirme Rubriği

Öğrencilerin tam sayılar konusunda Scratch programı ile algoritma üretme süreci “RBC Soyutlama Modeli” ve “Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması” çerçeveleri ile analiz edilmiştir. Bu çerçevelerin veri analizi kısmında nasıl kullanıldığı bir sonraki başlıklar altında verilmiştir.

3.6.1.1. RBC Soyutlama Modeli

Bu araştırmada matematiksel soyutlama sürecini anlamlandırmayı hedefleyen modellerden biri olan ve Hershkowitz, Schwarz ve Dreyfus tarafından 2001 yılında ortaya atılan RBC (Recognizing - Building with- Constructing) soyutlama modeli kullanılmıştır. Bu modelin oluşturulmasında Altun ve Memnun’un (2012) çalışmasında kullandığı kelimeler temel

olarak seçilmiş ve Tablo 3.2.'deki model geliştirilmiştir. Bu şekilde bulgular yorumlanırken, öğrencilerin ifadelerinin ve bilişsel davranışlarının eylem basamaklarına göre değerlendirilmesi amaçlanmıştır.

Tablo 3.2. RBC Soyutlama Modeli

Eylemler	Açıklamaları	Kod
Tanıma	Öğrencinin önceki bilgilerinin eldeki probleme uygun olduğunu fark etmesini ifade eder.	T
Kullanma	Bir problemin çözümünün gerekçelendirilmesi gibi yerel bir hedefe ulaşmak için tanınan nesneleri kullanan öğrenciyi ifade eder.	K
Oluşturma	Nispeten yeni bir yapı oluşturmak için önceki yapıların dikey veya yatay matematikselleştirme ile birleştirilmesidir.	O

Bu araştırma kapsamında algoritma üretme süreci esnasında, RBC soyutlama modeli kullanılarak öğrencilerin gözlenebilir bilişsel eylemler üzerinden bilgiyi oluşturmaları incelenmiştir. Öğrencilerin pozitif tam sayılar konusundaki ifadeleri bu bilişsel eylemlere göre değerlendirilmiştir. Ayrıca öğrencilerin Scratch programında algoritma üretme adına yaptıkları adımlar da değerlendirilmeye tabi tutulmuştur. Bu değerlendirmede Tablo 3.2.'de verilen soyutlama modeli kullanılmıştır. Verilerin analizi sonucunda elde edilen örnek eylemlerden birkaçı Tablo 3.3.'te verilmiştir.

Tablo 3.3. RBC Soyutlama Modeli Göre Tam Sayılarla İlgili Araştırmada Geçen Örnek Eylemler

Eylemler	Örnekler
Tanıma	Cem’in bir sayının asal olması için sadece 1’e ve kendisine tam bölünmesi gerektiğini söylemesidir.
Kullanma	Pozitif bölenleri listeleme etkinliğinde öğrencilerin ekrana 99999999999999999999999999999999 sayısını yazarak algoritmayı sınamasıdır.
Oluşturma	Erdem’in ardışık sayıların toplamı etkinliğinde; ilk önce ardışık en büyük sayıyı algoritmaya girmesi ve sonra “sürekli tekrarla” bloğunun içine “sayıyı –1 kadar değiştir” bloğunu eklemesidir.

3.6.1.2.Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması

Algoritma üretmek öğrenciler için zorlu bir süreçtir. Çünkü oluşturulan algoritma çözümün tüm özelliklerini sağlaması gerekmektedir. Algoritma basit bir çözümden ziyade bir problem için kapsamlı bir çözüm inşa etme sürecidir.

Tablo 3.4. Algoritma Üretme Süreci Çerçevesi (Moala, 2019)

Terim	Tanımı
Süreç	Görevin çözüm şekli
Algoritma	Bir görevi çözmek için kullanılabilecek açık bir kurallar dizini
Algoritmik Yaklaşım	Matematiksel problemlerin çözülmesine yönelik bir çözüm bulmak için kullanılabilecek bir algoritma açıkça oluşturulur. Bir algoritma görevi, algoritma yaklaşımını vurgulayan bir görevdir.
Bir çözümün özelliklerini dikkate almak	Bulunan çözümün belirli özelliklerini fark eden ve ardından algoritmanın fark edilmiş özelliklere sahip bir çözüm üretmesini garanti eden belirli kurallar oluşturan bir algoritma oluşturma mekanizmasıdır.
Etkin bir dizi özellik (çözümün)	Yalnızca bu özelliklerin hesaba katılmasıyla algoritmanın (uygulandığında) doğru bir çözüm üretmesi garanti edilirse çözüm yeterlidir.

Tabloda öğrencilerin algoritma üretme süreçlerinde farkında olarak veya olmayarak doğru, yanlış ifadeler kullanabildiği Moala (2019) tarafından ortaya konulmuştur. Bu çalışmada Moala'nın yaptığı çalışmadan yararlanılarak öğrencilerin algoritma üretme süreçlerinde ifade ettikleri argümanlar Tablo 3.5.'teki gibi sınıflandırılmıştır. Ayrıca veri analizinde Tablo 3.5.'te verilen kodlar kullanılmıştır.

Öğrencilerin algoritma üretme sürecinde ifade ettikleri düşünceler ve davranışlar öğrenciler tarafından bilinçli olarak ortaya konulmuşsa açıkça belirtilen argümanı içine alınmıştır. Aksi durumda öğrenciler farkında olmadan bir doğru düşünce ve davranışı ortaya koymuşlarsa açıkça belirtilmeyen argümanı içine alınmıştır. Daha sonra öğrencilerin bu düşünce ve

davranışları problemin çözümüne ilişkin doğru ve yanlış argümanlar olarak sınıflandırılmıştır.

Tablo 3.5. Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması

Özellikler	Kodu
Açıkça Belirtilen ve Doğru Olan Özellikler	A-D
Açıkça Belirtilmeyen ve Doğru Olan Özellikler	B-D
Açıkça Belirtilen ve Yanlış Olan Özellikler	A-Y
Açıkça Belirtilmeyen ve Yanlış Olan Özellikler	B-Y

Tablo 3.6.’da bu araştırma için her bir argüman için birer örnek bulunmaktadır.

Tablo 3.6. Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılmasına Örnekler

Özellikler	Örneği
Açıkça Belirtilen ve Doğru Olan Özellikler	Negatif tam sayılar pozitif bölen listesine alınmamalıdır.
Açıkça Belirtilmeyen ve Doğru Olan Özellikler	“Sayı defa tekrarla” bloğunu koymalıyız (Bu şekilde yaptıklarında algoritma sayının kendisini bölen olarak almamaktadır. Çünkü algoritmanın başında “böleni sıfır yap” bloğu kullanılmıştır. Bu şekilde bölme işlemi sıfırdan başlamakta ve sayının kendisine sıra gelince durmaktadır. Bu eksiliği anlamadan algoritmanın sonuna “sayının kendisini listeye ekle” bloğunu kullanarak çözmüşlerdir.).
Açıkça Belirtilen ve Yanlış Olan Özellikler	“Sayı bölü sayı eksi 1” bloğunu kullanmalıyız (“sayı bölü sayı eksi 1” bloğundaki yanlışlık araştırmacı tarafından açıkça belirtilmiştir. Çünkü sayı bölü sayı 1 yapar ve $1-1=0$ olur).
Açıkça Belirtilmeyen ve Yanlış Olan Özellikler	Bir sayının pozitif bölenleri 1’den değil 2’den başlamalıdır (asal sayılar konusu ile karıştırılmıştır.)

Öğrencilerin hangi argümanları daha fazla kullandığı sonuç ve tartışma bölümünde ortaya konulmuştur.

3.6.2. Algoritma Yapısı ile Yapılan Değerlendirme Rubriği

Öğrencilerin pozitif tam sayılar konusunda Scratch programı ile oluşturulan algoritmaların yapıları “Algoritma Yapısı Değerlendirme Rubriği” ve “Dr. Scratch Programı” rubrikleri ile analiz edilmiştir. Bu rubriklerin veri analizi kısmında nasıl kullanıldığı bir sonraki başlıklar altında verilmiştir.

3.6.2.1. Algoritma Yapısı Değerlendirme Rubriği

Oluşturulan algoritma başka bir müdahaleye gerek kalmadan çözümü üretmelidir. Bu araştırmada öğrencilerin oluşturduğu algoritmalar yapı bakımından iki değerlendirme rubriğine göre incelenmiştir. Birincisi Knuth (2014) tarafından ortaya konulan iyi bir algoritmanın beş özelliğidir. Bu özellikler ve açıklamaları Tablo 3.6.’da verilmiştir.

Tablo 3.7. Knuth (2014)’a Göre Algoritmaların Beş Temel Özelliği

Özellik	Açıklaması
Sonlu olma	Algoritma mutlaka bir cevap ortaya çıkartmalıdır.
Kesinlik	Algoritmadaki adımlar, onu inceleyen her kişinin anlayabileceği şekilde açık olmalıdır.
Girdi	Algoritmanın girdileri problem çözümüne uygun bir şekilde ve doğru tanımlanmalıdır.
Çıktı	Algoritmanın çıktıları problem çözümüne uygun bir şekilde ve doğru tanımlanmalıdır.
Etkileycilik	Oluşturulan algoritma istenen problemi çözmelidir.

Öğrencilerin oluşturduğu algoritmalar “Algoritma Yapısı Değerlendirme Rubriği” ile değerlendirilmiştir. “Algoritma Yapısı Değerlendirme Rubriği” Knuth (2014) tarafından ortaya konan iyi bir algorithmada bulunması gereken özellikler kapsamında matematik öğretimi alanında uzman bir öğretim üyesi ve alanında uzman bir bilişim teknolojileri öğretmeni ile oluşturulmuştur. Değerlendirme aşamasında algoritmalar yukarıdaki rubriğe göre puanlanmıştır. Bu şekilde öğrencilerin üç etkinlik için oluşturdukları algoritmaların yapısının detaylı analiz edileceği düşünülmüştür.

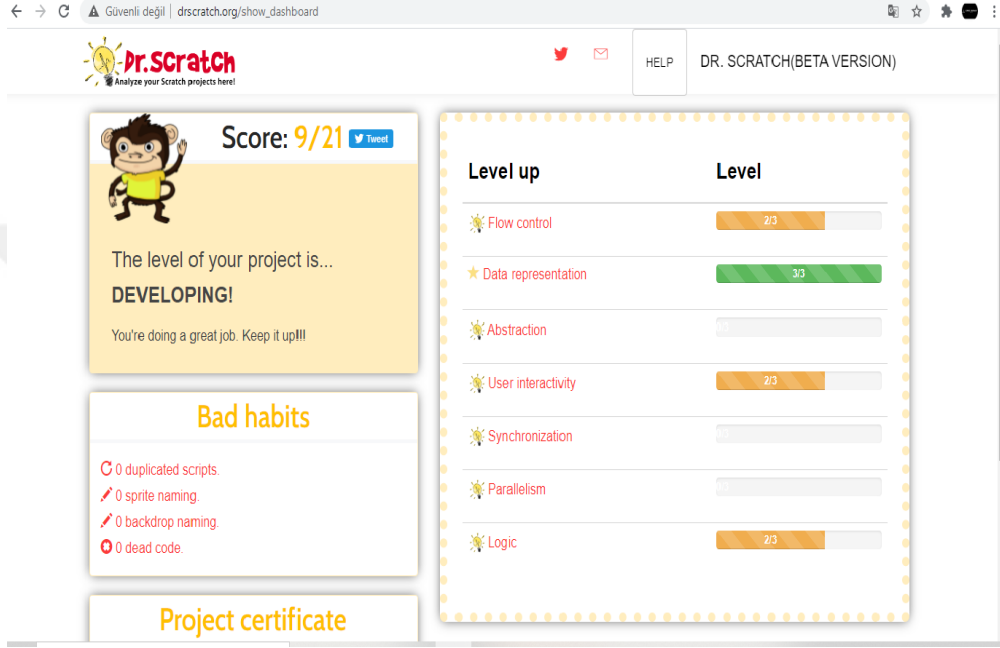
Tablo 3.8. Algoritma Yapısı Değerlendirme Rubriği

Puan	0	1	2
Özellikler			
Sonlu Olma	Algoritmanın sonu yoktur.	Algoritmanın birkaç aşaması vardır ama sonu yoktur.	Algoritmanın sonu vardır.
Kesinlik	Algorithmada bir kesinlik yoktur.	Algorithmada kesinlik kısmen vardır.	Algoritma tam olarak kesindir.
Girdi	Algoritmaların girdileri yoktur veya yanlıştır.	Algoritmanın girdisi kısmen doğrudur.	Algoritmanın girdileri tam ve doğrudur.
Çıktı	Algoritmanın çıktıları yoktur veya yanlıştır.	Algoritmanın çıktıları kısmen doğrudur.	Algoritmanın çıktıları tam ve doğrudur.
Etkileycilik	Algoritmanın adımları makul değildir.	Algoritmanın adımları kısmen makuldür.	Algoritmanın adımları makul ve etkilidir.

3.6.2.2. Dr. Scratch Programı

Scratch programı ile yapılan matematik eğitimi araştırmaları literatür kısmında verilmiştir. Bu kapsamda oluşturulan Scratch projelerin değerlendirmesi ile ilgili çeşitli araçlar araştırmalar mevcuttur. Bazı çalışmalarda araştırmacıların oluşturduğu araçlar kullanılırken bazı çalışmalarda Dr. Scratch gibi otomatik değerlendirme araçları kullanılmıştır. Bu araştırmada öğrencilerin oluşturduğu algoritmaları yapı bakımından değerlendirme araçlarından ikincisi olarak Dr. Scratch programı seçilmiştir. Dr. Scratch programı, oluşturulan Scratch programı projelerini birçok açıdan değerlendirmek için kullanılan bir

internet sitesidir. Literatürde Dr. Scratch programının kullanıldığı araştırmalar mevcuttur (Erdem, 2018; Moreno vd., 2015; Oluk vd., 2018). Bu araştırmalarda üretilen projeler Dr. Scratch programı ile analiz edilmiş ve bulgular elde edilmiştir. Moreno vd. (2015) yaptıkları araştırmada Dr. Scratch programının Scratch ile oluşturulan algoritmalarda hatalı kodlamayı, kullanımı gereksiz blokları, kodun yanlış tekrarı, hiçbir zaman çalıştırılmamış kod ve projenin yanlış başlatılması gibi olası hataları algıladığını belirtmişlerdir.



Şekil 3.9. Dr. Scratch Programı Değerlendirme Rubriği

Dr. Scratch programı değerlendirme yaparken yedi farklı analiz başlığı kullanmaktadır (Bkz Şekil 3.9.). Dr. Scratch programı görsellik, ses, senkronize hareketler, açısı hesaplamalarına göre yansımalar gibi özellikleri de dikkate almaktadır. Bu kapsamda Dr. Scratch programı nispeten daha kapsamlı projelerde etkin ve tutarlı değerlendirme olanağı sunmaktadır. Programda olan görsellik, ses, senkronize hareketler, açısı hesaplamalarına göre yansımalar gibi özelliklerin öğrencilerle yapılan pilot çalışmalarda kullanılmadığı görülmüştür. Bu araştırmadaki asıl amaç kapsamlı bir Scratch projesi oluşturmak değil etkinliklerden istenen matematiksel algoritmayı doğru bir şekilde oluşturmaktır. Bu kapsamda matematik öğretimi alanında uzman bir öğretim üyesi ile bilişim alanında uzman bir öğretmenin görüşleri alınarak değerlendirme rubriği Tablo 3.8.'deki gibi revize edilerek ilgili kısımlar alınmıştır. Oluşturulan bu değerlendirme rubriği hakkında öğrencilere etkinliklerden önce bilgi verilmemiştir. Rubrik verilmemesinin sebeplerinden biri öğrencilerin belli kod bloklarını kullanarak algoritma oluşturmaları değil, çözüme en kısa ve doğru yoldan ulaşacakları kodlar

kullanmalarınıdır. Rubrik verilmemesinin bir diğer sebebi ise öğrencilerin rubrikteki özelliklere sahip kodlara odaklanarak etkinliği yanlış yürütsele de yüksek puan almalarını önlemektir. Öğrencilerin oluşturdukları algoritmaların yapılarının objektif bir şekilde değerlendirmek için Dr. Scratch programının belirtildiği gibi kullanılması uygun görülmüştür.

Tablo 3.9. Revize Edilmiş Dr. Scratch Programına Göre Değerlendirme Rubriği

Puan	1 Puan	2 Puan	3 Puan
Özellik			
Flow Control (Akış Kontrolü)	Birbiri ardına yürütülen bir dizi blok oluşturulmalıdır. Kullanıcı yeşil bayrağa tıkladığında, tüm bloklar birbiri ardınca çalışmalıdır.	Aynı blokları tekrarlamak yerine, aynı etkiyi elde etmek için daha rahat ve yönetilebilir blok kullanılmalıdır.	Bir dizi bloğun kaç kez çalıştırılmasını istediğinizi önceden bilmiyoruz. Bu durumlarda, "... olana kadar tekrarlar" bloğu kullanılmalıdır.
Data Repesantation (Veri Temsili)	Oluşturulan her karakterin ve her bloğun bir değeri olmalıdır.	Oluşturulan verileri depolamak için "değişkenler" bloğu oluşturulmalıdır.	Değişkenlere ek olarak "listeler" bloğu kullanılmalıdır. Listeler aynı anda birden fazla değeri depolayabilir.
User interactivity (Kullanıcı Etkileşimi)	Kullanıcı ile etkileşimde çalışmaya başlayabileceğiniz en temel blok olan "yeşil bayrak tıkladığında" bloğu eklenmelidir.	Kullanıcıyla etkileşime izin veren hareket, ses, fare imleci gibi bloklar kullanılmalıdır.	Kullanıcı etkileşimi için web kamerası veya mikrofon kullanılmalıdır.
Logic (Mantık)	Scratch programı, bloktaki koşulu değerlendirir. Bu koşul doğruysa, içindeki bloklar çalışır.	Farklı durumları düşünerek "eğer ise değil ise" bloğu kullanılmalıdır.	Birden fazla koşulu aynı anda değerlendirmek gerekir. Bunun için "ve/veya" blokları kullanılmalıdır.

Dr. Scratch programı bu gibi başlıkların dışında Bad habits (Kötü Yapılar) adı altında oluşturulan projeyi şu açılardan da değerlendirmektedir;

- Duplicated scripts. (Gereksiz yinelenen veya bir blokla oluşturulabilecek görevi birden fazla blokla oluşturma)
- Sprite naming. (Karakterlerin adını projeye uygun vermeme, varsayılan adları kullanma)
- Backdrop naming. (Senaryoların adını projeye uygun vermeme, varsayılan adları kullanma)
- Dead Code. (Projede hiç çalıştırılmayan kod)

Dr. Scratch programı bu başlıkların içinde istenilen durumlar oluşmadığında o başlık altında değerlendirme yapmamaktadır. Bu yüzden o başlık sıfır puan olarak geçmektedir. Bu başlıklar altında değerlendirmeler yapılmış ve her bir çalışma için 0, 1, 2 veya 3 puanlama sistemi ile Dr. Scratch programı puanlama yapmıştır.

3.7. Geçerlik ve Güvenirlik

Kirk ve Miller (1986), nitel araştırmalar için geçerlik kavramını araştırmacının çalışmasını olduğu hali ile ortaya koyması ve olabildiğince objektif gözlemde bulunması olarak ifade etmektedir. Araştırmada bütüncül bir resim ortaya koyabilmek ve geçerliği sağlamak için veri çeşitlemesi, katılımcı teyidi, araştırma üzerinden elde edilen veriler ve sonuçlar üzerinden uzman görüşü almak, uzun süreli veri toplamak ve betimsel analizin kullanıldığı çalışmalarda doğrudan alıntılara yer vermek önemlidir (Yıldırım & Şimşek, 2011). Nitel araştırmalarda iç geçerlik araştırmacının belirlediği kategori ve yorumların gerçekleşen doğrularla örtüşmesine ve gerçeği yansıtmasına bağlıdır. Nitel bir araştırmada iç geçerlik elde edilen verilerin farklı kişi veya çeşitli program ve algoritmalar tarafından incelenerek karşılaştırılmasıyla arttırılabilir. Bu araştırmada bunun için Knuth (2014)'un algoritmaların temel özellikleri çalışması, revize edilmiş Dr. Scratch programında bulunan rubrik ve bilişim teknolojileri alanında uzman bir öğretmenin görüşleri kullanılmıştır. Dış geçerlik ise sonuçların genellenebilmesi ile alakalıdır. Nitel araştırmalarda belli bir durum ayrıntılı bir şekilde incelendiğinden, az katılımcı ile geniş bir örnekleme temsil etmediğinden ve seçilen yöntem yapılan araştırmaya özgü olduğundan genelleme güçtür. Nitel araştırmalarda karşılaştırılma ve dönüştürülme kavramları dış geçerlikle ilişkili olarak kullanılmaktadır. Bu

nedenle nitel arařtırmalardaki bilgiler, kategoriler ve sonuçlar ne kadar iyi tanımlanırsa diğerk arařtırmacıların sonuçları anlamlandırması ve benzer arařtırmalar yapmaları daha kolay olur (Büyüköztürk, Akgün, Demirel, Karadeniz & Çakmak, 2015).

Arařtırmada geçerliğı sağlamak için iki aşamalı uygulama yapılmıř ve Scratch programı üzerinden yapılacak olan çalıřmalar katılımcıların sınıf düzeyleri dikkate alınarak hazırlanmıřtır. Sadece “Mod Kavramı” katılımcıların bilmediğı bir kavram olarak belirlenmiř ve bu konuda öğrencilere kısa bir konu özeti yapılmıřtır. Ek 2’de verilen sınav ile öğrencilerin bu kavramları anladığı sonucuna varılmıřtır. Algoritmalar ile ilgili çalıřmalar için matematik öğretimi alanında uzman bir öğretim üyesi ile biliřim alanında uzman bir öğretmenin görüşlerine başvurulmuř ve revize edilerek asıl çalıřmaya yön verilmiřtir. Ayrıca veri çeřitilmesi adına hem Scratch programı üzerinden yapılan çalıřmalar titizlikle takip edilmiř hem de çalıřma esnasında öğrencilerden Sesli Düşünme Protokollerini kullanmaları istenmiřtir. Veriler betimsel analize tabi tutulmuř, bulgular program üzerinden kurulan algoritmaların resimleri verilerek yorumlanmıřtır. Arařtırmada iç geçerliğın sağlanması için veri toplama aracı ve veri toplama sonucunda ortaya çıkan verilerin kategorize edilmesinde uzman görüşüne başvurulmuř ve uzman görüşleri doğrultusunda analizlere yön verilmiřtir. Dıř geçerliğı sağlamak için ise elde edilen veriler belirlenen çerçeveye göre doğrudan alıntılama ile ayrıntılı olarak analiz edilmiřtir.

