

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ ÖĞRETMENLİĞİ PROGRAMI
YÜKSEK LİSANS TEZİ

**6.SINIF ÖĞRENCİLERİNE SCRATCH VE KODU
GAME LAB PROGRAMLAMA DİLLERİNİN
ÖĞRETİMİNDE ÖĞRENCİLERİN TUTUM, ÖZ
YETERLİLİK VE AKADEMİK BAŞARILARININ
KARŞILAŞTIRILMASI**

Anıl DİNÇER

**İzmir
2018**

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ EĞİTİM BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ ÖĞRETMENLİĞİ PROGRAMI
YÜKSEK LİSANS TEZİ

**6.SINIF ÖĞRENCİLERİNE SCRATCH VE KODU
GAME LAB PROGRAMLAMA DİLLERİNİN
ÖĞRETİMİNDE ÖĞRENCİLERİN TUTUM, ÖZ
YETERLİLİK VE AKADEMİK BAŞARILARININ
KARŞILAŞTIRILMASI**

Anıl DİNÇER

**Danışman
Doç. Dr. Bahar BARAN**

**İzmir
2018**

Yemin Metni

Yüksek lisans tezi olarak sunduğum “6. Sınıf Öğrencilerine Scratch ve Kodu Game Lab Programlama Dillerinin Öğretiminde Öğrencilerin Tutum, Öz-Yeterlilik ve Akademik Başarılarının Karşılaştırılması” adlı çalışmanın tarafımdan, bilimsel etik ve geleneklere aykırı düşecek bir yardıma başvurmaksızın bizzat kendim tarafından hazırlandığını ve yararlandığım yapıtların kaynakçada belirtilenlerden oluştuğunu, bunlara atıf yaparak yararlanmış olduğumu belirtir ve bunu onurumla doğrularım.

19/06/2018

Anıl DINCER

Değerlendirme Kurulu Üyeleri

Eğitim Bilimleri Enstitüsü Müdürlüğüne

İşbu çalışma, jürimiz tarafından Bilgisayar Ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Öğretmenliği Yüksek Lisans Programında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Başkan : Doç. Dr. Bahar BARAN

Üye : Prof. Dr. Ercan AKPINAR

Üye : Dr. Öğr. Üyesi Esra YECAN

Onay

Yukarıda imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

21.06.2018
YILMAZ

Prof. Dr. Süha YILMAZ
Enstitü Müdürü V.



Teşekkür

Bu çalışmayı yürütürken bana fikirleriyle yol gösteren, yardımlarını hiç esirgemeyen ve tezimin her aşamasında sürekli destek olan saygıdeğer danışmanım Doç. Dr. Bahar BARAN'a hoşgörüsü, sabrı ve emekleri için teşekkürlerimi saygıyla sunuyorum.

Üniversite hayatım boyunca eleştiri ve fikirlerine çok değer verdiğim, zorlandığım anda çekinmeden fikirlerine danıştığım ve çalışmamın oluşmasında kattığı fikirlerle her zaman destek olan Prof. Dr. Ercan AKPINAR'a ayrıca teşekkür ederim

Çalışmayı gerçekleştirdiğim Yeniceköy Ortaokulu'nda bana çalışmam esnasında uygun ortam sağlayan ve gerekli desteği veren Okul Müdürüm Vahdettin BESLEN, Müdür Yardımcım Tuncay GÜLER ve Reşat ARSLAN'a şükranlarımı sunuyorum.

Çalışmamın her aşamasında beni destekleyen, motive eden ve yardımlarını esirgemeyen sevgili eşime ayrıca teşekkürü bir borç bilirim.

Özet

6. Sınıf Öğrencilerine Scratch ve Kodu Game Lab Programlama Dillerinin Öğretiminde Öğrencilerin Tutum, Öz Yeterlilik ve Akademik Başarılarının Karşılaştırılması

Bu çalışmanın amacı Scratch ve Kodu Game Lab ile programlama öğretiminin 6. Sınıf öğrencilerin programlama akademik başarısı, tutumu ve öz-yeterlik algıları üzerine olan etkisini incelemektir. Araştırmanın çalışma grubunu Çanakkale ili Biga ilçesinde bulunan Yeniceköy Ortaokulunda iki şubede öğrenim gören 27 ortaokul 6. sınıf öğrencisi oluşturmaktadır. Araştırmada iki adet deney grubu bulunmaktadır. Bu deney gruplarında öğrenci seçiminde küme örnekleme yöntemi kullanılmıştır. Ders içeriğinin ve ders planlarının oluşturmasına katkı sağlamak amacıyla yürütülen ihtiyaç analizi kapsamında Bilişim Teknolojileri Öğretmenleri ile yarı yapılandırılmış mülakatlar yürütülmüştür. Programlama öğretimi toplam 20 ders saati olmak üzere, bir gruba Scratch programlama dili, diğer gruba Kodu Game Lab programlama dili anlatılmıştır. Deney gruplarına ön-test ve son test olarak, Programlama Dilleri Akademik Başarı Testi, Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğretimine Yönelik Tutum Ölçeği ve Programlamaya İlişkin Öz-Yeterlik Ölçeği uygulanmıştır. Ayrıca programlama dili öğretiminin kalıcılık yönünden incelenmesi amacıyla son-test uygulandıktan sekiz hafta sonra tekrar-test olarak Programlama Dilleri Akademik Başarı Testi uygulanmıştır. Araştırmada verileri analiz ederken ilişkisiz örneklem t-testi, ilişkili örneklem t-testi ve Shapiro-Wilk normallik testi uygulanmıştır. Araştırmanın bulguları incelendiğinde Scratch ile programlama öğrenen öğrencilerin, Kodu Game Lab ile öğrenim görenlere göre başarıları anlamlı derecede arttığı görülmüştür. Scratch programlama dili ile öğrenim gören öğrencilerin, Kodu Game Lab ile öğrenim gören öğrencilerin tutumu ve öz-yeterlik algıları arasında anlamlı derecede bir fark olmadığı saptanmıştır. Scratch ve Kodu Game Lab ile programlama eğitimi alan öğrencilerin programlama dillerine karşı olan öz-yeterlik algılarının anlamlı derecede arttığı da görülmüştür. Ayrıca Kodu Game Lab ile programlama eğitiminin tutumu anlamlı derecede arttığı da ortaya çıkmıştır.

Anahtar Kelimeler: Scratch, Kodu Game Lab, Programlama, Başarı, Öz-yeterlik, Tutum.

Abstract

The Comparison of 6th Grade Students' in Terms of Attitudes, Self-Efficacy and Academic Achievement on Teaching of Scratch and Kodu Game Lab Programming Languages

The aim of this study is to examine the impact of teaching programming with the Scratch and Kodu Game Lab on programming academic achievement, attitude and self-efficacy perceptions of 6th grade students. The study group of this research was composed of 27 middle-school students who were studying in two different classes in Yeniceköy Secondary School which is located in Biga, a district of Canakkale province. There were two experimental groups in this research. In these experimental groups, the Cluster Sampling Method has been applied for the selection of students. Semi-structured interviews with Information Technology Teachers were applied within the context of the needs analysis which aims to contribute to the creation of course content and lesson plans. Teaching programming to children One of the experimental group was taught In the programming teaching lesson for children scheduled for 20 hours, the scratch programming language was used in one of the experimental groups, while the cod game lab was used in the other group. Experimental groups were distributed “the Programming Language Academic Achievement Test”, “the Educational Computer Games Assisted Learning”, “the Coding Attitude Scale” and “the Computer Programming Self-Efficacy Scale” both as the pre-test and final-test. After eight weeks, the Programming Language Academic Achievement Test was applied as a re-test for testing retention. The independent sample t-test, dependent sample t-test and Shapiro-Wilk normality test have been applied to analyse data. When the findings of the study indicated that the students in the group of Scratch teaching have significantly improved the programming achievement when compared with the other group have studied with Code Game Lab to learn programming. It has been determined that there was no significant difference between the attitude and self-efficacy perceptions of the students in the group of the Scratch and the Code Game Lab. Self-efficacy perceptions in the groups of both Scratch and Code Game Lab have also increased significantly. It has also emerged that teaching programming with Code Game Lab has increased attitudes of students towards programming significantly.

Keywords: Scratch, Kodu Game Lab, Programming, Academic Achievement, Self-Efficacy, Attitude.

İçindekiler

Yemin Metni	i
Değerlendirme Kurulu Üyeleri	ii
Tez Veri Giriş Formu	iii
Teşekkür	iii
Özet	v
Abstract	vi
Tablo Listesi	xi
Şekiller Listesi	xiii
BÖLÜM I	1
1. GİRİŞ	1
1.1. Problem Durumu	1
1.2. Araştırmanın Amacı ve Önemi	4
1.3. Problem Cümlesi	7
1.3.1. Alt Problemler	7
1.4. Sayıtlılar	7
1.5. Sınırlılıklar	7
1.6. Kısaltmalar	8
1.7. Tanımlamalar	8
BÖLÜM II	9
2. İLGİLİ YAYIN VE ARAŞTIRMALAR	9
2.1. Bilgi İşlemsel Düşünce (Computational Thinking)	9
2.2. Programlama Nedir?	10
2.3. Programlama ile İlgili Temel Kavramlar	13
2.4. Dünyada ve Ülkemizde Programlama ve Kodlama Eğitimi	23
2.4.1. Kodlama ve Programlama Eğitimi	23
2.4.2. Dünyada Kodlama ve Programlama Eğitimi	24
2.4.3. Ülkemizde Kodlama ve Programlama Eğitimi	25
2.5. Alan Yazın Çalışmalar	27
2.5.1. Yükseköğretimde ve Öğretmen Eğitiminde Scratch ve Kodu Game Lab Kullanılması ile İlgili Alan Yazın	27
2.5.2. İlköğretim ve Ortaöğretimde Scratch ve Kodu Game Lab ile İlgili Alan Yazın ...	36
2.5.3. Alan Yazın Özet	52

BÖLÜM III	57
3. YÖNTEM	57
3.1.1. Araştırma Modeli	57
3.1.2. İhtiyaç Analizi: Bilişim Teknolojileri Öğretmenlerinin Programlama Öğretimi İle İlgili Görüşlerinin Belirlenmesi	58
3.1.3. Amaç	58
3.1.4. Yöntem.....	58
3.1.4.1. Mülakat Formu Hazırlama Süreci	58
3.1.4.2. Mülakat Formunun Uygulanması	59
3.1.4.3. İhtiyaç Analizine İlişkin Bulgular	60
3.1.4.3.1. Çalışma Grubuna.....	60
3.1.4.3.2. Programlama Öğretimine İlişkin Bulgular	61
3.1.4.3.3. Katılımcıların Görüşleri	63
3.1.4.4. Sonuç.....	68
3.1.4.4.1. Katılımcıların Görüşleri Doğrultusunda Yapılan Çalışmalar.....	69
3.2. Evren ve Örneklem	71
3.3. Veri Toplama Araçları	72
3.3.1. Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ)	72
3.3.2. Programlamaya İlişkin Öz-Yeterlik Ölçeği (PİÖYÖ).....	73
3.3.3. Programlama Dilleri Akademik Başarı Testi (PDABT)	73
3.4. Uygulamanın Gerçekleştirilme Süreci	77
3.5. Veri Çözümleme Teknikleri.....	87
3.5.1. Betimsel İstatistik Sonuçları	87
3.5.2. Değişkenler	89
3.5.2.1. Bağımsız Değişken	89
3.5.2.2. Bağımlı Değişken.....	89
3.5.3. Araştırmada Kullanılan İstatistiksel Testler.....	89
3.5.3.1. İlişkisiz Örneklemeler <i>t</i> -testi.....	89
3.5.3.2. İlişkili Örneklemeler <i>t</i> -testi.....	90
3.5.3.3. Shapiro-Wilk Normallik Testi	90
BÖLÜM IV	92
4. BULGULAR VE YORUMLAR.....	92

4.1.	Programlamaya Yönelik Başarının İncelenmesi	92
4.1.1.	PDABT Ön-Test Sonuçlarının İncelenmesi.....	92
4.1.2.	PDABT Son-Test Sonuçlarının İncelenmesi	94
4.1.3.	DG1 Grubuna Ait PDABT Ön-Test ve Son-Test Sonuçlarının İncelenmesi.....	96
4.1.4.	DG2 Grubuna Ait PDABT Ön-Test ve Son-Test Sonuçlarının İncelenmesi.....	97
4.1.5.	DG1 ve DG2 Grubuna Ait PDABT Kalıcılık Yönünden Tekrar-Test Sonuçlarının İncelenmesi	99
4.1.6.	DG1 Grubuna Ait PDABT Kalıcılık Yönünden Son-Test ve Tekrar-Test Sonuçlarının İncelenmesi.....	100
4.1.7.	DG2 Grubuna Ait PDABT Kalıcılık Yönünden Son-Test ve Tekrar-Test Sonuçlarının İncelenmesi.....	101
4.2.	Programlamaya Yönelik Tutumun İncelenmesi.....	102
4.2.1.	EBODKÖYTÖ Ön-Test Sonuçlarının İncelenmesi	102
4.2.2.	EBODKÖYTÖ Son-Test Sonuçlarının İncelenmesi.....	103
4.2.3.	DG1 Grubuna Ait EBODKÖYTÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi	104
4.2.4.	DG2 Grubuna Ait EBODKÖYTÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi	105
4.3.	Programlamaya Yönelik Öz-Yeterlik Algısının İncelenmesi	106
4.3.1.	PİÖYÖ Ön-Test Sonuçlarının İncelenmesi.....	106
4.3.2.	PİÖYÖ Son-Test Sonuçlarının İncelenmesi	107
4.3.3.	DG1 Grubuna Ait PİÖYÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi.....	108
4.3.4.	DG2 Grubuna Ait PİÖYÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi.....	109
	BÖLÜM V	110
5.	SONUÇ, TARTIŞMA ve ÖNERİLER	110
5.1.	SONUÇ	110
5.2.	TARTIŞMA	111
5.2.1.	Scratch ve Kodu Game Lab ile programlama öğretiminin programlama dilleri akademik başarısına ve kalıcılığa etkisi.....	111
5.2.2.	Scratch ve Kodu Game Lab ile programlama öğretiminin programlama dilleri tutuma etkisi.....	113
5.2.3.	Scratch ve Kodu Game Lab ile programlama öğretiminin programlama dilleri öz-yeterlik algısına etkisi	114
5.3.	ÖNERİLER.....	115
	KAYNAKÇA.....	117

EKLER.....	127
EK 1:Bilgisayar Programlamaya İlişkin Öz Yeterlilik Ölçeği.....	127
EK 1:Bilgisayar Programlamaya İlişkin Öz Yeterlilik Ölçeği Kullanım İzni	128
EK 2:Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği .	129
EK 2:Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği Kullanım İzni	130
EK 3:Bilişim Teknolojileri Öğretmenlerinin Programlama Öğretirken Nelere Dikkat Ettikleri ve Programlamanın Nasıl Öğretilmesi Konusunda Düşüncelerini İçeren Mülakat Formu..	131
EK 4:Programlama Dilleri Akademik Başarı Testi	134
EK 5:Anket ÖnUygulama İzni.....	138
EK 6: Anket Gerçek Uygulama İzni	139
EK 7:Scratch Ders Planları	140
EK 8: Kodu Game Lab Ders Planları.....	150
EK9. İntihal Raporu.....	160

Tablo Listesi

Tablo 1 Programlama Bilgisi	12
Tablo 2 Yükseköğretimde Öğretmen Eğitiminde Scratch ve Kodu Game Lab ile Yapılan Çalışmalar	32
Tablo 3 İlköğretim ve Ortaöğretimde Scratch ve Kodu Game Lab ile Yapılan Çalışmalar ..	45
Tablo 4 Uygulanan Yarı-Deneysel Desenin Görselleştirilmiş Hali	57
Tablo 5 Öğretmenlerin Cinsiyetine Ait Bilgiler.....	60
Tablo 6 Öğretmenlerin Yaşına Ait Bilgiler.....	60
Tablo 7 Öğretmenlerin Eğitim Durumlarına Ait Bilgiler.....	61
Tablo 8 Öğretmenlerin Görev Sürelerine Ait Bilgiler	61
Tablo 9 Öğretmenlerin Bilgisayar Sınıfına Sahiplik Durumuna Ait Bilgiler	61
Tablo 10 Öğretmenlerin Programlama Anlattığı Sınıf Seviyesi	62
Tablo 11 Öğretmenlerin Programlama Dilleri Öğretirken Kullandığı Programlara Ait Bilgiler	62
Tablo 12 Öğretmenlerin Ne Kadar Süreyle Programlama Anlattığına Ait Bilgiler.....	62
Tablo 13 Programlama Öğretimi Öğretme Zamanı	63
Tablo 14 Programlama Öğretiminde Gerekli Olan Ön Bilgiler.....	64
Tablo 15 Programlama Nasıl Öğretilmeli.....	64
Tablo 16 Programlama Öğretmek İçin Harcanması Gereken Süre.....	65
Tablo 17 Programlama Öğretmek İçin Kullanılması Geren Öğretim Yöntemleri.....	65
Tablo 18 Programlama Öğretiminden Öğrencilerin Motivasyon Durumları.....	66
Tablo 19 Bir Programlama Dili Tercih Sebepleri.....	66
Tablo 20 Öğretmenlerin Derste İşledikleri Konular	67
Tablo 21 Konuların Saatlere Göre Dağılımı	70
Tablo 22 Araştırmayı Oluşturan Örneklemeye Ait Frekanslar.....	71
Tablo 23 Maddelerin konulara göre dağılımı	74
Tablo 24 Deneme Testinde Yer Alan Maddelerin Güçlük ve Ayırıcılık İndeksleri	75
Tablo 25 Nihai Test Madde İstatistikleri	76
Tablo 26 Nihai Test Maddelerin konulara göre dağılımı.....	77
Tablo 27 Ön-test Betimsel İstatistikler	88
Tablo 28 Son-test Betimsel İstatistikler	88
Tablo 29 PDABT Ön-Testlere Ait Shapiro-Wilk Sonuçları	92
Tablo 30 PDABT Ön-testlere Ait İlişkisiz Örneklemeler t-test Sonuçları	93
Tablo 31 DG1 ve DG2 Gruplarına Ait PDABT Maddelere İlişkin Ön-test Sonuçları	94
Tablo 32 PDABT Son-testlere ait Shapiro-Wilk Sonuçları	95
Tablo 33 PDABT Son-testlere Ait İlişkisiz Örneklemeler t-test Sonuçları	95
Tablo 34 DG1 ve DG2 Grubuna Ait PDABT Maddelere İlişkin Son-test Sonuçları	96
Tablo 35 DG1 Grubu PDABT Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları ...	97
Tablo 36 DG2 Grubu PDABT Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları ...	97
Tablo 37 PDABT Maddelere İlişkin Ön-test ve Son-test Sonuçları	98
Tablo 38 PDABT Tekrar-Testlere Ait Shapiro-Wilk Sonuçları	99
Tablo 39 PDABT Tekrar testlere Ait İlişkisiz Örneklemeler t-test Sonuçları.....	99

Tablo 40 DG1 Grubu PDABT Son-Test ve Tekrar-Test İlişkili Örneklemeler t-test Sonuçları	100
Tablo 41 DG2 Grubu PDABT Son-Test ve Tekrar-Test İlişkili Örneklemeler t-test Sonuçları	101
Tablo 42 EBODKÖYTÖ Ön testlere ait Shapiro-Wilk Sonuçları	102
Tablo 43 EBODKÖYTÖ Ön-testlere Ait İlişkısız Örneklemeler t-test Sonuçları.....	102
Tablo 44 EBODKÖYTÖ Son testlere ait Shapiro-Wilk Sonuçları.....	103
Tablo 45 EBODKÖYTÖ Son-testlere Ait İlişkısız Örneklemeler t-test Sonuçları.....	104
Tablo 46 DG1 Grubu EBODKÖYTÖ Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları	104
Tablo 47 DG2 Grubu EBODKÖYTÖ Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları	105
Tablo 48 PİÖYÖ Ön testlere ait Shapiro-Wilk Sonuçları.....	106
Tablo 49 PİÖYÖ Ön-testlere Ait İlişkısız Örneklemeler t-test Sonuçları	106
Tablo 50 PİÖYÖ Son testlere ait Shapiro-Wilk Sonuçları	107
Tablo 51 PİÖYÖ Son-testlere Ait İlişkısız Örneklemeler t-test Sonuçları.....	108
Tablo 52 DG1 Grubu PİÖYÖ Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları ..	108
Tablo 53 DG2 Grubu PİÖYÖ Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları ..	109

Şekiller Listesi

Şekil 1. Akış şeması.....	14
Şekil 2. Scratch programına ait bir görsel.....	16
Şekil 3. Kodu Game Lab programlama diline ait bir görsel	18
Şekil 4. Hacker Can programlama diline ait bir görsel.....	20
Şekil 5.mBlock programlama diline ait bir görsel	21
Şekil6. Code Combat programlama diline ait bir görsel.....	22
Şekil 7. Code.org programlama diline ait bir görsel	23
Şekil 8. Scratch’ ta Örnek Bir Uygulamanın Ekran Görüntüsü	68
Şekil 9. Kodu Game Lab’ da Örnek Bir Uygulamanın Ekran Görüntüsü.....	68
Şekil 10. Programlama için gerekli ön bilgilere ait görsel.....	69
Şekil 11. Örnek bir algoritma görüntüsü.....	78
Şekil 12. Örnek bir Akış Şemasına ait bir örnek.....	78
Şekil 13. Scratch’ ta ekrana yazdırma becerisiyle ilgili bir örnek	79
Şekil 14. Kodu Game Lab ekrana yazdırma becerisiyle ilgili bir örnek	80
Şekil 15. Scratch ile örnek bir toplama işlemiyle ilgili bir örnek	80
Şekil 16. Kodu Game Lab ile örnek bir toplama işlemiyle ilgili bir örnek.....	81
Şekil 17. Scratch ile döngü kullanarak desen çıkartmaya ilişkin bir örnek	82
Şekil 18. Kodu Game Lab ile işaretli alanlarda sürekli hareket eden robotla ilgili bir örnek	82
Şekil 19. Scratch ile değişken kullanarak yapılan bir örnek	83
Şekil 20. Kodu Game Lab ile puan değişkeni kullanılarak yapılan bir örnek	84
Şekil 21. Scratch ile koşul ifadesi kullanılarak yapılan bir örnek.....	85
Şekil 22. Kodu Game Lab ile koşul ifadesi kullanılarak yapılan bir örnek	86
Şekil 23. Scratch’ ta elma toplama oyununa ilişkin bir örnek uygulama.....	86
Şekil 24. Kodu Game Lab ile motosiklet oyununa ilişkin bir örnek uygulama	87

BÖLÜM I

1. GİRİŞ

1.1. Problem Durumu

21. yüzyılda bireylerden, eleştirel düşünme ve problem çözme becerileri gibi bilişsel gelişimi destekleyen becerilerin kazanmaları beklenmektedir (Scott, 2017). Programlama eğitiminin bu becerilerin kazanılmasında oldukça etkili olduğu görülmektedir (Harlow ve Forret, 2010; Nam, Kim ve Lee, 2010; Oluk ve Korkmaz, 2016; Ozoran, Çağıltay ve Topallı, 2012; Shin ve Park, 2014; Taylor, Yünkül, Durak, Çankaya ve Mısırlı, 2017). Programlama eğitimi öğrencilere bazı beceriler kazandırmada etkili olsa da öğrenilmesi genel anlamda soyut düşünme gerektirdiği için öğrencilere zor ve karmaşık gelebilmektedir (Fesakis ve Serafeim, 2009; Ozoran, Çağıltay ve Topallı, 2012).

Programlama eğitiminde öğrencilerin yaşadığı daha farklı zorluklar da vardır. Cevahir ve Özdemir (2017) programlama eğitiminde karşılaştıkları zorlukları; ezbere dayalı eğitim, tekrar ve uygulama eksikliği, soyut düşünme yeteneğindeki eksiklik, aritmetik, matematiksel ve analitik düşünme yetersizliği, dizi ve döngü konularında öğrenmede yaşanan zorluk olarak ifade etmişlerdir. Özmen ve Altun (2014) ise yaptıkları bir çalışmada programlama eğitimde yaşanan zorlukları; pratik eksikliği, algoritma oluşturmama ve bilgi eksikliği olarak ortaya koymuşlardır. Ayrıca programlama öğretiminde öğrencilerin yaşadığı zorlukları; programlama bilgisi, programlama becerisi, programlama mantığını kavrama ve hata ayıklama olarak ifade etmişlerdir. Programlama eğitiminde yaşanan bu zorluklar, öğrenciler üzerinde kaygı ve stresi artırıp, derse katılma isteklerini azaltıp, programlama başarısı ve öz-yeterlik algıları üzerinde olumsuz bir etki de oluşturabilmektedir.

Programlamanın öğrenciler tarafından zor ve karmaşık olarak görülmesi önemli şirketleri, bu zorlukları gidermek ve öğrencilerin kolay bir şekilde programlamayı öğrenebilmeleri için görsel programlama araçları geliştirmeye yöneltmiştir. Programlama eğitiminde karşılaşılan zorlukları gidermek, öğrencilerin yaşayabileceği kaygı ve stresi azaltmak, öğrencilerin derse katılımını artırabilmek için, önemli şirketlerin geliştirdiği bir görsel programlama dili kullanılabilir (Demir, 2015; Fesakis ve Serafeim, 2009; Fowler, 2012; Genç ve Karagüç, 2011; Ozoran, Çağıltay ve Topallı, 2012). Scratch ve Kodu Game Lab gibi görsel programlama dilleri, programlama öğretimini soyut yapıdan çıkararak somutlaştırabilmektedir (Bassil ve Keller, 2001; Ersoy, Madran ve Gülbahar,

2011). Görsel programlama dillerinin yaygınlaşmasından sonra hangi programlama dilinin seçileceği bir sorun olarak karşımıza çıkmaktadır. Bu tez çalışması ile öğretmenlere, programlama dili öğretiminde Scratch ve/veya Kodu Game Lab programlama dilleri ile ilgili başarı, tutum ve öz-yeterlik gelişimi ile ilgili fikir vermesibeklenmektedir.

Programlamanın zor ve karmaşık olarak görülmesi öğrencilerin programlamaya karşı olan tutumlarını da olumsuz bir şekilde etkileyebilmektedir. Ayrıca bu durum, öğrencilerde başaramama hissi deyaratabilir. Bunun için Scratch gibi programlama öğrenmeyi kolaylaştırıcı araçların seçilmesi (Lewis, 2010; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Nam, Kim ve Lee, 2010) ve öğrenme ortamının eğlenceli hale getirilmesi (Gallardo ve Ruíz, 2013; Lopez, Gonzalez ve Cano, 2016), öğrencilerin tutumlarını (Gülbahar ve Kalelioğlu, 2014; Oh, Kim ve Lee, 2013; Papadakis, Kalogiannakis, Zaranis ve Orfanakis, 2016; Wang, Huang ve Hwang, 2014; Yüksel, 2017), ve öz-yeterlik algılarını olumlu yönde etkilemektedir (Yükseltürk ve Altıok, 2016).

Günümüzde en çok kazanan şirketleri incelendiğine bu şirketlerin başında Microsoft, Google, Facebook gibi firmaların olduğunu görülmektedir (“Dünyanın en değerli markası”, 2017). Bu firmaların yazılım üreterek bu noktalara gelmeleri, ülkelerinin kalkınma düzeylerine verdikleri destek, programlamanın nedenli önemli olduğu konusunda bize ipucu vermektedir (Sayın ve Seferoğlu, 2016). Ayrıca programlama öğrenmenin 21. yüzyıl bireyleri için bir zorunluluk haline geleceği de beklenmektedir (Sayın ve Seferoğlu, 2016). Bu durumda programlama öğrenmeye ne zaman başlanması gerektiği, programlama eğitiminin nasıl verilmesi gerektiği, hangi ön bilgiler gerektirdiği ve hangi programlama dili seçilmesi gerektiği konusunda bir sorun ortaya çıkmaktadır.

Dünyada programlama öğretimine verilen değer gün geçtikçe arttığı yadsınamaz bir gerçektir. Özellikle gelişmiş ülkelerin küçük yaşlardan itibaren müfredatlarına programlama ve kodlama eğitimini eklediği görülmektedir. Bu ülkelerden İngiltere, kodlama eğitimini beş yaşından itibaren çocukların öğrenmesi için harekete geçmiş ve 2014 Eylül ayında 5-14 yaş grupları için öğretime başlamıştır. Çocukların algoritma kavramından başlayarak, temel programlama mantığı geliştirmek ve bir programı etkin kullanabilmesi amaçlanmıştır (Öndeş, 2016). Ayrıca Avustralya, Güney Kore, Tayvan, ABD gibi ülkeler de küçük yaşlardan itibaren programlama öğretimine başlamışlardır (Demirer ve Sak, 2016; Sayın ve Seferoğlu, 2016). Ülkemizde de programlama öğretimiyle ilgili çalışmalar yapılmaktadır. İlköğretim seviyesinde programlama öğretimi, 2012 yılında zorunlu olan Bilişim Teknolojileri ve Yazılım dersinde, *Problem çözme*, *Programlama* ve *Özgün Ürün geliştirme* üitesiyle verilmeyle başlamıştır. 5.,6.,7. ve 8. sınıflar için uygun görülen

programlama eğitimi "*Bir problemi çözmek ve projeyi gerçekleştirmek için strateji geliştirebilir, çözüm üretirken farklı bakış açılarını ve yaklaşımları kullanabilir*" ve "*Yazarlık ve programlama dillerini tanıyabilir, en az bir yazarlık/ programlama dilini etkili biçimde kullanabilir*" yeterlilikleri kazandırmayı amaçlamıştır (TTKB, 2012). Programlamaya verilen değer giderek artsa da, programlama öğretiminin nasıl yapılacağına ilişkin MEB'in sunduğu bir kaynak bulunmamaktadır. Öğretmenlerin programlama öğretimi yapmaları için bir kaynağa sahip olmaması ve her öğretmenin kendine göre programlama öğretmesi, programlama eğitiminin verimli olarak işlendiği üzerine tartışmalara yol açmaktadır.

Büyük şirketlerin, A-12 (Anaokulundan 12. Sınıfa kadar) seviyesi için programlama pazarında Mbot, Makey Makey, Lego Mindstorms gibi robot kitleriyle yer almaya çalıştığı, bu durumda ticari çıkarlara dönüştüğü ve oluşturulan robotlar için hazır kodlar yayınladıkları görülmektedir. Yayımlanan hazır kodlar öğrencilerin programlama öğrenme gereksinimi azalmaktadır. Bu durum programlama öğretmek yerine ezberciliğe teşvik ettiğini söyleyebiliriz. Programlama öğreniminde asıl amaç, programlama mantığı geliştirmeyi sağlamak ve ezberci yapıdan uzaklaşmak olmalıdır (Altun ve Özmen, 2014; Cevahir ve Özdemir, 2017).

21.yy bireylerinin kazanması gereken becerilerden biri bilgi işlemsel düşünme becerisidir. Çünkü bilgi işlemsel düşünme (computational thinking) kavramı, düşünme ve problem çözme becerileri gibi bilişsel becerileri içinde barındırmaktadır (ISTE, 2018; Barr, Harrison ve Conery, 2011). Bilgi işlemsel düşünme, bilgisayar bilimlerinin temel kavramlarını kullanarak problemleri çözmeyi, sistemleri tasarlamayı ve insan davranışlarını anlamayı içeren bir kavramdır. Bilgi işlemsel düşünme becerisi sadece bilgisayar bilimi ile ilgilenenlerin değil, bütün insanların kazanmaları gereken bir beceridir (Wing, 2006). Bilgi işlemsel düşünme ayrıca programlama öğretimi için de önemli bir beceridir. Bilgi işlemsel düşünme becerileri ile programlama başarısının ilişkili olduğu bazı çalışmalarla ortaya çıkmıştır (Oluk ve Korkmaz, 2016; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017). Öğrencilerin program yazabilmeleri için bilgi işlemsel düşünme becerilerinden yararlanmaları gerekebilir. Çünkü bilgi işlemsel düşünme becerisi olan bireyler, bir problemi çözmek için problem ne kadar zor olursa olsun parçalara ayırır, önemli ayrıntıları çıkartır ve algoritmik düşünme kullanarak her bir adımı inceler ve çözüm yolu geliştirir. Böylece problem kolaylaştırılarak çözülebilir (ISTE, 2018). Bu durumda programa öğretiminde, öğrencilere algoritma ve akış şeması gibi algoritmik düşünme gerektiren konuların işlenmesinin programlama başarısı için bir gereklilik haline

geldiği savunulabilir. Ayrıca öğrencilere bilgi işlemsel düşünmeyi destekleyici etkinlikler planlanarak ders planlarına entegre edilebilir.

Sonuçolarak,programlama öğretiminin önemi gün geçtikçe artmakta ve bu alanda uygulamalı çalışmalara olan ihtiyaç daha da artmaktadır. Bu nedenle günümüzde önemi artan Scratch ve Kodu Game Lab gibi görsel programlama dilleriyle programlama öğretiminin öğrenciler üzerinde etkisinin ortaya konulması gerekmektedir. Yapılan bu araştırmada Scratch ve Kodu Game Lab ile programlama öğretiminin 6. sınıf öğrencilerinin programlama başarısı, programlamaya yönelik tutum ve programlamaya karşı öz-yeterlik algısına olan etkisini ortaya çıkarmak amaçlanmıştır.

1.2. Araştırmanın Amacı ve Önemi

Gelişen ve değişen teknolojik gelişmeler göz önüne alındığında öğrencilerin bu teknolojik gelişmeleri yakalamaları ve çağa ayak uydurabilmeleri için bazı standartlar yakalamaları gerekmektedir (ISTE, 2018). Bu standartlara göre öğrencilerden; öğrenme için hedef belirlemeleri ve bu hedeflere ulaşmak için teknolojiyi kullanarak daha güçlü öğrenmeler sağlamaları, dijital dünyaya ayak uydurmaları ve bir dijital vatandaş olmaları, kendine özgü araştırma stratejileriyle bilgiye ulaşmaları ve oluşturmaları, yenilikçi ve yaratıcı çözümler oluşturarak problemi tanımlamaları ve çözmek için çeşitli teknolojiler kullanmaları, problem çözme sürecinde teknolojiyi kullanmaları ve bilgi işlemsel düşünebilmeleri, öğrencilerin kendini ifade etmeleri için uygun ortamlar seçmeleri ve yaratıcı bir iletişimci olmaları, küresel işbirliğinde bulunmaları beklenmektedir. Bu becerilerin bazılarının kazandırılmasında programlama öğretiminin de katkısı olduğu görülmektedir. Programlama öğretiminin problem çözme becerilerini olumlu yönde etkilediğini gösteren çalışmalar olduğu (Hava, 2016; Nam, Kim ve Lee, 2010; Shin ve Park, 2014; Taylor, Harlow ve Forret, 2010; Yıldırım, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017) görülmektedir. Ayrıca Scratch ile programlama öğretiminin yaratıcılık yönünden anlamlı derecede etki ettiği (Kobsiripat, 2014; Oh, Lee ve Kim, 2013; Ozoran, Çağıltay ve Topallı, 2012; Taylor, Harlow ve Forret, 2010; Yıldırım, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017), işbirlikli öğrenme ortamı sağladığı (Papatğa, 2016; Taylor, Harlow ve Forret, 2010), öğrencilerin birbirleriyle etkileşime girmelerine yardımcı olduğu ve aralarındaki iletişimi olumlu etkilediği (Lopez, Gonzalez ve Cano, 2016; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Zavala, Gallardo ve Ruíz, 2013) yapılan çalışmalarda görülmüştür.

Programlama dilleri öğretiminin, son yıllarda oldukça adından söz ettiren, dünyada çeşitli ülkelerin öğretim programlarına giren ve zorunlu olarak okutulan bir ders olduğu görülmektedir (Demirer ve Sak, 2016; EuropeanSchoolnet, 2015; Sayın ve Seferoğlu, 2016). Bunun sebebide programlama dili öğretiminin; problem çözme, yaratıcılık, algoritmik ve bilgi işlemsel düşüncebecerilerini öğrencilere kazandırabilmesi olarak görülmektedir (Shin, Park ve Bae, 2013). Programlamayı öğrenmek problem çözme, yaratıcılık, algoritmik düşünme gibi özellikleri kazandırmasını sağlasa da kavramların soyut kalması bu özelliklerin kazanılmasını zorlaştırmaktadır. Programlamanın soyut yapısından kaynaklanan sorunların giderilmesi için, görsel programlama dillerinin kullanılabilir. Görsel programlama dillerinin kullanılması programlamayı somutlaştırabilmektedir (Ersoy, Madran ve Gülbahar, 2011). Programlamanın kavranması açısından, oyun temelli programlama dilleriyle programlama öğretimi oldukça bir hal almaktadır. Öğrenenler oyun temelli programlama dilleriyle programlama öğrenirken hem sürece aktif olarak katılabilmekte hem de daha kalıcı öğrenmeler gerçekleştirebilmektedirler. Ayrıca bu tür programlarla öğrenciler senaryo oluşturabilmekte, bu sayede de anlamlı ilişkiler kurarak bilgileri örgütleyebilmektedir (Eremekçi ve Fidan, 2012).

Görsel programlama dilleriyle küçük çocukların, karışık kod yapıları öğrenmelerine gerek kalmadan uygulamalar, oyunlar yapmaları ve tasarımları mümkün hale gelmektedir (Resnick ve diğerleri.,2009). Özellikle çocukların seviyelerine uygun olarak hazırlanmış olan Scratch ve Kodu Game Lab programlama dili tercih edilebilir. Scratch, Kodu Game Lab gibi görsel temelli programlama dillerinin kullanımının kolay, sade ve eğlenceli olması küçük çocukların da problem çözme, yaratıcılık, algoritmik ve bilgi işlemsel düşüncebecerileri gibi bazı özellikleri erken yaşlarda kazanmasına yardımcı olabilmektedir (Çatlak, Tekdal ve Baz,2015; Shin, Park ve Bae, 2013). Scratch programlama dili MIT tarafından geliştirilmiş olup görsel blokların sürüklenip bırakılması mantığına dayalı bir görsel programlama dilidir (Scratch, 2017). Kodu Game Lab programlama dili, basit görsel programlama dili özellikleri sayesinde bilgisayar ve Xbox platformlarında çocukların ve yetişkinlerin 3 boyutlu oyun geliştirmelerine yardımcı olan oyun temelli bir yazılımdır. Kodu Game Lab programlama dili sayesinde çocuklar, problem çözme, öykü oluşturma, yaratıcılık, neden sonuç ilişkisi kurma gibi önemli becerileri kazanmaları öngörülmektedir (Kodu Game, 2018). Görsel programlama araçları ile programlama eğitiminin eğlenceli bir ortamda geçtiği de (Fidan, 2016; Gallardo ve Ruíz, 2013; Lopez, Gonzalez ve Cano, 2016; Yıldırım, 2016; Ozoran, Çağıltay ve Topallı, 2012) görülmektedir. Görsel programlama aracı kullanılmasının programlama öğrenimini kolaylaştırdığı (Genç ve Karaguş, 2011; Lewis, 2010; Malan ve Leitner, 2007; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Nam, Kim ve Lee,

2010; Yıldırım, 2016), öğrencilerin programlama öğrenme isteklerini olumlu etkilediğini (Fesakis ve Serafeim, 2009; Fidan, 2016; Gülbahar ve Kalelioğlu, 2014; Kobsiripat, 2014; Malan ve Leitner, 2007; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016), olumlu tutum geliştirdiğini (Baltalı, 2016; Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011; Malan ve Leitner, 2007; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016) ve programlama için gerekli motivasyonu attırdığı (Erol, 2015; Fidan, 2016; Fowler, 2012; Genç ve Karaguş, 2011; Gülbahar ve Kalelioğlu, 2014; Lopez, Gonzalez ve Cano, 2016; Ruf, Mühling ve Hubwieser, 2014; Osman, Zakaria, Loke ve Downe, 2012) yönünde çalışmalar da mevcuttur. Ayrıca Gülbahar ve Kalelioğlu (2014) Scratch programlama dili kullanarak 5. Sınıf öğrencileri üzerinde yaptığı araştırmada öğrencilerin problem çözme becerilerine olan etkisinin anlamlı derecede katkı yapmamasına rağmen programlamayı sevdirdiğini ve öğrencilerin devam etmek istedikleri sonucunu ortaya çıkarmıştır (Yükseltürk ve Altıok, 2016). Bütün bu bulgulardan yola çıkarak görsel programlama aracının başarıyı da arttırdığı söylenebilir (Demir, 2015; Erol, 2015; Fesakis ve Serafeim, 2009; Osman, Zakaria, Loke ve Downe, 2012; Papadakis, Kalogiannakis, Zaranis ve Orfanakis, 2016; Ruf, Mühling ve Hubwieser, 2014; Yıldırım, 2017; Yüksel, 2017). Ayrıca başarı hissi yaşayan öğrencilerin öz-güvenlerinin de olumlu yönde etkileneceği (Yükseltürk ve Altıok, 2016) söylenebilir. Bu yüzden görsel programlama araçlarının özellikle programlamaya yeni başlayanlar için kullanılması uygun olduğu sonucuna ulaşılabilir.

Sonuç olarak, programlama öğretiminin bireylere önemli beceriler kazandırması ve bundan programlamaya verilen önemin gün geçtikçe artmaktadır. Ancak programlama öğretiminin nasıl işleneceği ilgili MEB'in öğretmenlere sunduğu bir kaynak bulunmamaktadır. Scratch ve Kodu Game Lab ile programlama öğretimine ilişkin bir çalışma yapılmadığı da görülmektedir. Kodu Game Lab ile programlama başarısına ilişkin bir çalışmaya rastlanmamıştır. Ayrıca Kodu Game Lab ile ilgili çalışmaların sayısının oldukça az olduğu ve ortaokul seviyesinde hem Kodu Game Lab hem de Scratch ile öz-yeterlik algıları üzerine bir çalışma yapılmadığı da ortaya çıkmıştır. Bu açıdan yapılan bu çalışmanın literatüre katkı sağlayacağı düşünülmektedir.

Yapılan araştırmanın amacı, ortaokulda öğrenim gören 6. sınıf öğrencilerine yönelik BTY dersinde programlama öğretimini, Scratch ve Kodu Game Lab programlama dilleri kullanarak gerçekleştirmek ve programlama dilleri öğretiminde bu iki yazılımın, öğrencilerin programlamaya yönelik tutum, akademik başarı ve öz-yeterlik algılarına etkisini araştırmaktır. Ayrıca programlama öğretiminin küçük yaşlardan itibaren BTY öğretim programına eklenmesinden sonra, programlama öğretiminin planlanması ve programlama

öğretimi için uygun programlama aracı seçimi hususunda öğretmenlere yön vermesi umulmaktadır.

1.3. Problem Cümlesi

Scratch ve Kodu Game Lab ile programlama öğretiminin 6. sınıf öğrencilerin akademik başarı, tutum ve özyeterlilik düzeylerine etkisi nedir?

1.3.1. Alt Problemler

- Scratch ve Kodu Game Lab ile programlama öğretiminde öğrencilerin programlama başarıları anlamlı derecede farklılaşmakta mıdır?
- Scratch ve Kodu Game Lab ile programlama öğretiminde öğrencilerin programlama başarıları kalıcılık testi puanları anlamlı derecede farklılaşmakta mıdır?
- Scratch ve Kodu Game Lab ile programlama öğretiminde öğrencilerin programlamaya yönelik tutumları anlamlı derecede farklılaşmakta mıdır?
- Scratch ve Kodu Game Lab ile programlama öğretiminde öğrencilerin programlamaya yönelik öz-yeterlilikleri anlamlı derecede farklılaşmakta mıdır?

1.4. Sayıtlar

Bu araştırmada aşağıda belirtilen varsayımlarla yürütülecektir.