Güvenirlik çalıřmanın tekrar edilebilirliğıdir. LeCompte ve Goetz (1982), dıř güvenirliğın sağlanmasına yönelik arařtırmacının ilk olarak arařtırma sürecindeki kendi konumunu açıklaması gerektiğini söylemektedirler. Bu şekilde arařtırmacı kendi konumu ile ilgili yapacağı açıklamalar ile aynı konuda çalıřacak diğerk arařtırmacılara yol göstermiř olur. Bu arařtırmada arařtırmacının yardımı sınırlıdır ve arařtırma sonuçlarına direkt etki etmediğı düşünülmektedir. LeCompte ve Goetz, iç güvenirlik konusunda ise bazı ipuçları önermektedirler. İç güvenirlik öncelikle toplanan verilerin betimsel bir yaklaşımla doğrudan sunulması ile ilgilidir. Yani arařtırmacı görüşme, gözlem ve dokümanlar ile elde ettiğı verileri herhangi bir yorum katmadan okuyucuya sunmalı ve yorumunu daha sonraya bırakmalıdır (Yıldırım & Şimşek, 2011).

Arařtırmada güvenirliğı sağlamak için amacıyla veriler kavramsal çerçeveye göre analiz edilmiřtir. Kavramsal çerçevede öğrencilerin Scratch programı üzerinden yaptıkları çalıřmalarda ortaya çıkan algoritma üretme süreçleri incelenmiřtir. Öğrencilerin verdikleri cevaplar ile çerçevede belirlenen davranıřlar ile uyumu incelenmiřtir (Hershkowitz vd., 2001 ve Moala, 2019). Arařtırmacının belirlediğı kategoriler için uzman görüşleri alınmıř, görüşler doğrultusunda bazı düzeltmeler yapılmıřtır.

BÖLÜM IV: BULGULAR

Bu bölümde akademik başarısı yüksek ortaokul öğrencilerinin Scratch programı üzerinden pozitif tam sayılar konusunda yaptığı etkinlikler ve bu etkinlikler sonucunda ortaya çıkan algoritmalara ilişkin bulgulara yer verilmiştir. Öğrencilerin etkinlik sürecinde kullandıkları ifadeler direkt alıntılanarak veya şekiller yardımı ile gösterilmiştir. Bu süreç veri analizinde açıklanan “RBC Soyutlama Modeli” ve “Öğrencilerin Algoritma Üretme Sürecinde Kullandıkları Argümanların Sınıflandırılması” değerlendirme çerçevesi kullanılmıştır. Veri analizinde belirtilen bu değerlendirme çerçeveleri cümle sonunda parantez içinde belirtilen kodlar ile gösterilmiştir. Algoritma üretme sürecinin sonunda öğrencilerin oluşturduğu algoritmalar “Algoritma Yapısı Değerlendirme Rubriği” ve “Dr. Scratch Programı” rubrikleri ile puanlanmıştır. Algoritma üretme süreci ve algoritmaların yapılarının incelenmesi her bir etkinlik için ayrı ayrı başlıklar altında sunulmuştur. Ayrıca algoritma üretme süreci ve algoritma yapıları ile ilgili etkinliklere ait genel bir değerlendirme sunulmuştur. Bu bölümde kişileri konuşmaları Araştırmacı: A, Erdem: E, Cenk: C ve Yalçın: Y şeklinde kodlanarak kullanılmıştır.

4.1. Pozitif Bölenleri Listeleme Etkinliğine Ait Bulgular

Etkinliğin başında öğrencilerle çevrimiçi bağlantı sağlanmıştır. Bu bağlantıdan sonra araştırmacı tarafından etkinlik öğrencilere açıklanmıştır. Etkinlik 34 dakika sürmüştür. Etkinlik sonunda öğrencilerden üretilmesi beklenen algoritmanın başarı ile oluşturulduğu gözlemlenmiştir. Algoritma üretme süreci ve öğrencilerin oluşturdukları algoritma ile ilgili bulgular bir sonraki kısımlarda verilmiştir.

4.1.1. Algoritma Üretme Sürecine Ait Bulgular

Öğrenciler etkinliğe başladıklarında ilk olarak algoritmanın girdilerini belirlemeyi amaçlamışlardır (Bkz [1-6]).

1. C: İlk önce değişkenleri oluşturarak başlayabiliriz (A-D).

Erdem, sayı değişkeni oluşturdu ve daha sonra “Diye sor ve bekle” ve “Tıklandığında” bloklarını ekledi.

2. C: Operatörlerden bir şey kullanabilir miyiz? Mesela “Diye sor ve bekle” bloğundan alınacak cevaptan küçük sayılar bölü cevap gibi bir şeyler olabilir (B-Y).
3. A: Hazırlık çalışmalarımızda Scratch programında ilk başta oluşturulması gerekenleri hatırlayın.
4. E: “sayı=cevap” bloğunu ekleyelim çünkü “Diye sor ve bekle” bloğundan alınacak cevap bizim sayımız olacaktır.

“Sayı=cevap” bloğu herhangi bir kısma eklenememiştir.

5. C: Eee kontrol bloğuna geçelim.
6. E: Bence “sayıyı cevap yap” yapmalıyız (A-D).

“Sayıyı yap” bloğu eklendi ve içerisine “cevap” bloğunu yerleştirildi.

Daha sonra ise bir sayının pozitif bölenlerini bulmak için arayışlara başlamışlardır. Bir sayının pozitif bölenlerinin sayının kendisinden küçük olması gerektiği Cenk tarafından vurgulanmıştır (T). Bunun için Scratch programındaki “operatörler” bloğunu kullanmak istemişlerdir [7-8].

7. C: Biz ne istiyoruz şimdi? Sayıdan küçük olan şeylerin hepsini bölmeli (A-Y). O zaman sayıdan küçük olanlar diye bir blok var mı?
8. E: Operatörlerden büyüktür bloğunu ekrana getireyim.

Daha sonra sayının kendisini ve kendisinden küçük sayıları almalarını gerektiğini düşünmüşler ve “küçüktür eşittir” diye bir blok olup olmadığını sormuşlardır. Scratch programında direkt bu şekilde bir blok yoktur. Ayrıca öğrenciler 6. sınıf öğrencisidir ve küçüktür eşittir ifadesi 8. sınıf eşitsizlik konusu ile ilgilidir. Bu kısımda Cem “eşittir” ifadesini sayının kendisi olarak değil 1 olarak algıladı ve bu sayının asal sayılar listesine ekleneceğini zannederek bölenlerin 1’den değil 2’den başlaması gerektiğini beyan etmiştir. Ayrıca bu bölen listesinin sayının bir eksiğine kadar devam edip orada bitmesi gerektiğini söylemiştir [9]. Bu şekilde öğrencinin pozitif bölenler ile asal sayıları birbirine karıştırdığı gözlemlenmiştir.

9. C: Eşit olanlara bakmasına gerek yok ki. O zaman asla asal diyemez. Asal olup olmadığını da anlamalıdır. Mesela 1 değil 2’den başlama ki 1’e illa asal demesin. 2’den başlamalı ve sayı eksi 1’de bitmeli (A-Y)

Öğrenciler “sayı bölü sayı küçüktür sayı” gibi bir blok oluşturarak bölenleri ifade etmeye çalışmışlardır [10].

10. E: Hocam buldum sanırım sayı bölü sayı büyüktür sayı (A-Y). Erdem Şekil 4.1.'deki gibi algoritmayı kurmuş ve ekrana “sayı bölü sayı büyüktür sayı” bloğunu getirmiştir (K).



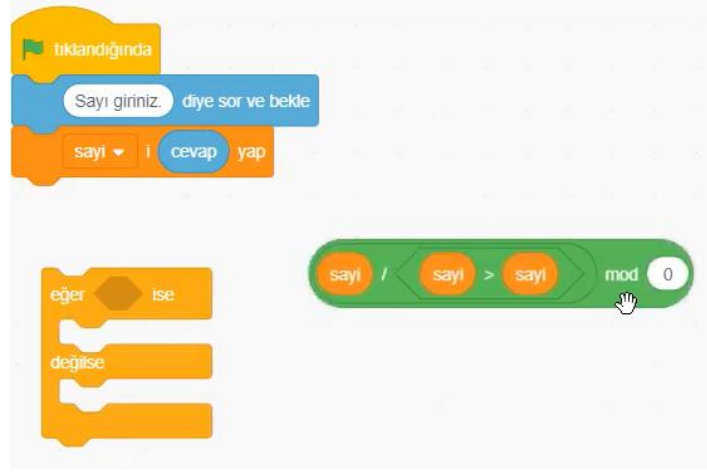
Şekil 4.1. “Sayı Bölü Sayı Büyüktür” Bloğu

Araştırmacı, aşağıdaki gibi bir açıklama yaptı. [11].

11. A: Bir sayının diğer bir sayıya tam bölündüğünü nasıl anlarız?
12. E: Bir dakika hocam burada tam bölünen bloğu olmalı (A-D).
13. E: Hocam bir saniye bu tam bölünme bloğu neydi? Nasıl yaptığımızı hatırladım ama adı aklıma gelmedi. Hatta 30 bir şey 5 diyorduk sıfır oluyordu (T). Evet hatırladım mod mod! (A-D)

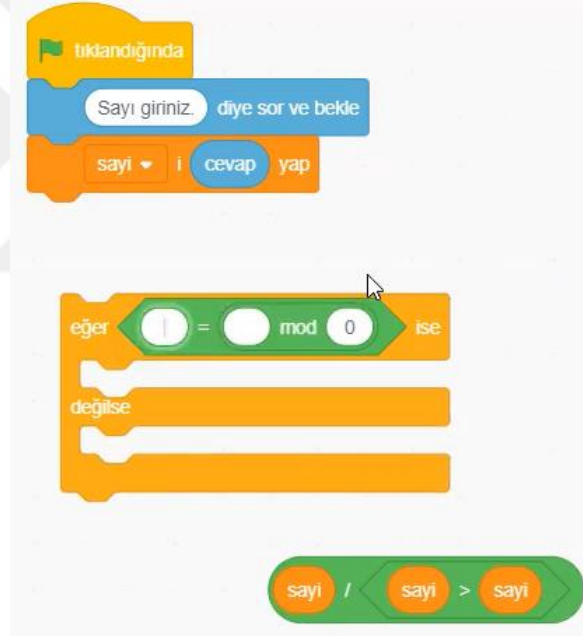
Erdem bu aşamada negatif sayıları gündeme getirerek negatif sayıların kullanılmaması gerektiğini belirtti (A-D). Erdem bu sırada “Mod bloğu” ekledi ve mod bloğunun sağ tarafına sıfır yazdı (B-Y). Erdem şekil 4.2’teki kod bloğunu oluşturdu (K).

Etkinlik başından itibaren Yalçın’ın grup çalışmasına hiç katılmadığı gözlemlendi. Yalçın’a bu durum soruldu ve kendi bilgisayarında algoritmayı oluşturmaya çalıştığını ifade etti. Fakat net bir şey ortaya çıkaramadığını belirtti.



Şekil 4.2. Mod Bloğu Oluştuktan Sonra Oluşturulan Kod Blokları

Daha sonra Erdem Şekil 4.3.’teki blokları birbirleri ile sürekli değiştirerek denedi (K).



Şekil 4.3. “Mod” Bloğu ile “Sayı Bölü Sayı Büyüktür Sayı” Bloğunun Yer Değiştirerek Denenmesi

Erdem “mod” kavramını tam olarak hatırladı ve mod bloğunu oluşturdu. Cem mod kavramını tam hatırlayamadı. Erdem, Cem için mod kavramını açıkladı. Bu açıklamadan sonra Cem mod kavramını hatırladı. Ancak mod bloğunun sağ ve sol kısımlarına hangi ifadeleri koyacakları konusunda kararsız kaldılar.

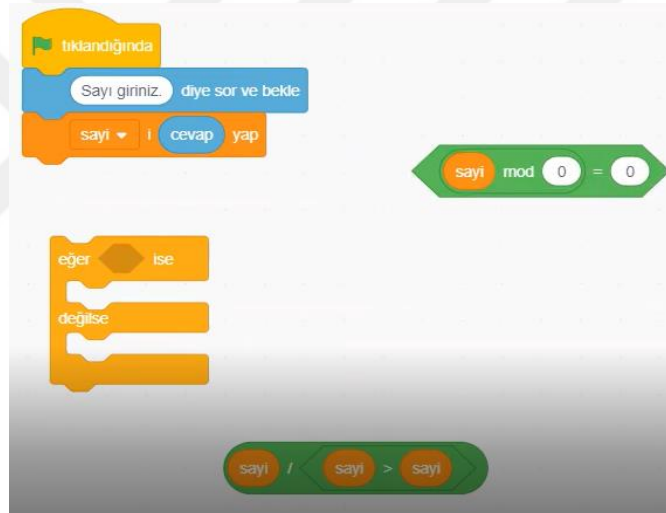
14. E: Bölen diye değişken oluşturmamız lazım sanırım (B-D).

15. C: Tamam o zaman bir tane liste oluşturmalyız (A-D).
16. E: O nereden yapılıyor?
17. C: Şey bir değişkenin altında bir liste oluştur var. Görüyor musun? Oradan yapılıyor. İsmini de bölenler yapalım (A-D).

Öğrenciler sayı değişkeni ve bölen listesi adı altında bir liste oluşturdu. Öğrencilerin “sayı mod sayı eksi” ve “sayı mod sayı küçüktür” gibi ifadeler ile uğraştılar (K). Blokları birkaç defa değiştirerek algoritmalar sınanmış ve ekranda herhangi bir sonuç çıkmadı.

18. E: Hocam ben şu modun işlemindeyim. Eğer sayı küçüktür sayı eşittir sıfır ise...
19. C: İyi de sayı mod sayı eksi... sayıdan küçükler bölü...(B-Y). Aaa durun öyle bir blok vardı. Şimdi hatırladım. Şundan şuna kadar diye bir şey var.

Erdem Şekil 4.4.’teki blokları oluşturdu.



Şekil 4.4. Erdem’in Oluşturduğu Blok

20. A: Evet mod bloğunun sol kısmına neler koymalıyız?
21. E: Sayı mod sayı küçüktür sayı olabilir mi? (B-Y)
22. A: Sayının diğer kısmına öyle bir şey koymalısın ki sürekli sayı değişkenini bölmeye çalışsın. Ne olabilir?

Erdem bölen değişkenini mod bloğunun sol tarafına ekledi (O).



Şekil 4.5. Erdem'in Doğru Oluşturduğu Mod Bloğu

Erdem “listeye ekle” bloğunu kullandı ve kod bloğu Şekil 4.5.’teki gibi oldu.

23. C: Şimdi yeşile basalım. Bakalım neler oluyor.

Yeşil bayrağa basıldı ve herhangi bir sonuç ekranda belirmedi. Cem, ekranda sonucun çıkmaması üzerine bölenlerin 1’den büyük ve sayıdan küçük olarak ayarlanması gerektiğini tekrar söyledi (B-Y). Bu durum hakkında araştırmacı, ile Cem arasında şu diyaloglar geçti [24-29].

24. C: Bölen sayıdan küçük olan ve 1’den büyük olan.

25. A: Böleni ne yapmanız gerekiyor?

26. C: Hocam bir şey sorabilir miyim? 1’i söylemeli miyiz? Çünkü bölenler listesinde 1 ve sayının kendisi çıkmayacaktır bu durumda (A-D), (K).

27. A: Sence bir sayının bölenlerinde en küçük sayı ve en büyük sayı nedir?

28. C: Hocam en küçük 1 olur ama o zaman asal sayıya asal demez (A-D).

29. A: İsterseniz tüm bölenleri en başta belirleyin. En son asal olup olmadığını belirleyebilirsiniz.

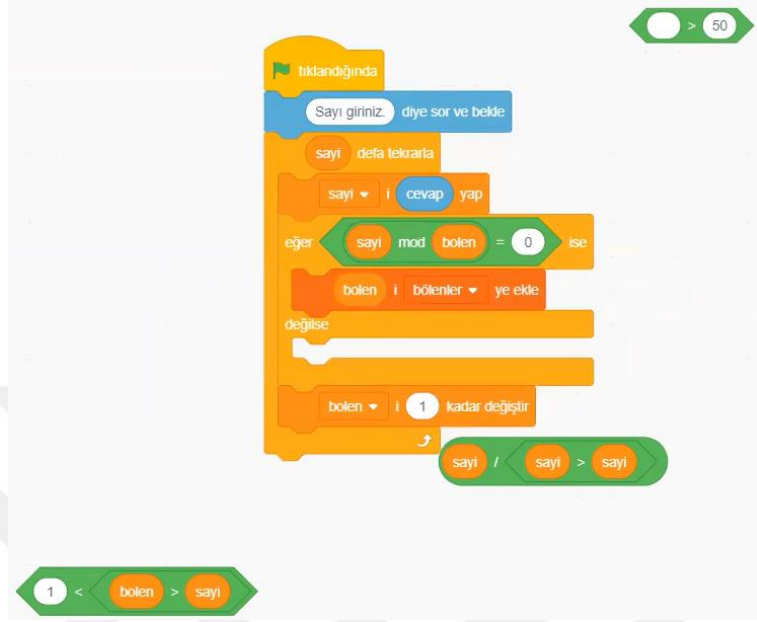
Erdem asal sayıların iki tane böleni olduğunu ifade etti ve Cem bu fikre katılıp bölen listesinde 2 tane bölen olduğunda sayının asal olacağını belirtti (T).

30. E: Asal olması için 2 tane böleni olması gerekir (A-D).

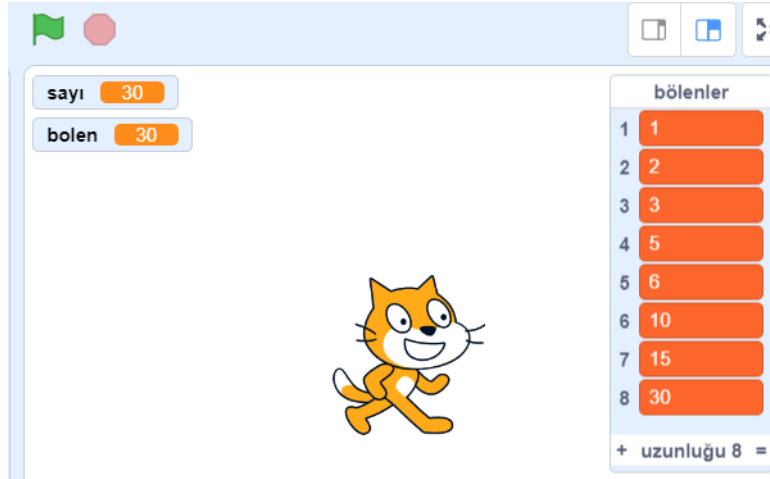
31. C: O zaman sayıyla 1 olduğunda iki tane oluyor ve sayı asal sayı oluyor. Öbür türlü 2’den az veya bir dakika 2’den az olamaz, ikiden fazla oluyor ve asal sayı olmamış oluyor. O zaman listede iki tane varsa asal sayıdır diyeceğiz (A-D). Tamam şimdi biz böleni nasıl yapacağız.

Bölen değişkenini nasıl tanımlamaları gerektiği üzerinde tartıştıktan “sayı defa tekrarlar bloğu” kullanıldı. Bloklar arasında dizilim üzerinde çalışılarak algoritma oluşturuldu ve çalıştırıldı. 30 sayısının tüm bölenlerinin ekranda doğru bir şekilde olduğu gözlemlendi.

Daha sonra farklı sayı denendiğinde ise önceki bölenlerin durduğu görüldü. Bu durumda “bölenlerin her şeyini sil” bloğu eklendi ama algoritma yine doğru sonucu vermedi. Erdem Şekil 4.6.’daki bloğu oluşturdu (K).



Şekil 4.6. Erdem’in Oluşturduğu Algoritma



Şekil 4.7. Algoritma Oluşturulduktan Sonra Ekranda Gözüken Bölenler Listesi

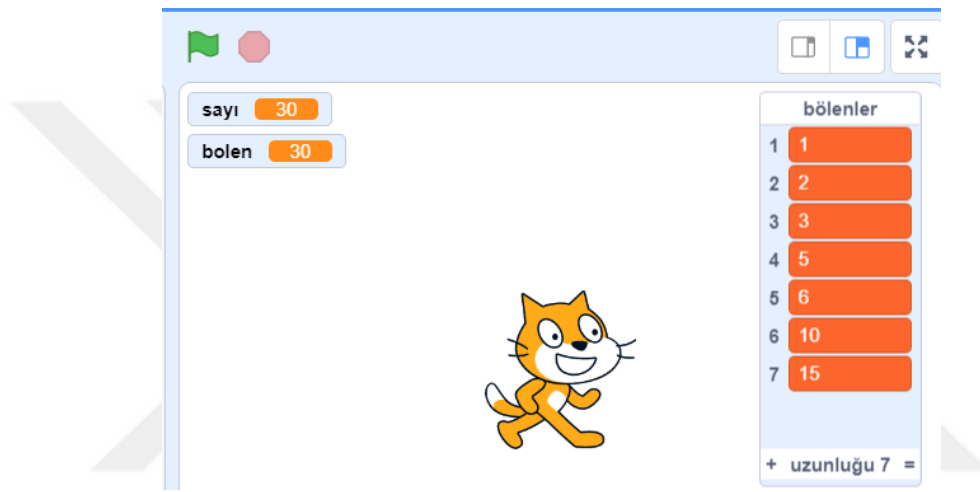
32. E: Hocam ama “değilse” bloğu...

33. E: Hocam mesela 7 yazdığında bölenler sadece 1 ve 7 olması gerekiyor. Fakat burada böyle olmadı. Önceki sayılar hala duruyor. Bunları temizlememiz lazım.

34. C: Daha kısa olsun diye evet değilse kısmını çıkartalım. Çünkü algoritmalar olabildikleri kadar kısa olması gerekir (A-D).

Öğrenciler gereksiz gördükleri “değilse” bloğunu kaldırdı (O). Algoritma kod blokları arasında tekrar bir düzenleme yapıldı ve tekrar çalıştırıldı. 30 sayısının kendisi hariç tüm bölenleri ekranda belirdi. Bu durumun yanlış olduğu 30 sayısının da listede olması gerektiği Erdem tarafından söylendi (A-D).

Erdem blokların yerlerini değiştirdi (K ve O). 30 sayısını algoritmaya girildi ve Şekil 4.8.’de gösterildiği gibi sahnede 30 sayısının kendisi hariç bölenleri oluştu.



Şekil 4.8. 30’un Bölenlerinde 30 Sayısının Olmadığı Durum

Etkinliğe fazla katılım sağlamayan Yalçın ekrandaki gibi bir algoritma kendi bilgisayarında oluşturduğunu ancak sonucu vermediğini söyledi. Bu sırada Erdem ve Cem “Algoritma sayının kendisini bölen listesine eklemiyorsa “sayıyı bölenler listesine ekle” bloğunu algoritmaya ekleyelim.” dedi [35].