1. Deney gruplarının akademik başarılarının benzer olduğu varsayılacaktır.
2. Öğrencilerin programlama ön-deneyimlerinin eşit olduğu varsayılacaktır.

1.5. Sınırlılıklar

Bu araştırma;

1. Çanakkale Biga Yeniceköy Ortaokulunda 2016-2017 eğitim-öğretim yılında öğrenim gören 27 Öğrenci ile

2. Programlama öğretiminde her iki öğrencinin bir bilgisayara sahip olmadurumu ile sınırlıdır.

1.6. Kısaltmalar

BTY: Bilişim Teknolojileri ve Yazılım

BÖTE: Bilgisayar ve Öğretim Teknolojileri Eğitimi

MEB: Milli Eğitim Bakanlığı

TTKB: Talim Terbiye Kurulu Başkanlığı

PDABT: Programlama Dilleri Akademik Başarı Testi

EBODKÖYTÖ: Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği

PİÖYÖ: Programlamaya İlişkin Öz-Yeterlik Ölçeği

DG1: Deney Grubu1

DG2: Deney Grubu2

A-12: Anaokulundan 12. Sınıfa kadar

1.7. Tanımlamalar

Programlama: Bir bilgisayarın belirli bir görevi yerine getirmesini, sorunları çözmesini ve insan etkileşimini sağlamasını mümkün kılmak için çeşitli komutlar geliştirme ve uygulama sürecidir.

Kodlama: Belirli bir mantığa göre görevi yerine getirmek için talimatların yazılması, algoritmanın belirlenen programlama dilinde kullanılmasını sağlayan programlamanın özel bir alt görev görevi olarak ifade edilebilir.

Programlama Dilleri: Bir problemin çözümünde bilgisayarda yazılacak çözümün adım adım ve belli bir kural dizisine uygun olarak algılayan tanım kümesine verilen isimdir.

Görsel Programlama Dilleri: Resim, grafik, blok gibi görsel öğeler kullanılarak yapılan programlama işlemine verilen isimdir.

BÖLÜM II

2. İLGİLİ YAYIN VE ARAŞTIRMALAR

2.1. Bilgi İşlemsel Düşünce (Computational Thinking)

Günümüzde değişen ve gelişen dünyada öğrencilerin, taşıması gereken en temel beceriler; yenilikçi, yaratıcı, algoritmik düşünebilen, araştırmacı, işbirliği yapabilen, problem çözebilen, eleştirel düşünebilen olmalarıdır. Ayrıca son zamanda popüler olan bilgi işlemsel düşünme (Computational Thinking) kavramının içerisinde barındırdığı bu temel özellikler, 21. yy bireylerinde olması gereken temel özelliklerle teknolojinin harmanlanması sonucu ortaya bir problem çözme yaklaşımı olarak ifade edilmektedir (ISTE, 2018). Başka bir ifadeyle bilgisayar bilimi kullanarak problem çözme yaklaşımı olarak adlandırılabilir (Küçüköğlu, 2014). Bilgi işlemsel düşünme kavramı sadece bilgisayarlılar için gerekli olmayıp tüm bireyler için oldukça önemli bir yere sahip olması gerektiği söylenebilir. Wing (2006)'e göre bilgi işlemsel düşünme becerileri 21. yy bireylerinin kullanacakları temel yetenekler arasına gireceği öngörülmektedir. Öngörülen bu beceriler; Sorunları çözmede bilgisayar gibi araçları kullanarak formülize etme, Verileri mantık çerçevesinde ele alma ve bu şekilde çözümleme, Soyut kavramları sunarken model veya simülasyonları kullanma, Adım adım düşünerek çözüme otomatik ulaşma, Problemi çözerken en uygun çözümü seçme ve uygulama, Problemin çözümü sürecini başka problemleri çözerken kullanma ve transfer etme olarak belirlenmiştir (Barr, Harrison ve Conery, 2011). Bu kavramların özünde algoritmik düşünmeyle büyük problemler, küçük parçalar haline ayrıştırılıp, sonuca ulaşmada en doğru yolun seçilip çözüme ulaşma yer almaktadır. Bilgi işlemsel düşünebilen bireylerin günlük hayatta karşılaştığı problemlerin çözüm safhasında da aynı yolu izleyerek problemlerini kolaylıkla çözebileceklerdir (Küçüköğlu, 2014). Ayrıca Barr, Harrison ve Conery (2011)'ye göre bu beceriler desteklenir ve geliştirilirse, bilgi işlemsel düşünme becerisinin aşağıdaki davranışların temelini oluşturduğu söylenebilmektedir:

1. Zor ve karmaşıklığa karşı duyulan güven
2. Zor problemlere karşı sabırla ve inatla çalışma
3. Belirsizliği karşı toleranslı olma
4. Açık uçlu bir sorunla başa çıkma yeteneği
5. İletişim kurma becerisi sergileyerek başkalarıyla birlikte çalışarak ortak bir çözüme ulaşmadır.

Bilgi işlemsel düşünme kavramına verilen önem gitgide artmaktadır. Literatürde yapılan araştırmaların sayısının artması, bilgi işlemsel düşünme kavramına verilen önemin bir göstergesi olarak kabul edilebilir. Korkmaz, Çakır, Özden, Oluk ve Sarioğlu (2015) tarafından yapılan çalışma, 1306 üniversite öğrencisinin katıldığı betimsel tarama modelindedir. Yaratıcılık (creativity), eleştirel düşünme (critical thinking), algoritmik düşünme (algorithmic thinking), İşbirliklilik (cooperativity) ve problem çözme (problem thinking) bilgi işlemsel düşünmenin alt boyutları üzerinden Korkmaz, Çakır ve Özden tarafından geliştirilen Bilgisayarca Düşünme Becerileri Ölçeği kullanarak, okul türü, bölüm, sınıf düzeyi, mezuniyet durumu, cinsiyet ve yaş değişkenine göre incelenmiştir. Araştırmanın sonucunda bireylerin bilgi işlemsel düşünme düzeyine ilişkin algıların orta ve yüksek düzeyde olduğu görülmektedir. Teknoloji fakültesi ve enstitüde öğrenim gören öğrencilerin bilgi işlemsel düşünme becerileri anlamlı düzeyde arttığı ve matematik, fen ve teknoloji alanlarının diğer bölümlere oranla anlamlı derecede farklılaştığı görülmektedir.

Programlama öğretiminin bilgi işlemsel düşünme becerileri üzerinde etkisinin olduğu yapılan bazı çalışmalarda kabul edilmektedir. Bu çalışmalardan Sırakaya (2017)'nin yaptığı çalışma, 54 üniversite öğrencisinin katıldığı tek grup öntest-sontest desen türündedir. Öğrencilere veri toplama aracı olarak kişisel bilgi formu ve bilgisayarca düşünme ölçeği uygulanmıştır. Çalışmaya göre bilgi işlemsel düşünme öntest ve sontest puanı açısından anlamlı fark olduğu sonucuna varılmıştır. Bilgi işlemsel düşünme becerilerinin alt faktörlerinden algoritmik düşünme ve problem çözmede anlamlı farklılık olduğu ancak eleştirel düşünme, işbirliklilik ve yaratıcılık faktörleri açısından anlamlı farklılık olmadığı görülmektedir. Sırakaya (2017)'ye göre bu faktörlerden ve sonuçlardan yola çıkılarak programlama dili öğretiminin bilgi işlemsel düşünme üzerinde olumlu etkisi olabileceği öngörülmektedir.

2.2. Programlama Nedir?

21. yüzyıl öğrenci becerileri incelendiğinde öğrenme ve yenilik becerileri, bilgi medya ve teknoloji becerileri, yaşam ve kariyer becerileri şeklinde ana başlık altında toplanabilir. Öğrenme ve yenilik becerileri incelendiğinde; yaratıcılık, eleştirel düşünme, problem çözme, iletişim, işbirliği olarak alt başlıklara ayrılabilir (Günüç, Odabaşı ve Kuzu, 2013). Öğrencilerin bu becerileri kazanmasındaki yöntemlerden biri de programlama öğretimidir. Programlama kavramı, ortaya çıkan bir problemin çözümü için oluşturulmuş kod parçalarının, bir programlama dili kullanılarak anlam kazanıp, çözüme ulaşmasına

verilen isimdir (Arabacıoğlu, Bülbül ve Filiz, 2007). Bilgisayarda yapılan iş ve işlemleri bilgisayarın anlaması, kişiye uygun özel işlerini kolaylaştırması için programlaması gerekmektedir. Günlük hayatta ve iş hayatında programlama ürünü olan programlar kullanılmaktadır (Tekin ve diğerleri., 2003).

Programlama tarihi incelendiğinde 1940-1950 yıllara kadar eskiye gidildiği görülmektedir. İlk olarak bilgisayarın insan ile iletişimini sağlayan makine dili olmuştur. O yıllarda makine diliyle programlar "0" ve "1" durumlarına göre hazırlanmaktaydı. "0" yok veya yanlış, "1" ise var veya doğru durumunu ifade etmekteydi. Programlamanın zor ve anlaşılması güç bir durumda olmasından dolayı makine diline yakın olan ve işlemcilerin algılayabileceği Assembly programlama dili kullanılmaya başlanmıştır. 1980'li yıllardan itibaren de üst seviyeli programlama dilleri hayatımıza girmeye ve yaygınlaşmaya başlamıştır (Tekin ve diğerleri., 2003).

Programlama için kişiler birden fazla bilgi ve beceriye ihtiyaç duyarlar. Bayman ve Mayer (1988) bu bilgi ve becerileri Sözdizimsel (Syntactic) Bilgi, Kavramsal (Conceptual) Bilgi ve Stratejik (Strategic) Bilgi olarak üç grupta ele almışlardır. Bu bilgi ve beceriler şu şekilde açıklanabilir:

1.Sözdizimsel Bilgi (Syntactic Knowlegde): Bir programlama dili içerisinde kodların hangi kurallara göre nasıl yazılacağına bilgisi. Dil bilgisi kuralları olarak tanımlayabiliriz. Örnek verecek olursak C# programlama dilinde her satırın sonuna noktalı virgül (;) konarken, Visual Basic programlama dilinde satırların sonuna herhangi bir noktalama işareti konulmaz. İki programlama diliyle de aynı probleme çözüm bulunurken, kod satırlarının yazılışlarının farklı olması durumudur. Sözdizimsel bilgi içerisinde probleme çözüm aranmaz sadece dilin yazılış bilgisi.

2.Kavramsal Bilgi (Conceptual Knowlegde): Programlama dilinde kullanılan kavramlara ve bu kavramların ne işe yaradıklarına ait bilgidir. Programlamada var olan kodların ve yapıların ne anlama geldiği kavramsal bilgi olarak açıklanabilir. Örnek olarak koşul ifadesi dediğimizde Visual Basic'te **İF (eğer)** kavramını kullanmamız gerekir. İf kavramını gördüğümüzde ne anlama geldiğini anlamamızın bilgisi olarak ta ifade edebiliriz. Ayrıca bu kavramlar diğer programlama dillerinde de aynı yapıda olabilirler. Örnek olarak C# programlama dilinde de koşul ifadesi **İF (eğer)** olarak yazılır. Bu durumdan kavramsal bilgilerin diğer programlama dillerine de transfer edilebileceği sonucunu çıkarabiliriz.

3.Stratejik Bilgi (Strategic Knowlegde): Programlama sürecinde sözdizimsel ve kavramsal bilgileri kullanarak problemin çözüme kavuşturulması için gerekli beceridir. Stratejik bilgide

hem programlama dilinin gramer yapısını bilmeye yani sözdizimsel bilgiye hem de kavramların ne işe yaradığını bilerek, kavramsal bilgiye ihtiyaç vardır. Bu aşamada program yazılarak sonuca ulaştırılır. Örnek olarak klavyeden girilen iki sayıdan hanginin büyük olduğunu bulduran bir program yazılması istendiğinde kullanılacağı kodları, kod satırlarının nasıl yazılacağını ve bunların nasıl bir bütün olarak yazılacağını bilmesi gerekir. Ancak bu şekilde yazılan program çözüme ulaşabilir.

McGill ve Volet (1997) programlama bilgisindeki söz dizimsel ve kavramsal bilgiyi ayrıştırarak dört grupta ele almışlardır. Bunları; Bildirimsel-Sözdizimsel, Bildirimsel-Kavramsal, İşlemsel-Sözdizimsel ve İşlemsel-Kavramsal olarak ayrıştırmıştır. Ayrıca bilginin nerede, ne zaman, nasıl ve niçin kullanılması gerektiğinin bilgisini de stratejik bilgi olarak ele almış ve Tablo1'deki gibi görselleştirilmiştir.

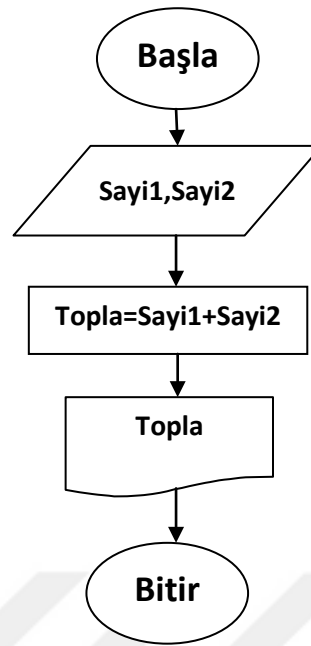
Tablo 1
Programlama Bilgisi

	Bildirge Bilgisi (Declarative Knowledge)	İşlemsel Bilgi (Procedural Knowledge)
Sözdizimsel Bilgi (Syntactic Knowledge)	Programlama dilindeki yazım kuralları bilgisidir. -Örnek olarak C# programlama dilinde satırı sonlandırmak için noktalı virgül konulduğunu, Visual Basic programlama dilinde satırı sonlandırmak için alt satıra geçmenin yeterli olduğunun bilinmesidir.	Programlama sürecinde söz dizimi kurallarına uyabilme becerisidir. - Örnek olarak Visual Basic'te If komutunun nasıl yazıldığının ve kullanıldığının bilinmesidir. - Bir metin dosyası açabilme ve Visual Basic tarafından görüntüleyebilmesidir.
Kavramsal Bilgi (Conceptual Knowledge)	Program çalışması için gerekli söz dizimi ve kavramların açıklama bilgisidir. - Örnek olarak mantıksal ve matematiksel sınama için koşul ifadesi olarak IF (eğer) komutunun kullanılmasıdır. - Tekrar eden işlemler için de For döngüsünün kullanılması gerektiğinin bilinmesidir.	Bir problemin çözümü için çeşitli çözüm yolları tasarlama becerisidir. - Örnek olarak bazı verilerin ortalamasını hesaplayabilmek için bir yapı oluşturabilmek.
Stratejik Bilgi (Strategic Knowledge)	Alışla gelmemiş bir problemle karşılaşılınca problemin çözümü için kodlama, programı test etme ve tasarlama becerisidir.	

2.3. Programlama ile İlgili Temel Kavramlar

Programlama dillerinde kullanılan ifadeler farklılık gösterse de programlama da bazı temel kavramlar mevcuttur. Tekin ve diğerleri. (2003)' e göre bu kavramları şu şekilde açıklayabiliriz.

- a. **Programlama Dili:** Bir problemin çözümünde bilgisayarda yazılacak çözümün adım adım ve belli bir kural dizisine uygun olarak algılayan tanım kümesine verilen isimdir. Kişinin ne yapmak istediğini bilgisayara yorumlayan dildir. Scratch, Kodu game lab, C++, C# gibi.
- b. **Program:** İhtiyaca uygun olarak geliştirilen, sorunları çözmek için ve rutin işleri daha kolay yapmayı sağlayan bilgisayar yazılımlarıdır.
- c. **Görsel Programlama:** Resim, grafik, blok gibi görsel öğeler kullanılarak yapılan programlama işlemine verilen isimdir. Örnek, Scratch, C#,Kodu Game Lab olabilir.
- d. **Editör:** Hızlı kod yazmak kodları tanıyan, yazılan kodlardaki hataları denetleyen, tanımlamalar yapmak için kullanılan yapılara verilen genel isimdir. Örnek, Visual Studio, Android Studio olabilir.
- e. **Algoritma:** Bir problemin çözümünde gerekli olan çözümlerin adım adım düşünülerek mantıksal bir sıraya dizilmesidir. Programın başlamasından sonlanmasına kadar geçen kısımların adım adım yazılmasını ifade eder. İki sayının toplamını bulduran bir algoritma örneği;
 1. *Başla*
 2. *İlk sayıyı gir*
 3. *İkinci sayıyı gir*
 4. *Sayıları topla*
 5. *Ekrana yazdır son*
 6. *Bitir*
- f. **Akış şeması:** Algoritmanın şekillerle görselleştirilmiş ve şekillerin birbirleriyle ilişkilendirilerek oluşturulan yapıya verilen isimdir. Akış şeması problemin çözümü için gerekli bir unsurdur. İki sayının toplamını bulduran bir akış şeması örneği Şekil1'de verilmiştir.



Şekil 1. Akış şeması

- g. Değişken:** Bir programda duruma göre farklı değer alabilen yapılara verilen isimlerdir. Örneğin "Topla" bir değişkendir ve sayılara göre farklı değer alır.
- h. Veri Tipi:** Değişkenlerin nasıl değerler içereceklerini belirlemek için kullanılan yapılardır. Bir değişkenin sayı, karakter, ondalıklı sayı veya tam sayı olduğu tanımlamak için kullanılır.
- i. Sabit:** Program içerisinde tek bir değer alabilen yapılara verilen genel isimdir. Örnek pi sayısı 3,14 olarak tanımlandığından başka değer almayan değerlerdir.
- j. Operatörler:** Programlama dillerinde matematiksel ve mantıksal işlemler yapmak için kullanılan karakterlere verilen isimlerdir. Örnek olarak "=", ">", "<", "+", "-", ",", "ve", "veya" gibi ifadelerdir.
- k. Koşul İfadeleri:** Programda yapılacak işlemlerin doğruluğunu, yanlışlığını veya bir işlemin yapılıp yapılmadığını kontrol eden ve mantıksal deyimlerdir.
- l. Fonksiyonlar:** Belirli sayıda verileri kullanarak bunları işleyip ortaya bir sonuç üreten komut grubuna verilen isimdir.
- m. Döngüler:** Belli bir kurala göre tekrar eden kurallar bütünüdür. Örnek 10 kez aynı ismi yazmak için kullanılabilir.
- n. Diziler:** Bir veya daha fazla boyutu olan veri kümelerine verilen isimdir. Benzer özellikteki verileri kaydetmek için kullanılır.

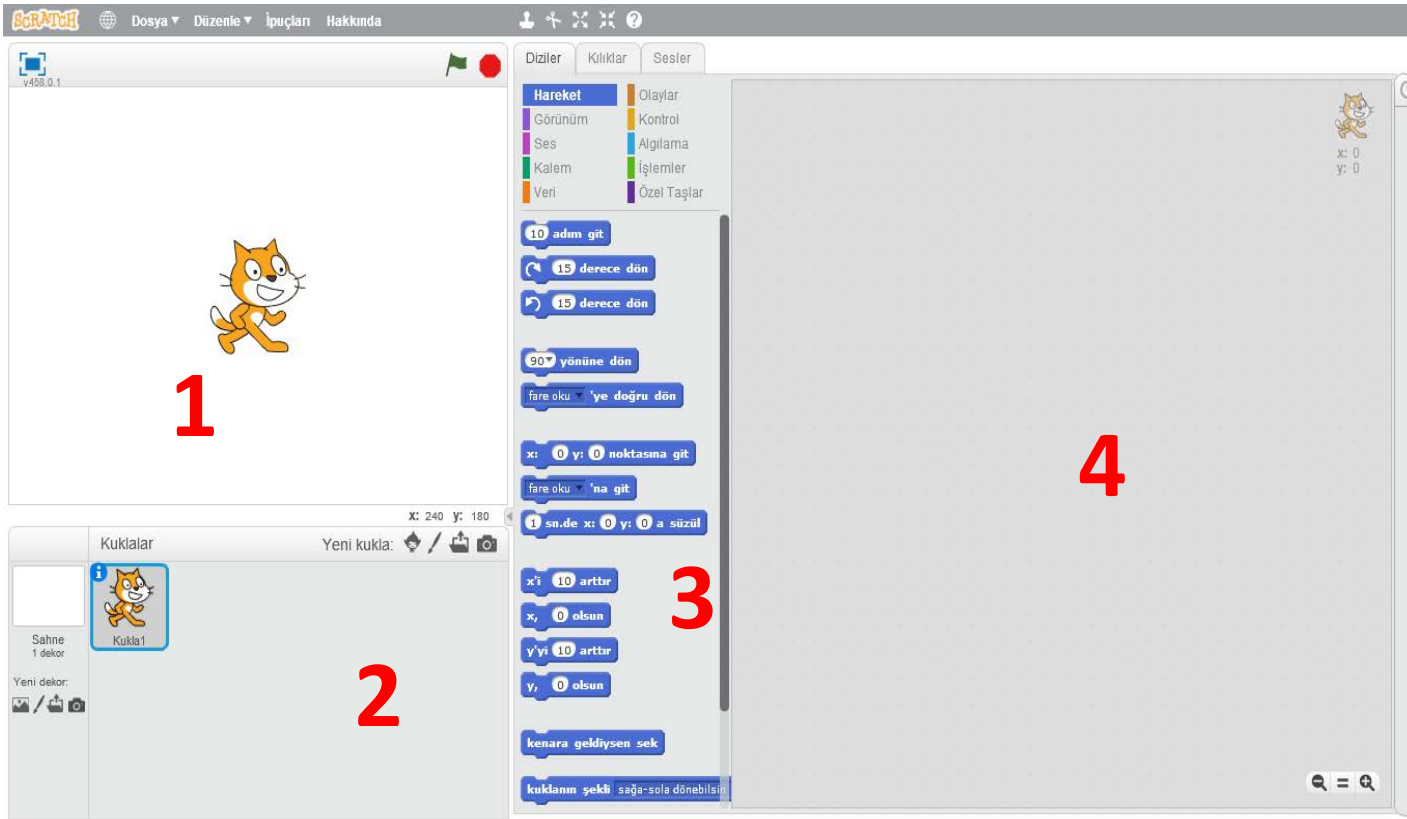
- o. Girdi Bilgisi:** Programa dışarıdan veri aktarımı işlemidir. Örnek, klavyeden girilen bir sayı olabilir.
- p. Çıktı Bilgisi:** Programda yapılan işlemlerin yorumlanarak sonuçlarının program dışına çıkarılma işlemidir. Örnek, iki sayının toplamının ekrana yazdırılması olabilir.
- q. Kod:** Program yazabilmek için kullanılan ve her programlama dilinde farklılık gösterebilen, programı oluşturan en küçük yapıdır.
- r. Atama İş:** Değişkene bir değer belirleme sürecidir. Örnek x değişkeninin değeri 100 olsun ($x=100$) denildiği zaman kullanılabilir.

Görsel Programlama Ortamları

Çalışmanın bu kısmında A-12 düzeyine uygun olarak tasarlanmış görsel programlama ortamları yer almaktadır. Öncelikle araştırmanın araçları olan Scratch ve Kodu Game Lab programlama dilleri, daha sonra da diğer programlama dilleri tanıtılmıştır.

a. Scratch

Scratch programlama dili masaüstü uygulama olarak 2007 yılında MIT (Massachusetts Institute of Technology) Medya Lab tarafından, çocukların hayallerindeki animasyonları, oyunları, etkileşimli hikâyeler oluşturabilmesi için oluşturulmuş görsel bir programlama dilidir. Yaklaşık 150 ülkede ve 40'tan fazla farklı dil seçeneğiyle, özellikle 8-16 yaş arası çocukların kullanımına sunulmuştur. Ayrıca tamamen ücretsiz bir yazılımdır. Scratch 2012 yılında web ortamında Scratch 2.0 sürümünü çıkararak hem çevrimiçi hem de çevrim dışı kullanılabilme özelliğine sahip olmuştur. Scratch'ın internet ortamına girmesi, dünya çapında daha fazla büyümesine ve daha fazla kişiye ulaşmasına olanak sağlamıştır. Özellikle kişilerin yaptığı çalışmalarını paylaşması ve bu çalışmaların geliştirilebilir olması öğrenciler için büyük bir fırsat olmuştur. Örneğin 2016 yılında 200 milyonun üzerinde çocuk kendi projelerini yapıp paylaşma şansı bulmuştur. Şuan Scratch programlama dili 196 ülkede 70'ten fazla dil seçeneğiyle dünyada çocuklar için en çok kullanılan programlama dillerinden biri olmaktadır (Scratch, 2017). Şekil 2'de Scratch programının ara yüzüne ait bir görsel verilmiştir.



Şekil 2. Scratch programına ait bir görsel

Scratch programı incelendiğinde dört temel kısımdan oluştuğu görülmektedir. Birinci kısım sahne kısmıdır. Bu kısımda, yapılan hikâye, oyun ve animasyon gibi içeriklerin son halini görebiliriz. İkinci kısımda karakterler ve dekorları oluşturabiliriz, hangi karakterle ilgili işlem yapılacağını bu kısımdan ayarlayabiliriz. Üçüncü kısım kod bloklarının bulunduğu kısımdır. Kod blokları yaptıkları işlere göre renklere ayrılarak menüler altında toplanmıştır. Dördüncü kısım kod yazım alanıdır. Üçüncü kısımdaki kod blokları buraya sürükleyip bırakarak kodlama işlemi yapılmaktadır.

Scratch programı, yapısı gereği kod yazmada kolaylık sağlamaktadır. Özellikle kod bloklarının sürükleyip bırak şeklinde olması kullanıcıları klavye ile kod yazmak zorunda bırakmamaktadır. Bundan dolayı olası sözel hatalardan arındırmaktadır. Ayrıca diğer programlama dillerinde olan sözdizimsel ifadeleri unutmak Scratch programında mümkün olmamaktadır. Çünkü bütün kodlar bloklar şeklinde programın içinde mevcut haldedir. Ayrıca kod bloklarının hangi bloklarla kullanılabileceği şekillerle anlaşılabilir yapıdadır. Uygun olmayan şekiller kod bloklarının içerisine girmediği için kullanılmadığı anlaşılmaktadır.

Scratch programlama dili ile programlama dili öğretiminin öğrencilere oldukça büyük katkıları vardır. Ozoran, Çağıltay ve Topallı (2012) mühendislik öğrencileriyle yaptıkları bir çalışmada Scratch programlama dilinin aşağıdaki katkıları sağladığı sonucuna ulaşmıştır;

- Scratch, programlama öğretimini daha eğlenceli hale getirmektedir,
- Scratch programlama olayını daha görsel hale getirmektedir,
- Scratch algoritma kavramını öğretmeye yardımcı olmaktadır,
- Scratch öğrencilerin yaratıcılıklarını artırmaktadır,
- Scratch sistemli düşünmeyi geliştirmektedir,
- Scratch, döngüler ve fonksiyonlar gibi programlamaya ait kavramları öğrenmede yardımcı olmaktadır,
- Scratch yorum yapmaya yardımcı olmaktadır,
- Scratch oyunlar ve animasyonlar konusunda yardımcı olmaktadır,

Scratch programlama dili ile çok sayıda ve kişilerin ihtiyacına uygun içerikler oluşturulabilmektedir. Hatta Scratch programlama dili ile oyun, animasyon ve oluşturulan hikâyeler derslerde de kullanılabilir. Ozoran, Çağıltay ve Topallı (2012) Scratch programlama dilini derslerde kullanarak aşağıdaki sonuçlara ulaşıldığını ifade etmişlerdir:

- Öğrencilerin notlarının arttığını ve dolayısıyla başarılarının arttığı gözlemlenmiştir
- Öğrencilerin başarısızlık oranı düşmüştür,
- Öğrencilerin derse olan katılımı artmıştır.

b. Kodu Game Lab

Kodu Game Lab, 2009 yılında Microsoft tarafından masaüstü cihazlar için üretilmiş bir üç boyutlu bir oyun yapma yazılımıdır. Toplam 24 dilde kullanıcılara sunulmuştur. Kodu Game Lab programlama dili özellikle çocukların kendi oyunları oluşturmalarını sağlayan görsel bir programlama dilidir. Kodu Game Lab programlama diliyle çocuklar yaratıcılıklarını kullanarak çeşitli senaryolar oluşturabilir veya çeşitli oyunlar üretebilirler. Kodu Game Lab, daha önce kodlama deneyimi olmayan çocukların bile hayal dünyalarına uygun ortamlar hazırlamalarına fırsat tanıyan tamamen ücretsiz bir yazılımdır (Kodu, 2017). Şekil 3'te Kodu Game Lab ara yüzüne ilişkin bir görsel verilmiştir.



Şekil 3. Kodu Game Lab programlama diline ait bir görsel

Kodu Game Lab ile programlama yapmak oldukça kolaydır. Nesne tabanlı bir yapısı vardır ve karmaşık kod yapılarından arındırılmıştır. Kod yapılarında görsellik oldukça fazladır. Bir kod yapısı seçildiğinde kullanıcıların yazacağı bir sonraki kod sadece ekleyebileceği kod yapılarıyla sınırlandırılmıştır. Bu sayede kullanıcılara kod seçerken veya yazarken hata yapma ihtimalini ortadan kaldırmaktadır. Kodlama mantığı olarak Eğer ise /Yap yapısındadır. Kodlar, eylemler, sıfatlar ve nesneler şeklinde barındırıldığından yapısı bir cümleye benzemektedir. Şekil 2 deki görsel de yazan kodlar incelendiğinde 2 numaralı kod satırında, *eğer bozuk paraya çarparsan ye* olarak bir cümle kurulabileceği görülmektedir. Ayrıca seçilen kodların üzerine fare ile gelindiğinde kodun ne işe yaradığı gösterilmektedir.

Kodu Game Lab programlama dili masaüstü bilgisayarlar ve Xbox 360 platformu için üretilmiş bir programlama dilidir. Uygulama şeklindedir ve kurulum gerekir. İnternet sitesinden kullanılabilir bir yapısı yoktur. Ayrıca şuan için mobil cihazlarda da kullanılamamaktadır. Buna rağmen kod yapısının kolay ve anlaşılır olmasından dolayı oldukça tercih edilmektedir. Ayrıca oyun temelli bir yapıya sahip olması da diğer bir tercih sebebi olmaktadır (Kodu,2017).

Fowler (2012) 10-13 yaş arası 60 ortaokul öğrencisiyle yaptığı bir araştırmada Kodu Game Lab programlama dilinin katkılarını şu şekilde ifade etmişlerdir:

- Kodu Game Lab ile öğrencilerin keyif sevilerinin artmıştır,
- Kodu Game Lab ile öğrencilerin motivasyonları artmıştır,
- Kodu Game Lab ile öğrencilerin can sıkıntısı azalmıştır,
- Kodu Game Lab ile öğrencilerin hayal kırıklığı seviyesi azalmıştır.

c. Hacker Can

Hacker Can ana dilde programlama dili öğretmek amacıyla oluşturulmuş yüzde yüz Türkçe bir görsel programlama dilidir. Şuan sadece Türkçe ve İngilizce olarak hizmet vermektedir. İnternet üzerinden www.hackercan.com internet sitesinden ücretsiz erişim sağlanmaktadır. Kullanıcılara mobil öğrenme ve uzaktan öğrenme fırsatı sunmaktadır. Oyun temelli olarak tasarlanmış olan Hacker Can programlama diliyle, “6-9 yaş” ve “9 yaş ve üzeri” yaş grupları kendilerine uygun bir şekilde algoritma ve programlamayı kolay bir şekilde öğrenmeye imkan sağlamaktadır. İçeriği yaş aralığına uygun olarak tasarlandığından Milli Eğitim Bakanlığı Talim Terbiye Kurulu tarafından 13 Ocak 2017 yılında yayınladığı BTY dersi müfredatında önerilen kaynak olarak gösterilmiştir (Hackercan, 2017).

Macera serisi şeklinde kurgulanmış olan Hacker Can zorluk seviyesine göre oluşturulmuş yaklaşık 200 bölümden oluşmaktadır. Bir bölümü geçmeden diğer bölümlere erişim imkanı sunmayan bir yol haritasına oluşturulmuştur. Öğrenciler kaldıkları yerden devam edebilmeleri için öğrenci panelleri oluşturulmuştur. Ayrıca öğrenciler tüm bölümleri bitirince Hacker Can Başarı Sertifikası almaya hak kazanmaktadır. Öğretmenlerin öğrencilerini kontrol edebilmeleri, hangi seviyede olduklarını, ne kadar sürede bölümleri geçebildiklerini, kaç satır kod yazdıklarını görebilmeleri için de eğitim paneli mevcuttur. Ancak bu özelliklere sahip olmak için belli bir miktar ücret ödeyip lisans almak gerekmektedir (Hackercan, 2017)..

Hacker Can görsel programlama ortamına ait bir görsel Şekil4’te verilmiştir.



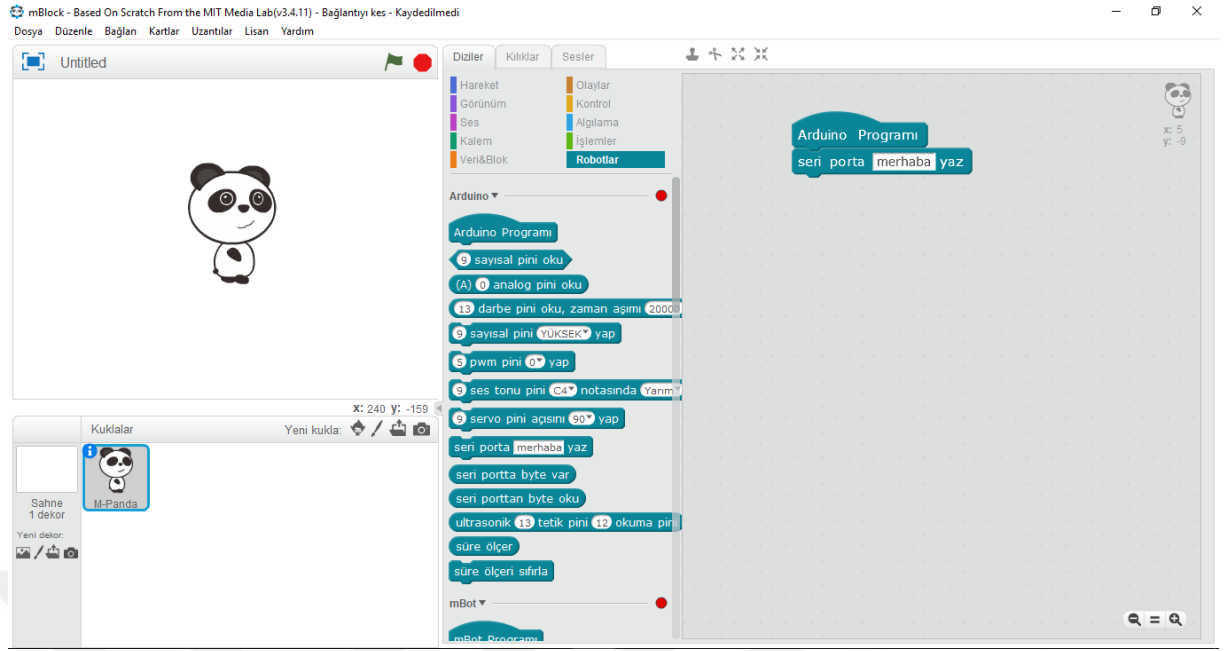
Şekil 4. Hacker Can programlama diline ait bir görsel

d. mBlock

Scratch tabanlı bir görsel programlama ortamı mBlock, daha çok mbot gibi robotları programlamak için kullanılmaktadır. Yapısal olarak blok tabanlı olup, Scratch'taki menülerden farklı olarak Robotlar menüsüne sahiptir. Onun dışında kod bloklarının renkleri, yerleşimi oldukça benzerdir. Bu menüde Arduino anakartlar ve onlara ait sensörler programlanmaktadır. Yapısal olarak mbot için üretildiğinden bu tarz robotlara ait kod blokları oldukça zengindir. Ancak Arduino kartlar için bu kod bloğu zenginliği söz konusu değildir. Sadece temel kodlar mevcuttur.

Çocuklar için üretilmiş ve tamamen ücretsiz olan mBlock 40'tan fazla dil desteği sunmaktadır. Kullanıcılar masaüstü ve mobil cihazlarına indirip kurabilecekleri uygulamaları www.mblock.cc internet sitesinden ulaşabilmektedir. Ayrıca bu internet sitesinden eğitimciler için kaynaklar ve materyallere ulaşabilmektedir. Mbot için kullanılacak örnek kod dizilerine de buradan ulaşmak mümkündür. Bu şekilde hem öğrenciler hem eğitimciler robotik kodlama için denenmiş kodları kullanarak yeni projelerin de geliştirebilmesi mümkün olacaktır (Mblock, 2018).

Block temelli görsel programlama aracı olan mBlock'a ait bir görsel Şekil5'te verilmiştir.



Şekil 5.mBlock programlama diline ait bir görsel

e. Code Combat

Code Combat dünyadaki her öğrenciye programlamayı öğretmeyi görev edinmiş ücretsiz bir programlama aracıdır. Code Combat www.codecombat.com internet sitesinden erişim sağlanabilen bir çevrimiçi yazılımdır. Mobil veya masaüstü cihazlardan kolaylıkla erişim sağlanabilmektedir. 2013 yılında hizmete giren ve 200 den fazla ülkede 50 den fazla dil desteği sunan Code combat, öğrencilere oyun oynarken programlamayı öğretme imkanı sunmaktadır. Öğrenciler kendine uygun karakterler seçerek, kolaydan başlayarak giderek zorlaşan bölümleri geçmeyi çalışmaktadır. Ayrıca öğrenciler karakterlerine ürün ekleyerek daha güçlü karakterler oluşturabilirler. Böylece karakterleri ekstra özellikler kazanabilmektedir. Öğrenciler ayrıca programlamayı kendi seçecekleri dört farklı programlama dilinde öğrenebilirler. Bu programlama dilleri Python, Java Script, Coffee Script ve Lua olarak karşımıza çıkmaktadır. Öğrenciler programlama dilini seçerek ilerleme şansına sahiptirler. Ancak kaldıkları yerden devam edebilmeleri için kayıt oluşturmak zorundalar. Öğretmenler ise öğrencilerini takip edebilmek ve görev verebilmek için sanal sınıf oluşturabilirler. Sınıflarına istedikleri öğrencileri seçebilirler. Ayrıca öğrencilere hazır planları uygulayabilir ve ders materyallerini ücretsiz bir şekilde edinebilirler. Öğrenciler ise isterlerse gerçek zamanlı olarak bir grup oluşturabilir ve o grupta beraber ilerleme şansına sahip olabilirler (CodeCombat,2018).

Yazım temelli görsel programlama aracı olan Code Combat'a ait bir görsel Şekil6'da verilmiştir.



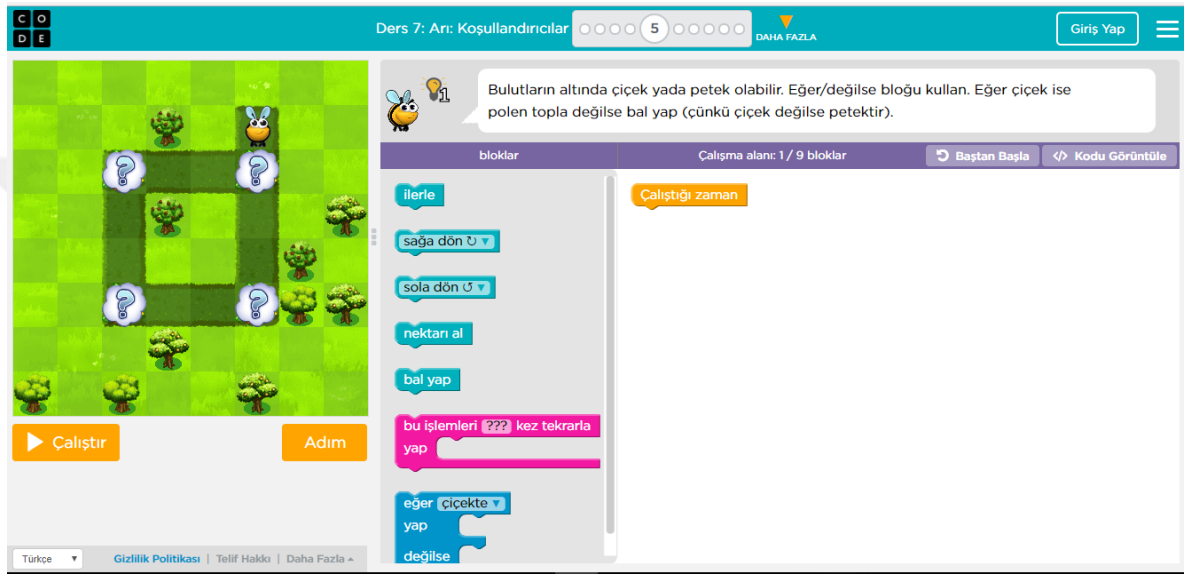
Şekil6. Code Combat programlama diline ait bir görsel

f. Code.org

50'den fazla dil desteği ve tamamen ücretsiz yapısıyla özellikle A-12 öğrenciler için üretilmiş bir görsel programlama ortamıdır. Code.org ortamı kod yapısı olarak hem blok hem de yazım temelli olarak yaş gruplarına uygun hale getirilmiştir. Her yaş grubu ve seviyeye uygun olarak hazırlanmış derslerle programlama öğretmeyi amaçlamaktadır. Özellikle her öğrencinin bilgisayar bilimleriyle tanışmasını sağlamak ve kadınların da bu alanda olan katılımlarını arttırmak amacıyla oluşturulmuştur. Açık kaynak kodlu bir görsel programlama aracı olan Code.org kullanıcıların katkı sağlamasına ve Code.org ortamını geliştirmesine imkan vermektedir. Temel olarak yaş grubu seviyelerine ayrılmış dört ders şeklinde oluşturulmuş Code.org ayrıca popüler olan oyunları da programlama öğretmek için kullanmaktadır. Oyunlardaki karakterleri belli senaryolarla hareket ettirmek için yaş gruplarına uygun komutlar sunarak öğrencilerin programlamayı daha çok sevmesine, eğlenmesine hem de öğrenmesine imkan sağlamaktadır (Code.org, 2018).

Code.org ortamı eğitimciler için oldukça kullanışlı bir ortam olarak karşımıza çıkmaktadır. Eğitimciler için ders planları ve materyaller sunmaktadır. Ayrıca eğitimcilere, öğrencilerini sınıflar oluşturarak ekleyebilecekleri, gelişimlerini takip edecekleri bir kontrol paneli sunmaktadır. Eğitimciler bu kontrol paneliyle öğrencilerin hangi seviyede kaldıkları, geçtikleri bölümleri, görebilmektedir. Code.org eğitimcilere de profesyonellerin katılımıyla kendilerini geliştirme imkanı sunmaktadır (Code.org, 2018).

Görsel programlama aracı olan Code.org'a ait bir görsel Şekil7'de verilmiştir.



Şekil 7. Code.org programlama diline ait bir görsel

2.4. Dünyada ve Ülkemizde Programlama ve Kodlama Eğitimi

2.4.1.Kodlama ve Programlama Eğitimi

Programlama ve Kodlama kavramları günümüzde birbirinin yerine kullanılabilen kavramlar olarak karşımıza çıkmaktadır. Ancak bu kavramlar birbirinin yerine kullanılsa da bazı kavramsal farklılık olduğu da bir gerçektir. Bilgisayar programlama, bir bilgisayarın belirli bir görevi yerine getirmesini, sorunları çözmesini ve insan etkileşimini sağlamasını mümkün kılmak için çeşitli komutlar geliştirme ve uygulama sürecidir. Ancak kodlama belirli bir mantığa göre görevi yerine getirmek için talimatların yazılması, algoritmanın belirlenen programlama dilinde kullanılmasını sağlayan programlamanın özel bir alt görev görevi olarak ifade edilebilmektedir (EuropeanSchoolnet, 2015). Başka bir tanımla kodlama, bilgisayarlara bir işlemi yerine getirmeleri için izlemeleri gereken yolu adım adım yazılan

komutlarla anlatmak olarak tanımlanıyor (“21. yy çocuklarının dili”, 2017). Bu tanımlardan yola çıkılarak kodlamanın programlama için gerekli olan en temel yapı olduğunu söyleyebiliriz. Yine de aralarında çok keskin bir ayrım olmadığı için birbirinin yerine kullanılabilir (EuropeanSchoolnet, 2015). Programlama ve kodlama kavramlarının birbirini yerine kullanıldığına dair alan yazında yapılan çalışmalar olduğu da görülmektedir (Sayın ve Seferoğlu, 2016).