Erdem bu blokları algoritmaya yerleştirir. Algoritma çalıştırılır ve sayının kendisi hariç tüm bölenleri ekranda görünür.

35. C: Bir şey buldum! Bir şey buldum! Madem kendisini eklemiyor.

36. E: Sayıyı bölenler listesine ekle (A-D).

37. C: Evet ondan.

Erdem “sayıyı bölenler listesine ekle” bloğunu yerleştirdi. Farklı sayılar denendi. Sonuçlardan anlaşıldı ki tüm bölenler oluşturulmuş oldu. Sayının kendisini en başa

The image shows a Scratch environment with a light blue background. In the top-left corner, there is a green flag icon and a red octagon icon. In the top-right corner, there are three icons: a window icon, a blue square icon, and a zoom icon. On the left side, there are two input fields: "sayi" with the value "25" and "bolen" with the value "25". In the center, there is a cartoon cat character (Scratch cat) with orange fur, white belly, and a white bow tie, standing and smiling. On the right side, there is a "bölenler" (divisors) list. It contains three items: "1", "5", and "25". Below the list, there is a text field with the text "+ uzunluğu 3 =".

Oluşturulan algoritma denendi ve doğru sonuçları verdiği öğrenciler tarafından gözlemlendi. Cem programın hata verip vermeyeceğini belirlemek için çok büyük bir sayının yazılmasını teklif etti [38]. Ekranı 99999999999999999999 sayısı yazıldı ve algoritma çalıştırıldı (K). Ekranı yazılan sayının çok büyük olmasından dolayı bölenlerinin oluşturulması çok uzun sürdüğü fark edilerek bu süreç burada kesildi.

[illegible]

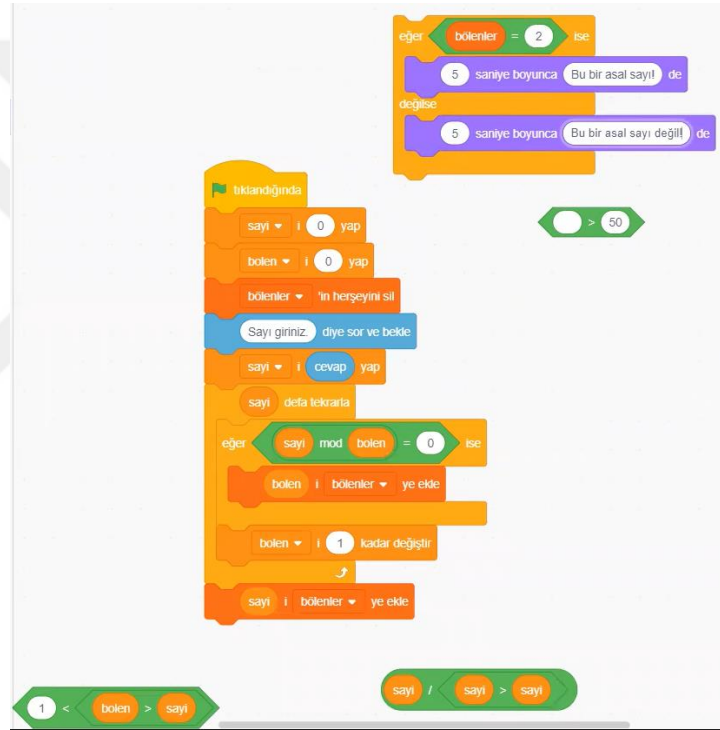
71

Algoritmada bir hatanın olmadığı belirlenerek bir sayının asal olup olmadığı ekrandaki karakter tarafından gösterilmesi aşamasına geçildi.

39. C: Evet crash (hata) da olmadı. Programın güzel tarafı da bu oldu. Bazı uzun programlarda Scratch donuyor ve bozuluyor. Bizde böyle bir şey olmadı. Bu da iyi...

40. A: Evet bölenleri doğru bir şekilde oluşturduunuz. Şimdi asal mı değil mi bunu belirlemeniz gerekiyor.

“Değişkenler” bölümünden “bölen listenin uzunluğu” öğrenciler tarafından kullanıldı. Erdem bölenler “listesinin uzunluğu eşittir 2” diye bir blok ekledi (O). Erdem Şekil 4.11’deki kod bloğunu oluşturdu.



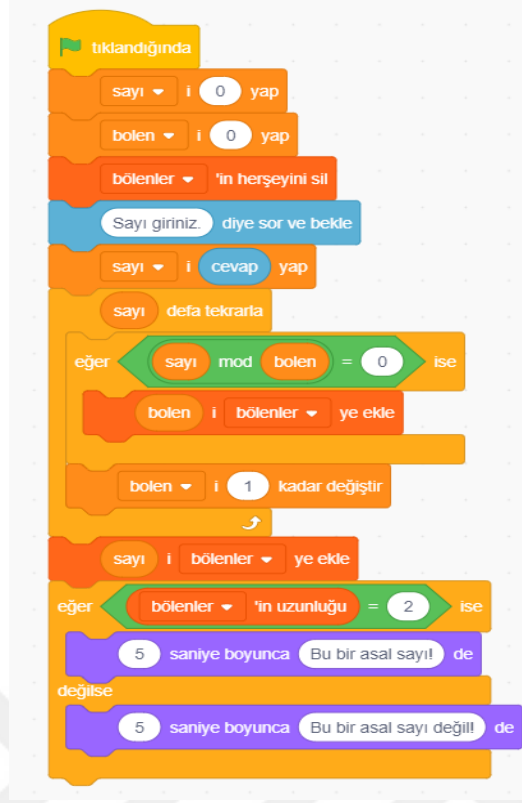
Şekil 4.11. Bölen eşittir 2 bloğunun oluşturulması

Asal sayı için yapılan blok en sona eklendi. Algoritma çalıştırıldı ve 30 için doğru sonucu verdiği görüldü.

41. C: Bir tane asal olmayan yap bir tane de asal olan yap ki tam anlayalım (A-D).

Algoritmaya 7 sayısı girilir. Bölenler doğru oluşur ama “7 asal sayı değil!” yazısı ekranda belirir.

42. C: Liste sekmesine bir göz atalım orada neler var acaba bu şekilde olmadı çünkü.



Şekil 4.12. Oluşturulan Algoritmanın Son Hali

43. C: Hocam bir program yapabildik sonunda.

44. A: Evet bir ara yapamayacak gibiydiniz ama yapabildiniz tebrik ederim.

Cem grup çalışmasının oldukça faydalı olduğunu belirtmiş ve etkinlik bu şekilde bitirilmiştir.

45. C: İpucundan bize yardım eden şey şu hocam; yalnız olunca beyin böyle sen yalnızsın yapmayacaksın gibi oluyor ama tartışmalı olunca mutlu oluyorsun. Bir de tartışa tartışa biraz fikirler çıkabiliyor. Beraber yapınca çok daha iyi oldu.

4.1.2. Algoritma Yapısına Ait Bulgular

Öğrenciler pozitif bölenleri listeleme etkinliğini tamamlamışlardır ve Şekil 4.12.’deki algoritmayı oluşturmuşlardır. Algoritma etkinlikten istenen sonuçları doğru bir şekilde vermektedir. Oluşan bu algoritmanın yapısını değerlendirmek için ilk olarak “Algoritma Yapısı Değerlendirme Rubriği” kullanılmıştır. Öğrencinin oluşturdukları algoritmanın her bir satırındaki bloklar 1’den başlanarak numara verilmiştir. Puanlama yapılırken her bir özellik için blokları verilen numaralardan yararlanılmıştır.



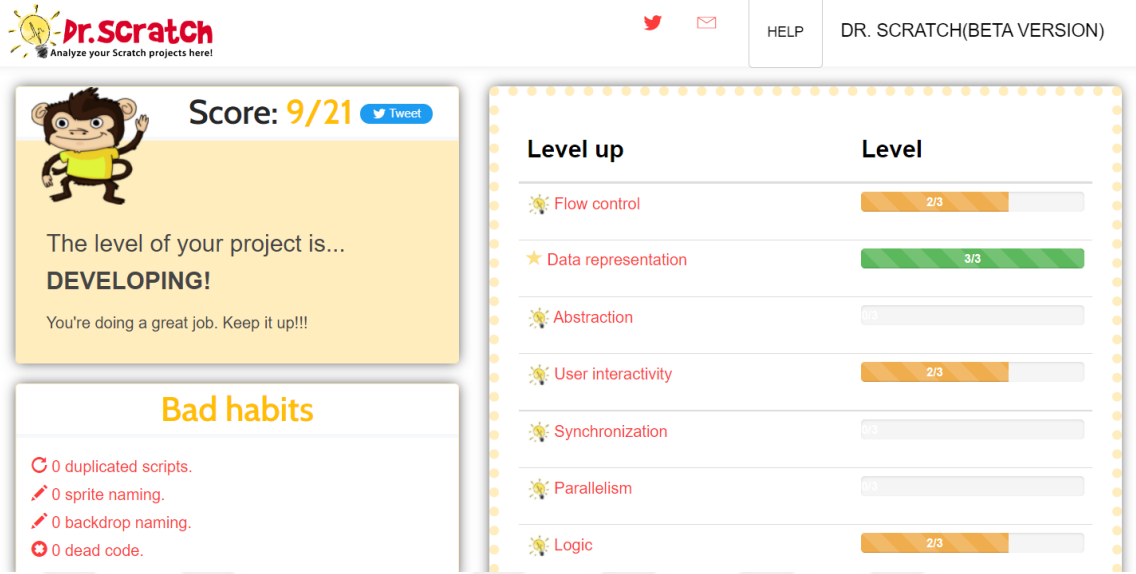
Şekil 4.13. Öğrencilerin Pozitif Bölenleri Listeleme Etkinliği İçin Oluşturdukları Algoritma ve Bloklara Verilen Numaralar

Tablo 4.1. Algoritma Yapısı Değerlendirme Rubriğine Göre Pozitif Bölenleri Listeleme Etkinlik Puanı

Özellikler	Puan	Blok Numarası
Sonlu Olma	2	7 ve 12
Kesinlik	2	Bütün Numaralar
Girdi	2	2,3 ve 4
Çıktı	2	9,11,13 ve 14
Etkileycilik	2	4,7,11 ve 12

Pozitif bölenleri listeme etkinliği “Algoritma Yapısı Değerlendirme Rubriğine” göre değerlendirildiğinde puan 10/10 olarak çıkmaktadır. Öğrencilerin oluşturdukları algoritma doğru sonucu verdiği için puanın yüksek çıktığı düşünülmektedir.

Öğrencilerin oluşturdukları algoritma Dr. Scratch sitesine yüklenmiş ve Şekil 4.14’te verilmiştir.



Şekil 4.14. Dr. Scratch Programına Göre Pozitif Bölenleri Listeleme Etkinliğinin Puanlanması

Dr. Scratch programı bu araştırma kapsamında revize edilerek veri analizi kısmında düzenlenmiştir. Buna göre flow control (akış kontrolü), data representation (veri temsili), user interactivity (kullanıcı etkileşimi) ve logic (mantık) başlıkları altında değerlendirilme yapılmıştır. Bu başlıklara göre değerlendirme yapıldığı öğrencilerin yaptıkları algoritmanın puanı 9/12 olmaktadır. Ayrıca “kötü yapı” kısmında hepsinin sıfır olduğu görülmektedir.

4.2. Ardışık Sayıların Toplamı Etkinliğine Ait Bulgular

Bu etkinlik kapsamında ilk olarak öğrencilerle çevrimiçi bağlantı sağlanmıştır. Bağlantı sorunsuz bir şekilde sağlandıktan sonra etkinlik öğrencilere açıklanmıştır. Etkinlik 18 dakika sürmüştür. Etkinlik sonunda öğrencilerden üretmeleri beklenen algoritmanın başarı ile oluşturulduğu gözlemlenmiştir. Algoritma üretme süreci ve öğrencilerin oluşturdukları algoritmalar ile ilgili bulgular bir sonraki kısımlarda verilmiştir.

4.2.1. Algoritma Üretme Sürecine Ait Bulgular

Öğrencilere etkinlik açıklandıktan sonra Cem hemen buradaki matematiksel ifadeyi anlamıştır (Bkz [1,3]). Ardışık sayıların toplamı olduğunu ifade eden Cem grubun etkinliğe doğru ve hızlı bir şekilde başlamasını sağlamıştır. Cem'in algoritma mantığını hemen anlaması çalışmanın sürecini kısalttığı düşünülmüştür. Çünkü öğrencilerin bu algoritma mantığını anlamalarının zaman alabileceği düşünülmüştür.

1. C: Anladım hocam 10 ve 10'dan küçük sayıları topluyor yani (T, K ve O).
2. A: Cem tebrik ederim. Algoritmanın mantığını hemen buldun.
3. C: Nedense ben 1'den 9'a kadar olan sayıların toplamının 45 olduğunu biliyordum. 10 daha eklersek 55 olur. O zaman çok kolay şimdi (A-D).

İlk olarak 10 ve 10'dan küçük sayıların toplanması şeklinde bir çalışma planı oluşturuldu. Scratch programının kontrolü Yalçın'da olduğu için Yalçın sahnenin dekorunu değiştirdi.

4. C: Önceki projedeki gibi yapacağız ama bu sefer toplama (A-D).

Cem'in önceki projeyi anımsatması üzerine projenin dosyası açıldı ve oradan ipucu yakalanmaya çalışıldı. Erdem tekrar "sayı artı sayı eşittir küçüktür sayı" bloğunu kullanmak istedi.

5. E: Hocam burada öncelikle sayı değişkenini oluşturmamız gerekiyor. Sonra sonuç değişkeni lazım. Sayı artı sayı eşittir küçük gibi bir şey yapmamız gerekiyor (B-Y).
6. A: Önceki araştırmada buradan fazla gidememiştiniz. Burada nasıl bir yol izlemeliyiz?
7. E: Evet hocam "sürekli tekrarla" bloğundan sayıyı 1 kadar değiştir yerine -1 kadar değiştir yapmamız lazım (A-D), (O).

Yalçın, Erdem'in dedikleri blokları düzenledi ve en başa "tıklandığında" bloğunu ekledi (K). Algoritmanın tekrarla blokları üzerinden işlemi kısaltacak bir blok aranmaya başlandı. Bunun için "... defa tekrarla" ve "liste" bloklarının eklenmesi düşünüldü (K).

8. E: Değişkenlerden "sayıyı cevap yap" bloğunu ekleyelim (A-D).
9. C: Bir şey soracağım. Bir liste yapsak? Bir liste yapalım ve listedeki tüm sayıları topla diyelim (A-Y).

Yalçın "liste" oluşturdu.

10. C: Listedeki bütün sayıları topla diye bir şey var mı?



Şekil 4.15.Yalçın'ın Oluşturduğu Bloklar Dizini

Liste sekmesinde böyle bir blok arandı ama bulunamadı.

11. E: Hocam ilk önce sayıyı giriyoruz. Sonra sayıyı cevap yapıyoruz. Sonra sürekli tekrarlar da sayı artı...Sayıyı -1 kadar değiştir ve bunu tekrarlamamız lazım (A-D), (O).

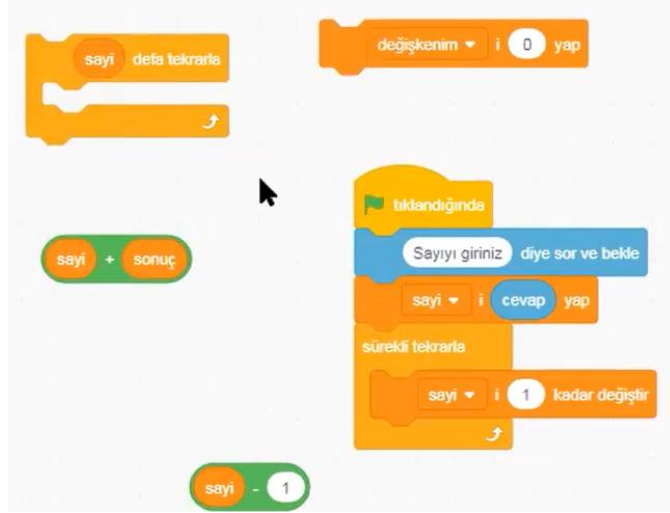
Yalçın “sayıyı 1 kadar değiştir” bloğunu ekrana getirdi.

12. A: -1 demenize gerek yok “1 kadar değiştir” demeniz yeterli.

Araştırmacı burada 1’den n’ye kadar sayıların toplamı şeklinde düşündüğü için böyle ifade kullandı [12]. Öğrenciler ise bunun tam tersi n’den 1’e kadar sayıların toplamı şeklinde düşündükleri için -1 ifadesini kullandılar [11].

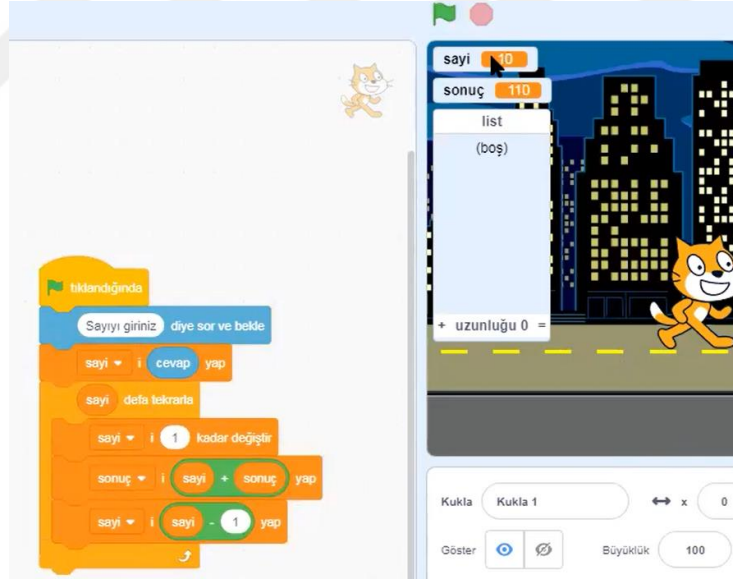
13. E: Sonuç diye oluşturmuştuk “sayı artı sonuç”. Hocam ama 1 kadar değiştir dediğimizde mesela 5 girdiğimizde 5,6,7 diye gider (A-D), (T).
14. C: Hocam şöyle yapabiliriz; “sayı artı sonuç” sonra “sayı eksi 1” böyle forever (sonsuz kadar) bir şey yapabiliriz. “Sayı eşittir sıfır” olana kadar (O).
15. A: Bir deneyin bakalım.
16. C: Ama bize forever lazım ya da sayı defa lazım.
17. E: “Sayı defa tekrarlar” yapacağız o zaman.

Yalçın Şekil 4.16.’daki bloklar zincirini oluşturdu (K).



Şekil 4.16. Yalçın'ın "Sayı Defa Tekrarla" Bloğunu Ekleme

Grup içinde kod blokların yer değişimi ile fikir ayrılığı yaşandı. Yalçın ve Cem algoritmayı Şekil 4.17.'deki gibi düzenleyip çalıştırmak istediler. Erdem ise algoritmanın doğru çalışmayacağını ifade etti (B-D). "Tıklandığında" ifadesine basıp 10 sayısını girdiler ve Scratch programı sonuç olarak 110 sayısını verdi.



Şekil 4.17. Algoritmaya 10 Sayısı Girildiğinde Beliren 110 Sonucu

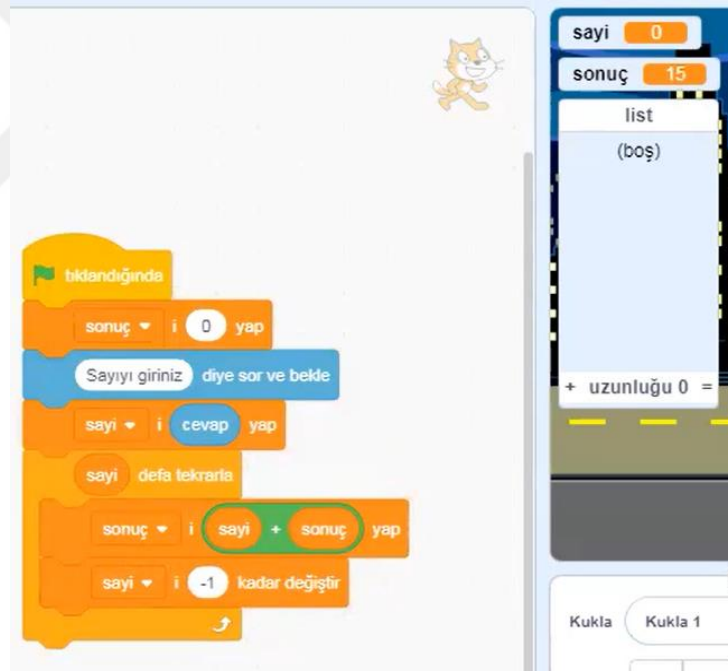
Erdem blokların yer değiştirmesi gerektiğini ifade etti. Cem ise "sonuç bölü 2 yapalım" diyerek doğru sonuç olan 55'e ulaşacaklarını ifade etti (A-Y). "Sonuç bölü 2" bloğu algoritmanın en alt kısmına eklendi (K). Algoritma denendi fakat yine doğru sonuç çıkmadı. Burada öğrenciler "sonuç" değişkenini sanki algoritmanın sonucu gibi algılandı. "Sonuç bölü 2" bloğu bu yüzden işe yaramadı.

18. E: Hocam sayıyı 1 arttırıyoruz sonra “sayı artı sonuç” yapıyoruz sonra tekrar 1 azaltıyoruz. Yani bu 2 katına sebep oluyor. “Sayı – 1” bloğuna gerek yok ve “sayıyı -1 kadar değiştir” dememiz gerekiyor. Bu şekilde her seferinde azalır ve sıfıra kadar gider (A-D).

Erdem burada algoritmanın sorununu hem matematiksel açıdan hem de blokların yerleri açısından doğru bir şekilde analiz etti (K ve O).

19. E: Hocam bence “sonuç bölü 2” ve “sayı – 1” bloklarını kaldırmalıyız. Sonrasında “sayıyı -1 kadar değiştir” ve “sonucu sayı artı sonuç yap” bloklarının yerlerini değiştirmeliyiz. Ayrıca sonucu sıfırlamalıyız (A-D).

Erdem’in dediği gibi [19] algoritma Şekil 4.18.’deki gibi oluşturdu (K ve O). Daha sonra algoritma çalıştırıldı ve doğru sonuçlar verdiği gözlemlendi. Örneğin 5 sayısı girildiğinde ekranda 15 yazısı belirdi.

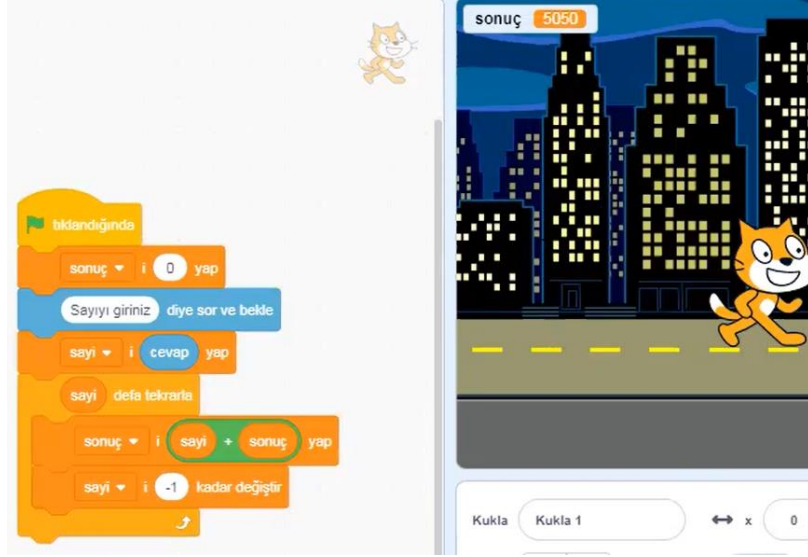


Şekil 4.18. Erdem’in Dedikleri [19] Kapsamında Oluşan Algoritma

20. Y: Ekranda sayı sıfır gözüküyor ve liste boş.

21. C: Sayı ve listeyi eleyin. Zaten bunlara gerek yok (A-D).

Cem’in dediği şekilde liste ve sayı bölümü sahneden silindi. Şekil 4.19.’daki gibi algoritma oluşturuldu ve çeşitli sayılar denendi. Algoritmanın doğru sonuçlar verdiği gözlemlendi.



Şekil 4.19. Algoritmanın Son Hali ve Algoritmaya 100 Sayısı Girilince Ekranda Beliren Sonuç

22. A: Bu etkinlik başarı ile bitirmiş oldunuz. Beklediğimden daha hızlı yaptınız.
23. C: Hocam çok eğlenceliydi. Bence başarma nedenimiz grup ile beraber çalışmak. Yalnız olsak yapamazdık bence.
24. E: Evet bence de herkesten farklı fikirler çıktı.

Bu etkinliğin hemen bitme sebebi bir önceki algoritmadan yararlanmaları olabileceği düşünülmüştür. Aynı zamanda Cem'in 1'den 10'a kadar olan tüm sayıların toplamını otomatik olarak zihnine getirebilmesi de çalışmayı hızlandırmıştır. Yalçın'ın bir önceki etkinliğe daha az katılmasından dolayı kontrol ona verilmiştir. Böylece öğrenci etkinliğe daha aktif katılabilmiş ve 3 kişinin el birliğiyle çalışma kısa sürmüştür.

4.2.2. Algoritma Yapısına Ait Bulgular

Öğrenciler ardışık sayıların toplama etkinliğini tamamlamışlardır ve Şekil 4.20.'deki algoritmayı oluşturmuşlardır. Algoritma etkinlikten istenen sonuçları doğru bir şekilde vermektedir. Oluşan bu algoritmanın yapısını değerlendirmek için ilk olarak "Algoritma Yapısı Değerlendirme Rubriği" kullanılmıştır. Öğrencinin oluşturdukları algoritmanın her bir satırındaki bloklar 1'den başlanarak numara verilmiştir. Puanlama yapılırken her bir özellik için blokları verilen numaralardan yararlanılmıştır.



Şekil 4.20. Öğrencilerin Ardışık Sayıların Toplamı Etkinliği İçin Oluşturdukları Algoritma ve Bloklara Verilen Numaralar

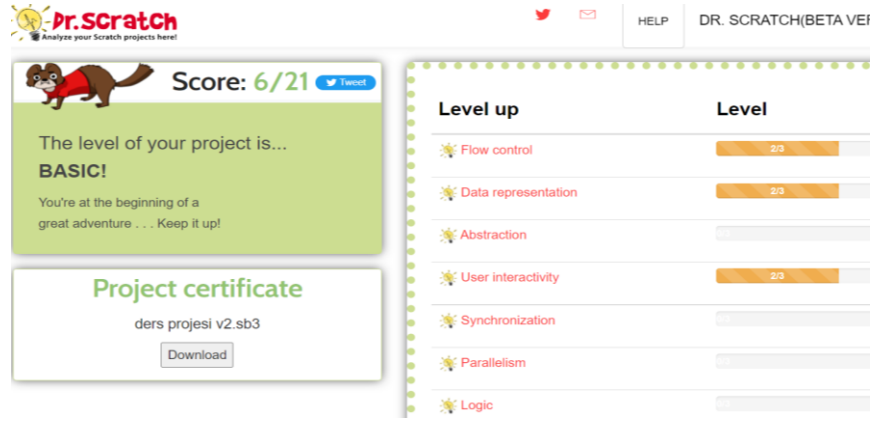
Tablo 4.2. Algoritma Yapısı Değerlendirme Rubriğine Göre Ardışık Sayıların Toplamı Etkinlik Puanı

Özellikler	Puan	Blok Numarası
Sonlu Olma	2	5
Kesinlik	2	Bütün Numaralar
Girdi	2	2,3 ve 4
Çıktı	2	6
Etkileycilik	2	5,6 ve 7

Ardışık sayıların toplamı etkinliği “Algoritma Yapısı Değerlendirme Rubriğine” göre değerlendirildiğinde puan 10/10 olarak çıkmaktadır. Öğrencilerin oluşturdukları algoritma doğru sonucu verdiği için puanın yüksek çıktığı düşünülmektedir.

Öğrencilerin oluşturdukları algoritma Dr. Scratch sitesine yüklenmiş ve Şekil 4.21.’de verilmiştir.

Dr. Scratch programı bu araştırma kapsamında revize edilerek veri analizi kısmında düzenlenmiştir. Buna göre flow control (akış kontrolü), data representation (veri temsili), user interactivity (kullanıcı etkileşimi) ve logic (mantık) başlıkları altında değerlendirilme yapılmıştır. Bu başlıklara göre değerlendirme yapıldığı öğrencilerin yaptıkları algoritmanın puanı 6/12 olmaktadır.



Şekil 4.21. Dr. Scratch Programına Göre Ardışık Sayıların Toplamı Etkinliğinin Puanlanması

4.3. Oyun Etkinliğine Ait Bulgular

Etkinliğin başında öğrencilerle çevrimiçi bağlantı sağlanmıştır. Bu bağlantıdan sonra araştırmacı tarafından etkinlik öğrencilere açıklanmıştır. Etkinlik 30 dakika sürmüştür. Etkinlik sonunda öğrencilerden üretilmesi beklenen algoritmanın başarı ile oluşturamadıkları gözlemlenmiştir. Algoritma üretme süreci ve öğrencilerin oluşturdukları algoritma ile ilgili bulgular bir sonraki kısımlarda verilmiştir.

4.3.1. Algoritma Üretme Sürecine Ait Bulgular

Öğrenciler oyunun kurallarını anladıklarını ifade ederek etkinliği başlamışlardır.

1. E: Hocam ben oyunu anladım ama algoritma nasıl yapacağımıza dair bir fikrim yok. Oyunda sayıyı alırken bir yanındaki sayı ile düşünmemiz gerekiyor. Örneğin 5, 8, 1, 3 sayı dizilişini düşünelim. Burada en sağdaki ve soldaki sayılara bakarsak 5 ve 3 var. 5'i almak mantıklı gibi duruyor büyük olduğu için ama biz 5'i alırsak diğer oyuncuya 8 alma şansı doğuyor. 8'i aldığında ise bizim önümüze geçmiş oluyor (A-D), (T, K ve O).
2. E: Hocam burada raunt başına alacağımız sayı yüksek olursa en sonda da puanlar toplanırken bizim sayımızda yüksek çıkar (A-Y).

Erdem burada ilk oluşturduğu yöntem ile çelişkili bir ifade kurdu. Başlangıçta ardındaki sayılara bakmamız gerekiyor dedi ama sonra bu fikrinden vazgeçti sadece sıra bize geldiğinde hangi sayı büyükse onu almamız gerekli ifadesini kullandı (B-D)

3. C: Nasıl yeneceğini tam olarak bulmalıyız. Yani bir algoritma lazım yenmek için. Mesela biz iki tane algoritma yapalım. Bu algoritmaları birbiri ile karşılaştıralım ve büyük olan kazanmış olur (K).
4. A: Evet güzel fikir. Peki bu algoritmaları nasıl oluşturabilirsiniz?
5. E: Hocam değişkenlerden ilk önce “sayı 1, sayı 2... sayı 10” oluşturmamız gerekiyor. Sonra sayı 1’e rastgele olarak 1’den 10’a kadar rastgele bir sayı verilmesi lazım. Bunu tüm sayılar için rastgele oluşturmalıyız (K).

Erdem burada algoritmanın girdilerini tanımlamaya yönelik bir strateji ortaya koydu (A-D). Fakat halihazırda tam bir başlama prosedürü bulunamadı.

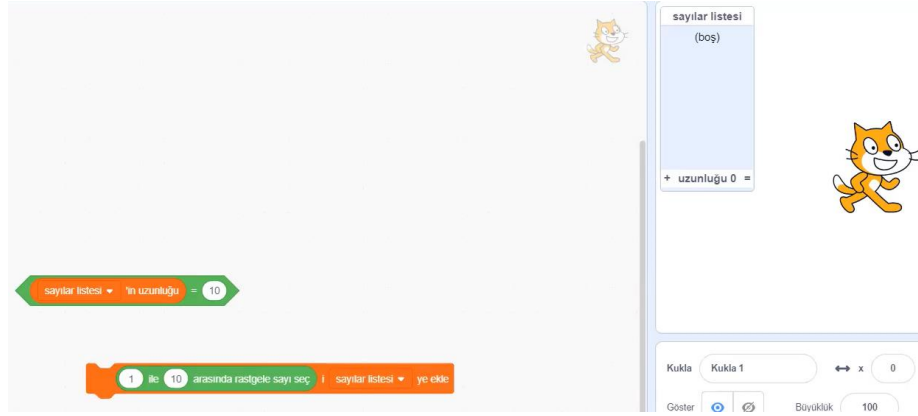
6. C: Listeden de 1’den 10’a kadar sayılar oluşturabiliriz.

Erdem’de bu fikre katıldı. Cem sayılar adında liste oluşturdu.

7. E: “Sayıların uzunluğu=10” yapabiliriz. Daha sonrasında ise “Sayıların ögesi” bloğundan “sayılar büyüktür sıfır küçüktür 11” gibi bir şey dememiz gerekiyor (K).
8. C: Hocam bunun için bir algoritma yapamayız ki imkânsız.
9. E: Yapabiliriz.
10. A: Bununla ilgili bir algoritma ben oluşturabildim. Sizlerin de biraz daha uğraşmanızı istiyorum.
11. E: İlk önce sayıları oluşturmamız gerekli. Daha sonrasında ise oyunu kuralları ile oynatmamız gerekiyor. Bunların sonuçlarına göre kazanma stratejisini bulabiliriz (T).

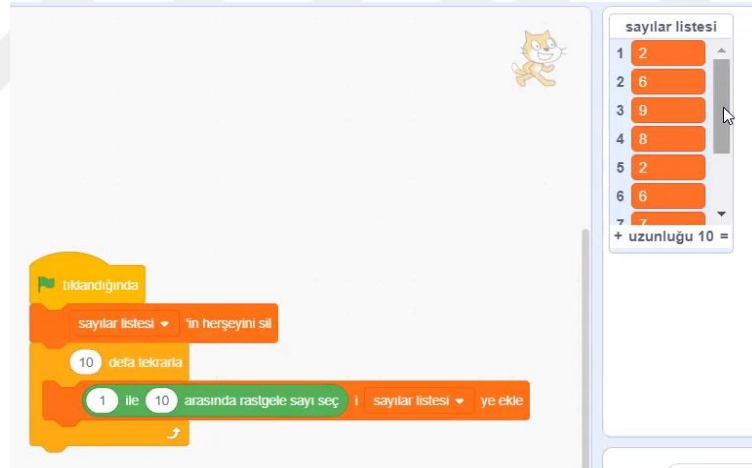
Etkinlik başlamasından 10 dakika kadar geçmesine rağmen öğrenciler henüz bir fikir, bir kod blokları örnekleri ortaya çıkaramadılar.

12. A: Dediğiniz gibi yapmaya çalışın. İlk önce oyunun kurallarına göre bir algoritma üretin. Daha sonra strateji geliştirmeye çalışırsınız.
13. E: Bizim için blok yapmışlar. “1 ile 10 arasında rastgele sayı seç” bloğunu kullanmalıyız.



Şekil 4.22. Etkinlikte Oluşturulan İlk Bloklar

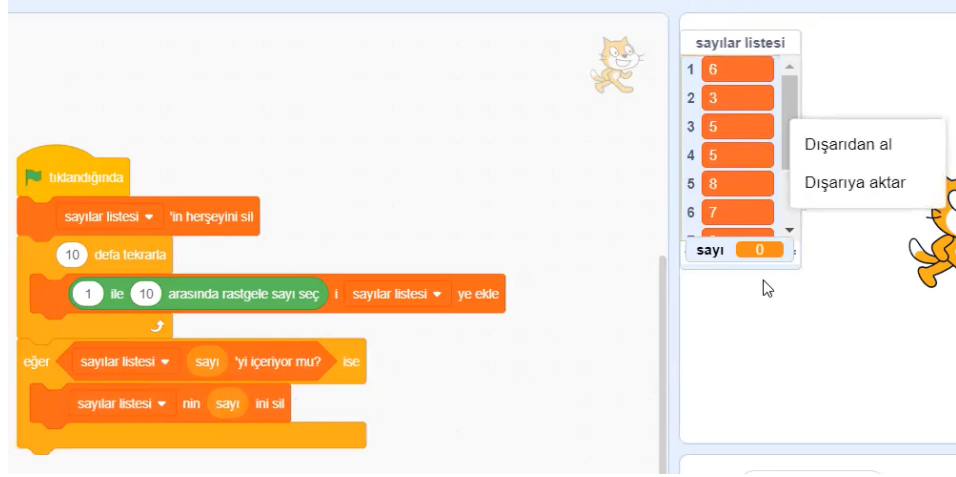
Daha sonra “tıklandığında”, “10 defa tekrarla” ve “sayılar listesinin her şeyini sil” blokları eklendi ve denemeler yapıldı (K). Oluşan kod bloklarına göre algoritma her seferinde 1 ile 10 arasında sayı üreterek listeye eklendi. Ancak buradaki sorun listeye aynı sayının birden fazla ekleniyor olmasıydı.



Şekil 4.23. Listede Aynı Sayıların Olması Sorunu

14. A: Evet sayıları oluşturduğunuz ama aynı sayılardan var. Her sayıdan birer tane olması gerekiyor. Bunu da hallettiğinizde oyunu oluşturmuş olacaksınız.
15. E: Burada bir blok var. “Sayılar listesi sayıyı içeriyor mu?” bloğunu kullanmayı deneyelim. Ayrıca “sayılar listesinin ... ini sil” bloğunu kullanabiliriz (K).

Bu kısımda Erdem tek başına etkinliği yürütmeye çalıştı. Erdem Şekil 4.24.’teki gibi bir algoritma oluşturdu ve tekrar denedi. Aynı sayıların listenin içinde olduğu görüldü.

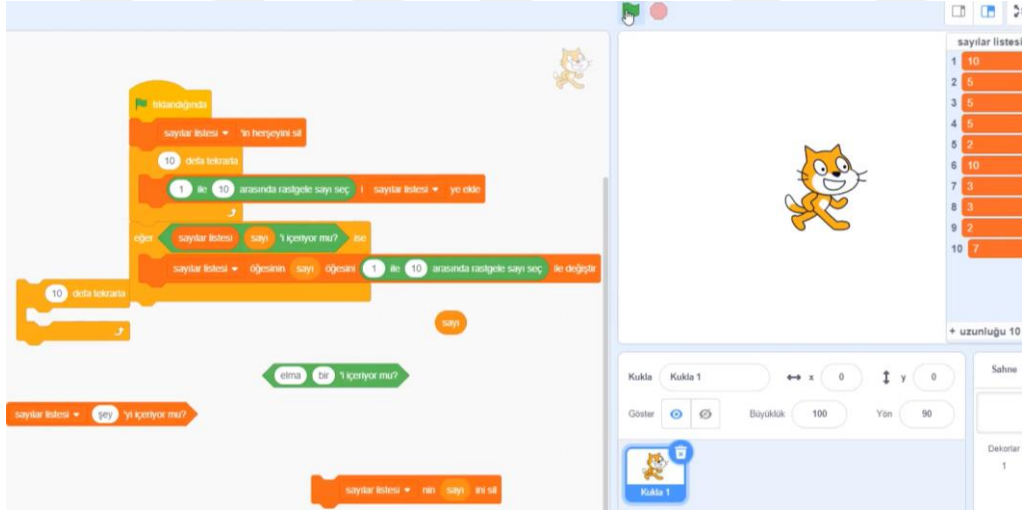


Şekil 4.24. Aynı Sayıların Listede Yer Alma Sorununun Devam Etmesi

16. E: Hocam tek başıma yapmaya çalışıyorum.

17. C: Hocam bu çok zor. Ben aslında kâğıt üzerinde bir algoritma yapmak için uğraşıyordum ama pes ediyorum ben de bu 10 tane sayı üretme işini giriyorum. Aslında tam bir şey bulduğumu düşündüğümde deniyorum. Yanlış bir şeyler çıkıyor.

Bu durumda öğrenciler birçok kod bloklarını deneyerek algoritmayı oluşturmayı denedi ama oluşan her algoritma aynı sayı çıkma sorununu çözemedi.



Şekil 4.25. Birçok Kod Blokları Denenmesine Rağmen Sorunun Devam Etmesi

18. E: Hocam biz bu şekilde 10^{10} defa deneyelim. Bu ihtimallerin birinde istediğimiz gibi her sayıdan birer tane gelmiş olacak (B-Y).

19. E: “Sayılar listesinde aynı sayı 2 defa içeriyor mu” bir blok olması lazım ama böyle bir blok yok.

20. C: Hocam pes edebiliyor muyuz?

21. A: Tabi ki, oluşturmayaacağınızı düşündüğünüzde bırakabilirsiniz.

22. C: Biraz daha düşünelim bakalım. Olmazsa pes edelim.

Biraz daha denemeler yapıldı fakat bir sonuç çıkmadı.

23. E: Hocam ilk önce 1'den 10'a kadar sayıları tekrarsız bir şekilde oluşturmamız lazım.

Daha sonra programa sağdan ve soldan sayılar seçeceğini içeren bir algoritma oluşturmamız lazım. Daha sonra her zaman kazanma algoritmasını üretip bu kodlara göre bloklar halinde yazmamız lazım ama daha sayıları oluşturmamıştık.

Öğrenciler daha fazla bir şey üretemeyeceklerini söylediler ve etkinlik sona erdi. Öğrencilere bu etkinliğin son etkinlik olduğu söylendi. Katılımları için teşekkür edildi.

24. C: Hocam biz de teşekkür ederiz. Ben bu etkinliklerden Scratch bloklarını daha iyi öğrenmiş oldum. Ayrıca etkinlikleri grup halinde beraber yapınca çok daha kolay oluyor.

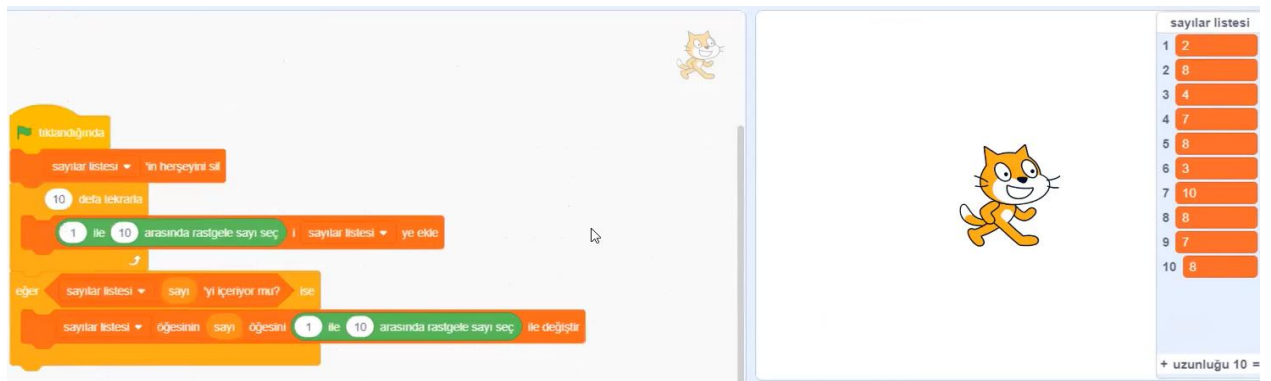
25. A: Matematik açısından sizce bu etkinliklerin nasıl faydası oldu?

26. E: Hocam matematik derslerimiz bu şekilde işlenirse formüllerin mantığını daha iyi anlarız ve daha akılda kalıcı olur konular.

27. C: Burada kendi sorumluluğumuzu aldığımız için çok daha iyi öğrendik. Derslerde bu şekilde olsa daha aktif katılım ile daha iyi öğreniriz.

28. E: Scratch programı yanlış blok koyulmasına izin vermediği için daha iyi anlamamızı sağlıyor.

Üçüncü etkinlik bu şekilde bitirildi. Öğrencilerin oluşturduğu algoritmanın son hali Şekil 4.26.'da verilmiştir.



Şekil 4.26. Üçüncü Etkinlikte Öğrencilerin Oluşturduğu Algoritmanın Son Hali

4.3.2. Algoritma Yapısına Ait Bulgular

Öğrenciler oyun etkinliğini tamamlamışlardır ve Şekil 4.27.’deki algoritmayı oluşturmuşlardır. Algoritma etkinlikten istenen sonuçları doğru bir şekilde verememektedir. Oluşan algoritmanın yapısını değerlendirmek için ilk olarak “Algoritma Yapısı Değerlendirme Rubriği” kullanılmıştır. Öğrencinin oluşturdukları algoritmanın her bir satırındaki bloklar 1’den başlanarak numara verilmiştir. Puanlama yapılırken her bir özellik için blokları verilen numaralardan yararlanılmıştır.