2.4.2.Dünyada Kodlama ve Programlama Eğitimi

Dünyada ve ülkemizde hızla yayılan ve son derece popüler olan kodlama ve programlama kavramlarının, eğitim programları içerisindeki yeri de giderek önem kazanmaktadır. Dünyada çeşitli ülkeler eğitim müfredatlarında programlama eğitimini entegre etmişler veya etmeyi planlamaktadır. Programlama eğitimini müfredatına alan ülkeler incelendiğinde Belçika, Estonya, İngiltere, Portekiz, İspanya, Fransa ve birçok Avrupa Birliği ülkesi ile Avustralya, Güney Kore gibi önemli ülkelerin yer aldığı görülmektedir (Demirer ve Sak, 2016; Sayın ve Seferoğlu, 2016).

Avrupa Komisyonu eski Başkan Yardımcısı Neelie Kroes “tek ihtiyacınız olan şey {C <3DE}” sloganıyla başlattığı kodlama seferberliği Avrupa’da politika haline gelmiştir. Bu politika sayesinde Avrupa’nın 20 ülkesi ve İsrail’in eğitim müfredatları incelenmiş, kodlama eğitiminin müfredata alınması ve müfredatta olan kodlama eğitiminin nasıl daha verimli olarak verilebileceği yönünde yapılan çalışmalar başlanmıştır. 2014 yılında European Schoolnet tarafından yapılan çalışmada 21 ülkeden 16’sının kodlamayı müfredatına aldığını göstermektedir. Bu 16 ülke Avusturya, Bulgaristan, Çek Cumhuriyeti, Danimarka, Estonya, Fransa, Macaristan, İrlanda, İsrail, Litvanya, Malta, İspanya, Polonya, Portekiz, Slovakya ve İngiltere’ den oluşmaktadır. Ayrıca Belçika’nın da kodlama eğitimini müfredatlarına almak için plan yaptığı da görülmüştür (EuropeanSchoolnet, 2015). Son zamanlarda Finlandiya kodlama eğitimini müfredatlarına almasıyla bu sayı 17’ ye yükselmiştir (Deruy, 2017). Sayın ve Seferoğlu (2016) yaptıkları çalışmada bu ülkelerin programlamayı eğitim müfredatlarına alma sebeplerini *Mantıksal Düşünmeyi Destekleme, Problem Çözmeyi Destekleme, Öğrencileri Bilişim Teknolojilerinin İçine Çekme, Kodlama Becerilerinin Desteklenmesi, Bilişim Teknolojileri İstihdamını Destekleme Ve Diğer Anahtar Bileşenleri Desteklemek* alt başlıkları altında toplamışlardır. Bu başlıklardan 15 ülke programlama eğitimini mantıksal düşünmeyi desteklediği için, 11 ülke anahtar yetenek ve kodlama becerilerini geliştirdiği

için ve 8 ülke de istihdama katkı sağladığı gerekçesiyle müfredatlarına aldıkları sonucuna ulaşmışlardır.

Kodlama eğitimi çeşitli yaş gruplarına verilebilmesinden dolayı ülkeler bu eğitimi eğitim müfredatlarına farklı yaş grupları için entegre etmişlerdir. İngiltere, Estonya, İsrail, Slovakya, İspanya, Fransa ve Finlandiya kodlama müfredatını ilkokuldan itibaren verilecek şekilde düzenlemişlerdir. Bulgaristan, Macaristan, Çek Cumhuriyeti ise sadece ortaokulda verilecek şekilde düzenlemiştir (EuropeanSchoolnet, 2015). Bu ülkeler müfredatlarına kodlama eğitimini; programlama, bilişimsel düşünme ve programlama, algoritmik problem çözme ve programlama gibi farklı isimlerle eklemişlerdir (EuropeanSchoolnet, 2015).

Kodlama eğitimi sadece okullarda ders olarak okutulmanın dışında çeşitli etkinlik ve organizasyonlarla tanıtılmaya ve sevdirmeye çalışılmaktadır. Bu etkinliklerin başında “Hour of Code” yani Kodlama saati etkinlikleri gelmektedir. Kodlama saati Code.org platformunun geliştirdiği, her yaş grubundaki öğrenciye uygun olarak tasarlanmış ve 45 dilde, 180’den fazla ülkedeki öğrenciler için tasarlanmış bir saatlik derslerdir (Kodsaati, 2018). Bu etkinliklerde çeşitli popüler oyunları ve çizgi filmleri öğrencilerin kodlamasına fırsat vererek kodlama öğretmeyi amaçlamaktadır. Ayrıca ABD’ de kodlamaya eğitimini yaygınlaştırmak ve geliştirmek amacıyla çeşitli yarışmalar da yapılmaktadır (Sayın ve Seferoğlu, 2016).

2.4.3. Ülkemizde Kodlama ve Programlama Eğitimi

Ülkemizde kodlama ve programlama eğitimi A-12 düzeyinde seçmeli ve zorunlu olarak verilmektedir. Ortaokullarda Bilişim Teknolojileri ve Yazılım dersinde 5 ve 6. sınıflarda zorunlu, 7 ve 8. sınıflar seçmeli olarak haftada 2 ders saati programlama eğitimi alabilmektedirler (TTKB, 2012). Ortaokulda programlama eğitiminin önemi giderek artmakta olduğu görülmektedir. Milli Eğitim Bakanlığı’nın yayınladığı öğretim programlarında Bilişim Teknolojileri ve Yazılım (BTY) dersi içerisindeki ağırlığı giderek artmaktadır. 2006 yılında yayınlanan öğretim programında İlköğretim Bilgisayar Dersi 4 ve 5. Sınıflarda haftada iki ders saati, 1, 2, 3, 6, 7 ve 8. sınıflarda ise haftada bir ders saati olarak verilmekteydi. Ayrıca ders seçmeliydi ve öğrencilere not verilmemekteydi. Programlama ve Kodlama eğitimi Bilişim Teknolojileri’nde İleri Uygulamalar öğrenme alanında programlamaya giriş, nesne tabanlı programlama ve web tasarımı eğitimi olarak belirlenmiş ve 6, 7 ve 8. sınıflarda uygulanmıştır (Seferoğlu, 2007). 2012 yılında dersin adı Bilişim Teknolojileri ve Yazılım dersi olarak değiştirilmiştir. Bu değişiklikte beraber 5 ve 6.

Sınıflarda haftada iki ders saati olarak zorunlu olmuş, 7 ve 8. Sınıflarda ise seçmeli olarak okutulmaya başlanmıştır. Programlama ve kodlama eğitimi içeriği genişletilerek *Problem Çözme, Programlama ve Özgün Ürün Geliştirme* ünitesiyle müfredattaki yerini almıştır (TTKB, 2012).

Ülkemizde ortaöğretimde meslek liselerinin Bilişim Teknolojileri bölümünde okuyan öğrencilere programlama eğitimi verilirken son zamanlarda programlamanın önem kazanmasıyla bazı ortaöğretim kurumlarında da programlama eğitiminin verilmesi gerekliliği tartışılır hale geldi. MEB 2016 yılında yayınladığı öğretim programıyla Bilgisayar Bilimleri dersini Kur-1 ve Kur-2 olarak belirlemiştir. Kur-1 ‘in içeriğine bakıldığında “*Problem Çözme ve Algotimlar*” ve “*Programlama*” ünitesiyle programlamanın bu kurun %93’ ünü kapladığını görmekteyiz. Kur-2 ise “*Robot Programlama*”, “*Web Tabanlı Programlama*” ve “*Mobil Programlama*” olarak tamamen programlama üzerine kurulmuştur (TTKB, 2016). Bilgisayar Bilimleri dersiyle Anadolu İmam Hatip Liselerinde, Anadolu Liselerinde, Güzel Sanatlar ve Spor Liselerinde seçmeli olarak haftada iki saat, hazırlık sınıfı bulunan Anadolu Liselerinde ve Sosyal Bilimler Liselerinde zorunlu olarak haftada dört saat, Fen Liselerinde zorunlu olarak haftada iki saat okutulmasına karar verilmiştir.

Sonuç olarak ülkemizde programlama eğitime verilen değerin günden güne arttığını görüyoruz. Bu artış programlama eğitiminin öğretim programlarındaki yerine bakarak anlaşılabileceği gibi yapılan etkinliklerle ve hayata geçirilen projelerle de anlaşılmaktadır. Özellikle illerde valiliklerin desteklediği bazı projeler kodlama öğretmek ve yaygınlaştırmak açısından oldukça ilgi uyandırmaktadır. Bu projelerden bazıları; Manisa Valiliği’nin desteğiyle yürütülen “*Kodla(MA)nisa*”, Rize Valiliği’nin desteğiyle yürütülen “*kodlarize*”, İzmir robotik kodlama projesi olan “*Robokod*”, Trabzon Valiliği koordinatörlüğünde, Karadeniz Teknik Üniversitesi ve Trabzon İl Millî Eğitim Müdürlüğü işbirliğinde yürütülen bir kodlama eğitimi projesi olan “*kodla{yap}*”, Düzce’ de “*Düzce Kodluyor*” projesi, Yozgat Valiliği himayesinde İl Milli Eğitim Müdürlüğü tarafından başlatılmış bir proje “*KodluYoz*”, Gümüşhane “*Gümüşkod*”, Tokat İl Milli Eğitim Müdürlüğü Kodlama Projesi “*Bir Yazılım Masalı*”, Gaziantep “*KodlAntep*” projesi, Aksaray “*KodlAksaray*” projesi, Muğla “*KodlaMuğla*” projesi ve Kocaeli “*Kodla Kocaeli*” projeleridir (EBA, 2018). Ayrıca bölgesel projeler dışında yapılan etkinlikler de vardır. Bu etkinliklerden bazıları; geldim, gördüm, kodladım sloganıyla CodeFest (Codefest, 2018), Robotik kodlama festivali olan “*Robocotfest*” (Bilgoritma, 2017), sınıfların yarıştığı kodlama yarışması olan “*Kodris*” olarak ifade edilebilir (Kodris, 2018).

2.5. Alan Yazın Çalışmalar

2.5.1.Yükseköğretimde ve Öğretmen Eğitiminde Scratch ve Kodu Game Lab Kullanılması ile İlgili Alan Yazın

Çalışmanın bu kısmında yüksek öğretimde ve öğretmen eğitiminde Scratch ve Kodu Game Lab programlama dillerinin kullanılması ile ilgili son 10 yıla ait yayınlanan makale ve tez çalışmaları incelenmiştir.

Yıldırım (2016) bilgisayar ile ilgili bölümde okuyan üniversite öğrencilerin programlama eğitiminde oyun ve dijital oyun tabanlı programlama dilleri kullanımı hakkında görüşlerini incelemek ve oyunların öğrenciler üzerindeki etkilerini araştırmak amacıyla 177 BÖTE ve Bilgisayar programcılığında okuyan üniversite öğrencisiyle bir çalışma gerçekleştirmiştir. Çalışma kapsamında BÖTE bölümünde okuyan ve rastgele seçilen 12 öğrenciye 40 saat süresince Scratch eğitimi verilmiştir. Diğer öğrencilere ise bu çalışma kapsamında herhangi bir eğitim verilmemiştir. Ancak toplamda 69 öğrencinin daha önce herhangi bir oyun tabanlı bir programlama dili kullandığı, 108 öğrencinin ise herhangi bir oyun tabanlı programlama dili kullanmadığı görülmüştür. Veriler öğrencilerden anket yoluyla toplanmıştır. Ayrıca öğrencilerin verilerinin doğruluğunu teyit etmek amacıyla 44 öğrenci ile birebir görüşme yapılmıştır. Araştırmanın sonuçları incelendiğinde katılımcılarının çoğunun oyun oynamayı sevdikleri, faydalı buldukları ve motive edici olarak buldukları söylenebilmektedir. Dijital oyun tabanlı program olarak en çok Scratch'ı tercih ettikleri ve Scratch hakkında olumlu bir görüş geliştirdikleri görülmektedir. Ayrıca Scratch eğitimi alan öğrencilerin bu eğitimi öğrencilerine vermek ve yaygınlaştırmak istedikleri, problem çözme, işlemsel düşünme, yaratıcılık, iletişim kurma ve takım çalışması gibi özellikleri geliştirdiği, eğlenceli öğrenme araçları ile kalıcı öğrenme sağladığını ifade etmişlerdir. Ancak dijital oyun tabanlı programları bilmedikleri ve bu yüzden de kullanılmadığı sonucuna ulaşılmıştır.

Baltalı (2016) ilköğretim seviyesinde kullanılacak programlama dillerinin seçilme sebeplerini belirlemek üzere 92 Bilişim Teknolojileri öğretmenlerinin görüşlerini alarak bir tez çalışması yürütmüştür. Çalışmada 4 programlama dili; Scratch, Microsoft Small Basic, Microsoft Kodu Game Lab ve Robomind incelenmek üzere seçilmiştir. Öğretmenlerin yazılımları değerlendirmeleri için her bir öğretmenin bu yazılımlardan birini kullanması sağlanmıştır. Bunun için öğretmenlere programın nereden ve nasıl indirebileceğine ve programının tanıtımının yapıldığı bir Youtube bağlantı linki e-posta hesaplarına bir gönderilmiştir. Öğretmenler daha sonra kendilerine gelen formu doldurup göndermişlerdir.

Çalışmanın sonuçları incelendiğinde görsel programlama aracı kullanmanın öğrenciler üzerinde olumlu bir etki bıraktığı ve ortaokul öğrencileri için uygun olduğu sonucuna ulaşılmıştır. Ayrıca öğretmenlere göre programlama dili seçiminde programlama dilinin tasarımının önemli olduğu ifade edilmiştir. Ayrıca programlama dili seçiminde kullanım kolaylığına dikkat edilmesi gerektiği ifade edilmiştir. Kodu Game Lab programlama dilinin, tasarım ve görsellik açısından ikinci olduğu ve onu Scratch programlama dilinin takip ettiği görülmektedir. Görsel programlama dili yazılımlarının genel olarak kullanılabilir ve tasarımlarının da iyi olduğu ancak bazı sorunlar da çıkardığı görülmüştür.

Yülseltürk ve Altıok (2016) Eğitsel oyun geliştirme sürecinde Scratch programlama aracının kullanımına ilişkin Bilişim Teknolojileri öğretmen adaylarının görüşlerini incelemek amacıyla Kırıkkale Üniversitesi BÖTE bölümünde öğrenim gören 169 öğretmen adayıyla bir çalışma gerçekleştirmişlerdir. Çalışma toplam bir dönem boyunca sürmüştür. Öğrencilere ilk sekiz hafta öncelikle oyun özellikleri anlatılmış; Scratch aracı tanıtılmış, temel kavramlar aktarılmış, çeşitli uygulamalar yapmaları ve son olarak da çeşitli eğitsel oyunlar geliştirmeleri istenmiştir. Öğretmen adayları tarafından geliştirilecek eğitsel oyunlar; hedef kitlesi, konusu ve oyunun sahip olması gereken özelliklere göre sınıflandırılmıştır. Veriler, öğretmen adaylarından üç şubeden altı öğrenci seçerek odak grup çalışması yoluyla nitel olarak toplanmıştır. Ayrıca öğrencilerin yaptıkları oyunlarda incelenerek araştırma sonuçlarına ilave edilmiştir. Araştırmanın sonuçları incelendiğinde Scratch ile eğitsel oyun geliştirmenin kolay olduğu görülmüştür. Öğrencilerin ilkökul seviyesinde oyunlar gerçekleştirdiği bunun sebebinin de mezun olan öğrencilerin bu seviyede eğitim vereceklerinden kaynaklandığı sonucuna ulaşılmıştır. Ayrıca öğrencilerin geliştirdikleri oyunların daha çok matematik dersine yönelik olduğu görülmüştür. Bu durumun sebebinin de öğrencilerin zor olan dersleri kolaylaştırmaya yönelmeleridir. Öğrencilerin projeleri incelendiğinde belirlenen kriterleri çoğunun gerçekleştirdiği ve projelerin büyük bir kısmının orta ve üzeri olduğu görülmüştür. Ayrıca öğrencilerin çoğunun da projelerini tamamladığından dolayı kendilerini başarılı gördükleri sonucuna ulaşılmıştır.

Fidan (2016) Scratch ile programlama öğretiminde oyunlaştırmanın öğrencilerin katılımı, motivasyonu ve başarısına olan etkisini incelemek amacıyla 37 üniversite ikici sınıf öğrencisiyle bir tez çalışması yürütmüştür. Çalışma Eğitimde Grafik ve Canlandırma dersinde bir dönem içerisinde 12 hafta boyunca gerçekleştirilmiştir. Haftalık konu, etkinlik ve görev tablosu hazırlanmış ve Edmodo üzerinden paylaşılmıştır. Süreç olarak öğrencilerin bir oyunun içinde olduğu hissettirilmiş puan, rozet ve skor tablosu gibi bileşenleri de sürece dahil etmiştir. Araştırmanın sonuçları incelendiğinde Scratch ile programlama eğitiminde

oyunlaştırmanın öğrenci katılımı ve motivasyonunu arttırdığı görülmüştür. Ayrıca dersin daha eğlenceli bir ortam ve süreç geçirildiği görülmüştür. Akademik başarı açısından da oyunlaştırmanın olumlu etki ettiği sonucuna varılmıştır.

Demir (2015) programlama öğretiminde farklı eğitsel programlama dillerinin derse entegrasyonunun öğrencilerin akademik başarısına, performansına ve bilgisayar programlamaya yönelik kaygılarına olan etkisini incelemek amacıyla 87 öğrenciyle bir tez çalışması yürütmüştür. Öğrenciler rastgele örneklem yoluyla seçilen öğrenciler 3 deney grubuna ayrılmışlardır. Çalışmada kontrol grubu kullanılmamıştır. Scratch ortamı, eğitsel programlama dili olarak seçilmiştir. Üç deney grubu, eğitsel programlama dilini derse şu şekilde entegre etmiştir; birincisi deney grubuyla programlama eğitimi teorik olarak işlenmiş, ikinci deney grubuna programlama eğitimi uygulamalı olarak, üçüncü deney grubuna ise programlama eğitimi hem teorik hem de uygulamalı olacak şekilde işlenmiştir. Çalışma eğitsel program dilinin seçimi ve hazırlık süreci iki hafta, uygulama süreci ise altı hafta olacak şekilde yürütülmüştür. Çalışmanın sonuçları incelendiğinde üç grupta da akademik başarının arttığı görülmektedir. Ayrıca eğitsel programlama dili ile programa eğitimi yapmanın öğrencilerin programlama öğretiminde kaygı düzeylerini olumlu yönde etkilediği sonucuna ulaşılmıştır.

Erol (2015) Scratch ile programlama öğretiminin öğrencilerin motivasyon ve programlama başarısına etkisini incelemek amacıyla 52 üniversite öğrencisi deneysel bir tez çalışması gerçekleştirmiştir. Deney ve kontrol gruplarındaki öğrenciler eşit sayıda ayrılmıştır. 14 hafta boyunca öğrencilere programlama eğitimi verilmiştir. İlk 7 haftada deney grubuna Scratch ile etkinlikler yaptırılmış, kontrol grubuna ise normal ders akışıyla programa mantığı verilmeye çalışılmıştır. Sonraki 7 hafta boyunca her iki gruba da C# programlama dili öğretilmiştir. Bu çalışmaya göre Scratch programlama dilinin öğrenciler üzerinde motivasyonu arttırmada anlamlı derecede etkili olduğu sonucuna ulaşılmıştır. Ayrıca programlama dillerine yönelik başarı konusunda da Scratch programlama dilinin anlamlı derece etkili olduğu sonucuna varılmıştır.

Ozoron, Çağıltay ve Topallı (2012) mühendislik öğrencilerine programlama giriş dersinde Scratch programlama dili kullanarak derse olan etkisini incelemek ve öğrencilerin görüşlerini almak amacıyla 55 mühendislik birinci sınıf öğrencisiyle bir çalışma gerçekleştirmişler. Eylem araştırması şeklinde gerçekleşen bu çalışmada öğrenciler bir dönem boyunca Scratch ile eğitim görmüş ve uygulama geliştirmişler. Öğrencilere 2 ara sınav ve final sınavı, ev ödevleri, laboratuvar etkinlikleri yapılmış ve geçmiş son iki yıla göre öğrencilere ait sonuçlar incelenmiştir. İncelemeler sonunda Scratch destekli programlama ile

öğrencilerin başarısızlık oranının düştüğü, katılım oranının arttığı görülmektedir. Bu çalışmanın genel sonuçları incelendiğinde, Scratch programının öğretimi daha eğlenceli hale getirdiği, daha çok görsellik kattığı, öğrencilerin yaratıcılıklarını arttırdığı, sistemli düşünceyi geliştirdiği ortaya çıkarılmıştır.

Genç ve Karakuş (2011), üniversite öğrencilerinin oyun tasarımı sürecinde Scratch kullanımına ait deneyimleri ile ilgili görüşlerini incelemek amacıyla Fırat üniversitesi BÖTE bölümü 2. Sınıfta öğrenim gören 109 öğrenciyle Eğitimde Bilgisayar Oyunları Tasarımı dersi kapsamında bir çalışma yürütmüşlerdir. Çalışmada nitel ve nicel veriler toplanarak analiz edilmiştir. Öğrenciler Programlama Dilleri dersi kapsamında Scratch programlama dili öğretilmiş ve dersin sonunda öğrencilerden oyun yapmaları istenmiştir. Çalışmanın sonuçları Scratch programının genel kullanımı, hoşlanma ve hoşlanmama durumları, öğrencilerin kendilerine olan güveni ve programlamaya karşı algıları alt başlıklarında incelenmiştir. Öğrencilerin Scratch programlama dili kullanımını basit bulduğu, bu sayede Scratch kullanırken kendilerini rahat hissettikleri sonucuna ulaşmışlardır. Ayrıca Scratch kullanımının öğrencilerin hoşuna gittiği ve temel programlamanın öğretilmesinde Scratch'ın diğer programlama dillerine oranla daha kolay öğrenilebilir bir dil olduğu görülmüştür. Scratch programlama dilinin motivasyonu arttırdığı ancak ileri düzeyde programlama yapanlar için uygun olmadığı görüşüne ulaşıldığı görülmüştür.

Fesakis ve Serafeim (2009) Yunanistan'da bir üniversitede programlamaya hakim olmayan 35 üniversite öğrencisinin Scratch programlama dili tutumlarını ve görüşlerinin incelemek amacıyla bir çalışma yapmışlardır. Bu çalışmaya göre öğrencilerin Scratch' ı kullanıcı dostu olarak gördüğü ve kullanımının kolay ve tatmin edici olarak gördüğü sonucuna ulaşılmıştır. Scratch kullandıktan sonra öğrencilerinin Scratch kullanma konusunda anlamlı düzeyde bir istek yaşadıkları görülmüştür. Ayrıca Scratch programlama dilinin öğrencilerin programlama öğrenmedeki kaygı ve stresi anlamlı ölçüde azalttığı sonucuna ulaşılmıştır. Bu durumdan yola çıkılarak Scratch'ın öğrencilerin başarıları ve tutumları üzerinde olumlu bir etki oluşturabileceği varsayımına ulaşılmıştır.

Malan ve Leitner (2007) ABD Harvard üniversitesi yaz kampında, Bilgisayar Bilimleri dersinde Scratch programlama dilini kullanarak öğrencilerin görüşlerini incelemek amacıyla bir çalışma yürütmüşlerdir. Bu çalışma kapsamında çoğu daha önce programlama öğretimi almamış 25 üniversite öğrencisinin görüşlerini incelemiştir. Öğrencilerden alınan görüşler doğrultusunda programlama öğretimi sürecinde Scratch programlama dilinin öğrencileri heyecanlandığını ve daha önce programlama dili eğitimi almamış kişilere kolay bir şekilde programlama öğrenmelerine imkân sağladığı sonucuna ulaşılmıştır. Ayrıca

bilgisayar bilimleri dersine girişte Scratch programlama dilinin olumlu bir etki sağladığı sonucuna ulaşılmıştır. Ancak Scratch programlama dilinin sınırlı kod yapısı olduğundan sadece giriş derslerinde kullanılması gerektiği sonucuna ulaşılmıştır.

Alan yazında incelenen çalışmalar Tablo 2’de özetlenerek görsel hale getirilmiştir.



Tablo 2
Yükseköğretimde Öğretmen Eğitiminde Scratch ve Kodu Game Lab ile Yapılan Çalışmalar

Çalışma Adı	Yazarları	Türü	Yılı	Amacı	Çalışma Grubu	Yöntem	Bulgular
Investigation of Pre-Service Information Technology Teachers' Game Projects Prepared with Scratch	Erman Yükseltürk, Serhat Altıok	Makale	2016	Eğitsel oyun geliştirme sürecinde Scratch programlama aracının kullanımına ilişkin Bilişim Teknolojileri öğretmen adaylarının görüşlerini incelemek	169 Üniversite Öğrencisi	Nitel	Scratch ile oyunlar daha kolay bir şekilde geliştirilebilir olduğu, Oyunların alan eğitiminde kullanılmasının alan derslerini daha iyi öğrenmeyi sağladığı, Öğrencilere orta ve üzeri oyun geliştirme becerisi kazandırdığı, Öğrencilerde başarıma hissi oluşturmaya yardımcı olduğu ortaya çıkmıştır.
Dijital Oyun Tasarım Programlarının Eğitimde Önemi	Ebru YILDIRIM	Makale	2016	Bilgisayar ile ilgili bölümde okuyan üniversite öğrencilerin programlama eğitiminde oyun ve dijital oyun tabanlı programlama dilleri kullanımı hakkında görüşlerini incelemek	177 Üniversite Öğrencisi	Nicel ve Nitel	En çok tercih edilen dijital oyun tabanlı programın Scratch olduğu, Scratch'ın öğretilmesi ve yaygınlaştırılması gerektiği, Scratch'ın kalıcı öğrenmeler sağladığı, Scratch eğitiminin yaratıcı, işlemsel düşünme, problem çözme gibi özellikleri geliştirdiği, Oyunların öğrencileri motive ettiği ve onlara fayda sağladığı sonucuna ulaşılmıştır.
Programlama Öğretiminde Kullanılabilecek Yazılımlara İlişkin Öğretmen	Salih Baltalı	Tez	2016	İlköğretim seviyesinde kullanılacak programlama dillerinin seçilme sebeplerini belirlemek	92 Bilişim Teknolojileri Öğretmeni	Nicel ve Nitel	Scratch ve Kodu Game Lab programlama dilinin ilköğretim seviyesine uygun olduğu, Scratch ve Kodu Game Lab programlama dilinin tasarım ve

Görüşleri							kullanılabilirlik açısından iyi olduğu Görsel programlama araçlarının programlama öğretiminde öğrenciler üzerinde olumlu etkiye sahip olduğu görülmüştür.
Scratch İle Programlama Öğretiminde Oyunlaştırmanın Öğrenci Katılımına Etkisi	Ayça Fidan	Tez	2016	Scratch ile programlama öğretiminde oyunlaştırmanın öğrencilerin katılımı, motivasyonu ve başarısına olan etkisini incelemek	37 Üniversite İkinci Sınıf Öğrencisi	Nitel	Scratch ile oyunlaştırmanın öğrenci katılımını ve motivasyonunu arttırdığı, Dersi daha eğlenceli hale getirdiği Oyunlaştırmanın akademik başarıyı arttırdığı ortaya çıkmıştır.
Programlama Öğretiminde Eğitsel Programlama Dilinin Farklı Kullanımlarının Programlama Başarısı Ve Kaygısına Etkisi	Faruk Demir	Tez	2015	Programlama öğretiminde farklı eğitsel programlama dillerinin derse entegrasyonunun öğrencilerin akademik başarısına, performansına ve bilgisayar programlamaya yönelik kaygılarına olan etkisini incelemek	87 Üniversite Öğrencisi	Nicel	Scratch ile programlama yapmanın başarıyı arttırdığı, Scratch ile programlama eğitiminin öğrencilerin kaygılarını azalttığı sonucuna ulaşılmıştır.
Scratch İle Programlama Öğretiminin Bilişim Teknolojileri Öğretmen Adaylarının Motivasyon Ve Başarılarına Etkisi	Osman Erol	Tez	2015	Scratch ile programlama öğretiminin öğrencilerin motivasyon ve programlama başarısına etkisini incelemek	52 Üniversite Öğrencisi	Nicel ve Nitel	Scratch'ın programlama öğretiminde motivasyonu arttırdığı, Scratch ile programlama yapmanın başarı üzerinde anlamlı derecede katkı sağladığı görülmüştür.

Using Scratch In Introduction To Programming Course For Engineering Students	Dinçer Ozoran, Nergiz Erçil Çağıltay, Damla Topallı	Makale	2012	Öğrencilerine programlama giriş dersinde Scratch programlama dili kullanarak derse olan etkisini incelemek ve öğrencilerin görüşlerini almak	55 Üniversite Mühendislik Birinci Sınıf Öğrencisi	Nicel	Scratch ile programlama yapmanın derse katılımı arttırdığı, Scratch'ın dersi eğlenceli hale getirerek, öğrencilerin yaratıcılıklarının ve sistemli düşünme becerilerini geliştirdiği Programlama öğretiminde Scratch'ın öğrenci başarı oranını arttırdığı ortaya çıkmıştır.
Tasarımla Öğrenme: Eğitsel Bilgisayar Oyunları Tasarımında Scratch Kullanımı	Zülfü Genç, Songül Karakuş	Makale	2011	Öğrencilerin oyun tasarımı sürecinde Scratch kullanım deneyimini incelemek	109 Üniversite 2. Sınıf Öğrencisi	Nicel ve Nitel	Scratch'ın kolay yapısı sayesinde programlama öğretiminde öğrencilerin kendilerini rahat hissettikleri, Programlama öğretiminde Scratch'ın öğrenci motivasyonunu arttırdığı, Scratch'ın temel programlama için etkili olduğu ancak ileri düzey için uygun olmadığı sonucuna ulaşılmıştır.
Influence of the Familiarization with “Scratch” on Future Teachers’ Opinions and Attitudes about Programming and ICT in Education	Georgios Fesakis, Kiriaki Serafeim	Makale	2009	Öğretmen adaylarının Scratch programlama dili hakkında tutumlarını ve görüşlerini incelemek	35 Üniversite Öğrencisi	Nicel	Scratch'ın kullanımının kolay olmasının öğrenciler üzerindeki kaygı ve stresi anlamlı ölçüde azalttığı, Programlama eğitiminde Scratch'ın kullanılmasının öğrencilerin programlamaya karşı isteklerini arttırdığı, Öğrencilerin programlama başarılarını ve programlamaya karşı olan tutumlarını olumlu yönde etkilediği ifade edilmiştir.
Scratch for Budding Computer	David J. Malan, Henry H.	Makale	2007	Bilgisayar bilimleri dersinde Scratch programlama dilini	25 Üniversite Öğrencisi	Nicel	Scratch'ın kolay yapısından dolayı öğrencilerin programlamayı daha kolay öğrenmelerine imkan sağladığı

Scientists	Leitner			kullanarak öğrencilerin görüşlerini incelemek			Scratch'ın öğrenciler üzerinde olumlu bir etki yarattığı, Kod yapılarının, üst düzey programlar yapabilecek şekilde olmadığı, bu yüzden de programlamanın temel kavramlarının öğretimi için uygun olduğu ifade edilmiştir.
------------	---------	--	--	---	--	--	---

2.5.2.İlköğretim ve Ortaöğretimde Scratch ve Kodu Game Lab ile İlgili Alan Yazın

Çalışmanın bu kısmında ilköğretim ve ortaöğretimde Scratch ve Kodu Game Lab programlama diline ait son yıllarda yayınlanan makale ve tez çalışmaları incelenmiştir.

Yünkül, Durak, Çankaya ve Mısırlı (2017) Scratch eğitimi alan öğrencilerin bilgisayarca düşünme becerileri ölçeği ile Bilişim Teknolojileri ve Yazılım dersi puanları arasındaki ilişkiyi belirlemek ve Scratch eğitiminin bilgi işlemsel düşünce puanları üzerine etkisini araştırmak amacıyla Balıkesir’de iki farklı ortaokul 6. sınıfta öğrenim gören toplam 69 öğrenciyle bir çalışma gerçekleştirmişlerdir. Yapılan deneysel çalışmada 69 öğrencinin 36’sı deney grubunda, 33’ü ise kontrol grubunda yer almaktadırlar. Her iki gruptaki öğrencilerin daha önce Scratch ortamıyla karşılaşmadığı görülmüştür. Deney grubu öğrencilerine bir dönem boyunca haftada iki ders saati Scratch eğitimi verilmiştir. Kontrol grubuna ise Scratch eğitimi verilmemiştir. Son olarak öğrencilere dönem sonu sınavı ve bilgisayarca düşünme becerileri ölçeği uygulanmıştır. Araştırmanın sonuçları incelendiğinde Scratch eğitimi alan öğrencilerin Bilgi işlemsel düşünce puanlarının almayanlara oranla daha yüksek puanlar aldığı görülmüştür. Bu durumdan dolayı Scratch’ın problem çözme, algoritmik düşünce ve yaratıcı düşünme becerileri üzerinde anlamlı bir etkisi olduğu varsayımında bulunulmuştur. Ayrıca deney grubu öğrencilerin derse ait puanları ile Bilgi işlemsel düşünce puanları karşılaştığında pozitif yönde yüksek bir ilişki olduğu görülmüştür. Bu durumdan yola çıkarak bilgi işlemsel düşünce becerileri yüksek olan öğrencilerin programlamada da başarılı olacağı sonucuna ulaşılabilmektedir.

Yıldırım (2017) ortaokul öğrencileriyle Scratch programlama dili kullanarak bir mobil uygulama geliştirerek öğrencilerin programlama becerilerinin geliştirilmesini sağlamak amacıyla ortaokul 6. sınıfta öğrenim gören toplam 94 öğrenciyle bir tez çalışması gerçekleştirmiştir. Gerçekleştirilen çalışmada yarı deneysel desen kullanılmıştır. Öğrencilerin 41’i deney grubuna, 53’ü ise kontrol grubuna ayrılmıştır. Gerçek uygulama öncesi 7. sınıflardan 10 öğrenci seçilerek, daha önceden yapılan mobil uygulamayı denemişlerdir. Gerçek uygulama ise 5 hafta boyunca deney grubundaki öğrencilerin telefonlarına mobil uygulama indirilerek Scratch programlama dili eğitimi almışlardır. Ayrıca ders öğretmeninden de programlama eğitimi almaya devam etmişlerdir. Kontrol grubu ise sadece ders öğretmeninden programlama eğitimi almıştır. Araştırmanın sonuçları incelendiğinde Scratch ile hazırlanmış programlama dili mobil uygulamasının programlama

dili öğretilmesi açısından anlamlı bir şekilde farklılık sağladığı görülmüştür. Ayrıca programlama eğitimine katkı sağladığı sonucuna ulaşılmıştır.

Yüksel (2017) Scratch programı öğretiminde ayrılıp birleştirme tekniği kullanımının öğrencilerin derse yönelik tutumuna, başarısına ve kalıcılığa olan etkisini incelemek amacıyla ortaokul 6. Sınıf öğrencisiyle bir tez çalışması yürütmüştür. Çalışmada 60 ortaokul 6. Sınıf öğrencisi yer almıştır. Yapılan araştırmada yarı deneysel bir desen kullanılmıştır. Öğrenciler deney ve kontrol gruplarına 30'ar kişi olarak ayrılmışlardır. Araştırma 7 hafta boyunca haftada 2 saat olmak üzere toplam 14 ders saati süresinde gerçekleştirilmiştir. Kontrol grubu öğrencilerine düz anlatım gerçekleştirilmiş ve ek olarak çalışma kâğıdı dağıtılmıştır. Deney grubu ise 5 kişilik 6 heterojen grup olarak ayrılmış ve ayrılıp birleşme tekniği kullanılarak ders işlenmiştir. Araştırmanın sonuçları incelendiğinde ayrılıp birleşme tekniği ile yapılan programlama eğitiminin ders başarısını, derse yönelik tutumu ve öğrenilen bilgilerin kalıcılığını olumlu yönde etkilediğini göstermektedir.

Papadakis, Kalogiannakis, Zaranis ve Orfanakis (2016) acemi programcılara programlama öğretiminde Scratch ve App Inventor programlama dillerini kullanmanın öğrencilerin performansına ve tutumuna olan etkisini araştırmak amacıyla Yunanistan'ın Attica bölgesinde öğrenim gören 87 lise öğrencisiyle bir çalışma gerçekleştirmişler. Yapılan deneysel çalışmaya katılan öğrencilerin 35'i deney grubu 1, 34'ü deney grubu 2 ve 18'i ise kontrol grubunu oluşturmuştur. Deney grubu 1'e App Inventor, deney grubu 2'ye Scratch programlama dili, kontrol grubuna ise tipik metin tabanlı programlama dili eğitimi dört ay boyunca verilmiştir. Araştırmanın sonuçları incelendiğinde Scratch ve App Inventor programlama dili kullanan öğrencilerin tutumları ve programlama bilgileri anlamlı derecede arttığı görülmektedir. Buna ek olarak App Inventor kullananların performanslarının Scratch kullanan gruba ve kontrol grubuna oranla daha fazla olduğu görülmüştür.

Mladenović, Krpan ve Mladenović (2016) Scratch ve Python programlama dillerinin programlama başarısı ve problem çözme becerilerine olan etkisini araştırmak amacıyla Hırvatistan'da ilköğretim 5. Sınıfta öğrenim gören 50 öğrenciyle bir çalışma gerçekleştirmiştir. Çalışmaya katılan öğrenciler iki farklı okuldan dört şubeden 34 erkek ve 16 kız öğrenci olarak seçilmiştir. Birinci okuldan toplam 36 öğrenci, ikinci okuldan da toplam 14 öğrenci araştırmaya dâhil olmuştur. Araştırmanın gruplarını okul durumu, cinsiyete göre belirlenmiştir.. Öğrenciler daha önce programlama öğretimi almadığından herhangi bir programlama bilgileri yoktu. Programlama öğretimini tüm sınıflarda gerçekleştiren öğretmen aynıydı. Tüm öğrencilere programlama eğitimi verilmeden önce problem çözme testi verilmiştir. Bu test öğrencilerin motivasyonlarını arttırmak amaçlı

olduğundan kolay bir test uygulanmıştır. Testin ardından dört hafta boyunca haftada iki saat olmak üzere toplam sekiz hafta süresince Python programlama dili öğretilmiştir. Sekiz haftada Python’ da değişkenler, diziler, koşullar, veri girişi ve ekrana yazdırma kavramları anlatılmıştır. Daha sonra Python kavramlarının ne kadar anlaşıldığını belirlemek için test uygulanmıştır. Python’ dan sonra öğrenciler üç hafta boyunca öğretmen rehberliğinde verilen basit oyunları Scratch’ta uygulamaya geçmiştir. Üç hafta sonra da Scratch kavramlarının ne kadar öğrenildiğini belirlemek için test uygulanmıştır. Araştırmanın sonuçları incelendiğinde problem çözme becerileri açısından incelendiğinde okul durumu ve cinsiyete göre öğrenciler arası anlamlı bir fark bulunamamıştır. Ayrıca Scratch ve Python test sonuçları incelendiğinde de gruplar arası anlamlı bir fark bulunamamıştır. Scratch test sonuçları ile problem çözme becerileri testi sonuçları incelenmiş ve Scratch puanlarının anlamlı derecede yüksek olduğu sonucuna ulaşılmıştır. Bu durumda Scratch’ta oyunları programlayarak Scratch programlama dili öğrenmenin problem çözme yeteneğine bağlı olmadığı sonucuna ulaşılmıştır. Scratch ve Python testleri incelendiğinde zayıf öğrencilerin Scratch’ta daha yüksek puanlar alındığı görülmüştür. Bu durumda Scratch’ta öğrencilerin Python’ a göre daha başarılı olacağı sonucuna ulaşılmıştır.

Oluk ve Korkmaz (2016) 5. Sınıf öğrencilerin Scratch becerileri ile Bilgi işlemsel düşünce becerilerinin farklı değişkenler açısından karşılaştırmak amacıyla Kastamonu’da 31 ortaokul 5. sınıf öğrencisiyle bir ilişkisel araştırma modelini kullanarak bir çalışma gerçekleştirmişler. Çalışmadaki 31 öğrencinin 16’sı kız, 15’i de erkek öğrencilerden oluşmuştur. Öğrenciler Bilişim Teknolojileri ve Yazılım dersi kapsamında 6 hafta ve haftada 2 ders saati süresince Scratch eğitimi almışlardır. Öğrencilere internet üzerinden Scratch projeleri gösterilmiş ve Scratch hakkında bilgiler verilmiştir. Ayrıca derslerin bir bölümünde Scratch ile uygulama yapma imkânı sunulmuştur. Ayrıca son iki hafta öğrencilerin kendi projelerini yapmaları istenmiştir. Öğretmen süreçte sadece bir rehber görevi yürütmüştür. Araştırmanın sonuçları incelendiğinde ölçüm araçlarından herhangi birini kullanarak elde ettikleri puanların cinsiyete veya bilgisayar kullanım süresine göre farklılık göstermediği, ancak öğrencilerin Scratch ile programlama becerilerinin sayısal hesaplama becerileri arasında anlamlı bir ilişki olduğu görülmüştür.

Papatğa (2016) okumada sorun yaşayan ilkökul öğrencilerine Scratch programı aracılığıyla okuduğunu anlama becerilerinin nasıl geliştiğini ortaya koymak amacıyla ilkökul dördüncü sınıf öğrencileriyle bir tez çalışması yürütmüştür. Çalışmayı sekiz adet dördüncü sınıf öğrencisiyle 15 hafta boyunca sürdürmüştür. 15 hafta boyunca yapılan araştırmanın ilk iki haftasında okul belirlenmiş, 3., 4., 5. ve 6. haftasında, haftada üç gün olmak üzere toplam

24 saat Scratch programlama dili öğretilmiştir. 7. ve 8. hafta ise pilot uygulama yapılmış ve geri kalan 6 hafta boyunca toplam 6 metin üzerinden 46 ders saati boyunca çalışmayı gerçekleştirmiştir. Çalışmada hem nitel hem de nicel veri toplama araçları kullanılmıştır. Araştırmanın hem nicel hem de nitel sonuçları incelendiğinde Scratch programıyla yapılan etkinliklerin öğrencilerin okumayı anlama becerilerini geliştirdiği sonucuna ulaşılmıştır. Ayrıca derste işbirlikli öğrenme ortamı sağlandığından, işbirlikli öğrenme yönteminin öğrencilerin Türkçe dersindeki okuduğunu anlama becerileri üzerinde geleneksel öğretim yöntemine göre daha etkili olduğu sonucuna ulaşılmıştır.

Hava (2016) bilgisayar oyunu geliştirme etkinliğini incelemek amacıyla üstün yetenekli 20 öğrenci ile bir tez çalışması yürütmüştür. Bu çalışma iki faz halinde ilk faz 12, ikinci faz 11 toplamda ise 23 hafta sürecek şekilde gerçekleştirilmiştir. Çalışmada oyun platformu olarak Kodu Game Lab kullanılmıştır. Öğrencilerden Kodu Game Lab ile bir eğitsel oyun tasarımları istenmiştir. Oyun geliştirme etkinliğini incelemek amacıyla nitel ve nicel veriler toplanmıştır. Çalışmanın sonuçları incelendiğinde oyun tasarım modelinin oyun geliştirme bakımından faydalı olduğu ve öğretim uygulamalarının bu süreçte önemli rol oynadığı görülmüştür. Ayrıca oyun geliştirme oyun geliştirme etkinliğinin öğrencilerin problem çözme, üretim becerisi kazandırma yönünden önemli çıktılar verdiği görülmüştür.