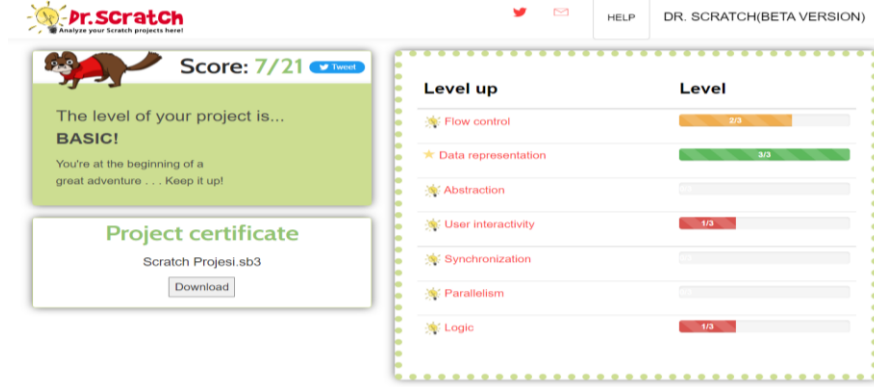
Şekil 4.27. Öğrencilerin Oyun Etkinliği İçin Oluşturdukları Algoritma ve Bloklara Verilen Numaralar

Tablo 4.3. Algoritma Yapısı Değerlendirme Rubriğine Göre Oyun Etkinlik Puanı

Özellikler	Puan	Blok Numarası
Sonlu Olma	1	3
Kesinlik	1	1,2,3 ve 4
Girdi	1	2
Çıktı	1	4
Etkileycilik	0	-

Oyun etkinliği “Algoritma Yapısı Değerlendirme Rubriğine” göre değerlendirildiğinde puan 4/10 olarak çıkmaktadır. Öğrencilerin oluşturdukları algoritma doğru sonucu veremediği için puanın düşük çıktığı düşünülmektedir.

Öğrencilerin oluşturdukları algoritma Dr. Scratch sitesine yüklenmiş ve Şekil 4.28’de verilmiştir.



Şekil 4.28. Dr. Scratch Programına Göre Oyun Etkinliğinin Puanlanması

Dr. Scratch programı bu araştırma kapsamında revize edilerek veri analizi kısmında düzenlenmiştir. Buna göre flow control (akış kontrolü), data representation (veri temsili), user interactivity (kullanıcı etkileşimi) ve logic (mantık) başlıkları altında değerlendirilme yapılmıştır. Bu başlıklara göre değerlendirme yapıldığı öğrencilerin yaptıkları algoritmanın puanı 7/12 olmaktadır. Etkinlik matematiksel açıdan istenilen sonucu vermemiştir. Ancak Dr. Scratch programına göre akış kontrolüne 2 puan vermesi algoritmanın içinde “... defa tekrarlar” bloğunun olmasıdır. Veri temsiline 3 puan vermesi de “değişken” ve “liste” kod bloklarının oluşturulmasıdır.

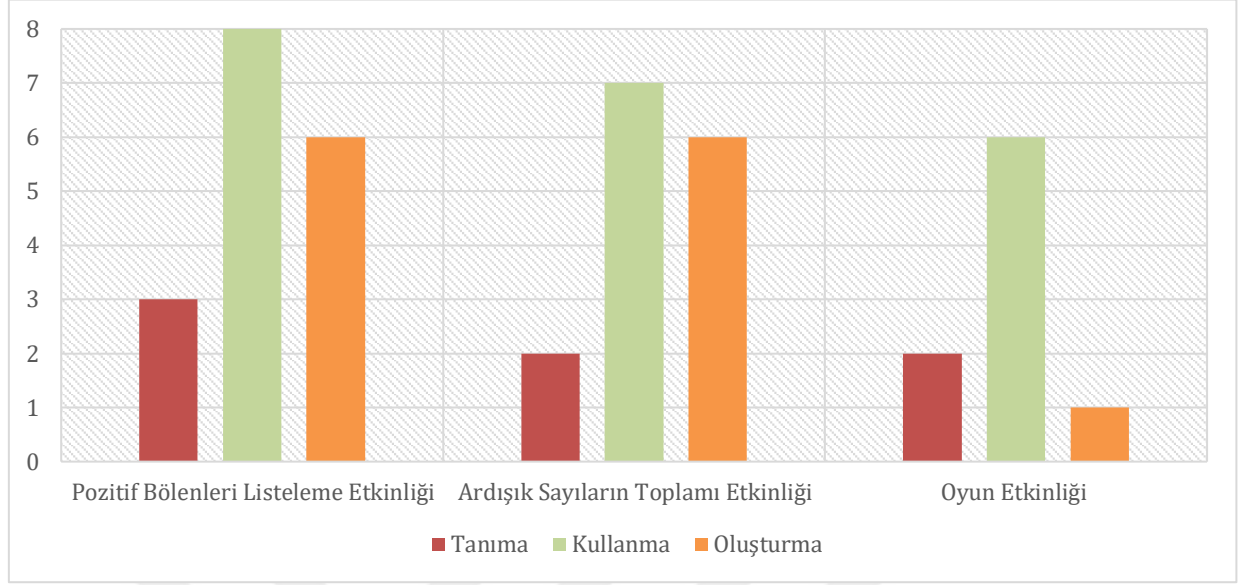
4.4. Etkinliklere Ait Genel Değerlendirme

Bu bölümde, öğrencilerin etkinlikler sürecinde sergiledikleri algoritma üretme süreçleri ve süreç sonunda oluşan algoritma yapıları ile ilgili genel bir değerlendirme sunulmuştur. Bu değerlendirme yapılırken araştırma soruları bağlamında ve araştırmada kullanılan dört adet veri analizi kapsamında değerlendirmeler sunulmuştur.

4.4.1. Etkinliklere Ait Algoritma Üretme Süreci ile İlgili Genel Değerlendirme

Öğrencilerin pozitif tam sayılar konusunda algoritma üretme süreçleri ile ilgili veriler RBC soyutlama teorisi ve öğrencilerin kullandıkları argümanların sınıflandırılması kapsamında değerlendirilmiştir. Bu değerlendirme de öğrencilerin etkinlikler sürecinde ifadelerinden doğrudan alıntılama yapılmış ve bu alıntılar cümle sonunda kodlar ile analiz edilmiştir.

Bu kodlamalar sonucunda öğrencilerin üç etkinlikte kullandıkları bilişsel eylemler Şekil 4.29’da verilmiştir.

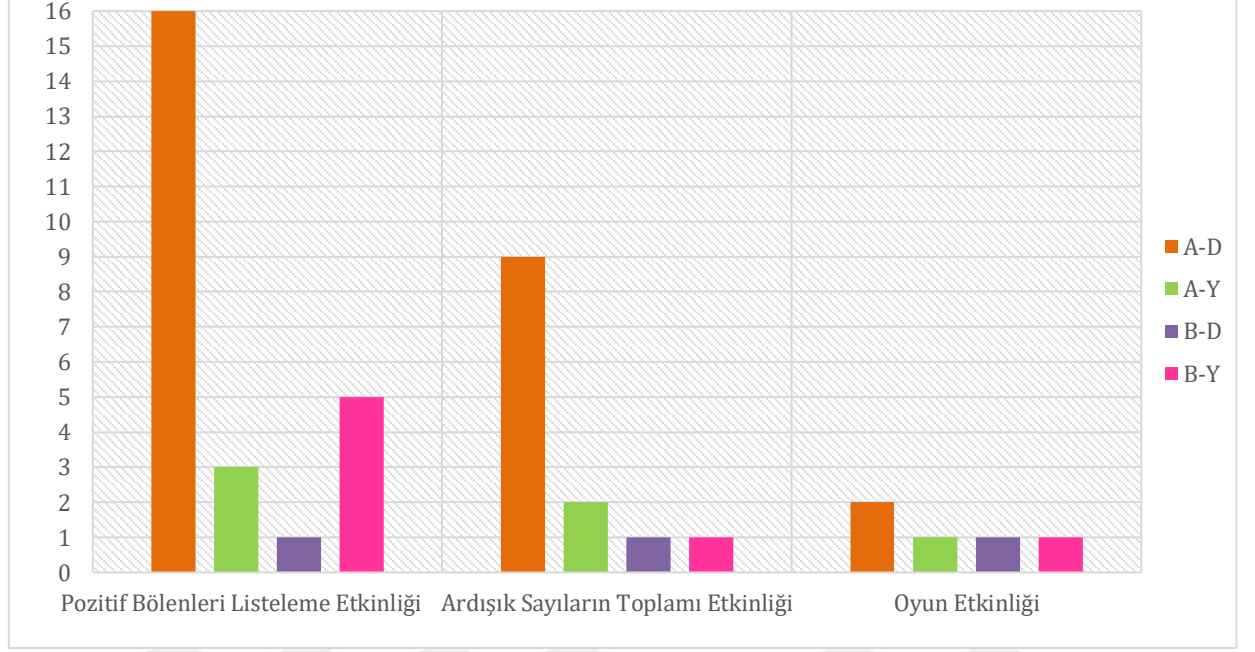


Şekil 4.29. Etkinliklerde Kullanılan RBC Soyutlama Teorisine Göre Bilişsel Eylemler

Öğrencilerin üç etkinlikte tanıma eylemlerinin birbirine yakın sayıda kullandığı görülmektedir. Öğrencilerin etkinliklerden istenen problem durumunu üç etkinlik içinde doğru anladığı ve bunu önceden bildiği pozitif tam sayılar konusu kapsamında tanıdığı görülmüştür. Öğrencilerin etkinlikler sürecinde kullandığı en çok bilişsel eylem kullanma eylemidir. Bunun nedeni olarak öğrencilerin algoritma üretmek için sürekli farklı yollar denemesidir. Üç etkinlik sürecinde gerçekleşen kullanma eylemlerine bakıldığında bu eylemin en çok pozitif bölenleri listeleme etkinliğinde olduğu görülmektedir. Çünkü bu etkinlikte öğrencilerin hem bir sayının pozitif tam bölenlerini hem de bu sayının asal olup olmadığını belirlemesi gerekiyordu. Başarısız olunan oyun etkinliğinde gerçekleşen kullanma eylemi başarılı olunan diğer etkinliklere göre çok düşük gözükmemektedir. Öğrenciler oyun etkinliğinde oyunun kurallarını anlayarak teorik olarak bir çözüm algoritması oluşturabilmişlerdir. Bu süreçte kullanma eylemini sıkça kullanmışlardır. Ancak bu teorik durumu Scratch programına aktaramayınca etkinlik başarısız olarak sonuçlanmıştır. Üç etkinlik sürecinde gerçekleşen oluşturma eylemleri incelendiğinde pozitif bölenleri listeleme ve ardışık sayıları toplamı etkinliklerinde eşit ve altı defa kullanıldığı görülmektedir. Oyun etkinliğinde ise oluşturma eylemi sadece 1 defa gözlemlenebilmiştir. Oluşturma eyleminin gerçekleşmesi için öğrencilerin dikey matematikleştirme ile nispeten yeni soyut bir ürün ortaya koyması gerekmektedir. Bu

durumu pozitif bölenleri listeleme ve ardışık sayıların toplamı etkinliklerinde öğrenciler ortaya koyarak başarılı bir şekilde algoritmayı üretebilmişlerdir. Oyun etkinliğinde ise bu eylemin 1 defa görülmesi etkinliğin başarısız sonuçlanmasına sebep olmuştur.

Öğrencilerin algoritma üretme süreçlerinde kullandıkları argümanlar sınıflandırılmış ve Şekil 4.30.'da verilmiştir.



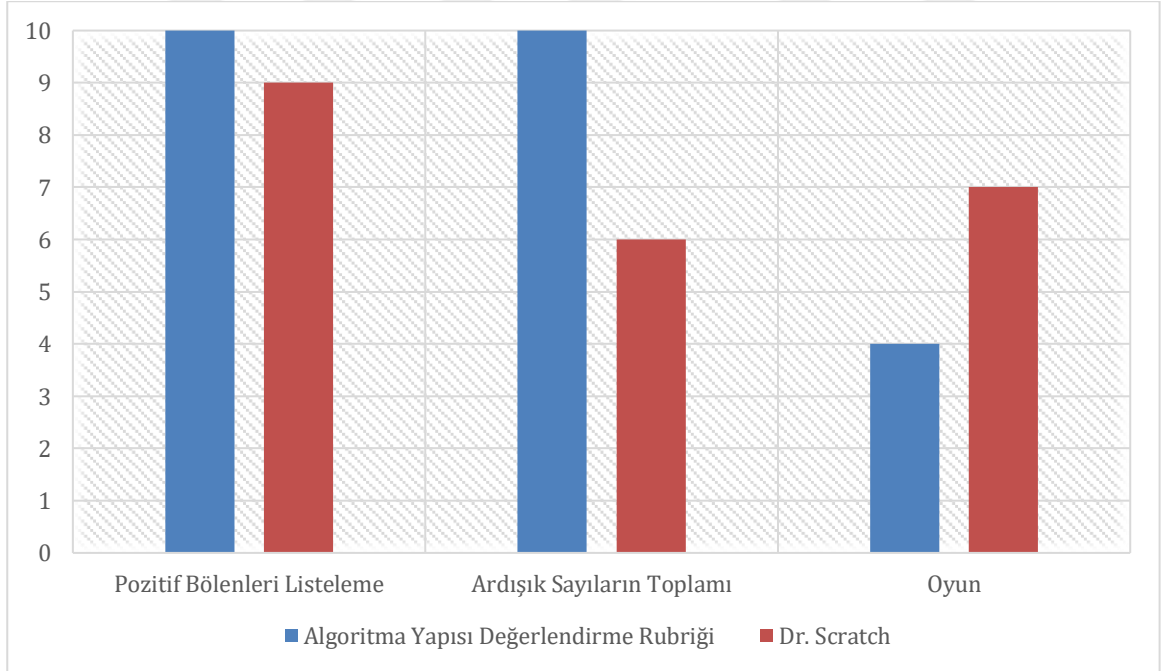
Şekil 4.30. Etkinliklerde Öğrencilerin Kullandıkları Argümanların Sınıflandırılması

Üç etkinlik kapsamında öğrencilerin en çok açıkça belirtilen ve doğru olan özellikleri kullandıkları görülmektedir. Öğrencilerin, pozitif tam sayılar konusunda doğru bilgilere sahip oldukları ve bunun farkında oldukları etkinlikler sürecinde kullandıkları ifadelerinden anlaşılmaktadır. Özellikle pozitif bölenleri listeleme etkinliğinde 16 defa açıkça belirtilen ve doğru olan özellikleri ifade ettikleri görülmektedir. Bu etkinlikte öğrenciler diğer etkinliklere göre çok daha fazla argümanlar üretip grup içinde tartışmışlardır. Bu nedenle açıkça belirtilen ve doğru olan özelliklerin sayısı daha fazla çıkmıştır. Başarılı olunan bir diğer ardışık sayıların toplamı etkinliğinde ise öğrenciler 9 defa açıkça belirtilen ve doğru olan özellikleri kullanmışlardır. Etkinliğin başında Cem'in algoritma mantığını çözmesi bu özelliği sınırlı sayıda tutmuştur. Oyun etkinliğinde ise öğrencilerin açıkça belirtilen ve doğru özellikleri sadece 2 defa kullanabildikleri görülmektedir. Öğrenciler bu etkinlikte doğru bir algoritma yapısı ortaya koyup etkinliği devam ettiremedikleri için bu özellik düşük sayıda kalmıştır. Öğrenciler üç etkinlik boyunca açıkça belirtilen ve yanlış olan özellikler ile açıkça belirtilmeyen ve doğru olan özellikleri sınırlı sayıda kullandıkları gözlemlenmiştir.

Öğrenciler açıkça belirtilmeyen ve yanlış olan özellikleri en çok pozitif bölenleri etkinliğinde kullanmışlardır. Başarılı oldukları bu etkinlikte öğrencilerin sürekli denemeler yapması ve sürekli farklı düşünceleri grup içinde tartışması bu sonucu ortaya çıkarmıştır. Ayrıca bu gibi etkenler pozitif bölenleri listeleme etkinliğinde diğer etkinliklere göre özelliklerin toplamda daha fazla ortaya çıkmasını sağlamıştır. Ardışık sayıların toplamı etkinliğinde açıkça belirtilen ve doğru özellikler dışındaki özelliklerin az görülmesi, etkinliğin kısa ve doğru bir süreç içinde gerçekleştiğinden dolayıdır. Oyun etkinliğinde ise tüm özellikler oldukça az kullanılmıştır. Bu etkinlikte öğrencilerin süreçte bir argüman ortaya çıkarmada zorlandıklarını göstermektedir.

4.4.2. Etkinliklere Ait Algoritma Yapıları ile İlgili Genel Değerlendirme

Bu bölümde öğrencilerin etkinlikler sonucunda ürettikleri algoritmaların yapıları ile genel bir değerlendirme sunulmuştur. Öğrencilerin ürettikleri algoritmalar algoritma yapısı değerlendirme rubriği ile 10 puan üzerinden bir değerlendirme, Dr. Scratch programı ile 12 puan üzerinden bir değerlendirmeye tabi tutulmuştur.



Şekil 4.31. Öğrencilerin Ürettikleri Algoritmaların Yapı Bakımından Değerlendirilmesi

Algoritma Yapısı Değerlendirme Rubriğine göre başarılı olunan pozitif bölenleri listeleme ve ardışık sayıların toplamı etkinliklerinde puanlar, tam puan çıkmıştır. Öğrencilerin bu iki etkinlikte ürettikleri algoritmalar sonlu olma, kesinlik, girdi, çıktı ve etkileycilik özellikleri kapsamında istenilen yapıdadır. Algoritma Yapısı Değerlendirme Rubriğine göre başarısız olunan oyun etkinliğinde ise puan, 10 üzerinden 4 puan çıkmıştır. Öğrencilerden istenilen sonucun oluşmadığı bu etkinlikte öğrencilerin oluşturduğu algoritma sonlu olma, kesinlik, girdi ve çıktı özelliklerinden 1'er puan almıştır. Etkileycilik özelliğinden ise bu algoritma hiç puan alamamıştır. Dr. Scratch programına göre puanlamalar sonucunda pozitif bölenleri listeleme etkinliği 12 üzerinden 9 puan, ardışık sayıları toplama etkinliği 12 üzerinden 6 puan ve oyun etkinliği ise 12 üzerinden 7 puan almıştır. Dr. Scratch programına göre pozitif bölenleri listeleme etkinliği yüksek bir almıştır. Ardışık sayıların toplamı ve oyun etkinliğinde ise puanlar birbirine yakın ve orta düzeyde çıkmıştır. Bu durum her iki algoritmada da yapıların doğru bir şekilde üretilmeye çalışılmıştır.

BÖLÜM V: SONUÇ, TARTIŞMA VE ÖNERİLER

Çalışmada elde edilen veriler, “Bulgular” bölümünde ayrıntılı olarak verilmiştir. Bu bölümde, “Akademik Başarısı Yüksek Ortaokul Öğrencilerinin Scratch Programı ile Algoritma Üretme Becerilerinin İncelenmesi” adlı çalışmanın bulguları, araştırma soruları bağlamında özetlenmiştir. Çalışma verilerinin değerlendirildiği bölüm olan sonuç bölümü, iki alt başlık altında ele alınmıştır. Bunlar; algoritma üretme süreçleri ve algoritma yapısı ile ilgili sonuçlardır.

5.1. Algoritma Üretme Süreci ile İlgili Sonuçlar

Bu kısımda algoritma oluşturma süreci ile ilgili sonuçlar her bir etkinlik için ayrı ayrı başlıklar altında sunulmuştur.

5.1.1. Pozitif Bölenleri Listeleme Etkinliğinin Algoritma Üretme Sürecine Ait Sonuçları

Pozitif bölenleri listeleme etkinliği, bir sayı girildiğinde sayının bölenlerini verip, asal olup olmadığını belirlemeye yönelik bir çalışmaydı. Öğrenciler etkinlik sürecinin başlarında bazı zorluklar yaşamışlardır. İlk başlarda tamamen yanlış yollar denenip sonuç elde edilmeye çalışılmıştır fakat Scratch programındaki birkaç başlangıç blokları hariç herhangi bir ilerleme sağlanamamıştır. Araştırmacı bu sırada öğrencilere bir sayının tam bölünmesi ile ilgili soru yöneltmiştir. Daha sonra öğrenciler bir sayının tam bölünmesi için kalanın sıfır olması gerektiğini ifade etmişlerdir. Ancak bunu Scratch programı üzerine aktaramamışlardır. Etkinliklerde bu durum sıklıkla yaşanmıştır. Matematiksel olarak doğru ifade edilen bir durumu Scratch programına aktarmada kod bloklarının anlamlarının tam bilinmemesi, daha önce Scratch programında matematiksel projeler üretmemelerinden kaynaklı bazı zorluklar yaşanmıştır. Bu durum Özkan’ın (2020) yaptığı çalışma ile benzer sonuçlar göstermektedir. Scratch programında tam bölen bloğunun olmadığı ve bunun yerine “mod” bloğunu kullanmaları gerektiği öğrenciler tarafından fark edilmiştir. Bu durumu kod bloklarına aktarma konusunda bazı zorluklar yaşansa da öğrenciler algoritmayı üretebilmişlerdir. Sonraki adımda bölen sayıların sınırlaması yapılmıştır. Bölmeye 1’den başlanması gerektiği ve bölünen sayıya kadar tüm sayıların denenmesi gerektiği

söylenmiştir. Burada kodlama yaparken “sayıyı sıfır yap” bloğundan başlamışlar ve “sayı defa tekrarlar” bloğunu kullandıkları için sayının kendisi bölenler listesine eklenememiştir. Sayının kendisinin de sayının bir böleni olduğunu bildikleri için “sayıyı bölen listesine” ekle bloğunu oluşturmuşlardır. Pozitif bölenleri listeleme algoritması bu şekilde oluşturulmuştur. Daha sonra algoritmanın doğru oluşup oluşmadığı hakkında denemeler yapılmış ve doğru sonuçların oluştuğu teyit edilmiştir. Bu durum Yılmazel (2016) yılında ortaya koyduğu bir algoritma her zaman geçerli ve doğru sonucu vermesi ilkesi ile ilgilidir. Öğrenciler bu durumu dikkate alarak algoritmayı farklı sayılarla denemişlerdir. Sayının asal olup olmadığı hakkındaki algoritma ise öğrenciler tarafından kısaca oluşturulmuştur. Bunun en büyük nedeni öğrencilerin asal sayıları iki tane böleni olması gerektiğini bilmeleridir. Listenin uzunluğunu “eğer ve değilse” blokları ile ayarlayıp algoritmayı bitirmişlerdir. Bu ekinlikte öğrencilerin pozitif tam sayılarla ilgili kullandıkları ifadeler incelendiğinde çoğunlukla ifadelerin doğru oldukları gözlemlenmiştir. Şengül ve Cantimer’den (2018) farklı olarak yapılan bu araştırmada ise öğrencilerin pozitif tam sayılar ve asal sayılar konularında matematiksel olarak herhangi bir hata yapmadıkları görülmüştür. Öğrenciler pozitif bölenleri listeleme etkinliğinde tanıma, kurma ve oluşturma süreçlerini sırayla değil dinamik bir şekilde kullanmışlardır. Örneğin, sayının bölenlerini listeme sürecinde ilk olarak doğru bir tanıma sürecinden sonra matematiksel olarak ifadeleri kurmuşlar ve Scratch programında oluşturma süreci ile bitirmişlerdir. Ancak programda sayının kendisini listeye alınmadığını görmüşler. “Sayıyı direkt listeye ekleyelim” kurma sürecinden sonra bunu oluşturma süreci ile kod bloklarını oluşturmuşlardır. Öğrenciler, algoritma oluştururken kullandıkları argümanlar 4 başlık altında toplanmış ve bulgulara sunulmuştur. Bu argümanların incelenmesi sonucunda açıkça belirtilen ve açıkça belirtilmeyen yanlış argümanların daha çok etkinliğin ilk safhalarında olduğu görülmüştür. Mod kavramına kadar açıkça belirtilen yanlış olan ifadeler oldukça fazla olduğu görülmüştür. Mod kavramında sonra açıkça belirtilen doğru ifadeler kullanılmasına rağmen mod kavramını Scratch programı üzerinde aktarmada yine açıkça belirtilen ve açıkça belirtilmeyen yanlış argümanların kullanıldığı görülmüştür. Bu argümanlar dinamik bir süreç içinde ilerlemiştir. Söylenen yanlış ifadeler öğrenciler tarafından fark edilerek doğru ifadelere dönüştüğü bir süreç yaşanmıştır. Örneğin, Cem başlangıçta pozitif bölenler ile asal sayıları karıştırmış, Erdem’in asal sayılar ile ilgili ifadeleri sonucunda kendi yanlışını fark edip doğru ifadeler oluşturmuştur. Etkinlik ilerledikçe ve doğru bloklar oluştuğunda argümanlar açıkça ve açıkça olmayan argümanlara dönüşmüştür. Özellikle asal sayıları konusunda öğrenciler açıkça doğru argümanları sıkça kullanmışlardır. Bu durum Moala’nın (2019) algoritmaların her durumda çalışması anlayışı ile uyumludur.