Wang, Huang ve Hwang (2014) Scratch ile bütünleştirilmiş proje senaryoları oluşturarak proje destekli öğrenme yoluyla problem çözme performansı, öğrenmeye ve Scratch karşı olan tutumu ve öğrenmeye karşı olan motivasyonunu araştırmak amacıyla 91 ortaokul öğrencisiyle bir çalışma yürütmüşlerdir. Bu öğrencilerin 43'ü ortalama puanlara sahip, 48'i ise yüksek matematik puanlarına sahip öğrencilerden oluşmaktadır. Bu iki grup aynı öğretmenin kontrolü altında çalışmalarını sürdürmüştür. Deney ve kontrol grubundaki öğrenciler kendi grupları içerisinde dörtlü işbirlikli gruplar halinde ayrılmışlardır. Sadece bir grup üç kişi olarak ayrılmışlardır. Araştırmanın sonuçlarına problem çözme becerileri, öğrenme motivasyonu, dersi alma yönündeki tutum ve Scratch kullanımına yönelik tutum bakımından hem matematik hem de ortalama öğrenciler öğrenme etkinliğinden sonra önemli gelişmeler gösterdiği görülmüştür. Ancak problem çözme becerileri, öğrenmeye karşı olan motivasyonları ve öğrenmeye ve Scratch karşı olan tutumu açısından iki grup arasında anlamlı bir farklılık bulunamamıştır.

Gülbahar ve Kalelioğlu (2014) Scratch programının problem çözme becerileri üzerindeki etkilerini araştırmak ve öğrencilerin Scratch programlama dili hakkındaki görüşlerini incelemek üzere 49 ortaokul 5. sınıf öğrencisiyle bir çalışma gerçekleştirmiştir. Öğretmenlere öncelikle video konferans sayesinde Scratch ile ilgili bilgiler anlatılmış ve

öğrencilerine aktarılması istenmiştir. Daha sonra öğrenciler Scratch ortamıyla karşılaşmış ve çeşitli uygulamalar yapmaları sağlanmıştır. Öğrenciler Scratch ortamıyla karşılaştıktan sonra öğrencilerden nitel ve nicel veriler toplanmıştır. Bu araştırmaya göre Scratch programlama dilinin ortaokul düzeyinde problem çözme becerilerinde anlamlı bir farklılığa neden olmadığı, ancak öğrencilerin problem çözme yeteneğinde kendilerine olan güvenlerinde küçük bir artış olduğu ortaya çıkmıştır. Ayrıca öğrencilerin görüşleri göz önüne alındığında öğrencilerin programlamayı sevdikleri ve programlamayı geliştirmek istedikleri sonucuna ulaşılmıştır. Son olarak öğrencilerin Scratch programlama dilinin kullanımını kolay bulduğu sonucuna ortaya çıkmıştır.

Kukul ve Gökçearslan (2014) daha önce programlama eğitimi almamış ilköğretim öğrencilerinin Scratch programlama dili kullanarak problem çözme becerilerini incelemek amacıyla 5. ve 6. sınıflarda öğrenim gören 304 ortaokul öğrencisiyle bir çalışma gerçekleştirmiştir. Problem çözme becerileri envanteri kullanarak, Problem Çözme Becerisine Güven, Öz denetim ve Kaçınma alt faktörlerine göre incelenmiştir. Bu çalışmaya göre öğrencilerin problem çözme becerilerinin öz denetim faktörünün normal onun dışındaki faktörlerin yüksek olduğu görülmektedir. Problem çözme becerisinin cinsiyet, bilgisayara sahip olma durumu ve sınıf düzeyine göre anlamlı bir farklılık göstermediği görülmüştür.

Shin ve Park (2014) Scratch programlama ile programlama çalıştıktan sonra öğrencilerin problem çözme becerilerini ölçmek amacıyla Kore’de öğrenim gören 46 ilkokul öğrencisiyle çalışmayı gerçekleştirmiştir. Bu çalışmada problem çözme becerisi; *Farklı düşünme, Karar verme, Planlama yeteneği, Geribildirim, Pratik yapma yeteneği* alt boyutlarıyla incelemiştir. Araştırmanın sonucuna göre farklı düşünme, karar verme, geribildirim ve planlama yeteneği alt boyutlarına göre anlamlı bir fark olduğu görülmüştür. Ancak pratik yapma alt boyutunda anlamlı bir farklılık olmadığı görülmüştür. Sonuç olarak Scratch programlama dilinin bilişsel yükü azaltarak problem çözme becerisini geliştirdiği görülmüştür.

Ruf, Mühling ve Hubwieser (2014) Scratch ve Karel programlama dillerini programlama becerisi kazandırma ve motivasyonlarına olan etkilerini karşılaştırmak amacıyla Almanya’nın Bavyera eyaletinde, ortaokul 7. Sınıflarda öğrenim gören 56 öğrenciyle bir çalışma gerçekleştirmişlerdir. Öğrencileri 2 gruba ayırıp, ön-test son-test kontrol gruplu bir çalışma gerçekleştirmişlerdir. Çalışmanın sonuçlarına göre programlama öğrenimini Scratch ile gerçekleştiren öğrencilerin programlamaya ait başarı puanlarının Karel programını kullanan öğrencilerin puanlarına göre daha yüksek olduğu görülmüştür.

Ayrıca Scratch kullanan öğrencilerin motivasyonlarının da Karel kullanan öğrencilere oranla daha yüksek olduğu sonucuna ulaşılmıştır.

Kobsiripat (2014) Scratch programlama dilinin ilköğretim öğrencilerinin programlama becerileri ve yaratıcılıklarına olan etkilerini incelemek amacıyla Tayland'da öğrenim gören 60 ilköğretim öğrencisiyle bir çalışma yürütmüşlerdir. Bu 60 kişi ilköğretimin her seviyesinde öğrenim gören öğrenciler arasından seçilmiştir. Öğrencilere uygulanacak dersler 16 saat süresince uygulanmıştır. Çalışmanın sonuçlarına göre Scratch programının eğitsel bir araç olarak öğrenciler için kullanılabileceği sonucuna ulaşılmıştır. Ayrıca Scratch programı öğrencilerin isteklerini artırıcı bir eğitim aracı olduğu ve öğrencilerin yaratıcılıklarını geliştirebileceği sonucuna ulaşılmıştır.

Brown ve diğerleri (2013) ortaokul sınıflarında Scratch kullanarak bilgisayar destekli öğretimle problem çözme becerilerini öğrencilere öğretmek, öğrencilerin matematiksel akıl yürütme ve problem çözme yeteneklerine olan etkisini incelemek amacıyla ABD'nin Philadelphia eyaletinde öğrenim gören 113 ortaokul 5. ve 6.sınıf öğrencisiyle çalışmayı yürütmüşlerdir. Çalışmaya katılan öğrenciler daha önce programlama deneyimi yaşamamış düşük geliri iki farklı okuldan seçilmiştir. Çalışmada 3 deney grubu 1 kontrol grubu yer almıştır. Deney grubu öğrencilere bir ay boyunca her gün bir buçuk saat matematik eğitimi ve 45 dakika Scratch müfredatı öğretilmiştir. Çalışmanın sonuçlarına göre Scratch' a dayalı derslerin problem çözme becerileri öğrenen öğrencilere faydalı olabileceği sonucuna varılmıştır. Ayrıca Scratch tabanlı ders işleyen öğrencilerin matematik puanlarının Scratch kullanmayanlara göre daha yüksek olduğu görülmüştür. Scratch kullananların etkinliklere katılma isteklerinin daha fazla olduğu sonucuna ulaşılmıştır.

Zavala, Gallardo ve Garcia-Ruiz (2013) üçüncü sınıf öğrencilerinde Scratch programı kullanarak sayı duygusunun geliştirilmesi, sayıların anlamını ve ilişkilerini kavramsallaştırma yeteneklerini incelemek amacıyla Meksika'da öğrenim gören 49 ilkokul öğrencisiyle bir çalışma gerçekleştirmişler. Öğrenciler 27 araştırma grubu, 22 kontrol grubu olarak aynı yaş grubu içerisinde farklı ilkokullarda öğrenim görmektedir. Bu çalışma için Scratch 2.0 programını kullanarak etkileşimli bir oyun geliştirilmiştir. Öğrenciler bu oyunları oynarken iki hafta boyunca 30 dakikalık üç oturum şeklinde gözlemlenmiştir. Bu çalışmanın sonuçları incelendiğinde Scratch'ın kullanıcı dostu bir ortam sağladığı, ancak programlama dili yapısından dolayı uygulanırlığı sınırlandırdığı gözlemlenmiştir. Ayrıca bir web kamerasının kullanılması öğrencilerin etkileşimini arttıracak, oyunun bir parçası olmasına katkı sağlayacağı öngörülmüştür. Bu özellikler öğrencilerin eğlenceli bir ortamda öğrenmelerine katkı sağlayacağı sonucuna ulaşılmıştır.

Oh, Lee ve Kim (2013) Scratch programlama dili kullanarak geliştirilen STEAM (Bilim, Teknoloji, Mühendislik, Sanat ve Matematik) eğitimini ilköğretimde kullanmak amacıyla Kore’de öğrenim gören 50 ortaokul 6. Sınıf öğrencisiyle bu çalışmayı gerçekleştirmişlerdir. Bu çalışmada 50 öğrenci deney ve kontrol gruplarına eşit olacak şekilde seçilmiştir. Çalışma 10 hafta boyunca “Enerji ve Araçları” ve "Yanma ve Söndürme” konuları üzerinden gerçekleştirilmiştir. Bu çalışmanın sonuçları incelendiğinde Scratch tabanlı STEAM eğitimi alan öğrencilerin yaratıcılık indeksi puanlarını ve bilime karşı olan tutumlarının anlamlı düzeyde arttığı görülmüştür. Son olarak Scratch ile STEAM eğitimi alan öğrencilerin yaratıcılıklarının arttığı görülmüştür.

Fowler (2012) Kodu Game Lab programlama dilinin motivasyona olan etkisini araştırmak için 60 ortaokul öğrencisiyle bir çalışma gerçekleştirmiştir. Çalışma grubu üç farklı ülkedeki, Yeni Zelanda’dan 19, Birleşik Krallıktan 23 ve ABD’den 18 kişi olmak üzere oluşturulmuştur. Dersler gündelik okul müfredatı içerisinde işlenmiştir. Öğrencilerin Kodu Game Lab kullandıktan sonra görüşleri hem nitel olarak hem de nicel olarak alınmıştır. Öğrencilerin verdikleri yanıtlar doğrultusunda öğrencilerin genelinin oldukça mutlu olduğu görülmüştür. Öğrencilerin görevleri yapma isteklerinin arttığı, can sıkıntılarının ve hayal kırıklığı seviyelerinin azaldığı görülmüştür. Ancak bazı memnuniyetsizliklerde görüldü.

Osman, Loke, Zakaria ve Downe (2012) farklı programlama öğrenme araçlarının ve mükemmeliyetçiliğin ortaokul öğrencilerin motivasyonu ve başarısı üzerindeki etkilerini incelemek amacıyla Malezya’nın Perak bölgesinde öğrenim gören 591 ortaokul öğrencisiyle bir çalışma yürütmüşlerdir. Bu çalışmada Visual Basic Express, Scratch ve 2D-Programlanabilir PyGame olmak üzere 3 farklı programlama dili kullanılmıştır. Bu çalışmaya göre programlama dillerinin öğrencilerin motivasyon ve başarıları açısından farklılık gösterdiği görülmüştür. Scratch ve PyGame görsel açıdan daha zengin olmasından ötürü Visual Basic programlama diline göre motivasyonu arttırdığı ve bundan dolayı başarıyı arttırdığı sonucuna ulaşılmıştır.

Nam, Kim ve Lee (2010) Scratch programlama öğrenimini kolaylaştırmak ve problem çözme becerileri üzerindeki etkilerini incelemek amacıyla Kore’de öğrenim gören 60 ortaokul 6. sınıf öğrencisinin katılımıyla bir iskele temelli kurs oluşturulmuştur. Kurs dört hafta boyunca haftada 2 ders saati şeklinde düzenlenmiştir. Araştırmada 2 deney grubuna da Scratch programlama dili kullanılmıştır. Ancak deney grubunun biri iskele temelli olarak tasarlanmıştır. Bu çalışmada sonuç olarak her iki grupta da problem çözme becerilerinde belirgin bir iyileşme gösterdiği ve programlama öğrenmenin öğretim yöntemlerinden bağımsız olarak problem çözme becerileri üzerinde olumlu bir etkisi olduğu ve Scratch

programlama dilinin problem çözme becerileri üzerinde anlamlı bir etkisi olduğu görülmüştür. Ayrıca iskele temelli olarak Scratch programının kullanılmasının programlamada ilerlemeyi kolaylaştırdığı görülmüştür.

Lewis (2010) Scratch ve LOGO programlama dilini, ortaokul 6. Sınıf öğrencilerin tutum ve öğrenme çıktılarına göre karşılaştırmak amacıyla 50 ortaokul öğrencisiyle bu çalışmayı yürütmüşlerdir. Bu çalışmaya katılan öğrenciler 5. Sınıfı tamamlamış olup ileri seviye için tasarlanmış bir yaz okulu programına başvurmuşlardır. Kurs bilgisayarda müzik, film ve oyun yapmak amacıyla oluşturulmuş olup, 12 gün içerisinde toplam 36 saat süresince devam etmiştir. Çalışmada 2 deney grubu yer almıştır. Deney grubu 1 toplam 26 kişiden oluşmuş ve Scratch programlama dilini öğrenmiştir. Deney grubu 2 ise 24 kişiden oluşmakta ve LOGO programlama dilini öğrenmiştir. Her iki deney grubu da ilk 6 gün boyunca Scratch ve LOGO programlama diliyle karşılaşmamış ve temel becerileri edinmişlerdir. Bu çalışmanın sonuçları incelendiğinde LOGO ortamının öğrencilerin özgüvenlerini, programlamaya olan ilgisini ve koşul yapısını desteklediği görülmektedir. Scratch ortamının da koşul yapısını öğrenen öğrenciler üzerinde etkili olduğu görülmüştür. Ayrıca Scratch programının öğrenmesi ve yorumlanması daha kolay olduğu görülmüştür. Son olarak öğrencilerin LOGO ve Scratch programlarında döngü kavramı üzerinde benzer performans gösterdiği görülmüştür.

Taylor, Harlow, Forret (2010) Scratch programlama dili ile programlama öğretiminin matematiksel ve teknolojik düşünmeye olan etkisini araştırmak amacıyla Yeni Zelanda’da ortaokulda öğrenim gören 9 ve 10 yaşlarındaki 60 öğrenci ile bir çalışma yürütmüşlerdir. Öğrenciler iki sınıfa, sınıflardaki öğrenciler de küçük gruplara ayrılmıştır. Öğrenciler gruplarda bilgi birikimlerini ve çeşitli kaynaklarını birbirleriyle paylaşmış, öğrencilere sorumluluk alabilecek şekilde ortam oluşturulmuştur. Her sınıfta, bir etkileşimli akıllı tahta ve dört bilgisayar bulunmaktadır. İlk gruba etkileşimli akıllı tahta kullanarak Scratch programını tanıtılmıştır. Öğrenciler için bazı uygulamalar yapılmıştır. İlk hafta sonunda öğrenciler kendi başına Scratch’ta bazı alıştırmalar yapacak seviyeye gelmiştir. İkinci grupta da aynı başlangıç yapılması planlanmış ancak akıllı tahtadaki yaşanan teknik sorundan dolayı akıllı tahta kullanılmadan öğretmen bilgisayarsız programlama yapmaya karar verdi. Onun için öğrencilere sınıfta talimat vererek kendisinin robot gibi davranması için yardım istedi. Sonra yaptıkları işlemlerin matematikteki hangi konuları içerdiğini sordu. Her iki öğretmen de çocukların Scratch programını tanıdıklarından emin olmak istedi. Emin olduktan sonra öğrencilere akıllı tahta veya bilgisayar kullanarak oyun yapmaları istendi. Yapılan çalışmaların sonucunda Scratch’ın bir sınıf ortamında öğrencilere işbirliği için fırsat

sağladığı sonucuna ulaşılmıştır. Scratch programlama dilinin çocuklara problem çözme stratejilerini geliştirmelerine, yaratıcı düşünceler yapmalarına, sistematik sebepleri sistematik bir biçimde tartışmalarına ve işbirlikçi çalışmasına yardımcı oldukları görülmektedir. Scratch'ın matematiksel düşünce üzerine etki etmediği görülmektedir ancak tam anlamıyla araştırılmamıştır.

Maloney, Resnick, Rusk, Silverman ve Eastmond (2010) acemi programcıların görsel tabanlı programlama dili olan Scratch programı deneyimlerini incelemek amacıyla kente yaşayan 8 – 16 yaş arası öğrencilerle bir çalışma yapmıştır. Bu çalışma için öğrencilerin okuldan sonra programlama öğrenebilecekleri bir teknoloji merkez oluşturulmuştur. Öğrenciler 18 ay boyunca bu merkezde çalışmışlar ve toplamda Scratch'ta 536 adet proje üretmişlerdir. Oluşturulan 536 projenin 111'inde senaryo kurulmadan gerçekleştirilmiştir. Bu durum Scratch'ın medya manipülasyonu ve kompozisyon aracı olarak kullanılmadığını göstermektedir. Geriye kalan 425 proje de ise birden fazla blok ve blok yapısı, değişken, koşul gibi mantıksal yapıların kullanıldığı görülmektedir. Yapılan çalışmanın sonuçları incelendiğinde kent çocuklarının teknoloji merkezinde programlama sayesinde bir etkileşime girdiği görülmektedir. Projelerde gençlerin mutlak değer ve karakök gibi kavramları kullanmadığı görülmüştür. Ancak değişkenler ve rastgele sayıların çok fazla kullanıldığı görülmüştür. Ancak değişkenler kullanılmadan önce bir rehber tarafından yardımcı olunmasına ihtiyaç olduğu görülmektedir. Ayrıca Scratch programlama dilinin çoklu ortam yönünün programlamaya katılımı kolaylaştırdığı düşünülmektedir.

Alan yazında incelenen çalışmalar Tablo3'te özetlenerek gösterilmiştir.

Tablo 3 İlköğretim ve Ortaöğretimde Scratch ve Kodu Game Lab ile Yapılan Çalışmalar

Çalışma Adı	Yazarları	Türü	Yılı	Amacı	Çalışma Grubu	Yöntem	Bulgular
The Effects of Scratch Software on Students' Computational Thinking Skills	Eyüp Yünkül, Gürhan Durak, Serkan Çankaya, Zeynel Abidin Mısırlı	Makale	2017	Scratch eğitimi alan öğrencilerin bilgi işlemsel düşünce becerileri ölçeği ile Bilişim Teknolojileri ve Yazılım dersi puanları arasındaki ilişkiyi belirlemek ve Scratch eğitiminin bilgi işlemsel düşünce puanları üzerine etkisini araştırmak	69 Ortaokul 6. Sınıf Öğrencisi	Nicel	- Scratch ile programlama eğitiminin problem çözme, algoritmik düşünce ve yaratıcı düşünme üzerinde olumlu etkisi olduğu, - Scratch başarı puanları ile bilgi işlemsel düşünce puanları arasında yüksek bir ilişki olduğu, - Bilgi işlemsel düşünce becerileri yüksek olan öğrencilerin programlamada da başarılı olacağı ifade edilmiştir.
Scratch Programlama Dili Eğitime Yönelik Bir Mobil Uygulamanın Geliştirilmesi	Emre Yıldırım	Tez	2017	Scratch programlama dili kullanarak bir mobil uygulama geliştirerek öğrencilerin programlama becerilerinin geliştirilmesini sağlamak	94 Ortaokul 6. Sınıf Öğrencisi	Nicel	Programlama öğretimi için Scratch'ta hazırlanıp kullanılan mobil uygulamanın öğrencilerin programlama becerilerini geliştirdiği sonucuna ulaşılmıştır.
Scratch Programı Öğretiminde Ayrılıp Birleşme Tekniği Kullanımının Öğrencilerin Derse Yönelik	Sezin Yüksel	Tez	2017	Scratch programı öğretiminde ayrılıp birleştirme tekniği kullanımının öğrencilerin derse yönelik tutumuna, başarısına ve kalıcılığa olan etkisini	60 ortaokul 6. Sınıf öğrencisi	Nicel	- Scratch ile programlama öğretimini ayrılıp birleşme tekniği kullanılmasının ders başarısını arttırdığı, - Motivasyonu ve derse karşı olan tutumu arttırdığı, - Kalıcı öğrenmeler sağladığı ortaya

Tutumuna, Akademik Başarısına ve Kalıcılığa Etkisi				incelemek			çıkıştır.
Visual Programming Languages Integrated Across The Curriculum In Elementary School: A Two Year Case Study Using “Scratch” in Five Schools	Jose-Manuel Saez-Lopez, Marcos Roman- Gonzalez, Esteban Vazquez-Cano	Makale	2016	İlköğretim öğrencilerine görsel programlama dilini öğretiminin, projeler ve etkinlikler yoluyla yapılmasının faydalarını ve öğrenciler üzerindeki etkisini incelemek	107 Ortaokul 5. Ve 6. Sınıf öğrencisi	Nicel ve Nitel	• Programlama dili öğretiminde Scratch kullanılması mantık, hesaplamalı uygulamalar ve programlama üzerinde etkili olduğu, • Scratch ile projeler yapmanın eğlenceli olduğu, • Proje ve etkinlikler yoluyla programlama öğretiminin öğrenciler arası iletişimi, etkileşimi, aktif öğrenmeyi ve motivasyonu arttırdığı ortaya çıkmıştır.
Using Scratch and App Inventor for teaching introductory programming in secondary education. A case study	Stamatios Papadakis, Michail Kalogiannakis, Nicholas Zaranis, Vasileios Orfanakis	Makale	2016	Acemi programcılara programlama öğretiminde Scratch ve App Inventor programlama dillerini kullanmanın öğrencilerin performansına ve tutumuna olan etkisini araştırmak	87 Lise Öğrencisi	Nicel	• Programlama öğretimi Scratch ve App Inventor ile gerçekleştirmenin öğrencilerin tutumlarını ve başarılarını olumlu yönde etkilediği, • App Inventor'ın Scratch'a oranla etkili olduğu sonucuna ulaşılmıştır.
Introducing Programming To Elementary Students Novices By Using Game Development In	Monika Mladenović, Divna Krpan, Saša Mladenović	Makale	2016	Scratch ve Python programlama dillerinin programlama başarısı ve problem çözme becerilerine olan etkisini araştırmak	50 Ortaokul 5. Sınıf Öğrencisi	Nicel ve Nitel	• Problem çözme becerileri ile Scratch arasında birbirine bağlı bir ilişki olmadığı, • Problem çözme becerilerini cinsiyet ve okul durumu değişkenlerine göre farklılaşmadığı,

Python And Scratch							Scratch ve Python programlama puanları arasında anlamlı bir farklılaşma olmamasına rağmen zayıf öğrencilerin Scratch'ta daha başarılı olduğu ortaya çıkmıştır.
Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables	Ali Oluk, Özgen Korkmaz	Makale	2016	5. Sınıf öğrencilerin Scratch becerileri ile Bilgi işlemsel düşünce becerilerinin farklı değişkenler açısından karşılaştırmak	31 Ortaokul 5. Sınıf Öğrencisi	Nicel	- Scratch ve bilgi işlemsel düşünce puanlarının cinsiyet ve bilgisayar kullanım süresi değişkeni açısından farklılık göstermediği, - Scratch ile bilgi işlemsel düşünce puanları arasında anlamlı bir ilişki olduğu ifade edilmiştir.
Okuduğunu Anlama Becerilerinin Scratch Programı Aracılığıyla Geliştirilmesi	Erdal Papatğa	Tez	2016	Okumada sorun yaşayan ilkökul öğrencilerine Scratch programı aracılığıyla okuduğunu anlama becerilerinin nasıl geliştiğini ortaya koymak	8 İlkokul 4. Sınıf Öğrencisi	Nicel ve Nitel	- Scratch programıyla yapılan etkinliklerin okumayı anlama becerisini geliştirdiği, - İşbirlikli öğrenme ortamının okuduğunu anlama becerileri üzerinde geleneksel öğretim yöntemine göre daha etkili olduğu ortaya çıkmıştır.
Tasarım tabanlı öğrenme ortamında bilgisayar oyunu geliştirme etkinliğinin öğretim sürecinde kullanılması	Kevser Hava	Tez	2016	Kodu Game Lab ile bilgisayar oyunu geliştirme etkinliğini incelemek	Üstün Yetenekli 20 Lise Öğrencisi	Nitel Ve Nicel	Kodu Game Lab ile bilgisayar oyunu geliştirme etkinliğinin öğrencilerin problem çözme, üretim becerisi kazandırması açısından etkili olduğu sonucuna ulaşılmıştır.
Constructing The Scratchjr Programming	Dylan J. Portelance, Amanda L.	Makale	2015	Küçük çocukların, kendilerini özel bir programlama	62 Anaokul, İlkokul 1.ve 2. Sınıf	Nicel	Öğrencilerin en çok hareket bloklarını kullandığı ancak cinsiyete göre anlamlı farklılık oluşturmadığı,

Language in The Early Childhood Classroom	Strawhacker, Marina Umaschi Bers			müfredatıyla öğrendikten sonra kendi ScratchJr projelerinde hangi programlama bloklarını kullanmayı tercih ettiğini belirlemek	Öğrencisi		- Kontrol bloklarını en çok 2. sınıf öğrencilerinin kullandığı, - Yaşları büyük öğrencilerin daha karmaşık blokları kullandığı ortaya çıkmıştır.
Effects of an Integrated Scratch and Project-Based Learning Approach on the Learning Achievements of Gifted Students in Computer Courses	Hsiu-Ying Wang, Iwen Huang, Gwo-Jen Hwang	Makale	2014	Scratch ile bütünleştirilmiş proje senaryoları oluşturarak proje destekli öğrenme yoluyla problem çözme performansı, öğrenmeye ve Scratch karşı olan tutumu ve öğrenmeye karşı olan motivasyonunu araştırmak	91 Ortaokul Öğrencisi	Nicel ve Nitel	- Matematik becerileri yüksek ve ortalama olan öğrencilerin Scratch ile bütünleştirilmiş proje senaryoları oluşturarak proje destekli öğrenme yoluyla problem çözme becerileri, öğrenme motivasyonu, dersi alma yönündeki tutum üzerinde olumlu etki ettiği, - Öğrenmeye karşı motivasyonları arasında farklaşma olmadığı, - Öğrenmeye ve Scratch karşı olan tutumu arasında farklılaşma olmadığı ifade edilmiştir.
The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective	Filiz Kalelioğlu, Yasemin Gülbahar	Makale	2014	Scratch programının problem çözme becerileri üzerindeki etkilerini araştırmak ve öğrencilerin Scratch programlama dili hakkındaki görüşlerini incelemek	49 Ortaokul 5. Sınıf Öğrencisi	Nicel ve Nitel	- Scratch'ın programlamayı sevdirmede etkili olduğu, - Scratch'ın kullanımının kolay olduğu - Scratch'ın ortaokul düzeyinde problem çözme üzerinde etkili olmadığı, - Öğrencilerin problem çözme yeteneğinde kendilerine güvenmelerine yardımcı olduğu ortaya çıkmıştır.
A Study on the	Seungki Shin,	Makale	2014	Scratch programlama	46 İlkokul		Scratch'ın farklı düşünme, karar

Effect affecting Problem Solving Ability of Primary Students through	Phanwoo Park			ile programlama çalıştıktan sonra öğrencilerin problem çözme becerilerini ölçmek	Öğrencisi		verme, geribildirim ve planlama yeteneği üzerinde etkili olduğu - Scratch'ın problem çözme alt boyutu olan pratik yapma konusunda etkili olmadığı, - Scratch'ın problem çözme becerisini arttırdığı,
Scratch vs. Karel – Impact on Learning Outcomes and Motivation	Alexander Ruf, Andreas Mühling, Peter Hubwieser	Makale	2014	Scratch ve Karel programlama dillerini programlama becerisi kazandırma ve motivasyonlarına olan etkilerini karşılaştırmak	56 Ortaokul 7. Sınıf Öğrencisi	Nicel	- Scratch ile programlama eğitimi alan öğrencilerin Karel'e göre daha başarılı olduğu, - Scratch kullananların motivasyonlarının daha yüksek olduğu ortaya çıkmıştır.
Effects of the media to promote the Scratch programming capabilities creativity of elementary school students	Worarit Kobsiripat	Makale	2014	Scratch programlama dilinin ilköğretim öğrencilerinin programlama becerileri ve yaratıcılıklarına olan etkilerini incelemek	60 ilköğretimin her seviyesi	Nicel	- Scratch'ın ilköğretim seviyesine uygun bir araç olduğu, - Scratch'ın motivasyonlarını ve yaratıcılıklarını arttırdığı sonucuna ulaşılmıştır.
Designing Interactive Activities Within Scratch 2.0 For Improving Abilities To Identify Numerical Sequences	Luis Armería Zavala, Sara C. Hernández Gallardo, Miguel Ángel García-Ruíz	Makale	2013	Scratch programı kullanarak sayı duygusunun geliştirilmesi, sayıların anlamını ve ilişkilerini kavramsallaştırma yeteneklerini incelemek	49 İlkokul Öğrenci	Nicel	- Scratch'ın kullanımının kolay olmasıyla birlikte programın sınırlılıklarından ötürü uygulanırlılığını etkilediği, - Bir web kamera eklenerek etkileşimin arttırılacağı, - Scratch kullanımının öğrencilerin eğlenceli bir ortam oluşturacağı ifade edilmiştir.
Development and	JungCheol Oh,	Makale	2013	Scratch programlama	50 ortaokul	Nicel	Scratch tabanlı STEAM eğitimi alan

Application of STEAM Based Education Program Using Scratch: Focus on 6th Graders' Science in Elementary School	JiHwon Lee, JongHoon Kim			dili kullanarak geliştirilen STEAM eğitimini ilköğretimde kullanmak	6. Sınıf öğrencisi		öğrencilerin yaratıcılıklarının olumlu yönde etkilendiği, Scratch tabanlı STEAM eğitimi alanların bilime karşı tutumlarının anlamlı derecede artış gösterdiği sonucuna ulaşılmıştır.
Enriching Student Learning Programming Through Using Kodu	Allan Fowler	Makale	2012	Kodu Game Lab programlama dilinin motivasyona olan etkisini araştırmak	60 Ortaokul Öğrencisi	Nicel ve Nitel	- Kodu Game Lab ile programlama eğitimi almanın öğrencileri mutlu ettiği, - Can sıkıntıları ve hayal kırıklığını azalttığı, - Görevleri yapmada istekli oldukları ortaya çıkmıştır.
Secondary Students' Perfectionism and Their Response to Different Programming Learning Tools	Mohamad Amin Osman, Mohamed Nordin Zakaria, Siew Phaik Loke, Alan G. Downe	Makale	2012	Farklı programlama öğrenme araçlarının ve mükemmeliyetçiliğin ortaokul öğrencilerin motivasyonu ve başarıları üzerindeki etkilerini incelemek	591 Ortaokul Öğrencisi	Nicel	- Görsel programlama dillerinin metin tabanlı programlama dillerine göre motivasyonu arttırdığı, - Scratch'ın başarıyı anlamlı derecede arttığı sonucuna ulaşılmıştır.
Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some	Merilyn Taylor, Ann Harlow, Michael Forret	Makale	2010	Scratch programlama dili ile programlama öğretiminin matematiksel ve teknolojik düşünmeye olan etkisini araştırmak	9 ve 10 yaşındaki toplam 60 öğrenci	Nicel ve Nitel	- Scratch'ın problem çözme becerilerini geliştirmede olumlu katkı sağladığı, - Yaratıcı ve sistematik düşünmeyi geliştirdiği, - Scratch'ın etkileşimi arttırdığı ve işbirlikli öğrenmeye yardımcı olduğu,

Mathematical Thinking							• Matematiksel düşünme üzerinde anlamlı bir etkisi olmadığı ifade edilmiştir.
The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill	Dongsoo NAM, Yungsik KIM, Taewook LEE	Makale	2010	Scratch programlama öğrenimini kolaylaştırmak ve problem çözme becerileri üzerindeki etkilerini incelemek	60 Ortaokul 6. Sınıf Öğrencisi	Nicel	• Scratch ile problem çözme becerisinde gelişim sağlandığı, • Programlama öğrenmenin problem çözme üzerinde etkili olduğu, • İskele temelli Scratch öğretiminin programlama öğretimini kolaylaştırdığı görülmüştür.
How Programming Environment Shapes Perception, Learning and Goals: Logo vs. Scratch	Colleen M. Lewis	Makale	2010	Scratch ve LOGO programlama dilini, ortaokul 6. Sınıf öğrencilerin tutum ve öğrenme çıktılarına göre karşılaştırmak	50 Ortaokul 6. Sınıf Öğrencisi	Nicel	• LOGO programının özgüven ve programlamama olan ilgisini desteklediğini, • Döngü ve koşul kavramları üzerinde Scratch ve Logo programının etkili olduğu, • Scratch programının kolay öğrenilebilir bir programlama dili olduğu,
The Scratch Programming Language and Environment	John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, Evelyn Eastmond	Makale	2010	Acemi programcıların görsel tabanlı programlama dili olan Scratch programı deneyimlerini incelemek	8 – 16 yaş arası öğrenciler	Nicel	• Scratch'taki projelerde öğrencilerin mutlak değer ve karekök ifadelerini kullanmadığı, • Değişkenler ve rastgele sayıların çok fazla kullanıldığı • Scratch ile proje oluşturma sürecinde öğrencilerin etkileşiminin arttığı, • Scratch'ın çoklu ortam özelliğinden programlamaya katılımını kolaylaştırdığı ortaya çıkmıştır.

2.5.3. Alan Yazın Özet

Alan Yazın Özet

Çalışmanın bu kısmında ilköğretim, ortaöğretim, yüksek öğretimde ve öğretmen eğitiminde Scratch ve Kodu Game Lab programlama diline ait son yıllara ait yayınlanan makale ve tez çalışmaları incelenmiştir. Çalışmalar temel olarak İlk-ortaöğretim; yükseköğretim ve öğretmen eğitimi olarak iki kısma ayrılmış olup ayrı ayrı incelenip tablolastırılmıştır (Tablo1 ve Tablo2).

Yükseköğretim ve öğretmen eğitiminde yapılan Scratch ve Kodu Game Lab programlama diline ait yapılan alan yazın incelendiğinde Scratch programlama dili ile programlamanın daha eğlenceli olduğu (Fidan, 2016; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016) ve programlamaya yönelik kaygı düzeyini olumlu yönde etkilediği (Demir, 2015; Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011) ortaya çıkarılmıştır. Ayrıca Scratch'ın kullanımının (Genç ve Karaguş, 2011; Fesakis ve Serafeim, 2009; Malan ve Leitner, 2007; Yükseltürk ve Altıok, 2016) ve öğretiminin kolay olarak görüldüğü (Genç ve Karaguş, 2011; Malan ve Leitner, 2007; Yıldırım, 2016) ve yüksek öğrenim öğrencilerinin Scratch'ı ilköğretim seviyesine uygun bulduğu (Baltalı, 2016; Yıldırım, 2016;) görülmüştür.

Öğrencilerin Scratch hakkında olumlu bir görüş geliştirdiği (Baltalı, 2016; Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011; Malan ve Leitner, 2007; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016), öğrencilerin programlamaya katılma isteklerini (Fidan, 2016; Fesakis ve Serafeim, 2009; Malan ve Leitner, 2007; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016) ve programlama öğrenmeye karşı olan motivasyonlarını (Fidan, 2016; Erol, 2015; Genç ve Karaguş, 2011) arttırdığı ortaya çıkarılmıştır.

Scratch programlama ile programlama öğretiminin (Demir, 2015; Erol, 2015; Fesakis ve Serafeim, 2009) ve oyunlaştırmanın (Yükseltürk ve Altıok, 2016; Yıldırım, 2016) öğrenci başarısını arttırdığı ortaya çıkarılmıştır. Ayrıca Scratch ile programlama dili öğrenen öğrencilerin bilgi işlemsel düşünme, yaratıcılık (Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016), problem çözme, iletişim kurma ve takım çalışması gibi özellikler geliştirdiği (Yıldırım, 2016) ortaya çıkarılmıştır. Ancak Scratch programlama dilinin ileri derece kodlama yapanlar için uygun olmadığını (Genç ve Karaguş, 2011), Scratch kod yapısının sınırlı olmasından ötürü sadece giriş derslerinde öğretilmesi gerektiği görüşü (Malan ve Leitner, 2007) ifade edilmiştir. Ayrıca alan yazın incelendiğinde yükseköğretim

ve öğretmen eğitimi programlama öğretiminde, Scratch programlama dilinin öğrencilerin öz-yeterlik algıları üzerine az sayıda çalışmaya (Yükseltürk ve Altıok, 2016) rastlanmıştır. Yükseköğretim ve öğretmen eğitiminde Kodu Game Lab ile ilgili alan yazında yapılan çalışmalar incelendiğinde çok sayıda çalışma bulunmadığı ve bulunan çalışmanın Kodu Game Lab programlama dilinin tasarım ve kullanışlı bir programlama dili olduğu sonucuna ulaşılmıştır (Baltalı, 2016). Ancak Kodu Game Lab ile programlama eğitiminde başarı, tutum ve öz-yeterlilik değişkeni açısından herhangi bir çalışmaya rastlanılmadığı görülmektedir.

İlk ve ortaöğretimde yapılan Scratch ve Kodu Game Lab programlama diline ait yapılan alan yazın incelendiğinde Scratch programlama dili ile bilgi işlemsel düşünce becerileri arasında pozitif yönde anlamlı bir ilişki olduğu (Oluk ve Korkmaz, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017) ve bilgi işlemsel düşünce becerileri yüksek olan öğrencilerin programlama öğreniminde başarılı olacağı (Yünkül, Durak, Çankaya ve Mısırlı, 2017) ortaya çıkmıştır. Scratch programlama dili ile programlama öğretimi ve bilgi işlemsel düşünce becerilerinin alt boyutlarından biri olan problem çözme becerileri üzerinde anlamlı derecede etkili olduğu (Shin ve Park, 2014; Nam, Kim ve Lee, 2010; Yünkül, Durak, Çankaya ve Mısırlı, 2017) ve olumlu katkı sağladığı (Taylor, Harlow ve Forret, 2010) ifade edilmiştir. Ancak Scratch'ın problem çözme becerileri üzerinde etkili olmadığı konusunda (Gülbahar ve Kalelioğlu, 2014; Mladenović, Krpan ve Mladenović, 2016) çalışmaları olduğu görülmüştür.

Scratch programlama dilinin algoritmik düşünme (Oluk ve Korkmaz, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017) ve yaratıcı düşünme üzerinde anlamlı bir etkisi olduğu (Kobsiripat, 2014; Oh, Lee ve Kim, 2013; Taylor, Harlow ve Forret, 2010; Yünkül, Durak, Çankaya ve Mısırlı, 2017) ortaya çıkmıştır. Ayrıca Scratch'ın işbirlikli öğrenme ortamı sağladığı (Papatğa, 2016; Taylor, Harlow ve Forret, 2010) ve öğrencilerin etkileşime girmelerine yardımcı olduğu (Lopez, Gonzalez ve Cano, 2016; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Zavala, Gallardo ve Ruíz, 2013;) ifade edilmiştir.

Programlama eğitiminin Scratch ile yapılmasının derse olan tutumu olumlu yönde etkilediği (Gülbahar ve Kalelioğlu, 2014; Oh, Kim ve Lee, 2013; Papadakis, Kalogiannakis, Zaranis ve Orfanakis, 2016; Wang, Huang ve Hwang, 2014; Yüksel, 2017) ortaya çıkmıştır. Ancak Scratch'a olan tutumları üzerinde bir etkisi olmadığı (Wang, Huang ve Hwang, 2014) sonucuna ulaşılmıştır.

Scratch ile öğrencilerin programlama öğrenmesi ortamın eğlenceli olmasına katkı sağladığı (Gallardo ve Ruíz, 2013; Lopez, Gonzalez ve Cano, 2016) ortaya çıkmıştır. Ayrıca Scratch'ın programlamaya olan isteklerinin arttığı (Gülbahar ve Kalelioğlu, 2014; Kobsiripat, 2014), öğrenciler üzerinde heyecan, bağlılık gibi duyguları olumlu yönde etkilediği (Lopez, Gonzalez ve Cano, 2016) belirtilmiştir. Scratch'ın programlama öğretiminde isteği etkilemekle birlikte motivasyon üzerinde anlamlı derecede etkili olduğu ve motivasyonu arttırdığı (Lopez, Gonzalez ve Cano, 2016; Osman, Zakaria, Loke ve Downe, 2012; Ruf, Mühling ve Hubwieser, 2014) ortaya çıkmıştır. Ancak Gülbahar ve Kalelioğlu (2014) yaptıkları bir çalışmada Scratch'ın programlama dili öğretiminde öğrenmeye olan motivasyonları üzerinde bir etkisi olmadığını belirtmişlerdir.

Scratch'ın kullanımının kolay olduğu (Gülbahar ve Kalelioğlu, 2014), programlamayı kolaylaştırdığı (Lewis, 2010; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Nam, Kim ve Lee, 2010) ve öğrenci başarısını anlamlı derecede olumlu yönde etkilediği (Osman, Zakaria, Loke ve Downe, 2012; Papadakis, Kalogiannakis, Zaranis ve Orfanakis, 2016; Ruf, Mühling ve Hubwieser, 2014; Yıldırım, 2017) yapılan çalışmalarda ortaya çıkarılmıştır. Ayrıca Yüksel'in (2017) yaptığı bir çalışmada Scratch ile programlama eğitiminin ayrılıp birleşme tekniğiyle birlikte kullanılmasının da başarıyı ve kalıcı öğrenmeyi anlamlı derecede arttırdığı ifade edilmiştir.

Scratch ile programlama öğretimine ilişkin yapılan çalışmalar incelendiğinde öz-yeterlik alanında bir çalışmaya rastlanmadığı görülmüştür. Ancak Gülbahar ve Kalelioğlu (2014) yaptıkları bir çalışmada öğrencilerin kendilerine olan öz güvenlerinde küçük bir artış olduğu görülmüştür.

Kodu Game Lab ile ilk ve ortaöğretim seviyesinde yapılan çalışmalarının sayılarının az olduğu ve deneysel düzeyde iki çalışmaya rastlanmıştır (Fowler, 2012; Hava, 2016). Yapılan çalışmalar incelendiğinde Kodu Game Lab ile programlama eğitimi alan öğrencilerin mutlu oldukları, can sıkıntılarının azaldığı, hayal kırıklığına uğrama seviyesinin azaldığı ve görevleri yapma isteklerini arttığı (Fowler, 2012) ortaya çıkmıştır. Ayrıca yapılan diğer çalışmada Kodu Game Lab ile oyun geliştirme bakımından faydalı olduğu, öğrencilerin problem çözme üretim becerileri kazandırma açısından olumlu etki sağladığı (Hava, 2016) ifade edilmiştir.