Çünkü öğrenciler açıkça belirtilen ve doğru olan özellikleri sıkça kullanarak algoritmanın temelini doğru oluşturmuşlar ve algoritmanın geliştirilmesinde herhangi bir zorluk yaşamamışlardır.

5.1.2. Ardışık Sayıların Toplamı Etkinliğinin Algoritma Üretme Sürecine Ait Sonuçları

Öğrenciler ardışık doğal sayıların toplamı etkinliğini başarıyla tamamlamışlardır. Etkinlik, araştırmacı tarafından algoritma bölümü olmadan öğrencilere bir Scratch projesi gösterilerek başlamıştır. Öğrencilerden bu projenin matematiksel karşılığının bulunması ve bir benzerinin Scratch programında oluşturulması istenmiştir. Cem hemen bu matematiksel ifadenin 1’den n’ye kadar olan sayıların toplamı olduğunu anlamıştır. Açıkça belirtilen ve doğru olan bu durumun etkinlik süresini ciddi şekilde kısalttığı düşünülmüştür. Gauss toplamı diye de anılan bu toplamın bir formülü vardır. Ancak öğrenciler bu formülü bilmiyorlardı ve bu formülü bulmak içinde uğraşmamışlardır. Burada algoritmik düşünerek matematiksel olarak ilerlemişlerdir. Bunu yaparken kod bloğunun en son kısmına “-1 kadar değiştir” bloğunu eklemişlerdir. Bu şekilde algoritma 1’den n’ye kadar değil n’den 1’e kadar olan sayıların toplamı şeklinde oluşmuştur. Bu durum dikey matematikleştirmeye güzel bir örnek olarak verilebilir (Hershkowitz vd., 2001). Cem’in başlangıçta algoritmayı anlayıp söylemesi tanıma sürecine iyi bir örnek olarak verilebilir. Bu etkinlikte öğrenciler kurma süreci içerisinde çok fazla yanlış yapmadan doğru algoritmaya ulaşmışlardır. Oluşturma süreci de kurma süreci gibi fazla uzun sürmeden tamamlanmıştır. Öğrencilerin bu etkinlikte oluşturdukları argümanlar ise genellikle açıkça belirtilen ve doğru olan argümanlardan oluştuğu gözlemlenmiştir.

5.1.3. Oyun Etkinliğinin Algoritma Üretme Sürecine Ait Sonuçları

Öğrenciler oyun etkinliğini başarılı bir şekilde bitirememiştir. Bu etkinlikte öğrenciler oyunun kurallarını rahatça anlamışlardır. Fakat hem Scratch programına oyunu kuralları ile aktarmada hem de oyunun kazanma stratejisini belirlemede başarıya ulaşamamışlardır. Öğrenciler bu etkinlikte seçilen blokları matematiksel açıdan doğru yere, doğru sıra ile Scratch programına aktarmada zorluklar yaşamışlardır. Bu durum Özkan’ın (2020) yaptığı çalışma ile benzer sonuçlar göstermektedir. Oyunun kazanma yolunun belirlenmesinde Erdem açıkça belirtilen ve doğru olan ifadeleri ile tanıma sürecini doğru oluşturmuştur. Ancak sonrasında açıkça belirtilen ve yanlış olan ifadeler ile tanıma süreci sekteye

uğramıştır. Bu sırada Cem oyunun kesin stratejisini bulmak için kâğıt üzerinde denemeler yapmıştır fakat bir sonuç elde edememiştir. Bu etkinlikte öğrencilerin grup olarak hareket etmede zorlandıkları gözlemlenmiştir. Oyunu kazanma yolunun bulunamaması üzerine oyunu ilk önce Scratch programında oluşturma fikri ön plana çıkmıştır. Bu plana göre eğer doğru bir şekilde algoritmik olarak Scratch programına oyun aktarılabilirse algoritmanın sonuçları ile bir kazanma yolunun bulunacağı düşünülmüştür. Bu aşamada 1’den 10’a kadar sayılar oluşturulmuştur. Lakin bazı sayılardan birden fazla olduğu görülmüştür. Bu durum düzeltilmeye çalışılmış ancak başarılı olunamamıştır. Etkinlikte daha fazla ileriye gidememiştir. Öğrenciler bu etkinlikte sıklıkla “neden olmuyor?”, “bu şekilde olması lazımdı” gibi ifadeler kullanmışlardır. Öğrencilere göre tanıma ve kurma süreçleri doğru işletilmiş ancak oluşturma sürecinde anlamadıkları bir şeyler yüzünden ilerleyemediklerini düşünmüşlerdir. Bu etkinlikte liste oluşturma kısmından ileriye gidemedikleri için süreçte öğrencilerin ifadeleri ile 4 başlık altında toplanan argümanların pek oluşmadığı görülmüştür. Öğrencilerin pozitif bölenleri listeleme ve ardışık sayıların toplamı etkinliklerinde ürettikleri algoritmaların yapıları matematiksel açıdan doğru oluşturulduğu görülmüştür. Bu süreçte öğrencilerin birçok defa açıkça belirtilen ve açıkça belirtilmeyen yanlış argümanlar ürettikleri görülmüştür. Bu yanlışların süreçte doğru ifadelere dönüştükleri gözlemlenmiştir. Bunun en önemli nedenlerinden biri Scratch programında yanlış blok ile algoritmanın çalışmaması gösterilebilir. Ayrıca Scratch programı hızlı bir şekilde sonucu verdiği için yanlış ifadeler ile ilgili anında geri dönüt alınmış ve değişikliklerle düzeltme yoluna girilmiştir. Knuth’ın (1981) algoritmaların hızlı test edilmesi için teknolojiye yararlanılabilir anlayışı öğrenciler tarafından Scratch programı ile etkinliklerin yapım sürecinde kullanılmıştır. Bu sayede algoritmanın çalışıp çalışmadığı program üzerinde rahatlıkla görülebilmektedir. Öğrenciler grup ile çalışmanın etkinliklerin çözümünde oldukça faydalı olduğunu ve bu şekilde hem eğlendiklerini hem de farklı fikirler ile çözüme ulaşabileceklerini belirtmişlerdir.

5.2. Algoritma Yapısı ile İlgili Sonuçlar

Bu kısımda algoritma yapısı ile ilgili sonuçlar her bir etkinlik için ayrı ayrı başlıklar altında sunulmuştur.

5.2.1. Pozitif Bölenleri Listeleme Etkinliğinin Algoritma Yapısına Ait Sonuçları

Araştırma sonucunda, pozitif bölenleri listeleme etkinliğinin başarıyla tamamlandığı görülmüştür. Öğrencilerin bu etkinlikte oluşturdukları algoritma istenilen sonuçları verdiği için yapı olarak da doğru bir yapıda oluşturulmuştur.

Knuth (2014) tarafından ortaya konulan iyi bir algoritmada olması gereken beş özellikten yararlanılarak yapılan değerlendirme sonucunda algoritma puanı 10 üzerinden 10 çıkmıştır. Oluşturulan algoritma sonlu olma, kesinlik, girdi, çıktı ve etkileycilik özelliklerinden tam puan almıştır. Bu etkinlikte oluşturulan algoritmanın birer başlangıç ve bitiş prosedürü vardır. Algoritma adımları kesin ve net bir şekilde oluşturulmuştur. Algoritmanın girdileri Scratch programına uygun olarak hep sıfırdan başlatılmıştır. Sıfırdan başlatılan algoritmanın çalışmasında ise bu girdi her adımda 1'er arttırılarak sayıyı bölmesi hedeflenmiştir. Böldüklerini mod kavramı ile ilişkilendirmişlerdir. Bu şekilde kalanın sıfır olduğu durumlarda sayı, bölen listesine eklenmiştir. Bu şekilde girdisi ve çalışma prensibi belirlenen algoritma “sayı defa tekrarla” bloğu ile sonlu olma özelliğini kazanmıştır. Ayrıca her adım Scratch programı yardımı ile kesin olarak ifade edilmiştir. Her adımda bir önceki adımdan kalan veriler silinerek algoritmanın her seferinde baştan başlatılması sağlanmıştır. Bu bakımdan algoritmaların girdileri doğru bir şekilde oluşturulmuştur. Algoritmaların çıktıları ise etkinlikten istenen sonucu istenildiği formatta net ve kesin olarak ortaya koymuştur.

İkinci değerlendirme aracı olan Dr. Scratch programına göre algoritma yapısı değerlendirilmesinde pozitif bölenleri listeleme etkinliği için 12 üzerinden 9 puan çıkmıştır. Akış kontrolünde öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi “... olana kadar tekrarla” bloğunun kullanılmamasıdır. Veri temsili başlığında öğrencilerin oluşturduğu algoritma 3 puan üzerinden 3 puan almıştır. Kullanıcı etkileşimi başlığında öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi ise öğrencilerin mikrofon veya kamera kullanılmasını gerektirecek algoritma oluşturmamalarıdır. Mantık başlığında ise öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi ise “ve/veya” bloklarının kullanılmamasıdır. Ayrıca gereksiz yinelenen komut dizisi ve algoritmada kullanmayan ölü kodların olmaması algoritmanın doğru yapıldığının bir kanıtıdır. Ayrıca Morene ve diğerlerinin (2015) belirttiği hatalı kodlama, kullanımı gereksiz bloklar, kodun yanlış tekrarı, hiçbir zaman çalıştırılmamış kod ve projenin yanlış başlatılması gibi hatalar bu etkinlikte görülmemiştir.

5.2.2. Ardışık Sayıların Toplamı Etkinliğinin Algoritma Yapısına Ait Sonuçları

Araştırma sonucunda, ardışık sayıların toplamı etkinliğinin başarıyla tamamlandığı görülmüştür. Öğrencilerin bu etkinlikte oluşturdukları algoritma istenilen sonuçları verdiği için yapı olarak da doğru bir yapıda oluşturulmuştur.

Knuth (2014) tarafından ortaya konulan iyi bir algoritmada olması gereken beş özelliğten yararlanılarak yapılan değerlendirme sonucunda algoritma puanı 10 üzerinden 10 çıkmıştır. Oluşturulan algoritma sonlu olma, kesinlik, girdi, çıktı ve etkileycilik özelliklerinden tam puan almıştır. Bu etkinlikte oluşturulan algoritmanın birer başlangıç ve bitiş prosedürü vardır. Algoritma adımları kesin ve net bir şekilde oluşturulmuştur. Bu etkinlikte sayıların geriye doğru toplama işlemini yapması gerektiği vurgulanmış ve bir eksiltme ile algoritma oluşturulmuştur. Aynı zamanda algoritma sıfıra kadar devam etmiş ve negatif tam sayılar algoritmaya dahil edilmemiştir. Bu şekilde algoritmanın girdileri doğru bir şekilde belirlenmiştir. “Sayı defa tekrarlar” bloğu ile algoritmaya sonlu olma özelliği kazandırılmıştır. Algoritma kısa bir şekilde oluşturulmuş ve her bir adımı Scratch programı ile kesinlik özelliğini kazanmıştır. Algoritmanın başlangıç kısmı klasik algoritma bloklarından oluştuğu için burada herhangi bir etkileycilik gözlemlenmemiştir. Ancak “sayı defa tekrarlar”, “sayı artı sonuç” ve “sayıyı eksi bir kadar değiştir” blokları ile algoritma istenen sonucu vermiştir. Bu üç blok ile hem algoritma kısa bir şekilde oluşturulmuş hem de algoritma etkileycilik özelliğini kazanmıştır.

İkinci değerlendirme aracı olan Dr. Scratch programına göre algoritma yapısı değerlendirilmesinde ardışık sayıların toplamı etkinliği için 12 üzerinden 6 puan çıkmıştır. Akış kontrolünde öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi “... olana kadar tekrarlar” bloğunun kullanılmamasıdır. Veri temsili başlığında öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi öğrencilerin liste oluşturmamalarıdır. Kullanıcı etkileşimi başlığında öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi ise öğrencilerin mikrofon veya kamera kullanılmasını gerektirecek bir algoritma oluşturmamalarıdır. Mantık başlığında ise öğrencilerin oluşturduğu algoritma 3 puan üzerinden 0 puan almıştır. Bunun sebebi ise öğrencilerin oluşturduğu algoritmada koşul ile ilgili herhangi bir bloğun yer almamasıdır. Bu koşul bloklarının öğrenciler tarafından kullanılmama sebebi, öğrencilerin oluşturdukları algoritmanın direkt sonuca götürdüğünü görmelerinden kaynaklıdır. Ayrıca Morene ve diğerlerinin (2015) belirttiği hatalı kodlama, kullanımı gereksiz bloklar, kodun yanlış tekrarı, hiçbir zaman çalıştırılmamış kod ve projenin yanlış başlatılması gibi hatalar bu etkinlikte görülmemiştir.

5.2.3. Oyun Etkinliğinin Algoritma Yapısına Ait Sonuçları

Araştırma sonucunda, oyun etkinliğinin başarıyla tamamlanamadığı görülmüştür. Öğrencilerin bu etkinlikte oluşturdukları algoritmanın istenilen sonucu veremediği gözlemlenmiştir.

Knuth (2014) tarafından ortaya konulan iyi bir algoritmada olması gereken beş özelliğten yararlanılarak yapılan değerlendirme sonucunda algoritma puanı 10 üzerinden 4 çıkmıştır. Oluşturulan algoritma sonlu olma, kesinlik, girdi, çıktı özelliklerinden birer puan almıştır. Bu etkinlikte öğrenciler Scratch programında 1’den 10’a kadar sayıları tekrarsız ve rastgele oluşturamamışlardır. En başta algoritmanın girdileri istenildiği şekilde oluşturulamadığı için çıktılarda istenilen sonucu vermemiştir. “10 defa tekrarla” bloğu ile algoritmaya sonlu olma özelliği kısmen kazandırılmıştır. Oyunun kurallarında olan 10 sayı oluşturulmuş ancak etkinliğin istenen sonucunu veremediği için sonlu olma özelliği tam olarak belirlenememiştir. Algoritmada sayı değişkeni sadece 10 sayı üretecek şekilde oluşturulmuş ancak sayılar birbirini tekrar etmiştir. Bu durum kesinlik özelliğinin de tam olarak kazanılmamasına neden olmuştur. Algoritmada ise etkileycilik özelliği gözlemlenememiştir.

İkinci değerlendirme aracı olan Dr. Scratch programına göre algoritma yapısı oyun etkinliği için 12 üzerinden 7 puan olarak çıkmıştır. Etkinlikten beklenen sonuç elde edilmemesine karşın bu puanın ortaya çıkışı, öğrencilerin oluşturdukları algoritmanın yapı bakımından içinde doğrular olduğunu göstermektedir. Akış kontrolünde öğrencilerin oluşturduğu algoritma 3 puan üzerinden 2 puan almıştır. Bunun sebebi “... olana kadar tekrarla” bloğunun kullanılmamasıdır. Veri temsili başlığında öğrencilerin oluşturduğu algoritma 3 puan üzerinden 3 puan almıştır. Kullanıcı etkileşimi başlığında öğrencilerin oluşturduğu algoritma 3 puan üzerinden 1 puan almıştır. Bunun sebebi ise hem öğrencilerin mikrofon veya kamera kullanılmasını gerektirecek algoritma oluşturmamaları hem de algoritmanın kullanıcıya herhangi bir soru sormamasıdır. Mantık başlığında ise öğrencilerin oluşturduğu algoritma 3 puan üzerinden 1 puan almıştır. Bunun sebebi ise öğrencilerin “ve/veya” ile “eğer ise değilse” bloklarını algoritmada kullanmamasıdır. Bu koşul bloklarının öğrenciler tarafından kullanılmama sebebi, öğrencilerin oluşturdukları algoritmanın direkt sonuca götürdüğünü görmelerinden kaynaklıdır. Ayrıca Morene ve diğerlerinin (2015) belirttiği hatalı kodlama, kullanımı gereksiz bloklar, kodun yanlış tekrarı, hiçbir zaman çalıştırılmamış kod ve projenin yanlış başlatılması gibi hatalar bu etkinlikte görülmemiştir.

5.3. Tartışma

Bu araştırmada, öğrencilerin günümüz dünyasını anlamlandırmak için (Aytekin vd., 2018) algoritma üretme süreçlerine ve algoritma yapılarına odaklanılmıştır. Bunu yaparken algoritmalar ile sıkı bir ilişki içinde ve ücretsiz bir program olan Scratch Programı kullanılmıştır.

Bu araştırmadaki etkinliklerde öğretmen rehber konumdayken öğrenciler aktif konumdadır. Bu durum 21. yüzyıl eğitim anlayışı ile örtüşmektedir (Dilaver, 2016). Küçükoğlu (2018) tarafından ortaya konulan 21. yüzyıl becerilerinden öğrenci işbirliği, dijital okuryazarlık ve problem çözme becerileri bu araştırmada gözlemlenebilmiştir. Etkinliklerin grup çalışması ile yapılması öğrenci işbirliği, Scratch programı ile yapılması dijital okuryazarlık ve pozitif tam sayıları ile çeşitli etkinlikler üzerinde çalışılması problem çözme becerisine ait örneklerdir.

Algoritmik düşünme; herhangi bir problemin çözümüne ulaşmak için oluşturulan sistematik ve ayrıntılı işlemler bütünü olarak açıklanmaktadır (Çatlak vd., 2015). Bu araştırmada pozitif bölenleri listeleme ve ardışık sayıların toplamı etkinliklerinde öğrenciler algoritmalarını sistematik ve ayrıntılı olarak oluşturmuşlardır. 21. yüzyıl matematik eğitim anlayışında algoritmalar, problemlere özgün çözümler üretilmesi olarak ifade edilmektedir (Erümit vd., 2018). Bu araştırmada özellikle oyun etkinliğinde öğrencilerden rutin olmayan matematik problemine özgün çözüm üretmeleri beklenmiştir. Ancak öğrenciler oyun etkinliğini başarılı olarak tamamlayamamışlardır.

Öğrenciler pozitif bölenleri listeleme ve ardışık sayıların toplamı etkinliklerinde oluşturdukları algoritmalar benzer problemlere uygulandığında da doğru sonucu verecek şekilde oluşturulmuştur. Oyun etkinliğinde ise genelleştirilebilir bir algoritma oluşturamamışlardır. Bu durumda Moala (2019)'nın yaptığı araştırmada ulaşılan, öğrencilerden istenen algoritmaları kolaylıkla oluşturdukları ama benzer problemlere uygulamaya çalıştıklarında her zaman doğru sonucu veremediği sonucuyla uyumludur.

Etkinlikler sürecinde oluşturulan algoritmalar sürekli denenmiştir ve daha kısa yolları bulunmaya çalışılmıştır. Bu sonucun son yıllarda yapılan çalışmalar ile örtüştüğü söylenebilir (Moala, 2019; Yoon, 2018). Öğrenciler bildikleri matematiksel ifade ve sonuçlarından nispeten bilmedikleri yeni matematiksel ifade ve sonuçlara ulaşmışlardır. Ardışık sayıların toplamı etkinliğinde oluşturulan algoritma buna örnek verilebilir. Bu durum dikey matematikleştirme olarak da adlandırılmaktadır (Hershkowitz vd., 2001).

Özdeş (2013) yaptığı araştırmada, öğrencilerin bir sayının pozitif tam bölenlerinin belirlenmesi sorusunda %50 oranında yanlış yaptıklarını belirlemiştir. Öğrencilerin bu konuda en çok yaptıkları hataların; pozitif tam bölenlerinin kaç tane olduğunu bulma, sıfırın pozitif tam bölen olduğunu düşünme ve bir sayının bölenleri yerine katlarını bulma olduğunu gözlemlemiştir. Aynı çalışmada, öğrencilerin asal sayı sorularında %23 oranında yanlış yaptığını ve en sık yaptıkları hataların asal sayıların negatif işaretlilerinin de asal sayı olduğunu düşünme, sıfırın ve 1'in asal sayı olduğunu düşünme olarak belirlemiştir. Özdeş'ten farklı olarak yapılan bu araştırmada ise öğrencilerin pozitif tam sayılar ve asal sayılar konularında matematiksel olarak herhangi bir hata yapmadıkları görülmüştür.

Teknoloji, öğrencilerin matematik dersini öğrenme süreçlerinde kullanılabilecek en etkili yollardan biri olarak ifade edilmektedir (Saka & Çelik, 2018). Bu kapsamda da bu araştırmada Scratch programı kullanılmıştır. Scratch programı ile yapılan bu araştırmada öğrencilere özellikle bir soru sorulmamasına karşın araştırma boyunca çok eğlendiklerini belirtmişlerdir. Bu durum Oluk vd., (2018) yaptığı çalışmanın sonucu ile uyumludur. Wilson ve Moffat (2012), Permatasari, Yuana ve Maryono (2018) tarafından daha önce yapılan araştırmalarda öğrenciler Scratch kullanımını eğlenceli bulmuşlar ve Scratch programında proje oluşturmak için bu durumun onları motive ettiğini gözlemlemiştir. Akademik başarısı yüksek olan öğrenciler normal okul derslerinden ve müfredatından kolayca sıkılabilmektedirler. Bu farklı çalışma onlar için hem eğlenceli hem de yeni bir zorluk getirmektedir ki, bu öğrencileri motive etmek için iyi bir yol olabilir (Koshy, 2002).