Alan yazın incelendiğinde ilk, ortaöğretim, yükseköğretim ve öğretmen eğitiminde Scratch ve Kodu Game Lab programlama dilinin başarı üzerindeki etkisi üzerine yapılan çalışmalar incelendiğinde (Demir, 2015; Erol, 2015; Fesakis ve Serafeim, 2009; Osman,

Zakaria, Loke ve Downe, 2012 Papadakis, Kalogiannakis, Zaranis ve Orfanakis, 2016; Ruf, Mühling ve Hubwieser, 2014; Yüksel, 2017; Yıldırım, 2017), Scratch ile programlamanın başarıyı olumlu etkilediği ortaya çıkarılmıştır. Ayrıca Scratch programlama dilinin bilgi işlemsel düşünce becerilerine olan etkisi incelendiğinde, Scratch'ın bilgi işlemsel düşünce becerileri üzerinde olumlu bir etkiye sahip olduğu (Ozoran, Çağıltay ve Topallı, 2012; Yünkül, Durak, Çankaya ve Mısırlı, 2017; Yıldırım, 2016), Scratch'ın problem çözme üzerine yapılan çalışmalar incelendiğinde olumlu etki gösteren çalışmalar olduğu gibi (Yıldırım, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017; Shin ve Park, 2014; Nam, Kim ve Lee, 2010; Taylor, Harlow ve Forret, 2010), olumlu etki göstermeyen (Gülbahar ve Kalelioğlu, 2014; Mladenović, Krpan ve Mladenović, 2016) çalışmaların da olduğu görülmüştür.

Scratch programlama dilinin algoritmik düşünme (Oluk ve Korkmaz, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017) ve yaratıcılık üzerinde anlamlı bir etkisi olduğu (Kobsiripat, 2014; Oh, Lee ve Kim, 2013; Ozoran, Çağıltay ve Topallı, 2012; Taylor, Harlow ve Forret, 2010; Yıldırım, 2016 Yünkül, Durak, Çankaya ve Mısırlı, 2017) ortaya çıkmıştır.

Motivasyon ile ilgili çalışmalar incelendiğinde (Erol, 2015; Fidan, 2016; Fowler, 2012; Genç ve Karaguş, 2011; Gülbahar ve Kalelioğlu, 2014; Lopez, Gonzalez ve Cano, 2016; Ruf, Mühling ve Hubwieser, 2014; Osman, Zakaria, Loke ve Downe, 2012) Gülbahar ve Kalelioğlu (2014) dışındaki çalışmaların öğrencilerin motivasyonunu olumlu yönde etkilediği belirtilmiştir. Ayrıca incelenen çalışmalarda Scratch' ın kullanımının kolay bulunduğu (Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011; Gülbahar ve Kalelioğlu, 2014; Yükseltürk ve Altıok, 2016; Malan ve Leitner, 2007) ve öğretiminin kolay olarak görüldüğü (Genç ve Karaguş, 2011; Malan ve Leitner, 2007; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Nam, Kim ve Lee, 2010; Lewis, 2010; Yıldırım, 2016;) belirtilmiştir.

Scratch öğreniminin eğlenceli olduğu (Fidan, 2016; Gallardo ve Ruíz, 2013; Lopez, Gonzalez ve Cano, 2016; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016), öğrencilerin programlama öğrenme isteklerini olumlu etkilediğini (Fidan, 2016; Fesakis ve Serafeim, 2009; Gülbahar ve Kalelioğlu, 2014; Kobsiripat, 2014; Malan ve Leitner, 2007; Ozoran, Çağıltay ve Topallı, 2012; Yıldırım, 2016), kaygı durumlarını azalttığını (Demir, 2015; Fowler, 2012; Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011;) ve olumlu tutum geliştirdiğini (Baltalı, 2016; Ozoran, Çağıltay ve Topallı, 2012; Genç ve Karaguş, 2011; Fesakis ve Serafeim, 2009; Malan ve Leitner, 2007; Yıldırım, 2016) belirtmişlerdir

Sonuç olarak alan yazın incelendiğinde, Scratch ve Kodu Game Lab ile programlama öğretiminde ortaokul öğrencilerinin programlamaya ilişkin öz-yeterlik algılarına ilişkin deneysel çalışmalara rastlanılmadığı görülmüştür. Ayrıca Kodu Game Lab ile akademik başarı üzerine herhangi bir çalışmaya rastlanılmadığı, tutum değişkeni üzerine de az sayıda çalışma yapıldığı görülmektedir. Scratch ve Kodu Game Lab programlama dilleriyle birlikte programlama öğretiminin gerçekleştirildiği bir çalışma da bulunmamaktadır. Scratch ve Kodu Game Lab ile programlama öğretimi ile ilgili çalışmalar incelendiğinde, akademik başarı, tutum ve öz-yeterlik algıları değişkenlerinin birlikte kullanıldığı bir çalışmaya da rastlanılmamıştır. Bu çalışmanın bu açıdan literatüre katkı yapması umulmaktadır.



BÖLÜM III

3. YÖNTEM

3.1.1.Araştırma Modeli

Scratch ve Kodu Game Lab Programlama dilleri kullanımının öğrencilerin programlama akademik başarılarına, programlamaya yönelik tutumlarına ve öz-yeterlik algılarına etkisini inceleyen bu çalışmada, nicel araştırma yöntemlerinden yararlanılmıştır. Bu çalışmada, ön test- son test kontrol gruplu yarı-deneysel desen kullanılmıştır. Yarı deneysel desen yansız atamanın kullanılmadığı durumlarda kullanılan araştırma desendir (Büyüköztürk, Çakmak, Akgün, Karadeniz ve Demirel, 2008). Deneysel desenler, değişkenler arasındaki neden-sonuç ilişkisini ortaya koyan araştırma alanıdır (Karasar, 2005). Deneysel desenler, araştırmacının bağımsız değişken üzerinde müdahalede bulunmasının sonucu, bağımlı değişkene ilişkin sonuçların karşılaştırıldığı araştırma deseni olarak ifade edilmektedir (Büyüköztürk, Çakmak, Akgün, Karadeniz ve Demirel, 2008). Araştırmada kullanılan yarı-deneysel desenin görselleştirilmiş hali Tablo 4 'te verilmiştir.

Tablo 4
Uygulanan Yarı-Deneysel Desenin Görselleştirilmiş Hali

Grup	Ön-test	Programlama Dili	Son-test	Tekrar-test
Deneysel Grup 1:	PDABT		PDABT	
	EBODKÖYTÖ	Scratch	EBODKÖYTÖ	PDABT
	PİÖYÖ		PİÖYÖ	
Deneysel Grup 2:	PDABT		PDABT	
	EBODKÖYTÖ	Kodu Game Lab	EBODKÖYTÖ	PDABT
	PİÖYÖ		PİÖYÖ	

- * Programlama Dilleri Akademik Başarı Testi (PDABT)
- * Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği(EBODKÖYTÖ)
- * Programlamaya İlişkin Öz-Yeterlik Ölçeği (PİÖYÖ)

Programlama öğretiminin planlanması ve programlama için gerekli ihtiyaçların belirlenmesi için bir ön çalışma gerçekleştirilmiştir. Yapılan bu ön çalışmada alanında uzman Bilişim Teknolojileri Öğretmenlerinin görüşlerinden faydalanılmıştır. Yapılan bu ön çalışma, aşağıdaki başlıklar altında açıklanmıştır.

3.1.2.İhtiyaç Analizi: Bilişim Teknolojileri Öğretmenlerinin Programlama Öğretimi İle İlgili Görüşlerinin Belirlenmesi

3.1.3.Amaç

Bu tez çalışmasında ilk aşamada,6.sınıf seviyesinde ortaokul öğrencilerine Scratch ve Kodu Game Lab’da “programlama öğretimi” geliştirmek için, mevcut durum ve ulaşılacak istenen hedef arasındaki farkı değerlendirmek amacıyla ihtiyaç analizi yürütülmüştür. Bu kapsamda, ders içeriğinin ve ders planlarının oluşturmaya katkı sağlamak amacıyla alanında uzman Bilişim Teknolojileri(BT) öğretmenlerinin görüşlerinden yararlanılmıştır.

3.1.4.Yöntem

Bu aşamada mülakat formunun nasıl hazırlandığını ve nasıl uygulandığı gösterilmektedir.

3.1.4.1. Mülakat Formu Hazırlama Süreci

Bu çalışmada 6.sınıf seviyesinden öğrencilere Scratch ve Kodu Gam Lab’ da programlama öğretimi geliştirmek için ders içeriğinin ve ders planlarının oluşturmaya için BT öğretmenlerinin görüşleri alınması kararlaştırılmıştır. Bu kapsamda yarı yapılandırılmış sekiz sorudan oluşan “Öğretmen Gözünden Programlama Öğretimi” mülakat formu geliştirilmiştir. Mülakat formunun geliştirilmesinde uzman görüşü alınmıştır. Uzman görüşleri doğrultusunda düzeltilen mülakat formu EK3’te yer almaktadır.

Bu mülakat formu deneme soruları araştırmacı tarafından yazılmıştır. Daha sonra bu deneme soruları üzerinde programlama eğitimi vermiş alanında uzman iki BT öğretmeni ve bir BÖTE öğretim üyesinin görüşleri alınmıştır. Görüşler doğrultusunda form üzerinde şu şekilde değişiklikler yapılmıştır;

- Üçüncü maddedeki “Sizce programlama nasıl öğretilmelidir? Ne kadar süre harcanmalı?” sorusu yerine “Sizce programlama nasıl öğretilmelidir? Hangi öğretim yöntem ve teknikler kullanılmalıdır?” sorusunun yazılmasının daha uygun olduğu

kararlařtırılmıřtır. Ayrıca üçüncü sorudaki “Ne kadar süre harcanmalı?” kısmının ayrı bir soru olarak yer alması uygun bulunmuřtur.

- Beřinci soruda “Hangi programlama dilini kullanıyorsunuz. Kullandığınız programlama dilinin avantajları nelerdir? “ yerine “Hangi programlama dilini öğretiyorsunuz? Öğrettiğiniz programlama dilini seçme sebebiniz nedir?” uygun görülerek deęiřtirilmiřtir.
- Mülakat formunda olmayan ancak öğretmenlerin görüşleri doęrultusunda “Programlama öğretimi sırasında öğrencilerin programlama öğrenmeye karşı tutum ve davranıřlarını gözlemlerinizi doęrultusunda nasıl deęerlendirirsiniz?” sorusu altıncı soru olarak eklenmiřtir.
- Sekizinci soru olarak “Programlama öğretirken nasıl örneklerle dersi iřliyorsunuz? Programlama öğretiminde kullandığınız örnekleri, projeleri, ek bir dosya olarak ulařtırabilir misiniz?” yerine “Scrach ve Kodu Game programlama dilini kullandınız mı? Kullandıysanız ders anlatımınızda hangi konuları iřlediniz, hangi örnekleri verdiniz?” sorusu daha uygun görülerek deęiřtirilmiřtir.
- Görüşler doęrultusunda arařtırmaya katılan öğretmenlerin demografik özelliklerinden; yař, cinsiyet, eğitim durumu, kaç yıldır öğretmenlik yaptığı, çalıştığı kurumda bilgisayar laboratuvarına sahip olma durumu, kaç yıldır programlama eğitimi verdięi, programlama eğitimini hangi yazılımlar kullanarak öğrettięi, hangi sınıf düzeyinde programlama eğitimi verdięi ve kaç hafta süresince programa eğitimi iřlediklerini sorularının uygun olduęu görölmüřtür.

3.1.4.2. Mülakat Formunun Uygulanması

Biliřim Teknolojileri Öğretmenlerinin programlama öğretirken nelere dikkat ettikleri ve programlamanın nasıl öğretilmesi konusunda düşüncelerine iliřkin Milli Eğitim Bakanlığında görev yapan ve daha önce programlama eğitimi vermiř ve arařtırmaya gönüllü olarak katılmıř alanında uzman Biliřim Teknolojileri öğretmenlerine Google Form yardımıyla gönderilmiřtir. Öğretmenlerden her soruyu cevaplamaları istemiřtir. Gönderilen mülakat formunu toplam 33 öğretmen yanıtlamıřtır. Ancak dört öğretmen tüm soruları

yanıtlamadığı için verileri kullanılmamıştır. Araştırmada sonuç olarak toplam 29 öğretmenin görüşlerinden yararlanılmıştır.

3.1.4.3. İhtiyaç Analizine İlişkin Bulgular

3.1.4.3.1. Çalışma Grubuna

İhtiyaç analizine 29 bilişim teknolojileri öğretmenleri katılmıştır. Çalışma grubunun demografik özellikleri cinsiyet, yaş, eğitim durumu, görev süreleri, bilgisayar sınıfına sahip olma durumu, hangi programlama dili tercihi ve dönem içerisinde programlama dilleri öğretme süresi şu şekildedir:

Tablo 5 Öğretmenlerin Cinsiyetine Ait Bilgiler
Öğretmenlerin Cinsiyetine Ait Bilgiler

	Cinsiyet		
	Kadın	Erkek	Toplam
Frekansı	14	15	29
Yüzdesi	48,3	51,7	100
Kümülatif Yüzdesi	48,3	100	

Tablo 6
Öğretmenlerin Yaşına Ait Bilgiler

	Yaş		
	21-27	28 ve üzeri	Toplam
Frekansı	18	11	29
Yüzdesi	62,1	37,9	100
Kümülatif Yüzdesi	62,1	100	

Tablo 7

Öğretmenlerin Eğitim Durumlarına Ait Bilgiler

	Eğitim Durumu		
	Lisans	Yüksek Lisans	Toplam
Frekansı	25	4	29
Yüzdesi	86,2	13,8	100
Kümülatif Yüzdesi	86,2	100	

Tablo 8

Öğretmenlerin Görev Sürelerine Ait Bilgiler

	Görev Süresi				Toplam
	1-3 yıl	4-6 yıl	7-10 yıl	10 yıl ve üzeri	
Frekansı	22	3	2	2	29
Yüzdesi	75,9	10,3	6,9	6,9	100
Kümülatif Yüzdesi	75,9	86,2	93,1	100	

Tablo 9

Öğretmenlerin Bilgisayar Sınıfına Sahiplik Durumuna Ait Bilgiler

	Bilgisayar Sınıfına Sahiplik Durumunu		
	Evet	Hayır	Toplam
Frekansı	25	4	29
Yüzdesi	86,2	13,8	100
Kümülatif Yüzdesi	86,2	100	

3.1.4.3.2. Programlama Öğretimine İlişkin Bulgular

Öğretmenlerin kullandığı programlama dilleri, dönem içerisinde programlama öğretme süreleri ve programlama anlattıkları sınıf seviyesine ait veriler şu şekildedir;

Tablo 10

Öğretmenlerin Programlama Anlattığı Sınıf Seviyesi

	Sınıf Düzeyi				
	5.Sınıf	6.Sınıf	7.Sınıf	8.Sınıf	Toplam
Frekansı	7	18	2	2	29
Yüzdesi	24,1	62,1	6,9	6,9	100
Kümülatif Yüzdesi	24,1	86,2	93,1	100	

Tablo 11

Öğretmenlerin Programlama Dilleri Öğretirken Kullandığı Programlara Ait Bilgiler

	Kullanılan Programlama Dilleri					Toplam
	Scratch	Code.org	Kodu Game Lab	Code Academy	Diğer	
Frekansı	24	16	6	6	7	59
Yüzdesi	40,7	27,1	10,2	10,2	11,9	100
Kümülatif Yüzdesi	40,7	67,8	77,9	88,1	100	

Tablo 12

Öğretmenlerin Ne Kadar Süreyle Programlama Anlattığına Ait Bilgiler

	Programlama Dilleri Öğretim Süresi				Toplam
	1-3 hafta	4-6 hafta	7-10 hafta	10 hafta ve üzeri	
Frekansı	5	5	7	12	29
Yüzdesi	17,2	17,2	24,1	41,4	100
Kümülatif Yüzdesi	17,2	34,4	68,6	100	

3.1.4.3.3. Katılımcıların Görüşleri

Öğretmenlerden elde edilen veriler içerik analizi kullanılarak elde edilmiştir. İçerik analizinin kullanılmasının sebebi gerçekleri olduğu gibi ortaya çıkarmaktır. Bunun için içerik analizinde birbirine benzeyen veriler birleştirilerek temalar altında toplamak ve anlaşılacak şekilde yorumlanması sağlanması gerekmektedir (Yıldırım ve Şimşek, 2011). Temalar belirlendikten sonra kodlar ortaya çıkarılmıştır.

Öğretmenlere uygulanan mülakat formunda öğretmenlerin verdikleri cevapların frekansları ve yaptıkları bazı yorumlar aşağıdaki tablolarda gösterilmektedir.

Tablo 13

Programlama Öğretimi Öğretme Zamanı

Tema/Kodlar	Sıklık (f)
Sınıf Düzeyi	
Ana Sınıfı	4
1- 3. Sınıf	10
4- 8. Sınıf	15
Toplam	29

Öğretmen4: “Anaokulu seviyesinde başlanabilir. 5 yaşındaki oğlum Light-bot'un temel seviyesini ve Code.org ilk seviyeyi yardımsız bitirebiliyor. Ama Scratch gibi diğer uygulamalarda okumayı öğrenmiş olması gerekiyor.”

Öğretmen7: “4. sınıftan itibaren öğretilmeye başlanmalı, çünkü programlama yabancı dil gibidir. Belli bir yaşa kadar bu eğitim alınmazsa ileride açığı kapatmak zor olacaktır.”

Öğretmen22: “İlkokuldan itibaren. Çünkü programlama problem çözme yeteneği sağlar alternatif düşünme kapasitesini yükseltir. Bu yaşlar kritik yaşlar.”

Öğretmen17: “6. Sınıftan itibaren. Çünkü daha erken kavratmak biraz daha zor olabiliyor köy okullarında özellikle. Merkezi okullarda daha rahat şekilde anlatılabilir; ama daha küçük sınıflara.”

Tablo 14

Programlama Öğretiminde Gerekli Olan Ön Bilgiler

Tema/Kodlar	Sıklık (f)
Programlama Ön Bilgiler	
Algoritma	12
Problem Çözme Yolları	6
Matematik	5
Bilgisayar Okur Yazarlığı	4
Akış Şeması	2
Toplam	29

Öğretmen15: "Problem çözme konuları işletilmeli problem nasıl çözülür, yabancı dilde nasıl cümle kurulur bunlardan bahsedilmeli. Çocuklar bir sistemin nasıl yürüdüğünü görebilmeli. Tersine mühendislik yapabilmeli"

Öğretmen24: "Algoritma örnekleriyle başlanmalı. Ders eğlenceli şekle getirilerek çocuklara sunulmalı."

Öğretmen6: "Öğrencilerin dört işlemi bilmesi iyi olacaktır programlama öğretimi için iyi olacaktır."

Tablo 15

Programlama Nasıl Öğretilmeli

Tema/Kodlar	Sıklık (f)
Programlama Öğretim Şekli	
Bilgisayarda Uygulama	18
Eğitici Oyun	6
Bilgisayarsız Aktivite	5
Toplam	29

Öğretmen26: "Bilişim laboratuvarlarının eksikliği maalesef aşikâr bir durum. Ancak yine de programlama uygulamalı öğretilmeli çıkan sonuçlardan alınan dönütler öğrencinin derse ilgisini arttırıp bir şeyler üretme isteğini ortaya çıkaracaktır."

Öğretmen1: "Lego gibi 3 boyutlu eğitim materyalleriyle tasvir edilerek, dil öğrenir gibi şemalarla uygulama yapılarak öğretilmelidir."

Tablo 16

Programlama Öğretmek İçin Harcanması Gereken Süre

Tema/Kodlar	Sıklık (f)
Programlama Öğretim Süresi	
1-6 hafta	4
6-12 hafta	16
13 hafta ve üzeri	9
Toplam	29

Öğretmen20: "Programlama bana göre çok geniş ama bu yolun varlığını göstermek 8 haftalık bir eğitim sürecini alabilir"

Öğretmen5: "Öğretim verilen öğrencilerin sosyo-ekonomik seviyelerine, hazır bulunuşluklarına göre değişkenlik gösterebilir. Ancak daha önceki uygulamalarda 1 dönemin yeterli olmadığını gördüm".

Tablo 17

Programlama Öğretmek İçin Kullanılması Gereken Öğretim Yöntemleri

Tema/Kodlar	Sıklık (f)
Öğretim Yöntem ve Teknik	
Gösterip Yaptırma	14
Proje	8
Grupla Öğretim	7
Toplam	29

Öğretmen19: "Uygulamalı yöntemler. Gösterip yaptırma özellikle tercihim."

Öğretmen22: "Problem temelli ve proje tabanlı öğretim yöntemlerinin uygun olabileceğini düşünüyorum."

Öğretmen16: "Düz anlatım değil tabii ki de. Elbette problem çözme, grup çalışması, olmalı."

Tablo 18

Programlama Öğretiminden Öğrencilerin Motivasyon Durumları

Tema/Kodlar	Sıklık (f)
Motivasyon	
İstekli	18
İsteksiz	11
Toplam	29

Öğretmen26: " ortaya çıkarılan ürünlerde ben yaptım algısı oluşan çocuklarda özgüven ve motivasyon artışı, tüketmeye karşı üretme olgusu gelişti. Algoritmik düşünce ile olayları çok yönlü düşünebilme ve sonuçları nedenleriyle tartışıp düzenleyebilme ve yaratıcılık özellikleri gelişti."

Öğretmen22: "Öğrenciler kesinlikle büyük heyecan ve istekle derse katılıyor. Bir şeyler üretmek onları mutlu ediyor."

Tablo 19

Bir Programlama Dili Tercih Sebepleri

Tema/Kodlar	Sıklık (f)
Tercih Sebebi	
Görsel	15
Basit	6
Eğlenceli	4
İlgi Çekici	4
Toplam	29

Öğretmen25: "Scratch öğretiyorum. Sürekli gelişiyor ve blok programlamanın tüm nimetleri var. Code org sitesini en temel düzeyde kullanıyorum aşinalık yaratıyor. Bir de kulüpte bu dersleri alan öğrencilere Python öğretiyorum. Yazım dili kolay basit teferruatı olmayan bir dil."

Öğretmen26: "Kodu game ve Scratch oyun formatında ve görsel ağırlıklı olduğu için"

Öğretmen18: "Code.org, çocuklar için programlama eğlenceli hale geliyor"

Öğretmen24: “Scratch öğretiyorum çocukların ilgisini çekeceğini düşündüğüm için bununla başladım.”

Tablo 20

Öğretmenlerin Derste İşledikleri Konular

Tema/Kodlar	Sıklık (f)
Konular	
Algoritma	10
Döngü	8
Koşul	6
Operatör	5
Toplam	29

Öğretmen13: “Algoritma mantığını anlamaları için küçük oyunlarla destekledim konuyu büyük oyunlarla devam ettim.”

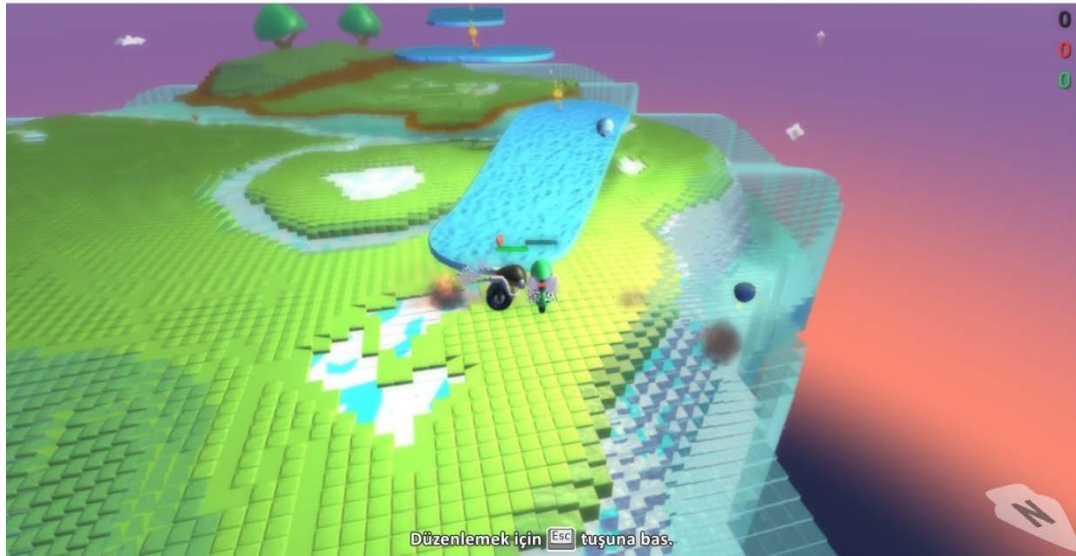
Öğretmen17: “Döngüler, eğer, mantıksal ve matematiksel işlemler hareket ettirme gibi konuları işliyorum”

Öğretmenlerden alınan veriler doğrultusunda programlama öğretimin ne kadar süre ile işleneceği planlanmıştır. Öğretimin süresi belirlendikten sonra hangi hedef ve davranışların öğrencilere kazandırılacağı öğretmenlerin görüşleri doğrultusunda belirlenmiştir. Kazanımlar belirlendikten sonra derste kullanılacak örnek uygulamalar, öğretmenlerin gönderdikleri örnek uygulamalardan yola çıkılarak hazırlanmıştır.

Öğretmenlerden gelen bazı örnek uygulamalar aşağıdaki görsellerde sergilenmektedir.



Şekil 8. Scratch' ta Örnek Bir Uygulamanın Ekran Görüntüsü



Şekil 9. Kodu Game Lab' da Örnek Bir Uygulamanın Ekran Görüntüsü

3.1.4.4. Sonuç

Araştırmanın bu kısmında katılımcıların görüşleri sonucunda ortaya çıkan sonuçlar ve sonuçlara göre yapılan çalışmalara yer verilmiştir.

3.1.4.4.1. Katılımcıların Görüşleri Doğrultusunda Yapılan Çalışmalar

Öğretmenlerden alınan görüşler doğrultusunda ilk olarak ders içeriğinde yer alan konular belirlenmiştir. Algoritma, akış şeması, ekrana yazdırma, operatör, döngü, değişken ve koşul kavramlarının ders akışında işlenecek konular olarak seçilmiştir.

Öğretmenlerden görüşler alındıktan sonra Bilişim Teknolojileri ve Yazılım dersi ortaokul müfredatı incelenmiştir. Öğretmenlerin görüşleri ve müfredatın incelenmesi sonucunda öğretiminde programlama eğitimi için gerekli olan ön bilgiler belirlenmiştir. Ön bilgiler belirlendikten sonra uygulamanın müfredata uygun olarak 2016-2017 Eğitim-Öğretim yılının ikinci dönemi olarak belirlenmiştir. Belirlenen ön bilgiler Şekil 10'da gösterilmiştir.



Şekil 10. Programlama için gerekli ön bilgilere ait görsel

Konular belirlendikten sonra programa öğretimi için kullanılan programlama dili belirlenmiştir. Öğretmenlerin görüşleri incelendiğinde Scratch, Code.org ve Kodu Game Lab programlama dillerinin en fazla tercih edilen programlama dilleri olduğu görülmektedir. Araştırmanın gerçekleştirildiği okulda internet bağlantısının yeterli olmaması ve Scratch ile kod yapılarının benzer olmasından dolayı Code.org uygulamasından vazgeçilmiştir. Kodu

Game Lab programlama dili ve Scratch programlama dili araştırmada kullanılmak üzere tercih edilmiştir.

Programlama dilleri için konular belirlendikten sonra ve programlama dili incelendikten sonra Ek7 ve Ek8’de gösterilen ders planları oluşturulmuştur. Öğretmenlerin gönderdikleri örneklerden yola çıkarak belirlenen konuların ne kadar sürede işleneceği belirlenmiştir. Programlama öğretimi için konuların toplam 20 ders saatinde işlenmesine karar verilmiştir. Konuların ne kadar sürede işleneceğine ilişkin veriler Tablo 21’de gösterilmektedir.

Tablo 21

Konuların Saatlere Göre Dağılımı

Konu	Süre (ders saati)	Yüzde
Algoritma	2	10
Akış Şeması	2	10
Ekrana Yazdırma Becerisi	2	10
Operatör Kavramı	2	10
Döngü Kavramı	2	10
Değişken Kavramı	2	10
Koşul Kavramı	4	20
Örnek Proje	4	20
Toplam	20	100

Sonuç olarak katılımcılardan alınan görüşler doğrultusunda alınan kararlar maddeler halinde aşağıdaki gibi verilmiştir;

1. Ders’te işlenecek konular algoritma, akış şeması, ekrana yazdırma, operatör, döngü, değişken ve koşul kavramı olarak belirlenip işlenmiştir.
2. Konular belirlendikten sonra öğrencilerin bu konuları öğrenmeden önceki sahip olması gereken ön bilgiler; bilgisayar okur-yazarlığı, problem çözme yolları, algoritma ve akış şeması öğrencilere verilmiştir.
3. Ön bilgiler verildikten sonra programlama eğitiminin 2016-2017 Eğitim-Öğretim yılının ikinci dönemi yapılması sağlanmıştır.

4. Scratch ve Kodu Game Lab programlama dilleri olarak seçilerek programlama dili öğretiminde kullanılmıştır.
5. Ek7 ve Ek8’de gösterilen ders planları hazırlanmıştır.
6. Programlama öğretimi süresi toplam 20 ders saati olarak belirlenip işlenmiştir.

3.2. Evren ve Örneklem

Çalışma grubunu Türkiye’de Milli Eğitim Bakanlığına bağlı Çanakkale ili Biga ilçesinde bulunan Yeniceköy Ortaokulunda iki şube de öğrenim gören 27 ortaokul 6. Sınıf öğrencisi oluşturmaktadır. Araştırmaya 28 öğrenci ile başlanmasına rağmen, bir öğrencinin araştırmanın son altı haftasına katılmamasından dolayı bu öğrenci araştırmadan çıkarılmıştır. Araştırmadan bir kişinin ayrılmasının, araştırma üzerinde etkisinin olacağı ve araştırma sonuçlarının doğruluğuna şüphe düşüreceği için bu şekilde bir yol izlenmektedir. Bu grupta yapılan çalışmalar 2016 – 2017 eğitim öğretim yılının 2. döneminde Bilişim Teknolojileri ve Yazılım Dersinde uygulanmıştır. Araştırmaya katılan öğrencilere ait frekans bilgileri Tablo 22’de gösterilmiştir.

Tablo 22

Araştırmayı Oluşturan Örneklem Ait Frekanslar

Gruplar	Kişi Sayısı
Deney Grubu 1	13
Deney Grubu 2	14
<i>Toplam</i>	27

Araştırmada iki adet deney grubu bulunmaktadır. Bu deney gruplarında öğrenci seçiminde küme örneklem yöntemi kullanılmıştır. Şubeler küme olarak atanmıştır. Küme örneklem yöntemi tek tek birey seçerek değil, seçkisiz yollarla oluşturulmuş gruplar yoluyla yapılan örneklemidir (Özen ve Gül, 2007). Programlama öğretimi toplam 20 ders saati olmak üzere, deney grubu 1’e (DG1) Scratch programlama dili, deney grubu 2’ye (DG2) Kodu Game Lab programlama dili anlatılmıştır. Deney gruplarına hem programlama dili anlatılmadan ön-test olarak, hem de programlama dili anlatıldıktan sonra son test olarak PDABT, EBODKÖYTÖ ve PİÖYÖ uygulanmıştır. Ayrıca programlama dili öğretiminin

kalıcılık yönünden incelenmesi yönünden son-test uygulandıktan sekiz hafta sonra tekrar-test olarak PDABT uygulanmıştır.

3.3. Veri Toplama Araçları

Bu çalışmada veri toplama aracı olarak 3 adet ölçek kullanılmıştır. Araştırmada tutum ölçmek için, Keçeci, Alan ve Zengin (2016) tarafından oluşturulan Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ) kullanılmıştır. Öz-yeterlik algılarını ölçmek için Altun ve Mazman' ın (2012) Programlamaya İlişkin Öz- Yeterlik (PİÖYÖ) algısı ölçeği kullanılmıştır. Akademik başarıyı ölçmek için uzman görüşü alınan, kapsam geçerliği ve güvenirlik analizi yapılan, çoktan seçmeli Programlama Dilleri Akademik Başarı Testi kullanılmıştır.

3.3.1.Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ)

Keçeci, Alan ve Zengin(2016), eğitsel bilgisayar oyunları destekli kodlama öğrenimine yönelik tutum ölçeği: geçerlilik ve güvenirlik çalışmasında öğrencilerin kodlama eğitimlerine yönelik tutumlarını ölçme amacıyla geçerli ve güvenilir bir ölçme aracına ihtiyaç duyulmasından dolayı Türkçe bir araç geliştirmişlerdir.

Ölçek 40 madde olarak, 28 olumlu ve 12 olumsuz maddeden ve 5'li Likert tipi (Kesinlikle Katılıyorum, Katılıyorum, Kararsızım, Katılmıyorum, Kesinlikle Katılmıyorum) şeklinde oluşmaktadır. Ayrıca 3 teorik boyuttan oluşmaktadır. Bu boyutlar kodlama öğretimine karşı istek, bilgisayar oyunlarının derslerde eğitim amaçlı kullanımına yönelik ilgi ve bilgisayarın asosyalleştirmesine yönelik endişeleridir. Ölçek maddeleri, Bilgisayar ve Öğretim Teknolojileri Öğretmenliği (BÖTE) bölümünde görev yapan 3 öğretim üyesi, 1 Türkçe Eğitimi öğretim üyesi ile farklı okullarda görev yapan 3 Bilgi ve İletişim Teknolojileri öğretmenlerinin görüşleri incelenerek son halini almıştır. Bu inceleme sonucunda 4 maddenin (2, 10, 19 ve 21. madde) planlanan alt boyutları ölçmediği gözlemlenmiş ve anketten çıkarılmıştır. Geliştirilen ölçek, ortaokul öğrencilerine hitap edeceği için anket maddelerinde sade, anlaşılır bir dil kullanılmıştır.

Geliştirilen ölçeğin geçerlilik ve güvenirliği hesaplanması için 240 ortaokul öğrenci ankete yanıt vermiştir. Bunun sonucunda anket maddeleri 28'e indirilmiştir. Bu şekilde ölçeğin bileşenlere yaptığı katkının %41.39 olduğu ve ölçeğin Cronbach Alfa güvenirlik

katsayısının ,833 olduğu görülmektedir. Sonuç olarak yapılan analizler sonucunda geliştirilmiş ölçeğin geçerli ve güvenilir olduğu görülmüştür.

3.3.2.Programlamaya İlişkin Öz-Yeterlik Ölçeği (PİÖYÖ)

Altun ve Mazman (2012), Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeğinin Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenirlik Çalışması adlı araştırmalarında Ramalingam ve Wiedenbeck (1998) tarafından geliştirilen “Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeği’nin”, Türkçe ’ye uyarlama ve bireylerin programlamaya ilişkin genel öz yeterlilik algılarının çeşitli değişkenler açısından incelenmeyi amaçlamışlardır. Türkçeleştirilmiş “Programlamaya İlişkin Öz-Yeterlik Algısı Ölçeği’nin” geçerlik ve güvenilirlik çalışması için farklı bölümlerde öğrenim gören 152 üniversite öğrencisine uygulanmıştır.

Programlamaya ilişkin öz-yeterlik ölçeğinin son hali yapılan geçerlik ve güvenilirlik analizler sonucunda 9 madde ve 2 faktör (basit programlama görevlerini gerçekleştirme ve karmaşık programlama görevlerini gerçekleştirme) olarak Türkçeleştirilmiş forma dönüştürülmüştür. Ölçeğin tamamı için Cronbach Alpha katsayısı .928 konjenerik ölçümler için daha uygun sonuçlar verdiği ifade edilen McDonald’ın ω (omega) katsayısı ise. 956 olarak hesaplanmıştır. Açıklayıcı faktör analizi sonunda dokuz madde toplam varyansın %80,814’ünü açıklamış ve bu yapıya ilişkin model doğrulayıcı faktör analizi ile de doğrulanarak programlamaya ilişkin öz yeterlilik algısını ölçen geçerli ve güvenilir bir araç elde edilmiştir.

3.3.3.Programlama Dilleri Akademik Başarı Testi (PDABT)

Bu çalışmada kullanılan başarı testi, öğrencilerin programlama başarılarını ölçmek üzere araştırmacı tarafından geliştirilmiştir. Test geliştirme çalışmalarına, öğretmenlerden alınan görüşler doğrultusunda algoritma, akış şeması, ekrana yazdırma, operatör, döngü, değişken ve koşul konularına uygun olarak, her konuya ilişkin en az iki soru olmak üzere toplam 22 soru hazırlanarak başlanmıştır. Oluşturulan form alanında uzman ve daha önce programlama eğitimi vermiş üç Bilişim Teknolojileri Öğretmeni ve üniversitelerin BÖTE bölümünde öğretim görevlisi kadrosunda görevli olan bir uzman tarafından incelenmiştir. Yazım hatalarının kontrolü için iki Türkçe Öğretmeni tarafından incelenip gerekli

düzenlemeler yapılmıştır. Deneme formunda yer alan maddelerin konulara göre dağılımını içeren tablo hazırlanmıştır (Tablo 23).

Tablo 23

Maddelerin konulara göre dağılımı

Konular	Hedefler	Madde numarası	Toplam soru	Yüzde
Temel algoritma becerisi		m1,m4, m5	3	13,64%
Döngü kavramı		m2, m3,m11	3	13,64%
Hata ayıklama becerisi		m6,m12, m8	3	13,64%
Koşul (Eğer) kavramı		m9,m15	2	9,09%
Operatör kavramı		m18	1	4,55%
Koşul (Eğer değilse)		m7,m13	2	9,09%
Yazdırma becerisi		m14,m19	2	11,11%
Eksik tamamlama becerisi		m20, m10	2	9,09%
Değişken+Döngü+Koşul		m21	1	4,55%
Döngü+Yazdırma becerisi		m22	1	4,55%
Koşul(Eğer değilse)+Operatör		m17	1	4,55%
Operatör+Döngü		m16	1	4,55%
Toplam			22	100,00%

Hazırlanan belirtke çizelgesine uygun olarak hazırlanan deneme testi 2016-2017 eğitim öğretim yılında Çanakkale Biga’da bulunan ortaokullarda öğrenim gören ve daha önce programlama eğitimi almış 115 öğrenciye uygulanmıştır. Programlama öğretimi alan öğrenci sayısı azlığı nedeniyle sınırlı sayıda öğrenciye uygulanmıştır. Bu sınırlılığı gidermek için testin madde analizinde diğer yöntemlere kıyasla daha duyarlı olan Henrysson yöntemi kullanılmıştır (Atılğan, Kan ve Doğan, 2013). Henryson yönteminde bütün testi yapan öğrencilerinden alınan cevaplar analiz edilmektedir (Güler, 2012). Madde analizinde tüm hesaplamalar için öncelikle öğrencilerin verdiği cevaplar Microsoft Excel 2007 programına girilmiş, daha sonra analizi yapmak için TAP (Test Analysis Program) kullanılmıştır. TAP programında elde edilen verilerden, madde güçlük indeksi ve madde ayırt edicilik indeksi Tablo 24’teki gibidir.

Tablo 24

Deneme Testinde Yer Alan Maddelerin Güçlük ve Ayırtıcılık İndeksleri

Davranış	No	Güçlük indeksi (p_j)	Ayırt Edicilik İndeksi(r_{jx})
1	1	0,82	0,24
	4	0,64*	0,48*
	5	0,83*	0,38*
2	2	0,43*	0,61*
	3	0,35*	0,61*
	11	0,42	0,55
3	6	0,47	0,27
	8	0,49*	0,78*
	12	0,51*	0,53*
4	9	0,56*	0,39*
	15	0,26	0,16
5	18	0,70*	0,51*
6	7	0,61*	0,53*
	13	0,65	0,15
7	14	0,80*	0,35*
	19	0,52	0,36
8	10	0,49*	0,64*
	20	0,32*	0,46*
9	21	0,42*	0,53*
10	22	0,34*	0,46*
11	17	0,24*	0,32*
12	16	0,51*	0,64*

***Başarı testine seçilen maddeler**

Deneme testinde yer alan maddelerden güçlük indeksine, ayırtıcılık indeksine ve çeldiricilerin çalışıp çalışmadığına bakarak madde seçimi yapılmıştır (Güler, 2012). Nihai teste madde seçerken öncelikle ayırt edicilik indeksleri 0,30'dan büyük olanları, varsa 0,40'tan büyük olanlar seçilmiştir (Atılgan, Kan ve Doğan, 2013). Yapılan istatistikler sonucunda dört madde atılmıştır. Ayrıca kazanımlar ve öğrencilerin testi cevaplama süreleri göz önüne alındığından iki madde daha testten çıkarılmıştır. Başarı testinin nihai halinin oluşmasını sağlayan madde istatistikleri Tablo 25'te gösterilmektedir.

Tablo 25

Nihai Test Madde İstatistikleri

Deneme testi	Nihai test	Madde güçlük indeksi	Madde varyansı	Madde standart sapması	Madde ayırt edicilik indeksi	Madde güvenirlik indeksi
Madde no	Madde no	(p _j)	$s_j^2 = p_j \cdot q_j$	$sj = \sqrt{p_j \cdot q_j}$	r_{jx}	$r_j = r_{jx} \cdot s_j$
2	1	0,43	0,25	0,50	0,72	0,36
3	2	0,35	0,23	0,48	0,65	0,31
4	3	0,64	0,23	0,48	0,45	0,22
5	4	0,83	0,15	0,39	0,45	0,17
7	5	0,61	0,24	0,49	0,65	0,32
8	6	0,49	0,25	0,50	0,84	0,42
9	7	0,56	0,25	0,50	0,39	0,19
10	8	0,49	0,25	0,50	0,71	0,35
12	9	0,51	0,25	0,50	0,53	0,26
14	10	0,80	0,16	0,41	0,36	0,15
16	11	0,51	0,25	0,50	0,71	0,35
17	12	0,24	0,18	0,43	0,43	0,18
18	13	0,70	0,21	0,46	0,58	0,27
20	14	0,32	0,22	0,47	0,43	0,20
21	15	0,42	0,24	0,49	0,52	0,26
22	16	0,34	0,22	0,47	0,50	0,24

Testin istatistiklerine bakıldığında, test puanları ortalaması 8,24; standart sapması (Sx) 3,54; varyansı (Sx²) 18,32 hesaplanmıştır. Testin ortalama güçlük düzeyinin ise 0,52 olduğu, bu durumda testin güçlüğünün ortalama güçlük düzeyine yakın olduğu görülmektedir. Ek-4'te sunulan başarı testinin KR-20 güvenirlik katsayısının 0,76 olması da güvenirliğinin iyi düzeyde olduğunu göstermektedir (Atılgan, Kan ve Doğan, 2013). Nihai testin belirtke tablosu Tablo 26'da verilmiştir.

Tablo 26

Nihai Test Maddelerin konulara göre dağılımı

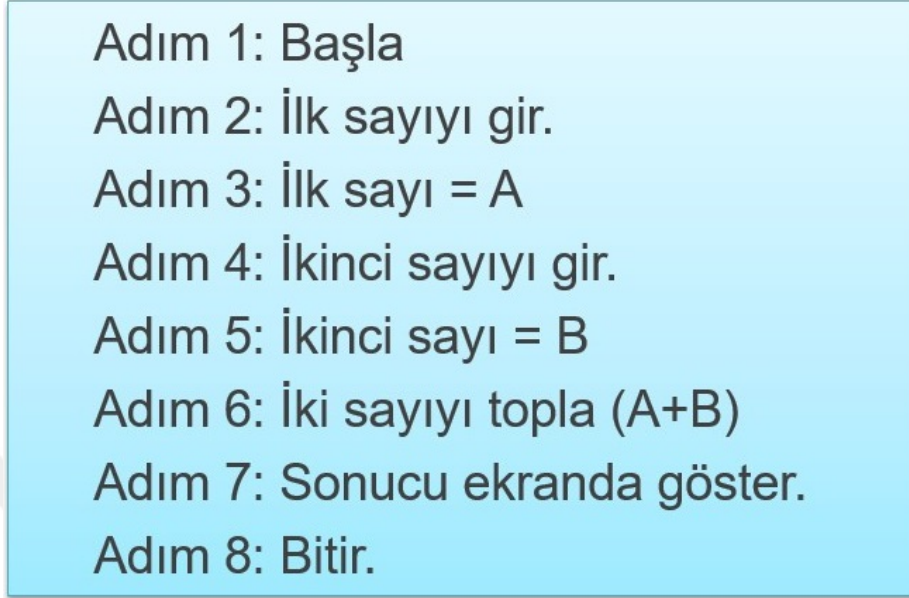
Konular	Hedefler	Madde numarası	Toplam soru	Yüzde
Temel algoritma becerisi		m3, m4	2	12,50%
Döngü kavramı		m1, m2	2	12,50%
Hata ayıklama becerisi		m9, m6	2	12,50%
Koşul (Eğer) kavramı		m7	1	6,25%
Operatör kavramı		m13	1	6,25%
Koşul (Eğer değilse)		m5	1	6,25%
Yazdırma becerisi		m10	2	12,50%
Eksik tamamlama becerisi		m14, m8	2	12,50%
Değişken+Döngü+Koşul		m15	1	6,25%
Döngü+Yazdırma becerisi		m16	1	6,25%
Koşul(Eğer değilse)+Operatör		m12	1	6,25%
Operatör+Döngü		m11	1	6,25%
Toplam			16	100,00%

3.4. Uygulamanın Gerçekleştirilme Süreci

Bu çalışma Milli Eğitim Bakanlığı'na bağlı okullardan Çanakkale iline bağlı Biga Yeniceköy Ortaokulu'nda 2016-2017 eğitim öğretim yılının ikinci döneminde toplam 20 ders saati süresince Bilişim Teknolojileri ve Yazılım dersi kapsamında ortaokul 6. Sınıf öğrencileriyle yürütülmüştür. Bilişim Teknolojileri ve Yazılım dersinin genel amaçlarından biri problem çözme, programlama ve özgün ürün geliştirmektir (TTKB,2012). Bu çerçevede yapılan çalışma Bilişim Teknolojileri ve Yazılım dersi kapsamında gerçekleştirilmiştir.

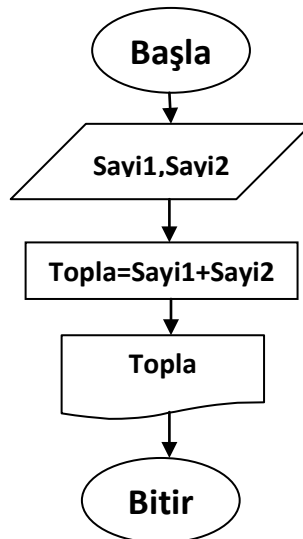
Programlama öğretimine başlamadan önce öğrencilere ön-test olarak Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ), Programlamaya İlişkin Öz Yeterlik Ölçeği (PiÖYÖ), Programlama Dilleri Akademik Başarı Testi (PDABT) uygulanmıştır.

Programlama öğretimine ilk hafta iki ders saatinde iki deney grubuyla Algoritma konusu işlenerek başlanmıştır. Öğrencilere verilen örnekleri adım adım düşünerek algoritma haline getirmeleri istenmiştir. Derste işlenen bir örnek Şekil 11’de verilmiştir.



Şekil 11. Örnek bir algoritma görüntüsü

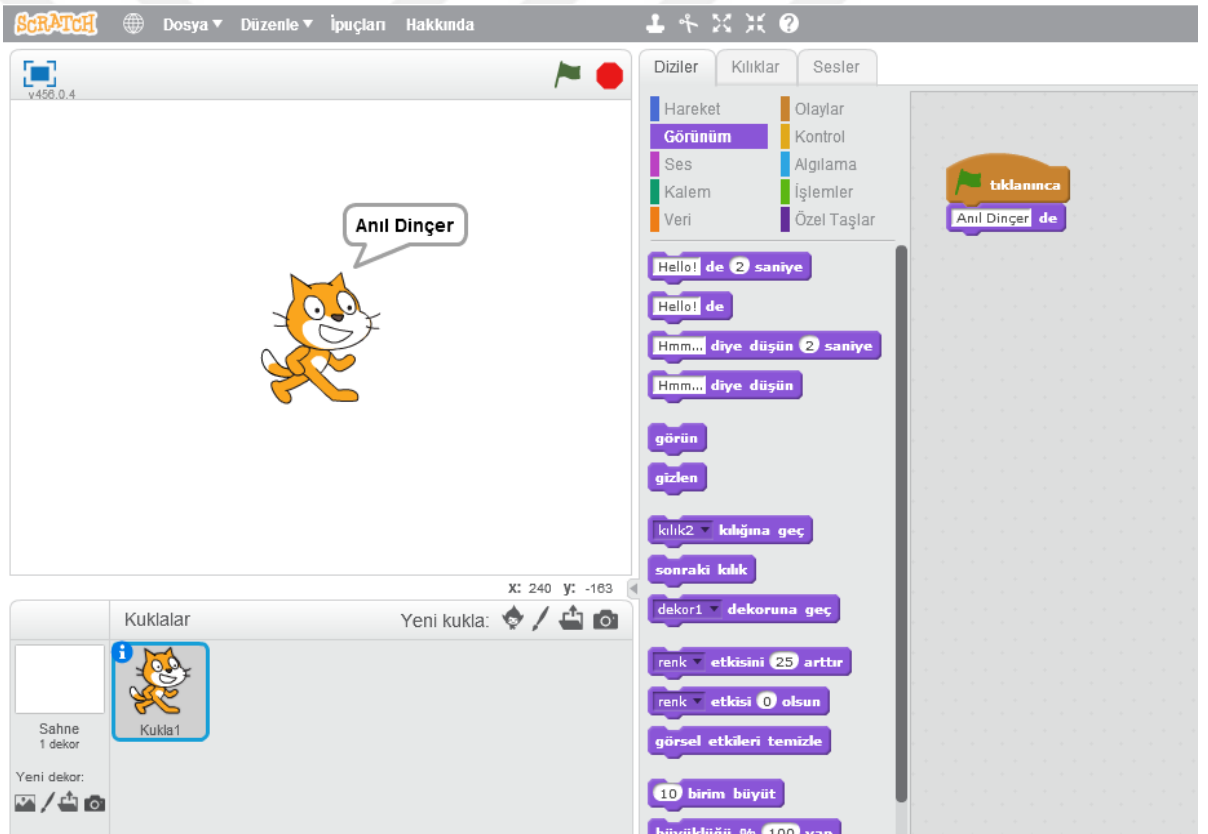
Programlama öğretimine ikinci hafta iki ders saatinde her iki deney grubuyla da Akış Şeması konusu işlenmiştir. Öğrencilere algoritmalarını yaptıkları örneklerin Akış Şemalarını çizmeleri istenmiştir. Derste işlenen bir örnek Şekil 12’de verilmiştir.



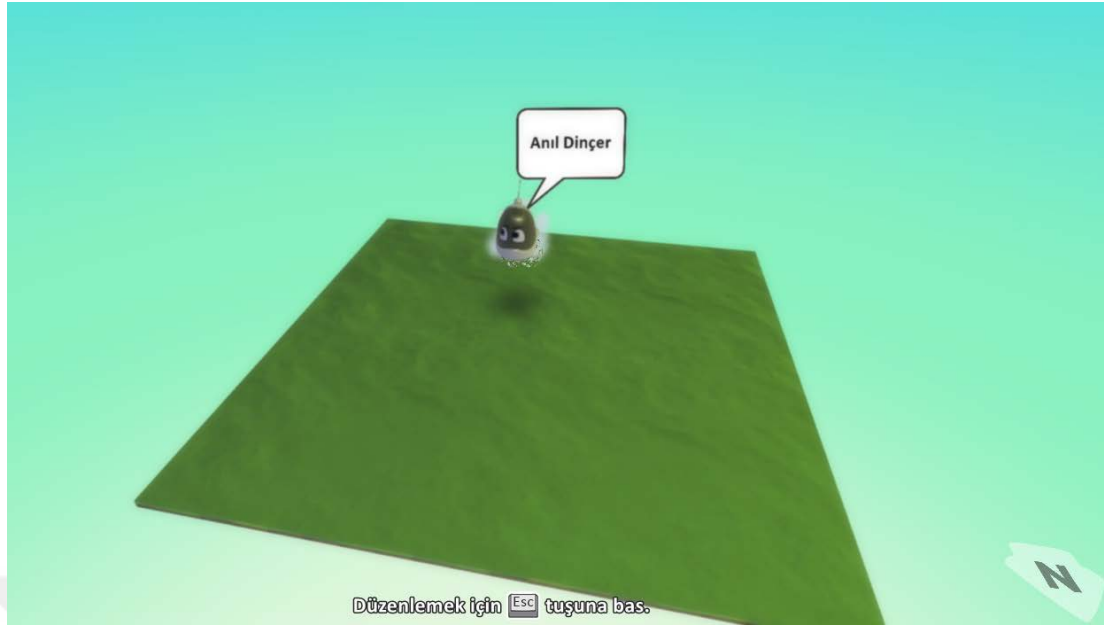
Şekil 12. Örnek bir Akış Şemasına ait bir örnek

Programlama öğretimine üçüncü hafta iki ders saati içerisinde DG1 grubuna Scratch programlama dili tanıtılmıştır. Menüleri anlatıldıktan sonra kod blokları gösterilmiştir. Kod bloklarına göre öğrencilere ekrana yazı yazdırma blokları gösterilmiş ve nasıl kullanmaları gerektiği anlatılmıştır. Her öğrenciden Scratch Programıyla adını soyadını yazmaları istenmiştir. Derste Scratch ile ekrana adını soyadını yazdırma etkinliğine ilişkin bir örnek Şekil 13'te verilmiştir.

Aynı hafta DG2 grubuna da Kodu Game Lab programlama dili tanıtılmıştır. Menüleri anlatıldıktan sonra kod yazım alanı ve kod görselleri gösterilmiştir. Daha sonra her öğrenciden Kodu Game Lab Programıyla adını soyadını ekrana yazdırmaları istenmiştir. Derste Kodu Game Lab uygulaması ile ekrana adını soyadını yazdırma etkinliğine ilişkin bir örnek Şekil 14'te verilmiştir.



Şekil 13. Scratch'ta ekrana yazdırma becerisiyle ilgili bir örnek



Şekil 14. Kodu Game Lab ekrana yazdırma becerisiyle ilgili bir örnek

Programlama öğretimine dördüncü hafta iki ders saati içerisinde DG1 grubuna Scratch programlama dilinde operatörler kavramı anlatılmıştır. Bu kavramla ilgili '<','>','ve','veya','değil' ,'+','x','-','/' işlemleri sırasıyla anlatılmıştır. Bununla ilgili Scratch'ta operatörler etkinliğine ilişkin bir örnek Şekil 15'te verilmiştir.



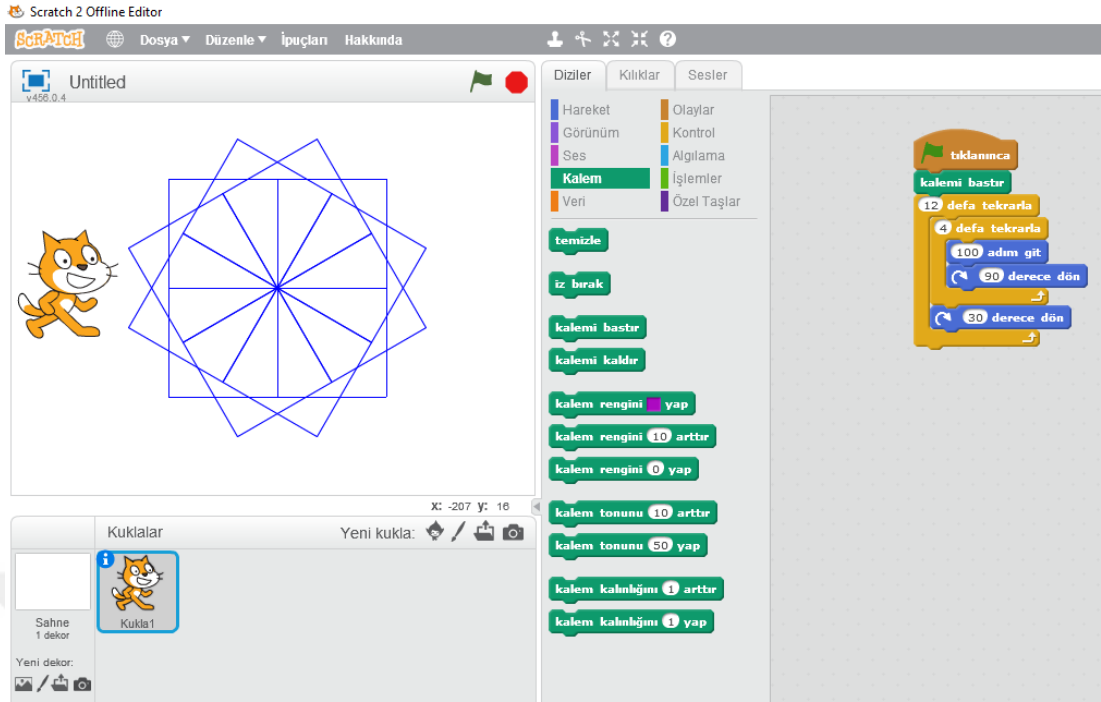
Şekil 15. Scratch ile örnek bir toplama işlemiyle ilgili bir örnek

Programlama öğretiminde aynı hafta iki ders saati içerisinde DG2 grubuna Kodu Game Lab Programlama dilinde operatörler kavramı anlatılmıştır. Bu kavramla ilgili '<','>','ve','veya','değil','+','x','-', '/' işlemleri sırasıyla anlatılmıştır. Bununla ilgili Kodu Game Lab'da operatörler etkinliğine ilişkin bir örnek Şekil 16'da verilmiştir.



Şekil 16. Kodu Game Lab ile örnek bir toplama işlemiyle ilgili bir örnek

Programlama öğretimine beşinci hafta iki ders saati içerisinde DG1 grubuna Scratch programlama dilinde döngüler kavramı anlatılmıştır. Bu kavramla ilgili tekrar eden davranışların döngüler yardımıyla daha kolay yapılacağı gösterilmiştir. Öğrencilerden daha sonra Scratch'ta döngüler kavramına ilişkin örnekler yapmaları istenmiştir. Scratch'ta döngüler etkinliğine ilişkin bir örnek Şekil 17'de verilmiştir.



Şekil 17. Scratch ile döngü kullanarak desen çıkartmaya ilişkin bir örnek

Programlama öğretimine aynı hafta iki ders saati içerisinde DG2 grubuna Kodu Game Lab Programlama dilinde döngüler kavramı anlatılmıştır. Bu kavramla ilgili tekrar eden davranışların döngüler yardımıyla daha kolay yapılacağı gösterilmiştir. Öğrencilerden Kodu Game Lab’da döngüler kavramına ilişkin örnekler yapmaları istenmiştir. Kodu Game Lab’da döngüler etkinliğine ilişkin bir örnek Şekil 18’de verilmiştir.



Şekil 18. Kodu Game Lab ile işaretli alanlarda sürekli hareket eden robotla ilgili bir örnek

Programlama öğretimine altıncı hafta iki ders saati içerisinde DG1 grubuna Scratch Programlama dilinde değişken kavramı anlatılmıştır. Bu kavramla ilgili olarak, bir değişkenin sabit bir değer veya değişen farklı değer alabileceği öğrencilere gösterilmiştir. Öğrencilerden daha sonra Scratch'ta değişken oluşturarak örnekler yapmaları istenmiştir. Scratch'ta değişken oluşturma etkinliğine ilişkin bir örnek Şekil 19'de verilmiştir



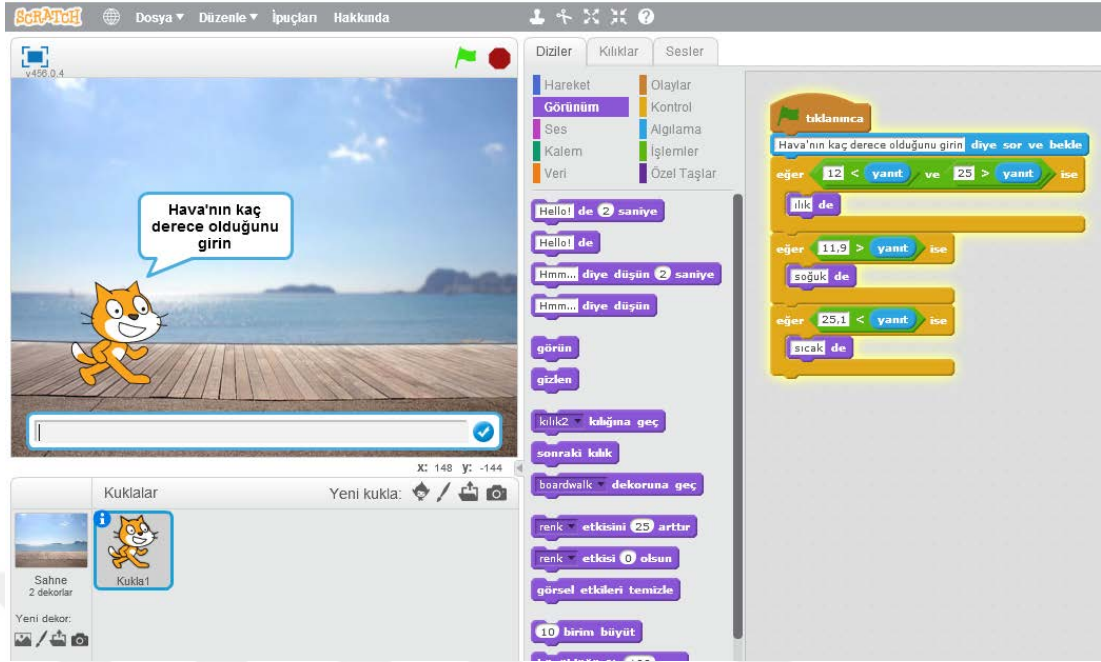
Şekil 19. Scratch ile değişken kullanarak yapılan bir örnek

Programlama öğretimine aynı hafta iki ders saati içerisinde DG2 grubuna Kodu Game Lab Programlama dilinde değişken kavramı anlatılmıştır. Bu kavramla ilgili olarak, bir değişkenin sabit bir değer veya değişen farklı değer alabileceği öğrencilere gösterilmiştir. Öğrencilerden daha sonra Kodu Game Lab'da skor değişkeni oluşturarak örnekler yapmaları istenmiştir. Kodu Game Lab'da değişken oluşturma etkinliğine ilişkin bir örnek Şekil 20'de verilmiştir.



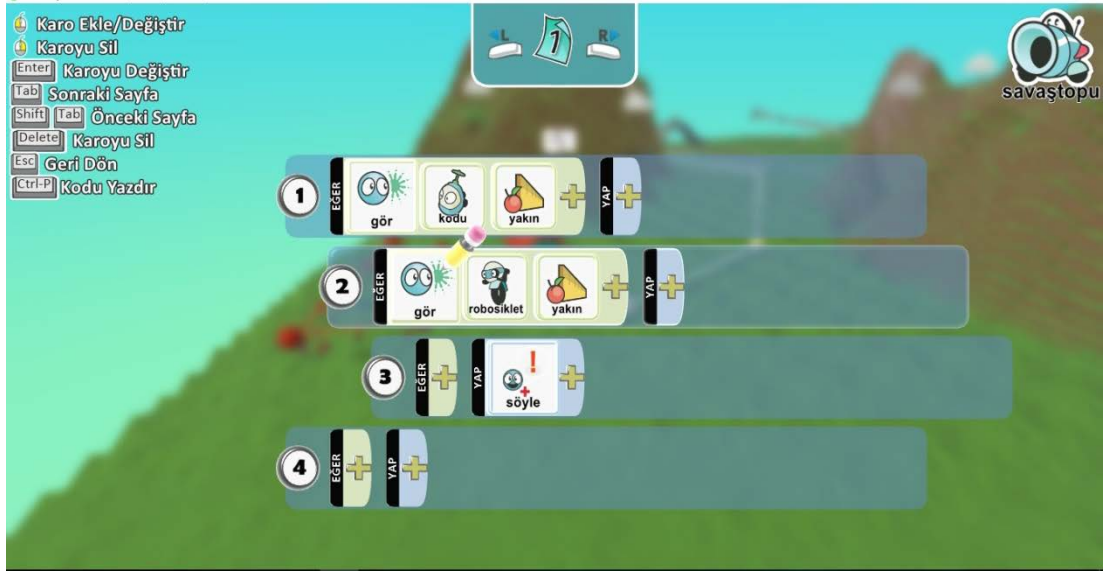
Şekil 20. Kodu Game Lab ile puan değişkeni kullanılarak yapılan bir örnek

Programlama öğretimine yedinci ve sekizinci haftalar toplam 4 ders saati içerisinde DG1 grubuna Scratch Programlama dilinde koşul kavramı anlatılmıştır. Bu kavramla ilgili öğrencilere günlük hayatta karar vermek zorunda olduklarının farkına varmaları sağlanmıştır. Ayrıca verdiği kararların bir sonucu olduğunu ve bu sonucun verdikleri kararlara göre değiştiği ifade edilmiştir. Bu kapsamda eğer komutu ve eğer değilse komutlarının kullanımı öğrencilere aktarılmıştır. Öğrencilerden daha sonra Scratch'ta koşullara ilişkin örnekler yapmaları istenmiştir. Scratch'ta koşul ifadelerine ilişkin bir örnek Şekil 21'te verilmiştir.



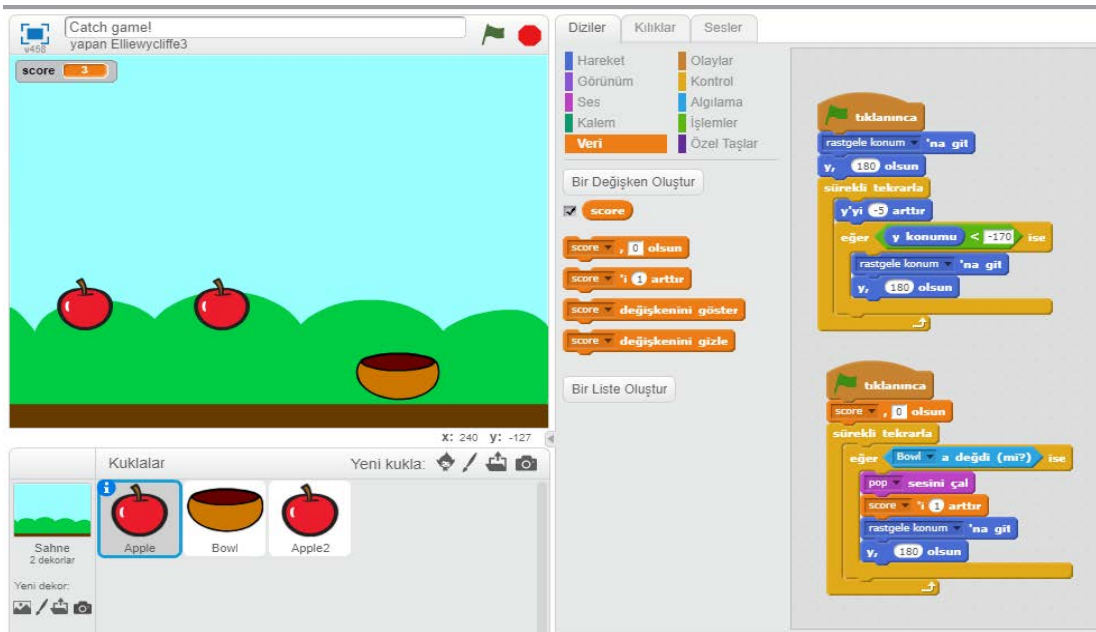
Şekil 21. Scratch ile koşul ifadesi kullanılarak yapılan bir örnek

Programlama öğretimine aynı haftalar toplam 4 ders saati içerisinde DG2 grubuna Kodu Game Lab Programlama dilinde koşul kavramı anlatılmıştır. Bu kavramla ilgili öğrencilere günlük hayatta karar vermek zorunda olduklarının farkına varmaları sağlanmıştır. Ayrıca verdiği kararların bir sonucu olduğunu ve bu sonucun verdikleri kararlara göre değiştiği ifade edilmiştir. Bu kapsamda eğer komutu ve eğer değilse komutlarının kullanımı öğrencilere aktarılmıştır. Öğrencilerden daha sonra Kodu Game Lab'da koşullara ilişkin örnekler yapmaları istenmiştir. Kodu Game Lab'da koşul ifadelerine ilişkin bir örnek Şekil 22'de verilmiştir.



Şekil 22. Kodu Game Lab ile koşul ifadesi kullanılarak yapılan bir örnek

Programlama öğretimine dokuzuncu ve onuncu haftalar toplam 4 ders saati içerisinde DG1 grubuna Scratch programlama dilinde öğrencilerin şuana kadar öğrendiği bütün kavramları bir arada kullanabilecekleri bir proje yapmaları istenmiştir. Projenin konusunun her öğrenci için aynı olmasına karar verilmiştir. Öğrencilerden elma toplama oyunu yapmaları istenmiştir. Scratch'ta öğrencilerin yaptığı projeye bir örnek Şekil 23'te verilmiştir.



Şekil 23. Scratch' ta elma toplama oyununa ilişkin bir örnek uygulama

Programlama öğretimine aynı haftalar toplam 4 ders saati içerisinde DG2 grubuna Kodu Game Lab programlama dilinde öğrencilerin şuna kadar öğrendiği bütün kavramları bir arada kullanabilecekleri bir proje yapmaları istenmiştir. Projenin konusunun her öğrenci için aynı olmasına karar verilmiştir. Öğrencilerden motosiklet yarışı oyunu yapmaları istenmiştir. Kodu Game Lab’da öğrencilerin yaptığı projeye bir örnek Şekil 25’te verilmiştir.



Şekil 24. Kodu Game Lab ile motosiklet oyununa ilişkin bir örnek uygulama

3.5. Veri Çözümleme Teknikleri

Elde edilen nicel verilerin analizi “SPSS for Windows 21.00” istatistik paket programı kullanılarak gerçekleştirilmiştir.

3.5.1. Betimsel İstatistik Sonuçları

Scratch (DG1) ve Kodu Game Lab (DG2) Programlama Dilleri öğretiminin öğrencilerin tutumlarına etkisini incelemek amacıyla Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ) uygulanarak Deney Grubu 1 (DG1) ve Deney Grubu 2 (DG2)’nin ön test ve son test; başarılarına olan etkisini incelemek amacıyla Programlama Dilleri Akademik Başarı Testi (PDABT) uygulanarak Deney Grubu

1 (DG1) ve Deney Grubu 2 (DG2)' nin ön-test ve son-test; öz yeterlik algılarına olan etkisini ölçmek amacıyla Programlamaya İlişkin Öz Yeterlik Ölçeği (PİÖYÖ) uygulanarak Deney Grubu 1 (DG1) ve Deney Grubu 2 (DG2)' nin ön-test sonuçları Tablo 27'de ve son-test sonuçlarının betimsel istatistiksel sonuçları ise Tablo 28'de verilmiştir.

Tablo 27

Ön-test Betimsel İstatistikler

	PDABT		EBODKÖYTÖ		PİÖYÖ	
	DG 1	DG 2	DG 1	DG 2	DG 1	DG 2
n	13	14	13	14	13	14
$\bar{X}$	6,23	6,64	3,56	3,26	1,84	2,23
Medyan	6,00	7,50	3,54	3,27	1,78	2,22
Minimum	3	3	2,46	2,07	,46	,59
Maksimum	12	10	4,04	4,39	1,22	1,22
Ranj	9	7	1,57	2,32	2,67	3,44
Standart Sapma	2,28	2,56	,40	,56	1,45	2,22
Çarpıklık	1,163	-,369	-1,730	-,068	,391	-,010
Basıklık	2,630	-1,326	4,607	1,260	-,858	,658

Tablo 28

Son-test Betimsel İstatistikler

	PDABT		EBODKÖYTÖ		PİÖYÖ	
	DG 1	DG 2	DG 1	DG 2	DG 1	DG 2
n	13	14	13	14	13	14
$\bar{X}$	12,62	8,43	3,52	3,59	4,73	4,78
Medyan	14	9	3,50	3,70	5,11	4,56
Minimum	6	3	2,96	2,93	2,33	2,78
Maksimum	16	14	4,43	4,46	7	6,,89
Ranj	10	11	1,46	1,54	4,67	4,11
Standart Sapma	2,82	3,67	,41	,44	1,45	1,39
Çarpıklık	-1,071	,046	,825	,71	-,258	,177
Basıklık	1,026	-1,179	,492	-,011	-,990	-1,147

3.5.2. Değişkenler

6. Sınıf öğrencilerine Scratch ve Kodu Game Lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarılarının karşılaştırılması isimli bu çalışmada toplamda iki bağımsız değişken ve üç bağımlı değişken yer almaktadır.

3.5.2.1. Bağımsız Değişken

Araştırmanın sonucunu etkileyen, araştırmacının ortama sokarak bağımlı değişkenin aldığı değeri değiştiren değişken olarak tanımlanabilmektedir (Büyüköztürk, 2017). Bu çalışmada yer alan bağımsız değişken, programlama öğretim yöntemidir. Programlama öğretim yöntemi olarak “Scratch programlama dili ile programlama öğretimi” ve “Kodu Game Lab programlama öğretimi” olarak belirlenmiştir.

3.5.2.2. Bağımlı Değişken

Ortamda olan bir durumdan etkilenecek farklı sonuçlar alabilen ve araştırmanın sonucu olarak kabul edilen değişken olarak tanımlanabilmektedir (Büyüköztürk, 2017). Bu çalışmada “Programlama Dilleri Başarısı”, “Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum” ve “Programlamaya İlişkin Öz Yeterlik” araştırmanın bağımlı değişkenleridir.

3.5.3. Araştırmada Kullanılan İstatistiksel Testler

3.5.3.1. İlişkisiz Örneklemeler *t*-testi

İlişkisiz Örneklemeler *t*-test, iki grubun birbirine eşit mi yoksa aralarında anlamlı bir farklılaşma var mı sorularına cevap arayan bir test olarak tanımlanabilmektedir. Bu testi kullanabilmek için iki grup olması esastır. Ayrıca İlişkisiz Örneklemeler *t*-test kullanabilmek için aşağıdaki bazı varsayımların da yerine getirilmesi gerekmektedir;

1. Veriler ölçek türü açısından aralık veya oransal olmalı
2. Grupların dağılımı normal olmalı
3. Birbirinden bağımsız iki grup olmalı

4. Ölçümlere ait varyanslar birbirine eşit olmalı (Çiçek,2017;Büyüköztürk,2017).

Bu araştırmada Scratch (DG1) ve Kodu Game Lab programlama dilleri öğretimi sırasında Shapiro-Wilk testi sonucuna göre normal dağılım gösteren grupların aralarındaki farklılaşma düzeyini araştırmak amacıyla, Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ) ön-test ve son-test, Programlamaya İlişkin Öz Yeterlik Ölçeği (PİÖYÖ) ön-test ve son-test, Programlama Dilleri Akademik Başarı Testi (PDABT) ön-test, son-test ve tekrar-test olarak ilişkisiz örneklemeler *t*-test kullanılmıştır.

3.5.3.2. İlişkili Örneklemeler *t*-testi

Bir gruba ait öncesine ve sonrasına ait ölçüm sonuçlarının aralarındaki farklılaşma düzeyini belirlemek için ilişkili örneklemeler *t*-test kullanılmaktadır. Örnek olarak bir sınıfın vize ve final notları arasındaki farklılaşma olup olmadığı bulunmak istenirse bu testin kullanılması gerekmektedir (Ak, 2017). Büyüköztürk (2014)'e göre ilişkili örneklemeler *t*-test kullanabilmek için bazı şartların yerine gelmesi gerekmektedir. Bunlar;

1. Ölçümler aralık ölçeğinde veya oran ölçeğinde olmalı
2. Puanlar normal dağılım göstermeli
3. Aynı örnekleme ait olmalı

Bu araştırmada Scratch (DG1) ve Kodu Game Lab programlama dilleri öğretimi sırasında Shapiro-Wilk testi sonucuna göre normal dağılım gösteren grupların kendi aralarındaki ön-test, son-test ve tekrar-test farklılaşma düzeyini araştırmak amacıyla, Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ) , Programlamaya İlişkin Öz Yeterlik Ölçeği (PİÖYÖ), Programlama Dilleri Akademik Başarı Testi (PDABT) olarak ilişkisiz örneklemeler *t*-test kullanılmıştır.

3.5.3.3. Shapiro-Wilk Normallik Testi

Grupların normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi kullanılmaktadır. Ancak Shapiro-Wilk normallik testi kişi sayısı 50'den az olan gruplar için kullanılabilir (Büyüköztürk, 2017).

Bu alıřmada, bağımlı deęiřkenlere ait Eęitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Öleęi (EBODKÖYTÖ) ön-test ve son-test puanları, Programlamaya İliřkin Öz Yeterlik Öleęi (PiÖYÖ) ön-test ve son-test puanları, Programlama Dilleri Akademik Başarı Testi (PDABT) ön-test, son-test ve tekrar-test puanlarının normal daęılım gösterip göstermedięini incelemek için kullanılmıřtır.



BÖLÜM IV

4. BULGULAR VE YORUMLAR

Araştırmanın bu kısmında Scratch (DG1) ve Kodu Game Lab (DG2) ile programlama öğrenimi gören öğrencilerin başarı, tutum ve öz-yeterlik algılarına ait istatistiksel sonuçlara yer verilmiştir.

4.1. Programlamaya Yönelik Başarının İncelenmesi

4.1.1. PDABT Ön-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch (DG1) ve Kodu Game Lab (DG2) programlama dillerini öğrenen deney gruplarının başarıları analiz edilmiştir. İlk olarak Programlama Dilleri Akademik Başarı Testi'nin (PDABT) ön-testlerine ait verilerinin normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testi uygulamak için örneklem sayısının 50' den az olduğu durumlar tercih edilmektedir (Büyüköztürk, 2017). Shapiro-Wilk testine ait sonuçlar Tablo 29'da verilmiştir.

Tablo 29

PDABT Ön-Testlere Ait Shapiro-Wilk Sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	6,23	2,28	,901	13	,137
DG2	14	6,64	2,56	,881	14	,060

Shapiro Wilk testi sonuçları ($p > 0,05$) anlamlılık düzeyinde test edilmiş olup Tablo 29 incelendiğinde DG1 grubu ($p = ,137$) ve DG2 grubu ($p = ,060$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 ön-test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklemeler *t*-test kullanılmıştır. İlişkisiz Örneklemeler *t*-test sonuçları Tablo 30' da verilmiştir.

Tablo 30

PDABT Ön-testlere Ait İlişkisz Örneklemeler t-test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	6,23	2,28	25	-,440	,663
DG2	14	6,64	2,56			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 5.1.1.2'deki başarı testi ön-test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}$ =6,23, SS=2,28), DG2 grubuna ait puanların ortalamasının ise ($\bar{X}$ =6,64, SS=2,56) olduğu görülmektedir. Ortalamalar incelendiğinde DG1 ve DG2 gruplarının ortalama puanlarının birbirlerine yakın olduğu görülmektedir. Ayrıca İlişkisz örneklemeler t-test sonuçları incelendiğinde grupların PDABT değişkeni açısından anlamlı bir fark olmadığı görülmektedir ($t(25)$ =-,440, p =,663 >,05).

Uygulama esnasında verilen Scratch ve Kodu Game Lab programlama dilleri eğitiminin öğrenciler üzerinde hangi konularda ne derecede bilgi sahibi olduğu incelemek amacıyla öğrencilerin PDABT ön-testlere verdikleri cevaplar incelenmiştir. Testlere verilen cevaplar tek kazanımları içerebildiği gibi birden fazla kazanımı da içerdği için birleştirilerek tablolştırılmıştır. Grupların ön-testlere ait verileri Tablo 31'de verilmektedir.

Tablo 31

DG1 ve DG2 Gruplarına Ait PDABT Maddelere İlişkin Ön-test Sonuçları

Konular	Madde numarası	DG1 (n=13)			DG2 (n=14)		
		Doğru		Yanlış	Doğru		Yanlış
		Sayı	Yüzde		Sayı	Yüzde	
Temel algoritma becerisi	M3	4	30,77	9	6	42,86	8
	M4	12	92,31	1	10	71,43	4
Döngü kavramı	M1	2	15,38	11	4	28,57	10
	M2	2	15,38	11	5	35,71	9
Hata ayıklama becerisi	M6	4	30,77	9	4	28,57	10
	M9	6	46,15	7	7	50,00	7
Koşul (Eğer) kavramı	M7	1	7,69	12	2	14,29	12
Operatör kavramı	M13	8	61,54	5	10	71,43	4
Koşul (Eğer değilse)	M5	5	38,46	8	7	50,0	7
Yazdırma becerisi	M10	12	92,31	1	13	92,86	1
Eksik tamamlama becerisi	M8	5	38,46	8	5	35,71	9
	M14	1	7,69	12	2	14,29	12
Değişken+Döngü+Koşul	M15	5	38,46	8	4	28,57	10
Döngü+Yazdırma becerisi	M16	4	30,77	9	3	21,43	11
Koşul(Eğer değilse)+Operatör	M12	3	23,08	10	3	21,43	12
Operatör+Döngü	M11	7	53,85	6	6	42,86	8

4.1.2.PDABT Son-Test Sonuçlarının İncelenmesi

DG1 ve DG2 gruplarının ön-test sonuçlarının incelenmesinin ardından bu kısımda Scratch ve Kodu Game Lab programlama dillerini öğrenen bu grupların başarıları analiz edilmiştir. Bu aşamada Programlama Dilleri Akademik Başarı Testi'nin (PDABT) son-testlerine ait verilerinin normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testine ait sonuçlar Tablo 32'de verilmiştir.

Tablo 32

PDABT Son-testlere ait Shapiro-Wilk Sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	12,62	2,82	,897	13	,123
DG2	14	8,43	3,67	,942	14	,445

PDABT son-testlere ait Shapiro-Wilk testi sonuçları ($p > 0,05$) anlamlılık düzeyinde test edilmiş olup Tablo 32 incelendiğinde DG1 grubu ($p = ,123$) ve DG2 grubu ($p = ,445$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 son-test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklemeler t -test kullanılmıştır. İlişkisiz Örneklemeler t -test sonuçları Tablo 33'te verilmiştir.

Tablo 33

PDABT Son-testlere Ait İlişkisiz Örneklemeler t -test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	12,62	2,82	25	3,305	,003
DG2	14	8,43	3,67			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 33'teki PDABT son-test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}=12,62$, $SS=2,82$), DG2 grubuna ait puanların ortalamasının ise ($\bar{X} =8,43$, $SS=3,67$) olduğu görülmektedir. Ortalamalar incelendiğinde DG1 grubunun ortalamasının DG2 grubunun ortalamasından daha büyük olduğu görülmektedir. Ayrıca İlişkisiz örneklemeler t -test sonuçları incelendiğinde grupların PDABT açısından aralarında anlamlı bir fark olduğu görülmektedir ($t(25)=-,3,305$ $p=,003>,05$).

Uygulama esnasında verilen Scratch ve Kodu Game Lab programlama dilleri eğitiminin öğrenciler üzerinde hangi konuların daha anlaşıldığını incelemek amacıyla öğrencilerin PDABT son-testlere verdikleri cevaplar incelenmiştir. Testlere verilen cevaplar tek kazanımları içerebildiği gibi birden fazla kazanımı da içerdiği için birleştirilerek tablolastırılmıştır. Grupların verdiği cevapların son-test verileri Tablo 34 'te verilmektedir.

Tablo 34

DG1 ve DG2 Grubuna Ait PDABT Maddelere İlişkin Son-test Sonuçları

Konular	Madde numarası	DG1 (n=13)			DG2 (n=14)		
		Doğru		Yanlış	Doğru		Yanlış
		Sayı	Yüzde		Sayı	Yüzde	
Temel algoritma becerisi	M3	12	92,31	1	8	57,14	6
	M4	11	84,62	2	14	100	0
Döngü kavramı	M1	13	100	0	5	35,71	9
	M2	11	84,62	2	5	35,71	9
Hata ayıklama becerisi	M6	8	61,54	5	5	35,71	9
	M9	12	92,31	1	9	64,29	5
Koşul (Eğer) kavramı	M7	8	61,54	5	9	64,29	5
Operatör kavramı	M13	12	92,31	1	11	78,57	3
Koşul (Eğer değilse)	M5	10	76,92	3	9	64,29	5
Yazdırma becerisi	M10	12	92,31	1	10	71,43	4
Eksik tamamlama becerisi	M8	12	92,31	1	5	35,71	9
	M14	11	84,62	2	7	50,00	7
Değişken+Döngü+Koşul	M15	5	38,46	8	5	35,71	9
Döngü+Yazdırma becerisi	M16	11	84,62	2	6	42,86	8
Koşul(Eğer değilse)+Operatör	M12	9	69,23	4	7	50,00	7
Operatör+Döngü	M11	7	53,85	6	6	42,86	8

4.1.3.DG1 Grubuna Ait PDABT Ön-Test ve Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında DG1 grubunun PDABT ön-test ve son-test sonuçlarının arasında anlamlı bir fark olup olmadığı incelenmiştir. Ön-test ve son-test sonuçları arasında fark olup olmadığını incelemek amacıyla İlişkili Örneklemeler *t*-test kullanılmıştır. İlişkili Örneklemeler *t*-testi kullanabilmek için gerekli olan grupların normal dağılım gösterdiği Tablo 29 ve Tablo 32 incelendiğinde görülmektedir. İlişkili Örneklemeler *t*-test sonuçları Tablo 35’ te verilmiştir.

Tablo 35

DG1 Grubu PDABT Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları

DG1	PDABT					
	n	$\bar{X}$	SS	t	p	Korelasyon
Ön-Test	13	6,23	2,28	-8,856	,000	,496
Son-Test	13	12,62	2,81			

Tablo 35'te DG1 grubuna ait PDABT ön-test ve son-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}=12,62$, $SS=2,81$) ön-test puanlarının ortalamasından ($\bar{X}=6,23$, $SS=2,28$) daha yüksek olduğu görülmektedir. Ayrıca İlişkili Örneklemeler t-test sonuçlarına göre grupların PDABT açısından aralarında anlamlı bir fark olduğu görülmektedir ($t(12)=-8,856$, $p=,000<,05$).

4.1.4.DG2 Grubuna Ait PDABT Ön-Test ve Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında DG2 grubunun PDABT ön-test ve son-test sonuçlarının arasında anlamlı bir fark olup olmadığı incelenmiştir. Ön-test ve son-test sonuçları arasında fark olup olmadığını incelemek amacıyla İlişkili Örneklemeler t-test kullanılmıştır. İlişkili Örneklemeler t-testi kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 29 ve Tablo 32 incelendiğinde görülmektedir. İlişkili Örneklemeler t-test sonuçları Tablo 36' da verilmiştir.

Tablo 36

DG2 Grubu PDABT Ön-Test ve Son-Test İlişkili Örneklemeler t-test Sonuçları

DG1	PDABT					
	n	$\bar{X}$	SS	t	p	Korelasyon
Ön-Test	14	6,64	2,56	-1,516	,153	,034
Son-Test	14	8,43	3,67			

Tablo 36' da DG2 grubuna ait PDABT ön-test ve son-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}=8,43$, $SS=3,67$) ön-test puanlarının ortalamasından ($\bar{X}=6,64$, $SS=2,56$) daha yüksek olduğu görülmektedir. Ancak İlişkili

Örneklemler t -test sonuçlarına göre grupların PDABT açısından aralarında anlamlı bir fark olmadığı görülmektedir ($t(12)=-1,516$, $p=,153>,05$).

Grupların arasında ilişki incelendiğinde, DG1 grubuna ait Tablo 35' te verilen ön-test ve son- test puanlarına ait korelasyon değerinin ($,496$), DG2 grubuna ait Tablo 36'da verilen ön-test ve son test puanlarına ait korelasyon değerinden ($,034$) yüksek olduğu görülmektedir.