Öğrencilerin oluşturduğu etkinliklerin değerlendirilmesinde RBC soyutlama modeli kullanılmıştır. Bu model kapsamında öğrencilerin tanıma, kullanma ve oluşturma eylemlerini sıkça kullandıkları gözlemlenmiştir. Bu durum Altun ve Memnun (2012)'un yapmış olduğu araştırma sonuçları ile uyumludur.

Bu araştırmada oluşturulan algoritmalar, Scratch programı kullanılarak kodlanmıştır. Bu durum öğrenciler için algoritma oluşturma yanında ayrı bir çaba gerektirmiştir. Çünkü öğrenciler tanıma ve kurma sürecinde oluşturdukları matematiksel işlemleri ve algoritmik yapıları Scratch programına aktarılırken çeşitli zorluklar yaşamışlardır. Bunun asıl sebebi Scratch programının blok tabanlı bir kodlama programı olmasıdır. Kodlamada yanlış blok kullanımı, bloklar arasında yanlış dizilim veya hatalı değişken kullanımı gibi birçok durumlar algoritmayı çalışmaz duruma getirebilmektedir (Aytekin vd., 2018).

Bu çalışmanın Scratch üzerinde yapılıyor olmasının öğrenciler açısından hem olumlu hem de olumsuz tarafları olmuştur. Olumlu tarafı, algoritma deneme işlemi daha hızlı ve daha net geri dönütler alınmasıdır. Bunun sonucunda algoritma hızlı bir şekilde revize edilmeye

çalışılmıştır. Olumsuz tarafı ise öğrencilerin sadece algoritma oluşturmak için değil, bu işlemi Scratch programı üzerine aktarma ile de uğraşmış olmalarıdır. (Scratch About, 2015). Grup çalışmalarının, öğrenciler üzerinde olumlu etkisi olduğu belirtilmektedir (Lou vd., 2001). Bu çalışmada da bahsedilen olumlu etkinin öğrenciler açısından oldukça faydası görülmüştür. Çalışmada öğrenciler grup içi etkileşim ve yardımlaşmayı oldukça iyi kullanmışlardır. Öğrenciler sıklıkla “Grup olmasaydık bu çalışmayı yapamazdık” demişlerdir. Grup içinde tartışma, mantık ve fikir üretiminin oldukça iyi düzeyde olduğu görülmüştür. Scratch programı ile algoritma üretme süreçlerini kapsayan bu araştırmada sadece algoritma üzerinde olan başarıları değil, öğrencilerin iş birliğinde olan başarısı da gözlemlenmiştir. Bu durumun benzeri Lewin, Somekh ve Steadman (2008) tarafından yapılan araştırmada da gözlemlenmiştir.

5.4. Öneriler

Bu araştırmada öğrencilerin pozitif tam sayılar konusunda algoritma üretme süreçleri ve süreç sonunda oluşturulan algoritmaların yapısı incelenmiştir. Öğrencilerin pozitif bölenleri listeleme ve ardışık sayıların toplamı etkinliklerinde başarılı oldukları gözlenirken, rutin olmayan oyun etkinliğinde ise başarısız oldukları gözlemlenmiştir. Bu kapsamda birçok araştırmaya konu olan pozitif tam sayılar konusunda, rutin olmayan problemler üzerinde araştırmalar arttırılabilir.

Araştırmaya katılan 6. sınıf öğrencileri kasıtlı olarak akademik başarısı yüksek öğrencilerden seçilmiştir. Bu öğrenciler aynı zamanda bir özel okulda öğrenim görmektedir. Bu konu üzerine araştırma yapacak olan araştırmacılar, farklı sınıf seviyelerinden veya bir devlet okulundan öğrenciler seçilebilir. Bu araştırma pozitif tam sayılar konusu üzerine yapılmıştır. Benzer bir çalışmada farklı müfredat konuları ile araştırmalar yapılabilir. Öğrenciler etkinlik sürecinde bazen doğru matematiksel ifadeleri söylemelerine karşın bu ifadeleri Scratch programına aktarmada bazı zorluklar yaşamışlardır. Bu kapsamda matematik perspektifinden öğrencilere Scratch programı eğitimi verilebilir. Matematik ve Scratch programı araştırmaları arttırılabilir. Scratch programı öğretmenler tarafından öğrenilip matematik derslerinde öğrencilere farklı öğrenme ortamları sunabilmek için kullanılabilir. Bu şekilde matematik dersi daha somut olarak işlenebilir. Ayrıca Scratch programının matematik derslerinde kullanılması ile öğrenci 21. yüzyıl becerilerinden biri olan dijital okuryazarlık becerisini kazanabilir. Öğrencilerin daha az başarılı olduğu matematiksel içerikler ve matematiksel süreçler, sınıf ortamında öğretmenler tarafından

tespit edilip o alanlara odaklanılarak öğrencilerin oyun oynar gibi Scratch programı üzerinde problem yaşadıkları kısımlar iyileştirilebilir.

Yapılan araştırma sonucunda öğrencilerin verilen etkinliklerde kullandıkları soyutlamalar yer yer incelenebilmiştir. Bu sebeple matematik eğitiminde yapılacak araştırmalarda, soyutlamanın öğrenci tarafından nasıl kullanıldığının gözlemlenebileceği bir uygulama tasarımı yapılabilir.

Amaca yönelik olarak, araştırmaya katılan öğrenci sayısı üçtür. Üç öğrenciyle çalışılması, büyük ölçekli karşılaştırmalar yapmaya ve bu karşılaştırmalara dayalı genellemelere ulaşmaya imkân vermemektedir. Bu çalışmanın amacı genelleme yapmak olmadığından gelecekte bu konuda çalışmalar yapacak araştırmacıların, karşılaştırmalara ve genellemelere uygun daha fazla öğrenci seçimi yapmaları önerilebilir.

Algoritma üretmek çözümün tüm özelliklerini dikkate almak demektir. Öğrencilerin pozitif tam sayılar konusunda algoritma tasarımları konuyu daha kapsamlı öğrenmesine olanak sağladığı düşünülmektedir. Bu yüzden matematik dersinde algoritma ile öğretim yapılması önerilmektedir. Ayrıca Scratch programının pozitif tam sayılar konusunda öğrencilerin hatalarını anlık düzeltmeye olanak sağladığı görülmüştür. Bu kapsamda pozitif tam sayılar konusu öğretiminde, Scratch programı kullanılabilir.

Bu araştırmanın eksik taraflarından biri öğrencilerin algoritmik düşünme becerilerinin, araştırmanın başında ölçülmemesidir. Bu kapsamda öğrencilerin algoritmik düşünme becerileri ölçülüp daha sonrasında belirli periyotlarla bir konuda eğitimler verilip öğrencinin matematik ders başarısı ölçülebilir.

Bu çalışmada öğrencilerin görüşleri, motivasyonları gibi duyuşsal özellikler araştırılmamıştır. Bu sebeple yapılacak çalışmalarda Scratch destekli matematik öğretimine yönelik öğrenci görüşleri, motivasyonları gibi konular araştırılabilir. Scratch etkinliklerine başlanılmadan önce öğrencilerle rubrikler paylaşılmamıştır. Benzer çalışmalarda öğrencilerle rubrikler önceden paylaşılabilir.

Öğrenciler Scratch etkinliklerini yaparken sonucun olumsuz olduğu durumlarda araştırmacının bir B planı yoktur. Bu alanda yapılacak olan araştırmalarda, etkinlik sonucunun olumsuz olması durumunda önceden bir B planı tasarlanabilir.

KAYNAKÇA

- Akay, H. (2006). Problem kurma yaklaşımı ile yapılan matematik öğretimin öğrencilerin akademik başarısı, problem çözme becerileri ve yaratıcılığı üzerindeki etkisinin incelenmesi. Doktora Tezi, Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara
- Akpınar, Y. & Altun, Y. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. İlköğretim Online, 13(1), 1-4.
- Altıparmak, K., ve Özdoğan, E. (2010). A study on the teaching of the concept of negative numbers. International Journal of Mathematical Education in Science and Technology, 41(1), 31-47.
- Altun, M. (2002). İlköğretim ikinci kademedeki (6, 7 ve 8. sınıflarda) matematik öğretimi. Bursa: Erkam Matbaası.
- Altun, M. & Memnun, D. S. (2012). RBC+C modeline göre doğrunun denklemi kavramının soyutlanması üzerine bir çalışma: özel bir durum çalışması. Uluslararası Cumhuriyet Eğitim Dergisi, 1, 17-37.
- Annamalai, S., & Salam, S. N. A. (2017). Facilitating programming comprehension for novice learners with multimedia approach: A preliminary investigation. In American Institute of Physics (AIP) (Ed.), AIP conference proceedings 1891.
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. IX. akademik bilişim konferansı bildirileri (s.193-197). Dumlupınar Üniversitesi, Kütahya.
- Aytekin, A. & Çakır S. F. & Yücel B. Y. & Kulaözü İ. (2018). “Algoritmaların Hayatımızdaki Yeri ve Önemi” Avrasya sosyal ve ekonomi araştırmaları dergisi (ASEAD), Cilt 5 Sayı 7 Yıl 2018, S 151-162.

- Baki, A. (1998). Matematik öğretiminde işlemsel ve kavramsal bilginin dengelenmesi. Atatürk Üniversitesi 40. Kuruluş Yıldönümü Matematik Sempozyumu, Atatürk Üniversitesi, Erzurum.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A dijital age skill for everyone. Available at: <http://files.eric.ed.gov/fulltext/EJ918910.pdf>, Erişim Tarihi: 21.07.2019.
- Baykul, Y. (2014). Ortaokulda matematik öğretimi 5, 6, 7 ve 8. sınıflar. Ankara: Pegem Akademi.
- Bikner-Ahsbahr, A. (2004). Towards the emergence of constructing mathematical meanings In M. J. Haines and A. B. Fuglestad (Eds.), Proceedings of the 28th Conference of the International Group for the Psychology of Mathematics Education, (Vol. 2, pp. 119-126). Bergen, Norway: International Group for the Psychology of Mathematics Education (PME).
- Boaler, J. (2016). Mathematical mindsets: Unleashing students' potential through creative math, inspiring messages, and innovative teaching. San Francisco, CA: Jossey Bass.
- Brown, W. (2015). Introduction to algorithmic thinking. Available at: www.cs4fn.com/algorithmicthinking.php, Erişim Tarihi: 21.07.2019
- Büyüköztürk, Ş., Akgün, Ö. E., Demirel, F., Karadeniz, Ş., ve Çakmak, E. K. (2016). Bilimsel araştırma yöntemleri (21. basım). Ankara: Pegem Akademi Yayınları.
- Cai, J. (2000). Mathematical thinking involved in US and Chinese students' solving of process-constrained and process-open problems. Mathematical Thinking and Learning, 2(4), 309-340.
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. design for teaching and learning in a networked world (pp. 17-27). Springer International Publishing.

- Çatlak Ş., Tekdal M. & Baz F.Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*. Vol 4 No 3 (2015),13-25
- Çekici E., & Yıldırım H. (2011). Matematik eğitimi üzerine bir inceleme. *Marmara Üniversitesi, İ.İ.B.F. Dergisi, CİLT XXXI, Sayı II, S. 175-196, 2011.*
- Çelik, T. & Karaaslan, E. (2004). Bilgi toplumuna geçiş süreci, Ege Üniversitesi İletişim Fakültesi.
- Chmiliar, I. (2010). Multiple-case designs. In A. J. Mills, G. Euepas & E. Wiebe (Eds.), *Encyclopedia of case study research* (pp 582-583). USA: SAGE Publications
- Çiltaş, A. & Yılmaz, K. (2013). İlköğretim matematik öğretmen adaylarının teoremlerin ifadeleri için kurmuş oldukları matematiksel modeller. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(2).
- Demir, Ü., & Cevahir, H. (2020). Algoritmik düşünme yeterliliği ile problem çözme becerisi arasındaki ilişkinin incelenmesi: Mesleki ve teknik anadolu lisesi örneği. *Kastamonu Education Journal*, 28(4), 1610-1619. doi: 10.24106/kefdergi.4179
- Denzin, N. K., & Lincoln, Y. S. (2000). *Handbook of qualitative research*. London: Sage Publications.
- Dilaver, Ü. (2016). Öğretim ilke ve yöntemleri. Ankara: Pegem Akademi.
- Dreyfus, T. (2007). Processes of Abstraction in Context the Nested Epistemic Actions Model.
- Dourish, P. (2016). Algorithms and their others: Algorithmic culture in context. *Big Data & Society* July–December 2016: 1–DOI: 10.1177/2053951716665128

- Erdem, E. (2018). Blok tabanlı ortamlarda programlama öğretimi sürecinde farklı öğretim stratejilerinin çeşitli değişkenler açısından incelenmesi. Yüksek Lisans Tezi. Başkent Üniversitesi. Eğitim Bilimleri Enstitüsü. Ankara.
- Erdem, U., & Uçak, H.H. (2020). Eğitimde yeni bir yön arayışı bağlamında “21. yüzyıl becerileri ve eğitim felsefesi”. Uşak Üniversitesi Eğitim Araştırmaları Dergisi, 6(1), 76- 93.
- Erümit, K. A., Karal, H., Şahin, G., Aksoy, D. A., Aksoy, A. & Benzer, A. İ. (2018). Programlama öğretimi için bir model önerisi: Yedi adımda programlama. Eğitim ve Bilim, 44,197.
- Fındıkçı, İ. (1998). Bilgi toplumunda eğitim ve öğretmen. Bilgi Toplumu Dergisi, 1. Türk Dünyası Araştırmaları Vakfı Yayını, s.83-91
- Foerster K.T. (2016) “Integrating programming into the mathematics curriculum: combining scratch and geometry in grades 6 and 7,” in proceedings of the 17th annual conference on information technology education, pp. 91–96.
- Förster E.-C, Förster “K.-T, Löwe T. (2018). Teaching programming skills inprimary school mathematics classes: an evaluation using game programming”9th IEEE Global Engineering Education Conference (EDUCON 2018)
- Frenkel, E. (2013). Loveand math: TheHeart of hidden reality. NY: Basic Books.
- Genç, Z. & Karakuş, S. (2011). Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı. 5th International Computer&Instructional Technologies Symposium (ICITS), (pp. 981-987) Elâzığ, Turkey.
- Gökçek, T. (2004). The role of technology in teaching and learning mathematics. Akademik Bilişim Konferansında sunulan sözlü bildiri, Karadeniz Teknik Üniversitesi, Trabzon.

- Gülbahar, Y. & Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 13(1).
- Gündüz, Y. S. (2016). Algoritmalar ve programlama. *Anadolu Üniversitesi Açıköğretim Yayınları*, Eskişehir, 70-71.
- Hart, E. W., & Sandefur, J. (Eds.). (2018). *Teaching and learning discrete mathematics worldwide: Curriculum and research*. Springer.
- Hativa, N. & Cohen, D. (1995). Self learning of negative number concepts by lower division elementary students through solving computer-provided numerical problems. *Educational Studies in Mathematics*, 28(2), 401–431.
- Hayes, B. & Stacey, K. (2003). *Teaching negative number using integer tiles*. Yayınlanmamış Doktora Tezi, Melbourne Üniversitesi, USA.
- Hershkowitz, R., Schwarz, B. B., & Dreyfus, T. (2001). Abstraction in context: Epistemic actions. *Journal for Research in Mathematics Education*, 32, 195–222.
- Hisar, F. M. (2020). Yedinci sınıf çokgenler konusunda 5E öğrenme döngüsüne göre epistemik eylemlerin RBC soyutlama modeliyle incelenmesi. Doktora Tezi. Eskişehir: Anadolu Üniversitesi Eğitim Bilimleri Enstitüsü
- ISTE Standards for Students. International Society for Technology in Education (ISTE), 2016, www.iste.org/standards/for-students
- İstanbul İl Milli Eğitim Müdürlüğü <http://istanbul.meb.gov.tr/www/temel-fikirlerden-temel-bilimlere-bilim-olimpiyatları/icerik/2150> [Erişim Tarihi: 30.05.2021].
- Kalyoncu, A. T. (2012). Yirmi birinci yüzyılda öğrencilerin sahip olması gereken bazı temel becerilere ilişkin yönetici ve öğretmen görüşleri. Yüksek Lisans Tezi. Yeditepe Üniversitesi. Sosyal Bilimler Enstitüsü. İstanbul.

- Karabak, D. & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. [Çevrim-ici: <http://jret.org/FileUpload/ks281142/File/21.karabak.pdf>, Erişim Tarihi: 20.06.2019]
- Kaup C. F. (2020) Examining educational staff's expansive learning process, to understand the use of digital manipulative artefacts to support the students' computational thinking and mathematical understanding Aalborg University, Faculty of Culture and Learning, Denmark, Proceedings of the 10th ERME Topic Conference MEDA 2020- ISBN 978-3-9504630-5-7 P. 53-60
- Kazak, V. (2012). İlköğretim altıncı sınıf öğrencilerinin kesirlerde toplama işlemine yönelik sözel problem kurma ve problem çözme becerilerinin incelenmesi. Atatürk Üniversitesi Eğitim Bilimleri Enstitüsü, Erzurum.
- Kılan, N. K. (2007). Bilgisayarın canı: Algoritma – Harzemiyet. 24. Bilişim Kurultayı Ankara, 14-16 Kasım 2007.
- Kilhamn, C. (2008). Making sense of negative numbers through metaphorical reasoning. Yayımlanmamış yüksek lisans tezi, Göteborgs University.
- Kirk J. & Miller M.L. (1986). Reliability and validity in qualitative research, qualitative research methods. Online Publication Date: January 1, 2011.
- Knuth, D. E. (1980). Algorithms in modern mathematics and computer science. Stanford Department Of Computer Science Report No. Stan-Cs-80-786, Ocak, 1980.
- Knuth, Donald E. (1981), The art of computer programming volume 2: Seminumerical algorithms, Addison-Wesley, Mass.
- Knuth, D. E. (2014). Art of computer programming, volume 2: Seminumerical algorithms: Addison-Wesley Professional.

- Kobsiripat, W. (2015). Effects of the media to promotethescratch programming capabilities creativity of elementary school students. *Procedia-Socialand Behavioral Sciences*, 174, 227-232.
- Koshy, V. (2002). *Teaching gifted children 4-7: A guide for teachers*. London: David Fulton Publishers Ltd.
- Köroğlu, H., & Yeşildere, S. (2004). İlköğretim yedinci sınıf matematik dersi tam sayılar ünitesinde çoklu zekâ teorisi tabanlı öğretiminin öğrenci başarısına etkisi. *GÜ, Gazi Eğitim Fakültesi Dergisi*, 24(2), 25-41.
- Köse N. & Yavuzsoy, N. (2008). İlköğretim 5. sınıf öğrencilerinin dinamik geometri yazılımı Cabri geometriyle simetriyi anlamlandırmalarının belirlenmesi: bir eylem araştırması. (Yayımlanmamış doktora tezi). Anadolu Üniversitesi, Eğitim Bilimleri Enstitüsü. Eskişehir.
- Küçüköğlu, A. (2018). 21. yüzyılda eğitim. ed. Özcan Demirel, Zeki Kaya, Eğitime Giriş, 14. baskı, Pegem A. Yay. Ankara 2018, ss.379-398.
- Külekcı, Ş., Çelik, F., Koçyiğit, E. & Macit, H. G. (2013). Scratch Eğitimi Ders Notları. 14.10.2020 tarihinde http://uekae.bilgem.tubitak.gov.tr/sites/images/scratch_kitabi.pdf adresinden alınmıştır.
- Lagrange, J. B. (2014). Algorithms. In S. Lerman (Ed.), *Encyclopedia of mathematics education* (pp. 32–36). Springer.
- Lagrange, J. B. & Laval, D. (2019). connected working spaces: the case of computer programming in mathematics education. Eleventh Congress of the European Society for Research in Mathematics Education, Utrecht University, Feb 2019, Utrecht, Netherlands.
- LeCompte, M. D., & Goetz, J. P. (1982). Problems of reliability and validity in ethnographic research. *Review of Educational Research* Spring, 52(1), 31-60.

- Levitin A. (2012) Introduction to the design & Analysis of algorithms, 3rd Edition, Pearson.
- Lewin, C., Somekh, B., Steadman, S. (2008). Embedding interactive whiteboards in teaching and learning: The Process of Change in Pedagogic Practice. *Education and Information Technologies*. 13, (4), 291- 303.
- Lockwood, E. E., DeJarnette, A., Asay, A. & Thomas, M. (2016). Algorithmic thinking: an initial characterization of computational thinking in mathematics. In M. B. Wood, E. E. Turner, M. Civil, & J. A. Eli (Eds.), *Proceedings of the 38th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education* (pp. 1588-1595). Tucson, AZ: The University of Arizona
- Lou, Y., Abrami, P. C., & d'Apollonia, S. (2001). Small group and individual learning with technology: A meta-analysis. *Review of Educational Research*, 71(3), 449-521.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming environment. *Transactions on Computing Education*, 10(4), 1- 15.
- Marimuthu, M. & Govender, P. (2018). Perceptions of Scratch programming among secondary school students in KwaZulu-Natal, South Africa. *The African Journal of Information and Communication (AJIC)*, 21, 51-80. <https://doi.org/10.23962/10539/26112>
- Marques, O. F., & Marques, M. T. (2012). No Problem? No research, little learning big problem! *Systemics, Cybernetics and Informatics*, 10(3), 60-62.
- Milli Eğitim Bakanlığı (2013). [MEB]. Ortaokul matematik dersi (5-8. sınıflar) öğretim programı. Ankara: MEB Yayınevi.
- Milli Eğitim Bakanlığı [MEB], (2018). Ortaokul matematik dersi (5, 6, 7 ve 8. sınıflar) sınıflar öğretim programı ve kılavuzu. Ankara: Ankara: Talim ve Terbiye Kurulu Başkanlığı.

- Moala, J.G. (2019). Creating algorithms by accounting for features of the solution: the case of pursuing maximum happiness. *Mathematics Education Research Journal*.
- Modeste, S. (2016). Impact of informatics on mathematics and its teaching. In F. Gadducci & M. Tavosanis (Eds.), *History and Philosophy of Computing* (pp. 243-255). Springer.
- Modeste S., Beauvoir S., Chappelon J., Durand-Guerrier V., León N and Meyer A . (2019). Proof, reasoning and logic at the interface between Mathematics and Computer Science : toward a framework for analyzing problem solving. In U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Eds.), *Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education* (pp. 284-291). Utrecht, the Netherlands: Freudenthal Group & Freudenthal Institute, Utrecht University and ERME.
- Moreno, J, Robles, G. & Román, M. (2015) Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED-Revista de Educación a Distancia*. Número 46. amore
- National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: Author.
- Oluk, A., Korkmaz, Ö. ve Oluk, H. A. (2018). Scratch'ın 5. sınıf öğrencilerinin algoritma geliştirme ve bilgi-işlemsel düşünme becerilerine etkisi. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 9(1), 54-71.
- Orhan, U. (2010). Algoritma ve programlamaya giriş ders notları Erişim Adresi: https://bmb.cu.edu.tr/uorhan/DersNotu/Algoritma_Ders_Notu.pdf Erişim Tarihi: 24.10.2021.
- Özdeş, H. (2013). 9. sınıf öğrencilerinin doğal sayılar konusundaki kavram yanılgıları. Yüksek lisans tezi. Adnan Menderes Üniversitesi Sosyal Bilimler Enstitüsü, pozitAydın.