Uygulama esnasında verilen Scratch ve Kodu Game Lab programlama dilleri eğitiminin öğrenciler üzerinde hangi konuların daha anlaşıldığını incelemek amacıyla öğrencilerin PDABT ön-test ve son-testlere verdikleri cevaplar incelenmiştir. Grupların maddelere verdikleri cevaplar üzerindeki etkileri Tablo 37'de verilmektedir.

Tablo 37 PDABT Maddelerinin Ön-test ve Son-test Sonuçları
PDABT Maddelere İlişkin Ön-test ve Son-test Sonuçları

Konular	No	DG1 (n=13)				DG2 (n=14)			
		Ön-test		Son-test		Ön-test		Son-test	
		Doğru	Yüzde	Doğru	Yüzde	Doğru	Yüzde	Doğru	Yüzde
Temel algoritma becerisi	M3	4	30,77	12	92,31	6	42,86	8	57,14
	M4	12	92,31	11	84,62	10	71,43	14	100
Döngü kavramı	M1	2	15,38	13	100	4	28,57	5	35,71
	M2	2	15,38	11	84,62	5	35,71	5	35,71
Hata ayıklama becerisi	M6	4	30,77	8	61,54	4	28,57	5	35,71
	M9	6	46,15	12	92,31	7	50,00	9	64,29
Koşul (Eğer) kavramı	M7	1	7,69	8	61,54	2	14,29	9	64,29
Operatör kavramı	M13	8	61,54	12	92,31	10	71,43	11	78,57
Koşul (Eğer değilse)	M5	5	38,46	10	76,92	7	50,0	9	64,29
Yazdırma becerisi	M10	12	92,31	12	92,31	13	92,86	10	71,43
Eksik tamamlama becerisi	M8	5	38,46	12	92,31	5	35,71	5	35,71
	M14	1	7,69	11	84,62	2	14,29	7	50,00
Değişken+Döngü+Koşul	M15	5	38,46	5	38,46	4	28,57	5	35,71
Döngü+Yazdırma becerisi	M16	4	30,77	11	84,62	3	21,43	6	42,86
Koşul(Eğer değilse)+Operatör	M12	3	23,08	9	69,23	3	21,43	7	50,00
Operatör+Döngü	M11	7	53,85	6	53,85	6	42,86	6	42,86

4.1.5.DG1 ve DG2 Grubuna Ait PDABT Kalıcılık Yönünden Tekrar-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch ve Kodu Game Lab programlama dillerini öğretimi gerçekleştirdikten sekiz hafta sonra öğrencilerin ne kadar kalıcı öğrenmeler gerçekleştirdiğini incelemek amacıyla DG1 ve DG2 gruplarına tekrar-test olarak PDABT uygulanmış ve sonuçları analiz edilmiştir. İlk olarak tekrar test sonuçlarının normal dağılım gösterip göstermediğini incelemek amacıyla Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testinden sonra grupların arasında anlamlı fark olup olmadığını incelemek amacıyla üzere İlişkisiz Örneklemeler *t*-test uygulanmıştır. Shapiro-Wilk testine ait sonuçlar Tablo 38’de verilmiştir.

Tablo 38
PDABT Tekrar-Testlere Ait Shapiro-Wilk Sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	11,93	1,93	,931	13	,350
DG2	14	7,71	3,24	,879	14	,056

Shapiro Wilk testi sonuçları ($p > 0,05$) anlamlılık düzeyinde test edilmiş olup Tablo 38 incelendiğinde DG1 grubu ($p = ,350$) ve DG2 grubu ($p = ,056$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 tekrar test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklemeler *t*-test kullanılmıştır. İlişkisiz Örneklemeler *t*-test sonuçları Tablo 39’ da verilmiştir.

Tablo 39
PDABT Tekrar testlere Ait İlişkisiz Örneklemeler *t*-test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	11,92	1,93	25	4,127	,000
DG2	14	7,71	3,24			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 39’daki kalıcılık testi ön-test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}=11,92$, $SS=1,93$), DG2 grubuna ait

puanların ortalamasının ise ($\bar{X}=7,71$, $SS=3,24$) olduğu görülmektedir. Ortalamalar incelendiğinde DG1 grubunun ortalama puanlarının DG2 grubunun ortalama puanlarının da daha yüksek olduğu görülmektedir. Ayrıca İlişkisiz örneklem t -test sonuçları incelendiğinde grupların PDABT değişkeni açısından aralarında anlamlı bir fark olduğu görülmektedir ($t(25)=4,127$, $p=,000<,05$).

4.1.6.DG1 Grubuna Ait PDABT Kalıcılık Yönünden Son-Test ve Tekrar-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch programlama dillerini öğretimi gerçekleştirdikten sonra gerçekleştirilen son-test ve belli bir süre sonra öğrencilerin ne kadar kalıcı öğrenmeler gerçekleştirdiğini incelemek amacıyla PDABT tekrar-test olarak uygulanmış ve sonuçları analiz edilmiştir. Son-test ve tekrar-test sonuçları arasında fark olup olmadığını incelemek amacıyla İlişkili Örneklem t -test kullanılmıştır. İlişkili Örneklem t -testi kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 32 ve Tablo 38 incelendiğinde görülmektedir. İlişkili Örneklem t -testi sonuçları Tablo 40’ da verilmiştir.

Tablo 40 DG1 Grubu PDABT Son-Test ve Tekrar-Test İlişkili Örneklem t -test Sonuçları

DG1	PDABT					
	n	$\bar{X}$	SS	t	p	Korelasyon
Son-Test	13	12,62	2,81	1,559	,145	,836
Tekrar-Test	13	11,92	1,93			

Tablo 40’ ta DG1 grubuna ait PDABT son-test ve tekrar-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}=12,62$, $SS=2,81$) tekrar-test puanlarının ortalamasından ($\bar{X}=11,92$, $SS=1,93$) daha yüksek olduğu görülmektedir. Ancak ilişkili örneklem t -test sonuçlarına göre son-test ve tekrar-test sonuçları arasında PDABT değişkeni açısından aralarında anlamlı bir fark olmadığı görülmektedir ($t(12)=1,559$, $p=,145>,05$).

DG1 grubuna ait son-test ve tekrar-test arasındaki ilişki incelendiğinde, DG1 grubuna ait Tablo 40’ ta verilen son-test ve tekrar-test puanlarına ait korelasyon değerinin (,836) yüksek olduğu ve aralarında güçlü bir ilişki olduğu görülmektedir.

4.1.7.DG2 Grubuna Ait PDABT Kalıcılık Yönünden Son-Test ve Tekrar-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Kodu Game Lab programlama dillerini öğretimi gerçekleştirdikten sonra gerçekleştirilen son-test ve belli bir süre sonra öğrencilerin ne kadar kalıcı öğrenmeler gerçekleştirdiğini incelemek amacıyla tekrar test olarak PDABT uygulanmış ve sonuçları analiz edilmiştir. Son-test ve tekrar-test sonuçları arasında fark olup olmadığını incelemek amacıyla İlişkili Örneklemeler *t*-test kullanılmıştır. İlişkili Örneklemeler *t*-test kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 32 ve Tablo 38 incelendiğinde görülmektedir. İlişkili Örneklemeler *t*-testi sonuçları Tablo 41’ de verilmiştir.

Tablo 41

DG2 Grubu PDABT Son-Test ve Tekrar-Test İlişkili Örneklemeler t-test Sonuçları

DG2	PDABT					
	n	$\bar{X}$	SS	t	p	Korelasyon
Son-Test	14	8,43	3,67	1,072	,303	,747
Tekrar-Test	14	7,71	3,24			

Tablo 41’de DG2 grubuna ait PDABT son-test ve tekrar-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}$ =8,43, SS=3,67) tekrar-test puanlarının ortalamasından ($\bar{X}$ =7,71, SS=3,24) daha yüksek olduğu görülmektedir. Ancak İlişkili Örneklemeler *t*-test sonuçlarına göre son-test ve tekrar-test sonuçları arasında PDABT değişkeni açısından aralarında anlamlı bir fark olmadığı görülmektedir ($t(13)=1,072$, $p=,303>,05$).

DG2 grubuna ait son-test ve tekrar-test arasındaki ilişki incelendiğinde, DG2 grubuna ait Tablo 41’de verilen son-test ve tekrar-test puanlarına ait korelasyon değerinin (,747) yüksek olduğu ve aralarında güçlü bir ilişki olduğu görülmektedir.

Grupların arasında ilişki incelendiğinde, DG1 grubuna ait Tablo 40’ ta verilen son-test ve tekrar-test puanlarına ait korelasyon değerinin (,870), DG2 grubuna ait Tablo 41’de verilen son-test ve tekrar-test puanlarına ait korelasyon değerinden (,747) yüksek olduğu görülmektedir.

4.2. Programlamaya Yönelik Tutumun İncelenmesi

4.2.1.EBODKÖYTÖ Ön-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch (DG1) ve Kodu Game Lab (DG2) programlama dillerini öğretimi gerçekleştirilmeden EBODKÖYTÖ ön-test olarak uygulanmış olup deney gruplarının tutumları analiz edilmiştir. İlk olarak Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ) ön-testlerine ait verilerinin normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testinden sonra grupların eşit düzeyde olup olmadığını belirlemek üzere İlişkisiz Örneklem t-test uygulanmıştır. Shapiro-Wilk testine ait sonuçlar Tablo 42’de verilmiştir.

Tablo 42 EBODKÖYTÖ ön-testlere ait Shapiro-Wilk sonuçları
EBODKÖYTÖ Ön testlere ait Shapiro-Wilk Sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	3,30	,28	,968	13	,863
DG2	14	3,04	,44	,929	14	,297

Shapiro Wilk testi sonuçları ($p > 0,05$) anlamlılık düzeyinde test edilmiş olup Tablo 42 incelendiğinde DG1 grubu ($p = ,863$) ve DG2 grubu ($p = ,297$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 ön-test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklem t-test kullanılmıştır. İlişkisiz Örneklem t-test sonuçları Tablo 43’ te verilmiştir.

Tablo 43
EBODKÖYTÖ Ön-testlere Ait İlişkisiz Örneklem t-test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	3,30	,28	25	1,845	,077
DG2	14	3,04	,44			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 43' teki tutum testi ön-test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}$ =3,30, SS=,28), DG2 grubuna ait puanların ortalamasının ise ($\bar{X}$ =3,04, SS=,44) olduğu görülmektedir. Ortalamalar incelendiğinde DG1 ve DG2 gruplarının ortalama puanlarının birbirlerine yakın olduğu görülmektedir. Ayrıca İlişkisiz Örneklemeler *t*-test sonuçları incelendiğinde grupların EBODKÖYTÖ değişkeni açısından anlamlı bir fark olmadığı görülmektedir ($t(25)=1,845$, $p=,77>,05$).

4.2.2.EBODKÖYTÖ Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch ve Kodu Game Lab programlama dillerini öğretimi gerçekleştikten sonra EBODKÖYTÖ son-test olarak uygulanmış olup DG1 ve DG2 gruplarının tutumları analiz edilmiştir. İlk olarak Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği (EBODKÖYTÖ) son-testlerine ait verilerinin normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testinden sonra grupların eşit düzeyde olup olmadığını belirlemek üzere İlişkisiz Örneklemeler *t*-test uygulanmıştır. Shapiro-Wilk testine ait sonuçlar Tablo 44' te verilmiştir.

Tablo 44

EBODKÖYTÖ Son testlere ait Shapiro-Wilk Sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	3,52	,41	,947	13	,549
DG2	14	3,59	,44	,938	14	,394

Shapiro-Wilk testi sonuçları ($p > 0,05$) anlamlılık düzeyinde test edilmiş olup Tablo 44 incelendiğinde DG1 grubu ($p =,549$) ve DG2 grubu ($p =,394$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 son test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklemeler *t*-test kullanılmıştır. İlişkisiz Örneklemeler *t*-test sonuçları Tablo 45' te verilmiştir.

Tablo 45

EBODKÖYTÖ Son-testlere Ait İlişkisiz Örneklem t-test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	3,52	,41	25	-,442	,662
DG2	14	3,59	,44			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 45’ teki tutum testi ön-test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}$ =3,52, SS=,41), DG2 grubuna ait puanların ortalamasının ise ($\bar{X}$ =3,59, SS=,44) olduğu görülmektedir. Ortalamalar incelendiğinde DG1 ve DG2 gruplarının ortalama puanlarının birbirlerine yakın olduğu görülmektedir. Ayrıca İlişkisiz Örneklem t-test sonuçları incelendiğinde grupların EBODKÖYTÖ değişkeni açısından aralarında anlamlı bir fark olmadığı görülmektedir ($t(25)$ =-,442, p =,662>,05).

4.2.3.DG1 Grubuna Ait EBODKÖYTÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında DG1 grubunun EBODKÖYTÖ ön-test ve son-test sonuçlarının arasında anlamlı bir fark olup olmadığı incelenmiştir. Ön-test ve son-test sonuçları arasında fark olup olmadığını incelemek amacıyla İlişkili Örneklem t-test kullanılmıştır. İlişkili Örneklem t-test kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 42 ve Tablo 44 incelendiğinde görülmektedir. İlişkili Örneklem t-test sonuçları Tablo 46’da verilmiştir.

Tablo 46

DG1 Grubu EBODKÖYTÖ Ön-Test ve Son-Test İlişkili Örneklem t-test Sonuçları

DG1	EBODKÖYTÖ					
	n	$\bar{X}$	SS	t	p	Korelasyon
Ön-Test	13	3,30	,28	-1,879	,085	,323
Son-Test	13	3,52	,41			

Tablo 46’ da DG1 grubuna ait EBODKÖYTÖ ön-test ve son-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}$ =3,51, SS=,41) ön-test

puanlarının ortalamasından ($\bar{X}=3,30$, $SS=,28$) daha yüksek olduğu görülmektedir. Ancak İlişkili Örneklemeler t -test sonuçlarına göre grupların EBODKÖYTÖ değişkeni açısından aralarında anlamlı bir fark olmadığı görülmektedir ($t(12)=-1,879$, $p=,085>,05$).

4.2.4.DG2 Grubuna Ait EBODKÖYTÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında DG2 grubunun EBODKÖYTÖ ön-test ve son-test sonuçlarının arasında anlamlı bir fark olup olmadığı incelenmiştir. Ön-test ve son-test sonuçları arasında fark olup olmadığını incelemek amacıyla İlişkili Örneklemeler t -test kullanılmıştır. İlişkili Örneklemeler t -test kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 42 ve Tablo 44 incelendiğinde görülmektedir. İlişkili Örneklemeler t -test sonuçları Tablo 47’ de verilmiştir.

Tablo 47 DG2 Grubu EBODKÖYTÖ Ön-Test ve Son-Test İlişkili Örneklemeler t -test Sonuçları

DG2	EBODKÖYTÖ					
	n	$\bar{X}$	SS	t	p	Korelasyon
Ön-Test	14	3,04	,44	-3,898	,002	,273
Son-Test	14	3,59	,44			

Tablo 47’de DG2 grubuna ait EBODKÖYTÖ ön-test ve son-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}=3,59$, $SS=,44$) ön-test puanlarının ortalamasından ($\bar{X}=3,04$, $SS=,44$) daha yüksek olduğu görülmektedir. Ayrıca İlişkili Örneklemeler t -test sonuçlarına göre grupların EBODKÖYTÖ açısından aralarında anlamlı bir fark olduğu görülmektedir ($t(13)=-3,898$, $p=,002<,05$).

Grupların arasında ilişki incelendiğinde, DG1 grubuna ait Tablo 46’da verilen ön-test ve son- test puanlarına ait korelasyon değerinin (,323), DG2 grubuna ait Tablo 47’de verilen ön-test ve son test puanlarına ait korelasyon değerinden (,273) yüksek olduğu görülmektedir.

4.3. Programlamaya Yönelik Öz-Yeterlik Algısının İncelenmesi

4.3.1.PIÖYÖ Ön-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch (DG1) ve Kodu Game Lab(DG2) programlama dillerini öğretimi gerçekleştirilmeden PIÖYÖ ön-test olarak uygulanmış olup deney gruplarının öz-yeterlik algıları analiz edilmiştir. İlk olarak Programlamaya İlişkin Öz Yeterlik Ölçeği (PIÖYÖ) ön-testlerine ait verilerinin normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testinden sonra grupların eşit düzeyde olup olmadığını belirlemek üzere İlişkisiz örneklem t-test uygulanmıştır. Shapiro-Wilk testine ait sonuçlar Tablo 48’de verilmiştir.

Tablo 48

PIÖYÖ Ön testlere ait Shapiro-Wilk Sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	1,84	,46	,954	13	,658
DG2	14	2,23	,59	,952	14	,594

ShapiroWilk testi sonuçları ($p > 0.05$) anlamlılık düzeyinde test edilmiş olup Tablo 48 incelendiğinde DG1 grubu ($p =,658$) ve DG2 grubu ($p =,594$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 ön-test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklem t-test kullanılmıştır. İlişkisiz Örneklem t-test sonuçları Tablo 49’ da verilmiştir.

Tablo 49

PIÖYÖ Ön-testlere Ait İlişkisiz Örneklem t-test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	1,84	,46	25	-1,924	,066
DG2	14	2,23	,59			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 49'daki öz-yeterlik testi ön-test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}=1,84$, $SS=,46$), DG2 grubuna ait puanların ortalamasının ise ($\bar{X}=2,23$, $SS=,59$) olduğu görülmektedir. Ortalamalar incelendiğinde DG2 grubunun ortalama puanlarının DG1 grubunun ortalama puanlarından daha yüksek olduğu görülmektedir. Ancak İlişkisiz Örneklemeler t -test sonuçları incelendiğinde grupların PİÖYÖ değişkeni açısından aralarında anlamlı bir fark olmadığı görülmektedir ($t(25)=-1,924$, $p=,066>,05$).

4.3.2.PİÖYÖ Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında Scratch ve Kodu Game Lab programlama dillerini öğretimi gerçekleştikten sonra PİÖYÖ son-test olarak uygulanmış olup DG1 ve DG2 gruplarının öz-yeterlik algıları analiz edilmiştir. İlk olarak Programlamaya İlişkin Öz Yeterlik Ölçeği (PİÖYÖ) son-testlerine ait verilerinin normal dağılım gösterip göstermediğini incelemek için Shapiro-Wilk normallik testi uygulanmıştır. Shapiro-Wilk testinden sonra grupların eşit düzeyde olup olmadığını belirlemek üzere İlişkisiz Örneklemeler t -test uygulanmıştır. Shapiro-Wilk testine ait sonuçlar Tablo 50'de verilmiştir.

Tablo 50 PİÖYÖ testine ait Shapiro-Wilk sonuçları

Grup	n	$\bar{X}$	SS	Shapiro-Wilk		
				İstatistik	sd	p
DG1	13	4,73	1,45	,955	13	,669
DG2	14	4,78	1,39	,938	14	,390

Shapiro-Wilk testi sonuçları ($p > 0.05$) anlamlılık düzeyinde test edilmiş olup Tablo 50 incelendiğinde DG1 grubu ($p =,669$) ve DG2 grubu ($p =,390$) değerlerinin normal dağılım gösterdiği görülmektedir. Normal dağılım gösteren testlerde DG1 ve DG2 son test grupları arasında grup seviyelerinin eşit düzeyde olup olmadığını incelemek amacıyla parametrik testlerden İlişkisiz Örneklemeler t -test kullanılmıştır. İlişkisiz Örneklemeler t -test sonuçları Tablo 51' de verilmiştir.

Tablo 51

PİÖYÖ Son-testlere Ait İlişkisiz Örneklem t-test Sonuçları

Grup	n	$\bar{X}$	SS	sd	t	p
DG1	13	4,73	1,45	25	-,094	,926
DG2	14	4,78	1,39			

DG1 ve DG2 gruplarındaki öğrencilerin Tablo 51'deki öz-yeterlik testi son test puanları incelendiğinde DG1 grubuna ait puanların ortalaması ($\bar{X}$ =4,73, SS=1,45), DG2 grubuna ait puanların ortalamasının ise ($\bar{X}$ =4,78, SS=1,39) olduğu görülmektedir. Ortalamalar incelendiğinde DG1 ve DG2 gruplarının ortalama puanlarının birbirlerine yakın olduğu görülmektedir. Ayrıca İlişkisiz Örneklem t-test sonuçları incelendiğinde grupların PİÖYÖ değişkeni açısından anlamlı bir fark olmadığı görülmektedir ($t(25)$ =-,094, p =,926>,05).

4.3.3.DG1 Grubuna Ait PİÖYÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında DG1 grubunun PİÖYÖ ön-test ve son-test sonuçlarının arasında anlamlı bir fark olup olmadığı incelenmiştir. Ön-test ve son-test sonuçları arasında anlamlı bir fark olup olmadığını incelemek amacıyla İlişkili Örneklem t-test kullanılmıştır. İlişkili Örneklem t-test kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 48 ve Tablo 50 incelendiğinde görülmektedir. İlişkili Örneklem t-test sonuçları Tablo 51' de verilmiştir.

Tablo 52

DG1 Grubu PİÖYÖ Ön-Test ve Son-Test İlişkili Örneklem t-test Sonuçları

DG1	PİÖYÖ					
	n	$\bar{X}$	SS	t	p	Korelasyon
Ön-Test	13	1,83	,46	-6,443	,000	-,226
Son-Test	13	4,73	1,45			

Tablo 52'de DG1 grubuna ait PİÖYÖ ön-test ve son-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}$ =4,73, SS=1,45) ön-test puanlarının ortalamasından ($\bar{X}$ =1,83, SS=,46) daha yüksek olduğu görülmektedir. Ayrıca İlişkili

Örneklemler *t*-test sonuçlarına göre grupların PİÖYÖ açısından aralarında anlamlı bir fark olduğu görülmektedir ($t(12)=-6,443$, $p=,000<,05$).

4.3.4.DG2 Grubuna Ait PİÖYÖ Ön-Test ve Son-Test Sonuçlarının İncelenmesi

Araştırmanın bu kısmında DG2 grubunun PİÖYÖ ön-test ve son-test sonuçlarının arasında anlamlı bir fark olup olmadığı incelenmiştir. Ön-test ve son-test sonuçları arasında anlamlı bir fark olup olmadığını incelemek amacıyla İlişkili Örneklemler *t*-test kullanılmıştır. İlişkili Örneklemler *t*-test kullanabilmek için gerekli olan grupların normal dağılım gösterdiği, Tablo 48 ve Tablo 50 incelendiğinde görülmektedir. İlişkili Örneklemler *t*-test sonuçları Tablo 4.3.4.1' de verilmiştir.

Tablo 53

DG2 Grubu PİÖYÖ Ön-Test ve Son-Test İlişkili Örneklemler t-test Sonuçları

DG2	PİÖYÖ					
	n	$\bar{X}$	SS	t	p	Korelasyon
Ön-Test	14	2,23	,59	-6,894	,000	,230
Son-Test	14	4,78	1,39			

Tablo 53' te DG2 grubuna ait PİÖYÖ ön-test ve son-test puanlarının ortalamaları incelendiğinde son-test puanlarının ortalaması ($\bar{X}=4,78$, $SS=1,39$) ön-test puanlarının ortalamasından ($\bar{X}=2,23$, $SS=,59$) daha yüksek olduğu görülmektedir. Ayrıca İlişkili Örneklemler *t*-test sonuçlarına göre grupların PİÖYÖ açısından aralarında anlamlı bir fark olduğu görülmektedir ($t(13)=-6,894$, $p=,000<,05$).

Grupların arasında ilişki incelendiğinde, DG1 grubuna ait Tablo 52'de verilen ön-test ve son- test puanlarına ait korelasyon değerinin (-,226), DG2 grubuna ait Tablo 53' te verilen ön-test ve son test puanlarına ait korelasyon değerinden (,230) düşük olduğu görülmektedir.

BÖLÜM V

5. SONUÇ,TARTIŞMA ve ÖNERİLER

5.1. SONUÇ

Bu çalışmada Scratch ve Kodu Game Lab programlama dilleri öğretiminin öğrencilerin programlama başarısına, programlamaya yönelik tutumlarına ve öz-yeterlik algılarına etkileri incelenmiştir. Toplanan veriler ve yapılan analizlere göre bu çalışmadan elde edilen sonuçlar aşağıdaki gibidir:

- Scratch ve Kodu Game Lab programlama dili ile öğrenim gören öğrencilerin, başarı testi ön-test puanları arasında anlamlı bir fark olmadığı sonucuna varılmıştır.
- Scratch programlama dili ile öğrenim gören deney grubundaki öğrencilerin, Kodu Game Lab programlama dili ile öğrenim gören öğrencilerin programlama başarısı arasında Scratch lehine anlamlı derecede fark olduğu sonucuna varılmıştır.
- Scratch ile programlama eğitimi alan öğrencilerin programlama dilleri başarısının anlamlı derecede arttığı sonucuna varılmıştır.
- Kodu Game Lab ile programlama eğitimi alan öğrencilerin programlama dili başarısının arttığı ancak bu artışın anlamlı derecede yüksek olmadığı sonucuna varılmıştır.
- Scratch ile programlama eğitimi alan öğrencilerin, Kodu Game Lab programlama dili ile öğrenim gören öğrencilerle kalıcılık yönünden aralarında Scratch lehine anlamlı bir fark olduğu sonucuna varılmıştır.
- Scratch ile programlama eğitimi alan öğrencilerin programlama dilleri başarısının son-teste göre kalıcılık yönünden azaldığı ancak bu etkilenmenin anlamlı derecede düşük olmadığı sonucuna varılmıştır.
- Kodu Game Lab ile programlama eğitimi alan öğrencilerin programlama dili başarısının son-teste göre kalıcılık yönünden azaldığı ancak bu etkilenmenin anlamlı derecede düşük olmadığı sonucuna varılmıştır.
- Scratch ve Kodu Game Lab programlama dili ile öğrenim gören öğrencilerin, tutum ön-test puanları arasında anlamlı bir fark olmadığı sonucuna varılmıştır.

- Scratch programlama dili ile öğrenim gören deney grubundaki öğrencilerin, Kodu Game Lab programlama dili ile öğrenim gören öğrencilerin tutumu arasında anlamlı derecede bir fark olmadığı sonucuna varılmıştır.
- Scratch ile programlama eğitimi alan öğrencilerin programlama dillerine yönelik tutumun arttığı ancak bu artışın anlamlı derecede yüksek olmadığı görülmüştür.
- Kodu Game Lab ile programlama eğitimi alan öğrencilerin programlama dillerine yönelik tutumun anlamlı derecede arttığı sonucuna varılmıştır.
- Scratch ve Kodu Game Lab programlama dili ile öğrenim gören öğrencilerin,öz-yeterlik ön-test puanları arasında anlamlı bir fark olmadığı sonucuna varılmıştır.
- Scratch programlama dili ile öğrenim gören deney grubundaki öğrencilerin, Kodu Game Lab programlama dili ile öğrenim gören öğrencilerin öz-yeterlik algısı arasında anlamlı derecede bir fark olmadığı sonucuna varılmıştır.
- Scratch ile programlama eğitimi alan öğrencilerin programlama dillerine yönelik öz-yeterlik algılarının anlamlı derecede arttığı sonucuna varılmıştır.
- Kodu Game Lab ile programlama eğitimi alan öğrencilerin programlama dillerine yönelik öz-yeterlik algılarının anlamlı derecede arttığı sonucuna varılmıştır.

5.2. TARTIŞMA

Araştırmanın bu kısmında Scratch ve Kodu Game Lab ile programlama öğrenimi gören öğrencilerin başarı, tutum ve öz-yeterlik algılarına ait sonuçlarla alan yazında olan çalışmalar arasındaki ilişkiler incelenerek tartışılmıştır.

5.2.1. Scratch ve Kodu Game Lab ile programlama öğretiminin programlama dilleri akademik başarısına ve kalıcılığa etkisi

Araştırmanın sonuçları incelendiğinde Scratch (DG1) ve Kodu Game Lab (DG2) ile programlama öğrenimi gören öğrencilerin başarılarının artış gösterdiği görülmektedir. Kodu Game Lab ile programlama öğrenimi gören öğrencilerin başarıları artmış olsa da bu artışın anlamlı düzeyde gerçekleşmediği, Scratch ile programlama öğrenimi gören öğrencilerin başarılarının ise anlamlı derecede artış gösterdiği görülmektedir. Scratch ile programlama öğreniminin başarı üzerine olumlu etki ettiğine ilişkin alan yazında benzer çalışmalarla paralellik gösterdiği görülmektedir (Demir, 2015; Erol, 2015; Fesakis ve Serafeim, 2009; Osman, Zakaria, Loke ve Downe, 2012; Papadakis, Kalogiannakis, Zaranis ve Orfanakis,

2016; Yıldırım, 2017; Yüksel, 2017). Bu bulgudan yola çıkılarak programlama öğretiminin Scratch ile gerçekleştirilmesinin programlama öğrenimi üzerinde daha etkili olduğu sonucuna ulaşılabilir. Bu durumda Scratch'ın bilgi işlemsel düşünme becerisini olumlu yönde etkilediği (Ozoran, Çağıltay ve Topallı, 2012; Yünkül, Durak, Çankaya Mısırlı, 2017; Yıldırım, 2016) bilgi işlemsel düşünme becerilerinin yüksek olmasının programlama başarısını geliştirmiş olabileceği (Yünkül, Durak, Çankaya ve Mısırlı, 2017) söylenebilir. Bilgi işlemsel düşünme becerilerinin alt boyutlarından biri olan problem çözme becerilerinin programlama başarısı üzerinde etkisi olduğu da bir gerçektir. Kodu Game Lab'ın problem çözme becerisi üzerine etki ettiği Hava (2016) üstün yetenekli öğrencilerle yaptığı bir çalışmada görülmüştür. Scratch'ın problem çözme üzerinde herhangi bir etkiye sahip olmadığı sonucuna ulaşan az sayıda çalışma olsa da (Gülbahar ve Kalelioğlu, 2014; Mladenović, Krpan ve Mladenović, 2016), Scratch ile programlama eğitimi üzerine yapılan çalışmaların çoğunda problem çözme üzerinde olumlu etkisi olduğu ortaya çıkmıştır (Shin ve Park, 2014; Taylor, Harlow ve Forret, 2010; Nam, Kim ve Lee, 2010; Yıldırım, 2016; Yünkül, Durak, Çankaya ve Mısırlı, 2017).

Yapılan bazı çalışmalarda Scratch'ın öğrencilerin birbirleriyle olan etkileşimlerini arttırdığı (Lopez, Gonzalez ve Cano, 2016; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Zavala, Gallardo ve Ruíz, 2013) görülmüştür. Öğrencilerin etkileşiminin artması, birbirleriyle bilgi alışverişi yapmalarını kolaylaştırabilir ve öğrenciler akran öğrenimi gerçekleştirerek programlama başarıları üzerinde olumlu bir etkiye sahip olduğu savunulabilir. Ayrıca Scratch programlama dili ile programlama öğretiminin işbirlikli çalışmayı olumlu yönde etkilediği de yapılan bazı çalışmalarda görülmektedir (Papatğa, 2016; Taylor, Harlow ve Forret, 2010).

Scratch programlama dili kullanımının (Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011; Gülbahar ve Kalelioğlu, 2014; Malan ve Leitner, 2007; Yükseltürk ve Altıok, 2016) ve öğreniminin de kolay olmasının (Genç ve Karaguş, 2011; Lewis, 2010; Malan ve Leitner, 2007; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Nam, Kim ve Lee, 2010; Yıldırım, 2016) programlama başarısını da arttığı sonucuna ulaşılabilir.

Bu çalışmada Scratch ile programlama eğitimi alan öğrencilerin kalıcı öğrenmeler gerçekleştirdiğini söyleyebiliriz. Yıldırım (2016) yaptığı çalışmada Scratch'ın kalıcılığa olumlu yönde etki ettiğine ilişkin bir çalışma da gerçekleştirmiştir. Yapılan çalışmada Scratch kalıcılık puanlarının sekiz hafta sonra anlamlı bir azalma olmadığı ve kalıcı öğrenmeler sağladığı savunulabilir. Bu çalışmayla Yıldırım (2016) çalışmalarının bu yönden paralellik gösterdiği, kalıcılık testi puanlarının yüksek olduğu ve öğrencilerin konuya hakim

olduğu söylenebilir. Ayrıca yapılan başka bir çalışmada Yüksel (2017), Scratch ile programlama eğitimi alan öğrencilerin kalıcı öğrenmeler gösterdiği sonucuna ulaşmıştır. Yapılan çalışma bu yönden Yüksel (2017)'nin yaptığı çalışmayla da örtüştüğü görülmektedir. Yüksel (2017)'nin yaptığı çalışmada Scratch programlama dilleri öğretimi Ayrılık Birleşme Tekniği kullanarak gerçekleştirdiği görülmüştür. Bundan yola çıkılarak Scratch ile programlama öğretiminde farklı öğretim tekniklerini kullanarak kalıcılığı arttırılabileceği şeklinde yorumlanabilir. Kodu Game Lab ile programlama eğitimi alan öğrencilerin programlama dili başarısının son-teste göre kalıcılık yönünden azaldığı ancak bu etkilenmenin anlamlı olmadığı da ortaya çıkmıştır. Bu durumda Kodu Game Lab programlama dilinin de kalıcı öğrenmeler sağladığını söyleyebiliriz.

Alan yazında Kodu Game Lab programlama dilinin başarı üzerine olan etkisiyle ilgili bir çalışmaya rastlanmamıştır.

5.2.2.Scratch ve Kodu Game Lab ile programlama öğretiminin programlama dilleri tutuma etkisi

Araştırmanın sonuçları incelendiğinde Scratch (DG1) ve Kodu Game Lab (DG2) ile programlama öğrenimi gören öğrencilerin programlamaya yönelik tutumlarının olumlu yönde değiştiği görülmektedir. İki deney grubu arasında tutum değişkeni açısından aralarında anlamlı bir farklılaşma olmadığı da ortaya çıkmıştır. Ancak iki deney grubunda da tutum olumlu yönde değişse de sadece Kodu Game Lab programlama dilleri ile bu artışın anlamlı derecede olduğu görülmektedir. Scratch ile programlama öğretiminde tutum değişkeni açısından anlamlı bir değişme olmasa da, Scratch'ın programlamaya karşı tutumu olumlu yönde etkilediği alan yazında yapılan (Baltalı, 2016; Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011; Malan ve Leitner, 2007; Ozoran, Çağıltay ve Topallı, 2012;Yıldırım, 2016) çalışmalarla da örtüşmektedir. Ancak araştırmanın genel sonucu incelendiğinde tutum değişkeni açısından Kodu Game Lab'ın programlamaya karşı tutumu arttırmada daha etkili olduğunu söylenilebilir. Çalışmanın bu sonucu Fowler (2012)'nin yaptığı çalışmayla da paralellik göstermektedir. Bu çalışmanın sonuçları incelendiğinde, Kodu Game Lab ile programlama eğitimi alan öğrencilerin mutlu oldukları, can sıkıntılarının azaldığı, hayal kırıklığına uğrama seviyesinin azaldığı ve görevleri yapma isteklerini arttırdığı (Fowler, 2012), bu sayede de programlamaya yönelik tutumun anlamlı derecede artmış olabileceği söylenilebilir. Yapılan ön çalışmada öğretmenlerin programlama dili tercihler sebepleri incelendiğinde daha çok görsel öğeler barındırdığı için programlama dili seçtikleri görülmektedir. Kodu Game Lab programlama dilinin de oyun tabanlı bir yapıya sahip olması

(Fowler, 2012) ayrıca tasarım ve görsellik açısından da Scratch programlama diline göre daha başarılı olmasının (Baltalı, 2016) öğrencilerin tutumları üzerinde anlamlı derecede etki ettiği söylenilebilir. Ayrıca Kodu Game Lab programlama aracının yapısının oyun temelli olması programlamayı öğrencilere oyun gibi göstermiş olabilir. Programlama sürecinde oyunlaştırmanın öğrencilerin tutumları üzerinde etkili olabileceği yönünde yapılan çalışmalar da mevcuttur (Yükseltürk ve Altıok, 2016; Yıldırım, 2016). Kodu Game Lab'ın tutumu olumlu yönde etkileyip, akademik başarıyı olumlu yönde etkilemesinin sebebinde, Kodu Game Lab'ın yapısının hareketli ve oyun temelli olmasının programlama öğrenmenin önüne geçmiş olabileceğinden kaynaklandığı düşünülebilir.

Scratch ile programlama öğretiminin, öğrenme ortamının eğlenceli olmasına katkı sağladığına (Gallardo ve Ruíz, 2013; Lopez, Gonzalez ve Cano, 2016) yönelik çalışmalar vardır. Gülbahar ve Kalelioğlu (2014), Scratch programlama dili kullanmanın ve öğreniminin kolay olmasını öğrencilerin programlamaya karşı olan isteklerini arttırdığı sonucuna ulaşmışlardır. Ayrıca Scratch'ın programlamaya karşı isteklerini arttırdığına ilişkin Kobsiripat (2014)'ün yaptığı bir çalışmada da benzer sonuçlarla paralellik gösterdiği görülmüştür. Scratch öğrencilerin programlama öğrenme isteklerini olumlu etkilese de Gülbahar ve Kalelioğlu (2014) yaptıkları çalışmada programlamaya yönelik motivasyonları üzerinde bir etkiye sahip olmadığı yönünde bir sonuca ulaşmışlardır. Yapılan bu çalışma, Gülbahar ve Kalelioğlu (2014) ve Kobsiripat (2014)'ün yaptığı çalışmalarla benzer sonuçlara sahip olduğu görülmektedir.

5.2.3. Scratch ve Kodu Game Lab ile programlama öğretiminin programlama dilleri öz-yeterlik algısına etkisi

Araştırmanın sonuçları incelendiğinde Scratch (DG1) ve Kodu Game Lab (DG2) ile programlama öğrenimi gören öğrencilerin programlamaya karşı olan öz-yeterlik algılarının arttığı görülmüştür. Bu artış her iki deney grubu için de anlamlı seviyede gerçekleşmiştir. Grupların birbirleri arasında bir farklılaşma olmadığı da görülmüştür. Her iki programlama aracının da öz-yeterlik algısını geliştirmesi açısından etkili olduğu sonucuna ulaşılabilir. Yapılan bu çalışma, Yükseltürk ve Altıok (2016)'un eğitsel oyun geliştirme sürecinde öğrencilerin kendilerini başarılı gördükleri yönündeki öz-yeterlik algılarıyla benzerlik gösterdiği görülmüştür. Programlama öğretimini Scratch gerçekleşmesinin öğrencilerin öz güvenlerini de arttırabileceği söylenebilir. Hem Scratch hem de Kodu Game Lab

programlama aracının programlama dili öğretiminde öz-yeterlik algılarının arttırmasının sebepleri düşünüldüğünde bu durumun görsel programlama araçlarının özelliklerinden olduğu söylenilebilir. Görsel programlama dillerinin genel olarak kullanımının kolay olması ve karışık kod yapısının bulunmamasının (Fesakis ve Serafeim, 2009; Genç ve Karaguş, 2011; Gülbahar ve Kalelioğlu, 2014; Malan ve Leitner, 2007; Yükseltürk ve Altıok, 2016) ve öğrenmesinin de kolay olmasının (Genç ve Karaguş, 2011; Lewis, 2010; Malan ve Leitner, 2007; Maloney, Resnick, Rusk, Silverman ve Eastmond, 2010; Nam, Kim ve Lee, 2010; Yıldırım, 2016) öğrencilerinin hem programlama başarısını hem de öz-yeterlik algısını da olumlu yönde etkileyeceği düşünülmektedir.

Kodu Game Lab ile programlama başarının artmamasına rağmen öz-yeterlik algılarının anlamlı derecede artmasının sebebinin, tasarım ve görsellik açısından başarılı olmasından kaynaklanabileceği söylenilebilir (Baltalı, 2016). Çünkü Kodu Game Lab'ın yapısının hareketli, görsel öğeler yönünden zengin ve kod yazmak dışında ortamın da kolay bir şekilde düzenlenebilir bir yapıya sahiptir. Ancak bu durum öğrencileri odaklanmaları gereken asıl noktadan uzaklaştırmış olabilir. Öğrenciler, Kodu Game Lab'ı ile programlama öğretimini bu şekilde algılamış olabilir ve öz-yeterlik algıları bundan dolayı yüksek çıkmış olabilir.

Alan yazın incelendiğinde ortaokul seviyesinde blok temelli programlama dilleri ve oyun tabanlı programlama dillerinin programlama dillerine karşı öğrencilerin öz-yeterlik algılarının incelendiği çalışma olmadığı da görülmektedir (Kasalak ve Altun, 2018). Bu durumdan yola çıkılarak bu çalışmanın alan yazın için bir kaynak oluşturacağı da söylenilebilir.

5.3. ÖNERİLER

1. Alan yazındaki çalışmalar incelendiğinde, Kodu Game Lab ile yapılan çalışmaların sayısının az olduğu görülmüştür. Az sayıdaki çalışmanın genellikle öğrencilerin motivasyonları üzerine yapıldığı, akademik başarı ve öz-yeterlik açısından herhangi bir çalışma yapılmadığı görülmüştür. Bu alanda deneysel çalışmalar yapılması önerilebilir.
2. Ortaokul düzeyinde Scratch'ın öğrencilerin öz-yeterlik algılarına ilişkin az sayıda çalışma olduğu görülmüştür. Bu alanda deneysel çalışmalar yapılması önerilebilir.

3. Öğrencilerin hangi konularda zorlandıklarını, hangi etkenlerin öğrencilerin tutumları etkilediklerini daha detaylı öğrenmek için nitel bulgular eklenerek bir çalışma yapmaları önerilebilir.
4. Programlama dili öğretiminde aynı derste farklı programlama dillerinin kullanılmasının öğrencilerin tutum, başarı ve öz-yeterlik algıları üzerinde etkileri ile ilgili çalışmalar yapılabilir.
5. Çalışmada öğrencilerin başarı testine verdiği cevaplara göre bazı sorulara her iki grubunda aynı soru üzerinde yanlışlar yaptıkları görülmüştür. Bu durumun ortaokul öğrencilerinin programlama öğretiminde hangi kavram yanlışlarına sahip olduklarının, belirlenip giderilmesi yönünde çalışmalar yapmaları önerilebilir.
6. Çeşitli programlanabilir robotların, Scratch ile yapılan oyunların içerisine programlama öğretime entegre edilmesinin programlama başarısına, motivasyonuna, tutumuna ve öz-yeterlik algılarına olan etkileri üzerine nitel ve nicel çalışmalar yapmaları önerilebilir.
7. A-12 eğitiminde Scratch ve Kodu Game Lab programlama aracıyla programlama eğitiminin farklı bağımsız değişkenlerle başarı, tutum ve öz-yeterlik algıları açısından etkisinin inceleneceği çalışmalar yapılabilir.
8. Farklı yaş gruplarıyla çalışarak bu çalışma genişletilebilir.