- Özkan, N. (2020). 8. sınıf öğrencilerinin scratch kullanarak tasarladıkları terazi etkinliklerinde eşittir işareti ve değişken kavramının incelenmesi. Yüksek Lisans Tezi. Mersin Üniversitesi Eğitim Bilimleri Enstitüsü, Mersin.
- Özmantar, M. F. & Monaghan, J. (2007). A Dialectical Approach to the Formation of Mathematical Abstractions. *Mathematics Education Research Journal*, 19(2), 89-112.
- Papert, S. (1980) *Mindstorms*, Brighton, Harvester Press.
- Patton, M. Q. (2018). Nitel araştırma ve değerlendirme yöntemleri (M. Butun ve S. B. Demir,Cev.) Ankara: Pegem Akademi.
- Permatasari, L., Yuana, R. A., & Maryono, D. (2018). Implementation of Scratch application to improve learning outcomes and student motivation on basic programming subjects. *Indonesian Journal of Informatics Education*, 2(2), 95–102.
- Polat, S. & Özcan, A. (2014). Akıllı tahta kullanımıyla ilgili sınıf öğretmenlerinin görüşleri, *Kastamonu Eğitim Dergisi*, 22(2), 439-455.
- Pusmaz, A. (2008). Matematik öğretmenlerinin problem çözme sürecinin belirlenmesi ve bu sürecin geliştirilmesinde web tabanlı mesleki gelişim çalışmasının değerlendirilmesi. *Marmara Üniversitesi, Eğitim Bilimleri Enstitüsü, İstanbul*.
- Rasmussen, C., Zandieh, M., King, K., & Teppo, A. (2005). Advancing mathematical activity: A practiceoriented view of advanced mathematical thinking. *Mathematical Thinking and Learning*, 7(1), 51–73.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch: Programming for All*. *Communications of the ACM*, November 2009.

- Sabella, M.S., Redish, E. F., (1995). Student understanding of topics in linear algebra, Physics Education Research Group University of Maryland Physics Department College Park, 1-6
- Saka, E., & Çelik, D. (2018). Öğretmen adaylarının matematiksel modelleme problemlerini çözme sürecinde teknolojinin rolü. AdıYalçın University Journal of Educational Sciences, 8, 116-150.
- Sarıoğlu, T., & Kartal, G. (2017). Bir yöntem olarak drama bilişim teknolojileri öğretiminde iyi bir seçenek olabilir mi? İlköğretim Online, 16(1).
- Scratch About. (2015). [Çevrim-ici: <https://scratch.mit.edu/about/>, Erişim tarihi: 20.06.2019].
- Shin, S., Park, P., & Bae, Y. (2013). The effects of an information-technology gifted program on friendship using Scratch programming language and clutter. International Journal of Computer and Communication Engineering, 2(3), 246. [Çevrim-ici: <http://www.ijcce.org/papers/181-J028.pdf>, Erişim Tarihi: 20.06.2019].
- Shin, S., & Park, P. (2014). A study on the effect affecting problem solving Ability of Primary Students through the Scratch Programming. [Çevrim-ici: http://onlinepresent.org/proceedings/vol59_2014/27.pdf, Erişim Tarihi: 20.06.2019].
- Skemp, R. E. (1986). The psychology of learning mathematics. UK: Penguin Books.
- Stein, M. K., Correnti, R., Moore, D., Russell, J. L. ve Kelly, K. (2017). Using the oryand me asurement to sharpen conceptualizations of mathematics teaching in the common coreera. AERA Open, 3(1), 1-20. doi:10.1177/2332858416680566
- Stewart, I. (2012). In pursuit of the unknown: 17 equations that changed the world. Hardcover – March 13, 2012.
- Şahin & Namlı (2017) Algoritma eğitiminin problem çözme becerisi üzerine etkisi, Recep Tayyip Erdoğan Üniversitesi Sosyal Bilimler Dergisi) 5: 135-153

Şan, M. K. (2006). Sanayi sonrası toplum kuramları, Sakarya Üniversitesi Sosyoloji Bölümü, 2006.

Şengül, S. & Zengin N. (2005). “Tam öğrenme ilkeleri doğrultusunda farklı öğretim yöntemleri ile işlenen matematik dersinin ilköğretim 7.sınıf öğrencilerinin matematik başarı düzeylerine etkisi”, M.Ü. Eğitim Bilimleri Enstitüsü. 2. Lisansüstü Eğitim Sempozyumu, 26-28 Eylül, İstanbul, 2005.

Şengül, S. ve Cantimer, G. G. (2018). Öğrenciler tam sayı kavramından ne anlıyor? Öğrenci gözüyle tam sayılarda kavram imajı. International Journal of Social Science, 65, 29-50. doi: 10.9761/JASSS7444

Şimşek, H. (1997). 21. yüzyılın eşiğinde, paradigmlar savaşı kaostaki türkiye. İstanbul: Sistem

Theodorou, C., & Kordaki, M. (2010). Super Mario: a collaborative game for the learning of variables in programming. International Journal of Academic Research, 2(4), 111-118.

Thomas, M. O. J. (2014). Algorithms. In S. Lerman (Ed.), Encyclopedia of mathematics education (pp. 36–38). Springer.

Trilling, B. & Fadel, C. (2009). 21st Century skills; learning for life in our times. Jossey Bass Company Press.

Ünür, E., & Barut, S (2020). Tam sayıların öğretiminde karikatürlerle zenginleştirilmiş eğitsel matematik hikâyelerinin kullanılmasının öğrencilerin akademik başarısına etkisi. Academy Journal of Educational Sciences, 4(1), 45-52.

Vogt, W. P., Gardner, D. C., & Haefele, L. M. (2012). When To Use What Research Design. New York: Guilford Press.

Yalın, H. İ. (2006). Öğretim teknolojileri ve materyal geliştirme. Ankara: Nobel.

- Yeşildere, S. (2006). Farklı matematiksel güce sahip ilköğretim 6,7 ve 8. Sınıf öğrencilerinin matematiksel düşünme ve bilgiyi oluşturma süreçlerinin incelenmesi. Yayınlanmamış Doktora Tezi. İzmir: Dokuz Eylül üniversitesi, Eğitim Bilimleri Enstitüsü.
- Yıldırım, A. & Şimşek, H. (2011). Sosyal bilimlerde nitel araştırma yöntemleri (8. basım). Ankara: Seçkin Yayınları.
- Yıldırım, A. & Şimşek, H. (2013). Sosyal bilimlerde nitel araştırma yöntemleri (geliştirilmiş 9. baskı). Ankara: Seçkin Yayınları
- Yıldız, M., Çiftçi, E., & Karal, H. (2017). Bilişimsel düşünme ve programlama. Eğitim Teknolojileri Okumaları, TOJET.
- Yılmazel, B. (2016). Algoritmalar ve programlama. Anadolu Üniversitesi Açıköğretim Yayınları, Eskişehir, 3-4.
- Yünkül, E., Durak, G., Çankaya, S., & Mısırlı, Z. A. (2017). Scratch yazılımının öğrencilerin bilgisayarca düşünme becerilerine etkisi. Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi, 11(2), 502-517.
- Yoon, C., Chin, S. L., Moala, J. G., & Choy, B. H. (2018). Entering into dialogue about the mathematical value of contextual mathematising tasks. Mathematics Education Research Journal, 30(1), 21–37.
- Yoon, C., Moala, J. (2018). Making mathematical thinking visible. teaching & learning research initiative.
- Wang, H. Y., Huang, I., & Hwang, G. J. (2014). Effects of an integrated scratch and project based learning approach on the learning achievements of gifted students in computer courses. In Advanced Applied Informatics (IIAIAI), 2014 IIAI 3rd International Conference on (pp. 382-387). IEEE. [Çevrim-ıç: <http://goo.gl/jxQfx0>, Erişim Tarihi: 20.06.2019]

- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.
- Wilson A. and Moffat D. C. (2012). David C. Evaluating scratch to introduce younger schoolchildren to programming school of engineering and computing, Glasgow Caledonian University, Glasgow, Scotland, UK
- Wilson A., Noss R., Hoyle C., Saunders P. & Laura B. (2019). Key factors for successfully embedding a programming approach to the primary maths curriculum at scale, Glasgow Caledonian University, Glasgow.
- Wing, J. M. (2006). Computational thinking. *Communication of ACM* 49:33-35.
- World Economic Forum. (2016). the future of jobs employment, skills and workforce strategy for the fourth industrial revolution. Erişim adresi: http://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf
- Wood, M. B., Turner, E. E., Civil, M., & Eli, J. A. (Eds.). (2016). Proceedings of the 38th annual meeting of the north American chapter of the international group for the psychology of mathematics education. tucson, AZ: The University of Arizona.

EKLER

EK-1 Merkezi Sınavlarda Çıkmış Algoritma Soru Örnekleri

17. $a \neq 0$ ve m, n tam sayılar olmak üzere
 $a^n \cdot a^m = a^{n+m}$ ve $\frac{a^m}{a^n} = a^{m-n}$
 $(a^n)^m = a^{n \cdot m}$ dir.

Aşağıda sadece ön yüzlerinde birer üslü ifadenin yazılı olduğu 4 mavi ve 4 kırmızı kart verilmiştir.

Mavi Kartlar			
2^{-2}	2^3	2^{-1}	2^4

Kırmızı Kartlar			
4^{-1}	4^{-3}	4^2	4^0

Mavi kartlardaki her bir üslü ifade kırmızı kartlardaki kendisine denk olmayan her bir üslü ifade ile birer kez çarpılarak yeni üslü ifadeler elde ediliyor.

Elde edilen bu üslü ifadelerden ikisinin birbirine oranı en çok kaçtır?

A) 2^{12} B) 2^{15} C) 2^{16} D) 2^{17}

2019 LGS sınavında matematik bölümünde sorulan bu soruda öğrencilere üst kısımda formül verilmiş ve ezber durumdan ziyade soruyu anlayıp çözmesi istenmiştir. Özellikle altı çizili kısım en çok ifadesi öğrenciden bir çözüm yolu, algortima, prosedür oluşturması istenmiştir.

EK Şekil 1

LGS 2019 Matematik 17.soru

MATEMATİK

9. Aşağıda dört farklı renkteki kartların her birinden üçer adet verilmiştir. Aynı renkteki kartların üzerinde aynı kareköklü ifade yazmaktadır.

Mavi	Kırmızı
$\sqrt{0,09}$	$\sqrt{0,25}$

Yeşil	Turuncu
$\sqrt{0,64}$	$\sqrt{1,44}$

Eymen, bu kartlardan seçerek üstlerinde yazan kareköklü ifadeleri topladığında bir doğal sayı elde etmektedir.

Buna göre Eymen en fazla kaç kart seçmiştir?

A) 8 B) 9 C) 10 D) 11

2019 LGS sınavında matematik bölümünde sorulan bu soruda öğrenciler ifadeleri kök dışına çıkarıp teker teker deneyerek cevaba ulaşabilirler. Ancak sınav süreli olduğundan dolayı hızlı çözümler bulmak sınavda süre kazandırmaktadır. Bu soruda da hepsini toplayıp çıkarma yoluna gidebilirler. Bu da farklı düşünmek olarak algoritmik düşünce ile ilgilidir.

EK Şekil 2

LGS 2020 Matematik 9.soru

6.

Yukarıdaki şekilde verilen her bir dairenin içine birbirinden farklı birer doğal sayı yazılacaktır. Bu sayılardan ikisi şekilde verilmiştir. Bulundukları dörtgenin köşelerindeki dairelerde yazan dört sayının çarpımına eşit olan A ve B sayıları aralarında asaldır.

Buna göre $A + B$ en az kaçtır?

A) 162 B) 191 C) 258 D) 289

2019 LGS sınavında matematik bölümünde sorulan bu soruda öğrencilerden sayı yerleştirmeleri istenmiştir. İki sayı arasındaki ilişkiyi anlayıp ve küçük sayılar yerleştirmek gerekmektedir. bu durum normal matematik derslerinde çok fazla gösterilemeyeceğinden dolayı öğrenciden dikey matematikleştirme yapması istenmiştir.

EK Şekil 3

LGS 2021 Matematik 6.soru

10. Rakamları birbirinden ve sıfırdan farklı üç basamaklı bir doğal sayının onlar basamağındaki rakam diğer basamaklarındaki rakamları tam bölüyorsa bu sayıya ortakatlı sayı denir.

Örneğin, 428 bir ortakatlı sayıdır.

Buna göre, en büyük ortakatlı sayı ile en küçük ortakatlı sayının farkı kaçtır?

A) 723 B) 727 C) 736 D) 742 E) 745

2020 YKS sınavında matematik bölümünde sorulan bu soruda öğrencilere hiç bilmedikleri bir ifade anlatılmış ve bir örnek verilmiştir. Bu anlatımı ve örneği anlayıp öğrencinin soruyu çözmesi beklenmektedir. Bu durum ise farklı durumları analiz edip algoritmalar oluşturması algoritmik düşünce ile yakından ilgilidir.

EK Şekil 4

YKS 2020 Temel Yeterlilik Sınavı Matematik 10.soru

EK-2 Modüler Aritmetik Sınavı

1. $128 \equiv x \pmod{5}$ denkleğini sağlayan en küçük x doğal sayı değeri kaçtır?
2. $172 \equiv x \pmod{7}$ denkleğini sağlayan en küçük üç x pozitif tam sayı değerinin toplamı kaçtır?
3. $318 \equiv x \pmod{6}$ denkleğini sağlayan en küçük x tam sayı değeri kaçtır?
4. $49 \equiv x \pmod{7}$ denkleğini sağlayan en küçük pozitif x tam sayı değeri kaçtır?
5. $195 \equiv x \pmod{9}$ denkleğini sağlayan iki basamaklı en küçük iki x doğal sayı değerinin toplamı kaçtır?

EK-3 Milli Eğitim Bakanlığı Araştırma İzin Yazısı



T.C.
İSTANBUL VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : E-59090411-20-18834207
Konu : Anket ve Araştırma İzin Talebi

06/01/2021

VALİLİK MAKAMINA

- İlgi : a) Marmara Üniversitesinin 16.12.2020 tarihli ve 2000331766 sayılı dilekçesi.
b) Bakanlığımızın 21.01.2020 tarih ve 1563890/ 2020/2 No'lu genelgesi
c) Müdürlüğümüz Araştırma ve Anket Komisyonunun 04.01.2021 tarihli tutanağı.
d) Bakanlığımız Ortaöğretim Genel Müdürlüğünün 18.12.2020 tarihli ve 18219678 sayılı yazısı.

Marmara Üniversitesi Yüksek Lisans Öğrencisi Emre DURASI'nın "Ortaokul Öğrencilerinin Scratch Programı İle Algoritma Üretme Süreçlerinin ve Matematiksel Muhakeme Becerilerinin İncelenmesi" konulu tezine dair araştırma çalışmasını İstanbul genelinde ortaokullarda akış şeması, anket uygulama isteği hakkındaki ilgi (a) yazı ve ekleri müdürlüğümüzce incelenmiştir.

Bakanlığımızın ilgi (d) yazısında 22.01.2021 tarihine kadar uzaktan eğitime devam edileceği belirtilmiştir.

Araştırmacının söz konusu talebi **yüz yüze eğitime geçilmesine müteakiben**, bilimsel amaç dışında kullanılmaması, uygulama sırasında bir örneği müdürlüğümüzde muhafaza edilen mühürlü ve imzalı veri toplama araçlarının kurumlarımıza araştırmacı tarafından ulaştırılarak uygulanması, katılımcıların gönüllülük esasına göre seçilmesi, araştırma sonuç raporunun kamuoyuyla paylaşılması koşuluyla, okul idarelerinin denetim, gözetim ve sorumluluğunda, eğitim-öğretimi aksatmayacak şekilde ilgi (b) Bakanlık emri esasları dâhilinde uygulanması, araştırma bittikten sonra 2 (iki) hafta içerisinde araştırma sonuç raporu hakkında Müdürlüğümüz istemanket@meb.gov.tr adresine mail yoluyla bilgi verilmesi kaydıyla Müdürlüğümüzce uygun görülmektedir.

Makamlarınızca da uygun görülmesi halinde olurlarınıza arz ederim.

Levent YAZICI
İl Millî Eğitim Müdürü

OLUR
06/01/2021
Dr. Hasan Hüseyin CAN
Vali a.
Vali Yardımcısı

- Ek:
1- Genelge.
2- Komisyon Tutanağı.

Adres : İstanbul Millî Eğitim Müdürlüğü-Strateji Geliştirme
Şb.Md.Binbirdirek Mah.İmran Öktem Cad.No:1 Sultanahmet-Fatih/İST.

Bu belge güvenli elektronik imza ile imzalanmıştır.

Belge Doğrulama Adresi : <https://www.turkiye.gov.tr/meb-ebys>

Bilgi için:

Telefon No : 0 (212) 384 36 28
E-Posta: ist.sgb34@gmail.com
Kep Adresi : meb@hs01.kep.tr

Internet Adresi:

Unvan : Dilek ALAI
Faks:

Bu evrak güvenli elektronik imza ile imzalanmıştır. <https://evraksorgu.meb.gov.tr> adresinden 69b6-7746-3657-9a11-8e7b kodu ile teyit edilebilir.



EK-4 Veli İzin Formu

Sayın Veli; Çocuğunuzun katılacağı bu çalışma, “Ortaokul Öğrencilerinin Scratch Programı ile Algoritma Üretme Süreçlerinin ve Matematiksel Muhakeme Becerilerinin İncelenmesi” adıyla, 15/02/2021 – 28/02/2021 tarihleri arasında yapılacak bir araştırma uygulamasıdır.

Araştırmanın Hedefi: Bu araştırmanın hedefi, akademik başarısı yüksek ortaokul öğrencilerinin Scratch programı ile algoritma oluştururken sergiledikleri süreci ve matematiksel muhakeme becerilerini incelemektir.

Araştırma Uygulaması: Görüşme / Gözlem (Çevrimiçi) şeklindedir.

Araştırma T.C. Millî Eğitim Bakanlığı’nın ve okul yönetiminin de izni ile gerçekleştirilmektedir. Araştırma uygulamasına katılım tamamıyla gönüllülük esasına dayalı olmaktadır. Çocuğunuz çalışmaya katılıp katılmamakta özgürdür. Araştırma çocuğunuz için herhangi bir istenmeyen etki ya da risk taşımamaktadır. Çocuğunuzun katılımı **tamamen sizin isteğinize bağlıdır**, reddedebilir ya da herhangi bir aşamasında ayrılabilirsiniz. Araştırmaya katılmamama veya araştırmadan ayrılma durumunda öğrencilerin akademik başarıları, okul ve öğretmenleriyle olan ilişkileri etkilemeyecektir.

Çalışmada öğrencilerden kimlik belirleyici hiçbir bilgi istenmemektedir. Cevaplar tamamıyla gizli tutulacak ve sadece araştırmacılar tarafından değerlendirilecektir.

Uygulamalar, genel olarak kişisel rahatsızlık verecek sorular ve durumlar içermemektedir. Ancak, katılım sırasında sorulardan ya da herhangi başka bir nedenden çocuğunuz kendisini rahatsız hissederse cevaplama işini yarıda bırakıp çıkmakta özgürdür. Bu durumda rahatsızlığın giderilmesi için gereken yardım sağlanacaktır. Çocuğunuz çalışmaya katıldıktan sonra istediği an vazgeçebilir. Böyle bir durumda veri toplama aracını uygulayan kişiye, çalışmayı tamamlamayacağını söylemesi yeterli olacaktır. Anket çalışmasına katılmamak ya da katıldıktan sonra vazgeçmek çocuğunuza hiçbir sorumluluk getirmeyecektir.

Onay vermeden önce sormak istediğiniz herhangi bir konu varsa sormaktan çekinmeyiniz. Çalışma bittikten sonra bizlere telefon veya e-posta ile ulaşarak soru sorabilir, sonuçlar hakkında bilgi isteyebilirsiniz. Saygılarımızla,

Araştırmacı : Emre DURASI

İletişim bilgileri:

Adres:

Telefon:

Velisi bulunduğum sınıfı numaralı öğrencisi

.....

.....’in yukarıda açıklanan araştırmaya katılmasına izin veriyorum. (Lütfen formu imzaladıktan sonra çocuğunuzla okula geri gönderiniz*).

...../...../.....

EK-5 Katılımcı Formu

Sayın Katılımcımız

Katılacağınız bu çalışma, Ortaokul Öğrencilerinin Scratch Programı ile Algoritma Üretme Süreçlerinin ve Matematiksel Muhakeme Becerilerinin İncelenmesi” adıyla, Emre Durası tarafından 15/02/2021 – 28/02/2021 tarihleri arasında yapılacak bir araştırma uygulamasıdır.

Araştırmanın Hedefi: Bu araştırmanın hedefi, akademik başarısı yüksek ortaokul öğrencilerinin Scratch Programı ile algoritma oluştururken sergiledikleri süreci ve matematiksel muhakeme becerilerini incelemektir

Araştırmanın Nedeni: Yüksek Lisans Tez Çalışması

Araştırmanın Yapılacağı Yer(ler): Ev ortamı

Araştırma Uygulaması: ☐ Görüşme ☐ Gözlem
(Çevrimiçi)

Araştırma T.C. Millî Eğitim Bakanlığı’nın ve okul/kurum yönetiminin izni ile gerçekleştirilmektedir. Araştırma uygulamasına katılım tamamıyla gönüllülük esasına dayalı olmaktadır. Çalışmada sizden kimlik belirleyici hiçbir bilgi istenmemektedir. Cevaplar tamamıyla gizli tutulacak ve sadece araştırmacılar tarafından değerlendirilecektir. Veriler sadece araştırmada kullanılacak ve üçüncü kişilerle paylaşılmayacaktır.

Uygulamalar, kişisel rahatsızlık verecek sorular ve durumlar içermemektedir. Ancak, katılım sırasında sorulardan ya da herhangi başka bir nedenden rahatsız hissederseniz cevaplama işini yarıda bırakabilirsiniz.

Katılımı onaylamadan önce sormak istediğiniz herhangi bir konu varsa sormaktan çekinmeyiniz. Çalışma bittikten sonra bizlere telefon veya e-posta ile ulaşarak soru sorabilir, sonuçlar hakkında bilgi isteyebilirsiniz. Saygılarımızla

Araştırmacı : Emre DURASI

İletişim bilgileri:

Adres :

Telefon:

Yukarıda bilgileri bulunan araştırmaya katılmayı kabul ediyorum.

...../...../.....
İsim-Soyisim

İmza:

Katılımcı Adı-Soyadı :

Telefon Numarası :