KAYNAKÇA

21. Yüzyıl çocuklarının dili. (2017, 2 Şubat). 21. Yüzyıl çocuklarının dili: Kodlama. Erişim adresi: <https://www.bbc.com/turkce/haberler-dunya-38844062>.
- Ak, B. (2017).*Parametrik hipotez testleri*.Şeref Kalaycı (Ed.) . SPSS uygulamalı çok değişkenli istatistik teknikleri(8.Baskı)(s. 73-82) içinde. Ankara: Dinamik Akademi.
- Altun, A. ve Mazman, G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formunun geçerlilik ve güvenirlik çalışması.*Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3(2), 297- 308.
- Arabacıoğlu, T. , Bülbül, H. İ. ve Filiz, A. (2013, Ocak). Bilgisayar Programlama Öğretiminde Yeni Bir Yaklaşım. *Akademik Bilişim'07 - IX. Akademik Bilişim Konferansında sunulan bildiri*.Dumlupınar Üniversitesi, Kütahya. Erişim adresi: http://ab.org.tr/ab07/kitap/arabacioglu_bulbul_AB07.pdf.
- Atılğan, H.,Kan, A. ve Doğan, N. (2013). *Eğitimde ölçme ve değerlendirme* (6. Baskı). Ankara: Anı Yayıncılık.
- Baltalı, S. (2016). *Programlama öğretiminde kullanılabilecek yazılımlara ilişkin öğretmen görüşleri* (Yayımlanmamış yüksek lisans tezi). Uludağ Üniversitesi, Bursa.
- Barr, D., Harrison, J. ve Conery,L. (2011). Computational thinking: A digital age skill for everyone.*International Society for Technology in Education*, 20-23. Erişim adresi: <https://www.iste.org/docs/learning-and-leading-docs/march-2011-computational-thinking-ll386.pdf>.
- Bayman, P., ve Mayer, R. (1988). Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology*, 80(3),291-298. doi: 10.1037/0022-0663.80.3.291
- Bilgoritma (2017). Erişim adresi:<http://www.bilgoritma.org/>.
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., and Fontecchio, A. (2008, Ocak). Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom. *2008 Annual Conference & Expositionda sunulan bildiri. Pennsylvania, ABD*. Erişim adresi: <https://peer.asee.org/3826>.

- Büyüköztürk, Ş. (2017). *Sosyal bilimler için veri analizi el kitabı* (23. Baskı). Ankara:Pegem Akademi.
- Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö.E., Karadeniz, Ş. ve Demirel, F. (2008). *Bilimsel araştırma yöntemleri* (14. Baskı). Ankara: Pegem Yayınları.
- Cevahir, H., ve Özdemir, M. (2017, Mayıs). Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri. *Conference: 11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu sunulan bildiri*.İnönü Üniversitesi, Malatya.Erişim adresi: <http://icits2017.inonu.edu.tr/dosya/1493636695061912000.pdf>.
- Codecombat (2018). Erişim adresi: <https://codecombat.com/>.
- Codefest (2018). Erişim adresi: <https://www.codefesttr.org/>.
- Code.org (2018). Erişim adresi: <https://code.org/about>.
- Çatlak, Ş., Tekdal, M. ve Baz, F. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir doküman inceleme çalışması. *Journal of Instructional Technologies & Teacher Education*, 4(3),13-25. Erişim adresi:<http://dergipark.gov.tr/jitte/issue/25088/264774>.
- Çiçek, (2017) *Tanımlayıcı istatistikler*. Şeref Kalaycı (Ed.) . SPSS uygulamalı çok değişkenli istatistik teknikleri (8.Baskı)(s. 51-61) içinde. Ankara: Dinamik Akademi.
- Demir, F. (2015). *Programlama öğretiminde eğitsel programlama dilinin farklı kullanımlarının programlama başarısı ve kaygısına etkisi* (Yayımlanmamış doktora tezi). Atatürk Üniversitesi, Erzurum.
- Demirer, V. ve Sak, N. (2016). Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Deruy, E. (2017, 24 Şubat). In Finland, Kids Learn Computer Science Without Computers. Erişim adresi: <https://www.theatlantic.com/education/archive/2017/02/teaching-computer-science-without-computers/517548/>.
- Dünyanın en değerli 20 markası (2017, 2 Şubat). Dünyanın en değerli 20 markası. Erişim adresi: <https://www.ntv.com.tr/galeri/ekonomi/dunyanin-en-degerli-20-markasi>

2017,EFPyXrN7JECqLdMbeLcxg/3HsYqmhKUUqNuHxq96D9aA#3HsYqmhKUUqNuHxq96D9aA.

EBA (2018). Erişim adresi: <http://www.eba.gov.tr/kod/illerimiz>.

Erekmekçi, M.ve Fidan, Ş. (2012). Oyunun tasarım platformları: Oyunun eğitim ve kültüre etkisi. *Yaşam Bilimleri Dergisi* 1(1) 851-861. <http://www.yasambilimleridergisi.com/makale/pdf/1356289798.pdf> adresinden 2017 yılında erişilmiştir.

Erol, O. (2015). *Scratch ile programlama öğretiminin Bilişim Teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisi* (Yayımlanmamış doktora tezi). Anadolu Üniversitesi, Eskişehir.

Ersoy, H., Madran, R. O. ve Gülbahar, Y. (2011, Şubat). Programlama dilleri öğretimine bir model önerisi: Robot programlama. *Akademik Bilişim Konferansı 2011' de sunulan bildiri*, İnönü Üniversitesi, Malatya. Erişim adresi: ab.org.tr/ab11/kitap/ersoy_madran_AB11.pdf.

EuropeanSchoolnet (2015). Computing Our Future. Erişim adresi: http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0.

Fesakis, G., and Serafeim, K. (2009, Temmuz). Influence of the familiarization with "Scratch" on future teachers' opinions and attitudes about programming and ICT in education. *Paper presented at the Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education da sunulan bildiri*. Paris, Fransa. doi.org/10.1145/1595496.1562957.

Fidan, A. (2016). *Scratch ile programlama öğretiminde oyunlaştırmanın öğrenci katılımına etkisi* (Yayımlanmamış yüksek lisans tezi). Uludağ Üniversitesi, Bursa.

Field, A. (2009). *Discovering statistics using SPSS* (3. bs.). London: Sage Publications Ltd. [Adobe Digital Editions sürümü]. Erişim adresi: <https://faculty.psau.edu.sa/filedownload/doc-6-pdf/a46e10627f49a5333cdcd0b700790f8b-original.pdf> adresinden 2017 yılında erişilmiştir.

Fowler, A. (2012, Kasım). Enriching student learning programming through using Kodu. *3rd Annual Conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2012)'da sunulan bildiri*. Christchurch, New

Zealand. http://www.citrenz.ac.nz/conferences/2012/pdf/2012CITRENZ_Fowler01-Kodu.pdf

Genç, Z., ve Karakuş, S. (2011, Eylül). Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımında Scratch kullanımı. *5th International Computer & Instructional Technologies Symposiumda sunulan bildiri*. Fırat Üniversitesi, Elazığ. Erişim adresi: <http://www.bitliseren.edu.tr/Medya/YazarMedya/385/Tasar%C4%B1mla%20%C3%96%C4%9Frenme%20%20E%C4%9Fitel%20Bilgisayar%20Oyunlar%C4%B1%20Tasar%C4%B1m%C4%B1nda%20Scratch%20Kullan%C4%B1m%C4%B1.pdf>.

Günüç, S., Odabaşı, H.F. ve Kuzu, A., (2013). 21. yüzyıl öğrenci özelliklerinin öğretmen adayları tarafından tanımlanması: Bir Twitter uygulaması. *Eğitimde Kuram ve Uygulama*, 9(4), 436-455. Erişim adresi: <http://acikerisim.lib.comu.edu.tr:8080/xmlui/handle/COMU/1145>.

Gülbahar, Y. ve Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 13(1), 33-50. Erişim adresi: <http://files.eric.ed.gov/fulltext/EJ1064285.pdf>.

Güler, N. (2012). *Eğitimde ölçme ve değerlendirme* (4. Baskı). Ankara: Pegem A Yayıncılık.

Hackercan (2017). Erişim adresi: <http://www.hackercan.com/tr/>.

ISTE (2018).ISTE Standards for Students. Erişim adresi: <http://www.iste.org/standards/for-students>.

Karasar, N. (2005). *Bilimsel araştırma yöntemi*. Ankara: Nobel Yayıncılık.

Kasalak, İ. ve Altun, A. (2018). Blok temelli programlamaya ilişkin öz-yeterlik algısı ölçeği geliştirme çalışması: Scratch örneği. *Eğitim Teknolojisi Kuram ve Uygulama*, 8(1),209-225. doi: 10.17943/etku.335916.

Keçeci, G., Alan, B. ve Zengin, F. (2016). Eğitsel bilgisayar oyunları destekli kodlama öğrenimine yönelik tutum ölçeği: Geçerlilik ve güvenirlik çalışması. *Education Sciences (NWSAES)*, 11(3),184-194. doi.org/10.12739/NWSA.2016.11.3.1C0661.

Kevser, H.(2016). *Tasarım tabanlı öğrenme ortamında bilgisayar oyunu geliştirme etkinliğinin öğretim sürecinde kullanılması* (Yayımlanmamış doktora tezi). Gazi Üniversitesi, Ankara.

- Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia-Social and Behavioral Sciences*, 174, 227–232. doi.org/10.1016/j.sbspro.2015.01.651.
- Kodsaati (2018). Erişim adresi: <https://hourofcode.com/tr>.
- Kodu (2017). Erişim adresi: <https://www.kodugamelab.com/about/>.
- Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A. ve Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *OMÜ Eđt. Fak. Derg. / OMU J. Fac. Educ.*, 34(2),68-87. Erişim adresi: <http://dergipark.ulakbim.gov.tr/omuefd/article/view/5000163470>.
- Kukul, V. ve Gökçearsan, Ş. (2014, Kasım). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. 8. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumunda sunulan bildiri*.Trakya Üniversitesi, Edirne.doi: 10.13140/RG.2.1.4563.7606.
- Küçüköđlü, Ö. (2014). 21.yüzyıl becerileri için en etkin oluşumlar: “Kod okur-yazarlığı” ve “FLL”. Erişim adresi. <http://www.egitimdeteknoloji.com/21yuzyil-becerileri-kod-okur-yazarligi-ve-fll/>.
- Lewis, C., M. (2010, Mart).How programming environment shapes perception, learning and goals: Logo vs. Scratch. *41st ACM Technical Symposium On Computer Science Education, SIGCSE 2010 da sunulan bildiri. Milwaukee, ABD*.<https://doi.org/10.1145/1734263.1734383>.
- Lopez, J. M., Gonzalez, M. R. ve Cano, E. V. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools.*Computers & Educations*,97, 129-141. doi.org/10.1016/j.compedu.2016.03.003
- Malan, D. J. ve Leitner, H. H. (2007, Mart). Scratch for Budding computer scientists. *Sigcse 2007:Proceedings of the Thirty-Eighth Sigcse Technical Symposium on Computer ScienceEducation da sunulan bildiri. Kentucky, ABD*.doi.org/10.1145/1227504.1227388.

- Maloney, J., Resnick, M., Rusk, N., Silverman, B., ve Eastmond, E. (2010). The Scratch programming language and environment. *Trans. Comput. Educ.*, 10(4), 1-15. doi>10.1145/1868358.1868363.
- Mblock (2018). The educational programming software that knows teacher better in the Age of AI. Erişim adresi: <http://www.mblock.cc/>.
- McGill T. J. ve Volet S.(1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Resarch in Education*, 29(3), 276-297,doi: 10.1080/08886504.1997.10782199.
- Mladenović, M., Krpan, D. ve Mladenović, S. (2016, Temmuz). Introducing programming to elementary students novices by using game development in Python and Scratch. *Conference: International Conference on Education and New Learning Technologies da sunulan bildiri*.Barselona, İspanya. doi: 10.21125/edulearn.2016.1323
- Nam, D., Kim, Y., and Lee, T. (2010). The effects of scaffolding-based courseware for the scratch programming learning on student problem solving skill. *Proceedings of the 18th International Conference on Computers in Education' da sunulan bildiri*. Putrajaya, Malaysia, Erişim adresi :<http://www.icce2010.upm.edu.my/papers/c6/short%20paper/C6SP153.pdf>.
- Oh J., Lee J. and Kim J. (2013). Development and application of STEAM based education program using Scratch: Focus on 6th graders' science in elementary school. In: Park J., Ng JY., Jeong HY., Waluyo B. (eds) *Multimedia and Ubiquitous Engineering. Lecture Notes in Electrical Engineering*,240, 493-501. doi.org/10.1007/978-94-007-6738-6_60
- Oluk, A.ve Korkmaz, Ö. (2016). Comparing students' Scratch skills with their computational thinking skills in terms of different variables. *I.J. Modern Education and Computer Science*, 11,1–7. <http://doi.org/10.5815/ijmeecs.2016.11.01>.
- Osman, M. A., Zakaria, M. N., Loke, S. P., and Downe, A. G. (2012, Aralık). Secondary students' perfectionism and their response to different programming learning tools. *Paper presented at the Humanities, Science and Engineering (CHUSER), 2012 IEEE Colloquium on sunulan bildiri*.Kota Kinabalu,Malezya.doi: 10.1109/CHUSER.2012.6504380.

- Ozoran, D., Cagiltay, N. Ve Topalli, D. (2012, Eki). Using Scratch in introduction to programming course for engineering students. *In 2nd International Engineering Education Conference' da sunulan bildiri*. Antalya, Türkiye. Erişim adresi: https://www.researchgate.net/publication/274632820_USING_SCRATCH_IN_INTRODUCTION_TO_PROGRAMMING_COURSE_FOR_ENGINEERING_STUDENTS.
- Öndeş, Ö., (2016, 29 Şubat). İngiltere ve ABD' de kodlama eğitimi. *Hürriyet*. Erişim adresi: <http://www.hurriyet.com.tr/egitim/ingiltere-ve-abd-de-kodlama-egitimi-40061604>.
- Özen, Y.ve Gül, A. (2007). Sosyal ve eğitim bilimleri araştırmalarında evren-örneklem sorunu. *Kazım Karabekir Eğitim Fakültesi Dergisi*, 15, 394-422. Erişim adresi: <http://e-dergi.atauni.edu.tr/ataunikkefd/article/viewFile/1021004163/1021003987>.
- Özmen, B.ve Altun, A . (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 1-27. Erişim adresi: <http://dergipark.gov.tr/tojqi/issue/21404/229409>.
- Papadakis, S., Kalogiannakis, M., Zaranis, N. ve Orfanakis, V. (2016). Using Scratch and App Inventor for teaching introductory programming in secondary education. *A Case Study International Journal of Technology Enhanced Learning*, 8(3/4), 217-233. doi: 10.1504/IJTEL.2016.10001505.
- Papatğa, E. (2016). *Okuduğunu anlama becerilerinin Scratch programı aracılığıyla geliştirilmesi* (Yayımlanmamış doktora tezi). Anadolu Üniversitesi, Eskişehir.
- Portelance, D. J., Strawhacker A. L. ve Bers, M. U. (2015). Constructing the Scratchjr programming language in the early childhood classroom. *International Journal of Technology and Design Education*, 26(4), 489-504. doi: 10.1007/s10798-015-9325-0.
- Resnick, M., Maloney, J., Hernandez, A., Easmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. ve Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67. <http://doi.org/10.1145/1592761.1592779>
- Ruf, A., Muhling, A., ve Hubwieser, P. (2014, Kasım). Scratch vs. Karel: impact on learning outcomes and motivation. *Paper presented at the Proceedings of the 9th Workshop*

in Primary and Secondary Computing Education. Berlin, Almanya. doi>10.1145/2670757.2670772.

Seferoğlu, S. S. (2007). İlköğretim bilgisayar dersi öğretim programı: Eleştirel bir bakış ve uygulamada yaşanan sorunlar. *Eurasian Journal of Educational Research*, 29, 99-111. Erişim adresi: http://yunus.hacettepe.edu.tr/~sadi/yayin/Seferoglu_EJER-2007_Bilgisayar-Dersi-Programi.pdf.

Sayın, S. ve Seferoğlu, S. S. (2016, Şubat). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansında sunulan bildiri*. Adnan Menderes Üniversitesi, Aydın.

Scott, A. L. (2017). The 21st Century Skills Early Learning Framework. http://www.p21.org/storage/documents/EarlyLearning_Framework/P21_ELF_Framework_Final.pdf adresinden 2017 yılında erişilmiştir.

Scratch (2017). Erişim adresi: <https://scratch.mit.edu/about/>.

Shin, S. ve Park, P. (2014). A study on the effect affecting problem solving ability of primary students through the scratch programming. *Advanced Science and Technology Letters*, 117-120. doi.org/10.14257/astl.2014.59.27.

Shin, S., Park, P. ve Bae, Y. (2013). The Effects of an information-technology gifted program on friendship using Scratch programming language and clutter. *International Journal of Computer and Communication Engineering*, 2(3), (pp. 246-249). Erişim adresi: www.ijcce.org/papers/181-J028.pdf.

Sırakaya, D. (2017). Bilgi işlemsel düşünme becerisi üzerine programlama öğretiminin etkisi [Öz]. Icits2017 konferansında sunulan bildiri. İnönü Üniversitesi, Malatya. Erişim adresi: <http://icits2017.inonu.edu.tr/dosya/1493894911055732500.pdf>.

Taylor, M., Harlow, A. ve Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences*, 8, 561-570. doi.org/10.1016/j.sbspro.2010.12.078.

Tekin, A., Akbal, A., Sevinç, B., Ertam, F., Tuzsuzoğlu, H. H., Serhatlıoğlu, İ., Balıkçı, K., Talu, M. F., Çıbuk, M., Özdemir, O., Daş, R., Akbulut, Y. ve Genç, Z. (2003). “C” ile programlamaya giriş. [Adobe Acrobat Reader sürümü]. Erişim adresi: http://web.firat.edu.tr/enfders/file/c_programlama/c.pdf.

- TTKB,(2012). Bilişim Teknolojileri Ve Yazılım Dersi Öğretim Programı. https://www.bilgisayarbilisim.net/attachments/bilisimteknolojileri_ortaokul-rar.29815/ adresinden 2017 tarihinde erişilmiştir.
- TTKB,(2017). Bilişim Teknolojileri Ve Yazılım Dersi Öğretim Programı. <http://bilgisayarbilim.com/wp-content/uploads/2017/01/Bili%C5%9Fim-Teknolojileri-ve-Yaz%C4%B1l%C4%B1m-Dersi-%C3%96%C4%9Fretim-Program%C4%B1.pdf> adresinden 2018 tarihinde erişilmiştir.
- Wang, H. Y., Huang, I., and Hwang, G. J. (2014, Ağustos). 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'da sunulan bildiri*. Kitakyushu, Japonya. doi: 10.1109/IIAIAI.2014.85
- Wing., J. M. (2006). Computational thinking. *Communications of the ACM* 49(3), 33-35. doi:10.1145/1118178.1118215.
- Yıldırım, E. (2016). Dijital oyun tasarım programlarının eğitimde önemi. *Mesleki Bilimler Dergisi*, 5(2),12 – 19. Erişim adresi:<http://dergipark.gov.tr/mbd/issue/34074/377099>.
- Yıldırım, E. (2017). *Scratch programlama dili eğitime yönelik bir mobil uygulamanın geliştirilmesi*(Yayınlanmamış yüksek lisans tezi). Onsekiz Mart Üniversitesi, Çanakkale.
- Yıldırım, A., ve Şimşek, H. (2011). *Sosyal bilimlerde nitel araştırma yöntemleri* (8. Baskı). Ankara: Seçkin Yayıncılık.
- Yüksel, S. (2017). *Scratch programı öğretiminde ayrılıp birleşme tekniği kullanımının öğrencilerin derse yönelik tutumuna, akademik başarısına ve kalıcılığa etkisi* (Yayınlanmamış yüksek lisans tezi). Adnan Menderes Üniversitesi, Aydın.
- Yükseltürk, E. ve Altıok, S. (2016). Investigation of pre-service information technology teachers' game projects prepared with Scratch. *SDU International Journal of Educational Studies*, 3(1), 59-66. Erişim adresi: <http://dergipark.gov.tr/sdujes/issue/20865/223885>.
- Yünkül, E., Durak, G., Çankaya, S. ve Mısırlı, Z. A. (2017). the effects of Scratch software on students' computational thinking skills. *Necatibey Eğitim Fakültesi Elektronik*

Fen ve Matematik Eğitimi Dergisi, 11(2), 502-517. doi: 10.17522/balikesirnef.373424.

Zavala, L.A., Gallardo S. H. and Ruiz M.A. (2013, Haziran). Designing interactive activities within Scratch 2.0 for improving abilities to identify numerical sequences. *IDC '13 Proceedings of the 12th International Conference on Interaction Design and Children'da sunulan bildiri*. New York, Usa. doi: 10.1145/2485760.2485831.



EKLER

EK 1: Bilgisayar Programlamaya İlişkin Öz Yeterlilik Ölçeği

Merhaba, bu anket sizin programlama konusunda kendinize ilişkin öz yeterlilik algınızı belirlemek üzere hazırlanmıştır. Programlamaya ilişkin verilen görevleri yaparken kendinize olan güveninizi 1 ile 7 arasında derecelendirerek belirtiniz. (“1= Hiç Güvenmiyorum”, “2=Genellikle Güvenmiyorum”, “3= Biraz Güveniyorum”, “ 4= %50/50” “5= Oldukça Güveniyorum”, “6=Genellikle Güveniyorum”, “7=Tamamen Güveniyorum”)

Anıl Dinçer

	Hiç güvenmiyorum	Genellikle güvenmiyorum	Biraz güveniyorum	50/50	Oldukça güveniyorum	Genellikle güveniyorum	Tamamen güveniyorum
	1	2	3	4	5	6	7
1. "Merhaba Dünya" mesajının görüntülenebileceği bir program yazabilirim..							
2. Üç sayının ortalamasını hesaplayan bir program yazabilirim.							
3. Verilen herhangi bir sayı dizisinin ortalamasını hesaplayan bir program yazabilirim..							
4. İstenilenler açıkça tanımlandığında bir problemin çözümüne yönelik oldukça karmaşık ve uzun bir program yazabilirim.							
5. Yazacağım bir programı modüler bir biçimde organize edip tasarlayabilirim							
6. Yazdığım uzun ve karmaşık bir programdaki tüm hataları ayıklayabilir ve çalışabilir hale getirebilirim.							
7. Uzun, karmaşık ve birden fazla dosya gerektiren bir programı kavrayabilirim.							

8. Bir programın daha okunabilir ve açık olması için uzun ve karmaşık kısımları yeniden yazabilirim.							
9. Çevrede bir sürü dikkat dağıtıcı olsa bile programa odaklanma yollarını bulabilirim.							

EK 1: Bilgisayar Programlamaya İlişkin Öz Yeterlilik Ölçeği Kullanım İzni

Re: İlt: Ölçek kullanım izni



Arif Altun
1.11 (Sal) , 20:31
Siz



Yanıtla

Anıl merhaba,
Tabi kullanabilirsiniz. Kullanım ile ilgili sorun olrsa Güzin Mazman hoca ile de temasa geçebilirsiniz. Eposta adresi s.mazman@gmail.com
İyi çalışmalar
A

31 Ekim 2016 Pazartesi tarihinde, Anıl Dincer <a_n_dincer@hotmail.com> yazdı:

İyi günler Arif Hocam,

Ben Anıl DİNÇER, Çanakkale Biga Yeniceköy ortaokulunda Bilişim teknolojileri öğretmeni olarak çalışmaktayım.

"6.sınıf öğrencilerine Sctrach ve Kodu Game Lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarıları yönünden karşılaştırılması" adlı tezim üzerinde çalışıyorum.

2013 yılında yayınlamış olduğunuz **Programlamaya ilişkin Öz Yeterlilik Algısı Ölçeği**'ni tezimde kullanmak üzere izninizi talep eder, teşekkürlerimi şimdiden sunarım.

Saygılarımla.

İletişim :0578522010

Re: Ölçek izin



Güzin Mazman
Dün, 18:20
Siz



Yanıtla

3.11.2016 10:27 tarihinde yanıt verdiniz.

Kullanabilirsiniz tabiki.
İyi çalışmalar, kolaylıklar.

Yrd. Doç. Dr. S. Güzin Mazman Akar

2 Kasım 2016 10:48 tarihinde Anıl Dincer <a_n_dincer@hotmail.com> yazdı:

İyi günler Güzin Hocam,

Ben Anıl DİNÇER, Çanakkale Biga Yeniceköy ortaokulunda Bilişim teknolojileri öğretmeni olarak çalışmaktayım.

"6.sınıf öğrencilerine Sctrach ve Kodu Game Lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarıları yönünden karşılaştırılması" adlı tezim üzerinde çalışıyorum.

2013 yılında yayınlamış olduğunuz **Programlamaya ilişkin Öz Yeterlilik Algısı Ölçeği**'ni tezimde kullanmak üzere izninizi talep eder, teşekkürlerimi şimdiden sunarım.

Saygılarımla.

İletişim : [0507 8522010](tel:05078522010)

EK 2:Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği

Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği

Sevgili öğrenciler bu test eğitsel bilgisayar oyunları destekli kodlama öğrenimine karşı tutumlarınızı tespit etmek için kullanılacak bir ölçme aracı geliştirilmesi amacıyla gerçekleştirilmiştir. Ölçekteki maddeler kodlama öğrenimine karşı istek, bilgisayar oyunlarının derslerde eğitim amaçlı kullanımına yönelik ilgi ve bilgisayarın asosyalleştirmesine yönelik endişeleri kapsayacak şekilde hazırlanmıştır. Ölçeğe cevap verirken aşağıdaki hususlara dikkat ediniz.

1. Ölçekte geçen bilgisayar oyunu sadece bilgisayarda oynanan oyunlar değil tablet, telefon vs. gibi oynanan oyunlar olarak da düşünülmelidir.
2. Kodlama: Bilgisayar ve benzeri düzeneklere bir işlem yaptırmak için verilen komut dizisidir.
3. Sınıf: Cinsiyet: Yaş :

	Maddeler	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılmıyorum	Kesinlikle Katılmıyorum
1	Bilgisayarla (tablet, telefon vb.) oyun oynamak okul başarıma katkı sağlar.					
2	Bilgisayar oyunlarından yararlanılarak ders çalışmak faydalıdır.					
3	Dersleri bilgisayarla oyun oynayarak işlemeyi isterim.					
4	Bilgisayar oyunlarındaki kahramanların dış görünüşünü kendim ayarlamak isterim.					
5	Bilgisayarda kendi oyunumu tasarlamak isterim.					
6	Bilgisayarda kodlama yapmayı öğrenmek isterim.					
7	Bilgisayar ile ilgili meslek sahibi olanlar aktif değildir.					
8	Oyunların seviyeli olması ilgimi çeker.					
9	Oyunlardaki seviyelerin çokluğu oyuna olan ilgimi artırır.					
10	Oyun hazırlarken kahramanın başına neler gelebileceğini kendim belirlemek isterim.					
11	Kendi hazırladığım bilgisayar oyununu arkadaşlarımla da oynamasını isterim.					
12	Bilgisayarda kendi oyunumu hazırlamak ilgimi çekmez.					
	Maddeler	Kesinlikle Katılıyorum	Katılıyorum	Kararsızım	Katılmıyorum	Kesinlikle Katılmıyorum
13	Arkadaşlarımla tasarladığım oyunları oynamak eğlenceli olabilir.					
14	Bilgisayarda kod yazmak benim için zordur.					
15	Bilgisayarda kod yazmayı öğrenmek istemem.					
16	Sokakta oynanan oyunlar benim için tehlikelidir.					
17	Kodlama öğreniminin benim için faydalı olacağını düşünüyorum.					
18	Bilgisayar oyunları ile eğitim öğretim olmaz.					
19	Kodlama öğrenimi problem çözme becerimi geliştireceği için sınavlarda başarımla artar.					
20	Kendi oyunumu tasarlamak yaratıcılığımı geliştirecektir.					
21	Bilgisayarla oyun oynamak boş zamanları değerlendirmek için idealdir.					
22	Dışarıda (futbol, basketbol, evcilik vb.) oynanan oyunlardan daha bilgisayarla oynanan oyunlar daha iyidir.					
23	Arkadaşlarımla oyun oynamaktansa bilgisayar oyunu oynamayı tercih ederim.					
24	Bilgisayar oyunlarında ikili oyunları daha çok severim.					
25	Kodlama öğrenimi zeka geliştirir.					
26	Bilgisayarda oyun oynamak bireyleri tembelleştirir.					
27	Derslerin bilgisayar oyunları ile işlenmesi derse olan ilgimi artırır.					
28	Bilgisayarda oyun kodlamayı öğrenirsem derslerim de başarımla artar.					

EK 2:Eğitsel Bilgisayar Oyunları Destekli Kodlama Öğrenimine Yönelik Tutum Ölçeği Kullanım İzni

Ynt: Ölçek kullanım izni



BURCU ALAN
31.10 (Pzt) , 14:53
Siz



Yanıtla

31.10.2016 14:58 tarihinde yanıt verdiniz.

merhabalar hocam ölçeğimizi elbette kullanabilirsiniz. iyi çalışmalar dilerim...

Gönderen: Anıl Dincer <a_n_dincer@hotmail.com>

Gönderildi: 31 Ekim 2016 Pazartesi 10:49:20

Kime: burcualan@outlook.com

Konu: İlt: Ölçek kullanım izni

İyi günler Burcu ALAN Hocam,

Ben Anıl DİNÇER, Çanakkale Biga Yeniceköy ortaokulunda Bilişim teknolojileri öğretmeni olarak çalışmaktayım.

"6.sınıf öğrencilerine Scratch ve Kodu Game Lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarıları yönünden karşılaştırılması" adlı tezim üzerinde çalışıyorum.

2016 yılında yayınlamış olduğunuz **EĞİTSEL BİLGİSAYAR OYUNLARI DESTEKLİ KODLAMA ÖĞRENİMİNE YÖNELİK TUTUM ÖLÇEĞİ**'ni tezimde kullanmak üzere izninizi talep eder, teşekkürlerimi şimdiden sunarım.
Saygılarımla.

iletişim :0578522010

EK 3:Bilişim Teknolojileri Öğretmenlerinin Programlama Öğretirken Nelere Dikkat Ettikleri ve Programlamanın Nasıl Öğretilmesi Konusunda Düşüncelerini İçeren Mülakat Formu

Verdiğiniz bilgiler akademik amaçlı kullanılmak üzere tezimde kullanılacaktır. Bilgiler kişi ismi paylaşılmadan, verilerin analizinde kullanılacaktır.

1. Yaşınız?

21 - 27()

28 - 34()

35 - 41()

42 ve üzeri ()

2. Cinsiyetiniz?

Kadın ()

Erkek ()

3. Eğitim Durumunuz?

Lisans ()

Yüksek Lisans ()

Doktora ()

Diğer

4. Kaç yıldır bilgisayar öğretmenisiniz?

1- 3 yıl ()

4- 6yıl ()

7- 9 yıl ()

10 yıl ve üzeri ()

5. Çalıştığınız okulda bilgisayar laboratuvarına sahip misiniz?

Evet ()

Hayır ()

6. Kaç yıldır programlama öğretiyorsunuz?

1- 3 yıl ()

4- 6yıl ()

7- 9 yıl ()

10 yıl ve üzeri ()

7. Programlama öğretirken hangi yazılımları kullanıyorsunuz?Birden fazla seçebilirsiniz.

Scratch ()

Code.org ()

Kodu Game ()

Code Combat ()

Code Academy () Diğer (lutfen belirtiniz)

8. Kaçınıcı sınıflarla programlama dersi işliyorsunuz?

5. Sınıf ()

6. Sınıf ()

7. Sınıf ()

8. Sınıf ()

9. Kaç hafta boyunca programlama anlatıyorsunuz?

1-3 Hafta () 4-6 Hafta () 7-9Hafta () 10 Hafta ve üzeri ()

Programlama öğretimi açısından sizlerden alacağımız verilerle programlama nasıl daha iyi öğretilir ve nasıl daha verimli öğretilir sorusuna cevap bulmaya çalışacağız.

1. Programlama öğretimi kaçınıcı sınıftan itibaren öğretilmeli? Neden?

2. Programlama öğretmeden önce sizce hangi konular işlenmeli, hangi ön bilgiler öğrencilere kazandırılmalı?

3. Sizce programlama nasıl öğretilmelidir? Hangi öğretim yöntem ve teknikler kullanılmalıdır?

4. Programlama öğretmek için ne kadar süre harcanmalı?

5. Programlama hangi öğretim teknik ve yöntemlerle daha verimli hale getirilebilir?

6. Programlama öğretimi sırasında öğrencilerin programlama öğrenmeye karşı tutum ve davranışlarını gözlemlerinizi doğrultusunda nasıl değerlendirirsiniz?

7. Hangi programlama dilini öğretiyorsunuz? Öğrettiğiniz programlama dilini seçme sebebiniz nedir?

8. Scrach ve Kodu Game programlama dilini kullandınız mı? Kullandıysanız ders anlatımınızda hangi konuları işlediniz, hangi örnekleri verdiniz?

Eğer mümkün ise, programlama öğretiminde kullandığınız örnekleri, projeleri, ek bir dosya olarak ulaştırabilir misiniz?

Anıl DİNÇER Yeniceköy Ortaokulu

Bilişim Teknolojileri Öğretmeni

İletişim = a_n_dincer@hotmail.com

Adınız Soyadınız: _____

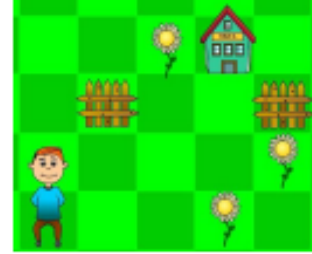
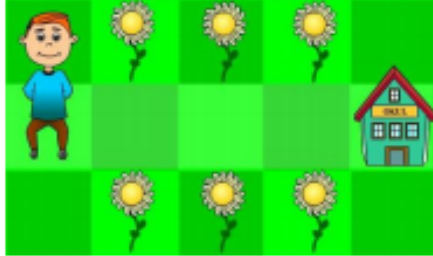
Çalıştığınız Kurum: _____



EK 4:Programlama Dilleri Akademik Başarı Testi

Ad Soyad:

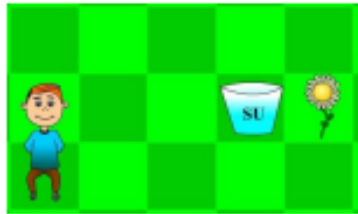
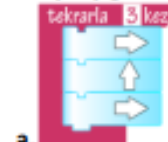
Sınıf:



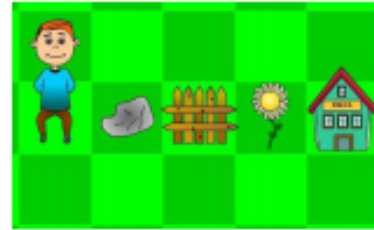
1. Yukarıdaki resimdeki Ali'yi döngü kullanarak okula istiyoruz. Bunun için aşağıdaki seçeneklerden hangisini seçmeliyiz?



2. Yukarıdaki resimdeki Ali'yi nesnelere değdirmeden okula ulaştırmak için döngü kullanmak istiyoruz. Bunun için aşağıdaki seçeneklerden hangisini seçmeliyiz?



3. Ali çiçeği sulamak istiyor. Ali'nin çiçeği sulaması için aşağıdaki kodlardan hangilerini kullanması gerekmektedir?



4. Ali resimdeki çiçeği öğretmenine hediye etmek istiyor. Ancak taşı ve çiti geçmekte zorlanıyor. Ali'nin çiçeği alıp okuldaki öğretmenine götürmesi için aşağıdaki hangi yolu izlemesi gerekir?

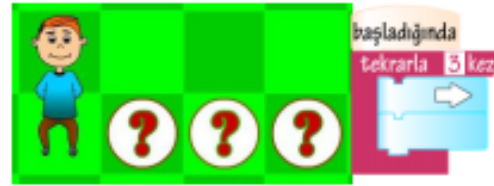


Ad Soyad:

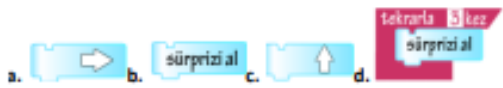
Sınıf:



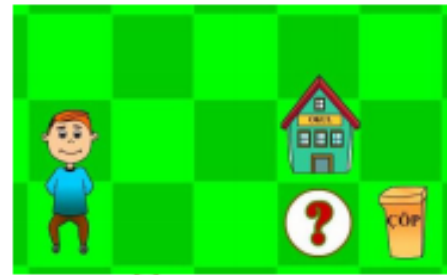
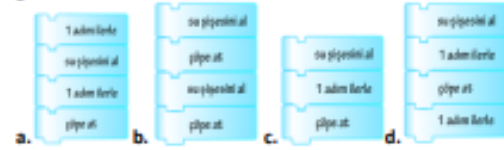
5. Ali ileride bir kova görmektedir. Kovanın içinde su olup olmadığını bilmemektedir. Ali kovanın içinde su bulursa çiçeği sulamak, su yoksa boş kovayı çöpe atmak için hangi kodları kullanmamız gerekir?



6. Ali'yi yolda 3 sürpriz beklemektedir. Ali sürprizlere ulaşmak için yandaki kodu kullanmıştır. Ancak döngünün içerisindeki 2. kod yazamamıştır. Ali'nin boş olan yere yazmak istediği kod aşağıdakilerden hangisidir?



8. Ali su şişelerini çöpe atmak için yandaki kodu yazmış ancak döngünün içerisine ne yazacağını bilmemektedir. Ali'ye yardım etmek için döngünün içerisine ne yazmamız gerekir?

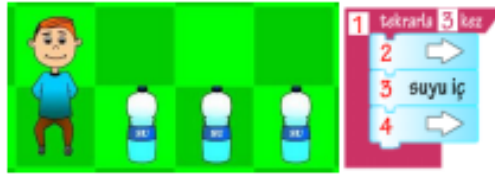


7. Resimdeki ? yerde Ali'nin çantası veya boş süt şişesi vardır. Eğer çanta varsa Ali okula gidecektir. Süt şişesi varsa da süt şişesini çöpe atacaktır. Aşağıdaki kılklardan hangisi bu koşulları sağlamaktadır.



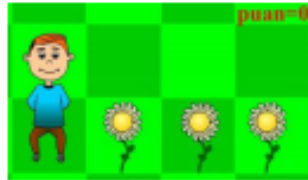
Ad Soyad:

Sınıf:



9. Ali çok susamış, 3 şişe su içmek istemiştir. Ali bunun için yandaki kod satırını yazmıştır ancak fazladan yaptığı 1 satırdan dolayı amacına ulaşamamıştır. Ali'nin fazladan yazdığı kod satırı aşağıdakilerden hangisidir?

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



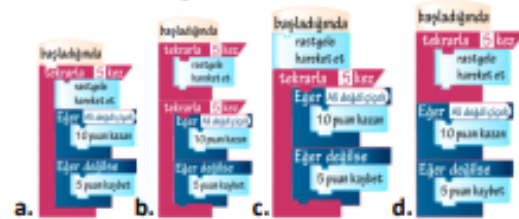
11. Ali her bir çiçeği topladığında 10 puan kazanacaktır. Ali'ye çiçekleri toplamak ve puan kazandırmak için aşağıdaki hangi kodları kullanması gerekir?



10. Ali programlama dünyasına yeni giriş yapmıştır. Ali'nin ilk programında 1 kez "Merhaba Dünya" yazması için aşağıdaki hangi kodu yazması gerekmektedir?



12. Ali rastgele 5 kez hareket etmektedir. Çiçeklere değdiği zaman 10 puan kazanmakta, başka bir nesneye değdiği zaman ise 5 puan kaybetmektedir. Ali'nin yaptığı hareketleri ve sonuçlarını görebilmesi için aşağıdaki hangi kodları kullanması gerekir?



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

Ad Soyad:

Sınıf:

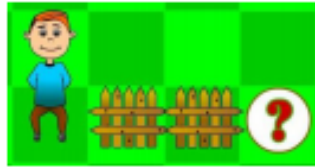
Bilişim Teknolojileri ve Yazılım



1. Sınav ?
2. Sınav ?
Ders notu ?

13. Ali Bilişim Teknolojileri ve Yazılım dersinden iki sınav olmuştur. Bu iki sınavın ortalamasını hesaplayıp ekrana yazdırmak istemektedir. Bunun için aşağıdaki hangi kod satırını yazması gerekir?

- a. **başladığında**
1. notu gir
2. notu gir
2 notu topla
toplamı
2'ye böl
sonucu ekrana yaz
- b. **başladığında**
1. notu gir
2. notu gir
1. notu
2. notu
2'ye böl
sonucu ekrana yaz
- c. **başladığında**
1. notu gir
2. notu gir
2 notu topla
toplamı
2'ye böl
sonucu ekrana yaz
- d. **başladığında**
1. notu gir
2. notu gir
1. ve 2. notu
çarp
çıkart sayısını
2'ye böl
sonucu ekrana yaz



14. Ali'yi çitlerin arkasında bir sürpriz beklemektedir. Ali sürprize ulaşmak için yandaki kod satırını yazmış ancak son kısma yazılacak kodu yazamamıştır. Ali'nin sürprize ulaşması için yukarıdaki kod satırına aşağıdakilerden hangisini eklememiz gerekir?

- a. **sağa dön**
sürprizi al
- b. **1 adım ilerle**
sağa dön
sürprizi al
- c. **sağa dön**
1 adım ilerle
sürprizi al
- d. **1 adım ilerle**
sağa dön
1 adım ilerle
sürprizi al



15. Ali resimdeki çiçekler bitene kadar toplamak istiyor ama kaç tane çiçek olduğunu bilmiyor. Bütün çiçekler bitene kadar toplamaları için Ali'nin aşağıdaki hangi kodları kullanması gerekir?

- a. **başladığında**
sürprizi al
çiketi al
1 çiçek asıldı
Eğer çiçek = 0
toplamayı bitir
- b. **başladığında**
çiketi al
1 çiçek asıldı
Eğer çiçek = 0
toplamayı bitir
- c. **başladığında**
sürprizi al
çiketi al
1 çiçek asıldı
Eğer çiçek = 0
toplamayı bitir
- d. **başladığında**
1 çiçek asıldı
Eğer çiçek = 0
toplamayı bitir



16. Ali'nin kuşu, sürekli olarak 3 saniye arayla "Seni Seviyorum Babacık" diye konuşmaktadır. Ali kuşunun yaptığı bu eylemi kod olarak yazmıştır. Ali'nin yazdığı kod satırı aşağıdakilerden hangisidir?

- a. **3 saniye bekle**
tekrarla sürekli
"Seni Seviyorum Babacık" diye konuş
- b. **tekrarla sürekli**
3 saniye bekle
"Seni Seviyorum Babacık" diye konuş
- c. **tekrarla 3 kez**
"Seni Seviyorum Babacık" diye konuş
3 saniye bekle
- d. **tekrarla sürekli**
"Seni Seviyorum Babacık" diye konuş
3 saniye bekle

EK 5:Anket ÖnUygulama İzni



T.C.
ÇANAKKALE VALİLİĞİ
İl Millî Eğitim Müdürlüğü

101. yıl
ÇANAKKALE

Sayı : 60305806-44-E.4008706
Konu: Anket Çalışması

24.03.2017

MİLLÎ EĞİTİM MÜDÜRLÜĞÜNE ÇANAKKALE

İlgi : Dokuz Eylül Üniversitesi Rektörlüğü Öğrenci İşleri Daire Başkanlığının 14/03/2017 tarihli ve 0304 sayılı yazısı.

Dokuz Eylül Üniversitesi Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Öğretmenliği Yüksek Lisans Programı Öğrencisi Anıl DİNÇER tarafından "6. Sınıf Öğrencilerine Scratch ve Kodu Game Lab Programlama Dillerinin Öğretiminde Öğrencilerin Tutum, Öz Yeterlilik ve Akademik Başarılarının Karşılaştırılması" konulu tez çalışması kapsamında, 24/03/2017 - 31/05/2017 tarihleri arasında, Biga Yeniceköy Ortaokulu 6. sınıf öğrencilerine yönelik uygulama çalışması yapılma isteği ilgi yazısıyla teklif edilmekte olup, Müdürlüğümüz Anket-Araştırma İnceleme Komisyonunca incelenerek uygun görülmüştür. Makamlarınızca da uygun görüldüğü takdirde, Olurlarınıza arz ederim.

Önder YILMAZ
Şube Müdürü

OLUR
24.03.2017

Erdal DOĞANCI
Millî Eğitim Müdürü V.

Ek : Komisyon Raporu (1sayfa)

Mehmet AYDIN
V.M.E.İ.
Güvenli Elektronik İmza
Aslı ile Aynıdır.
20.03.2017

Millî Eğitim Müdürlüğü Valilik Binası 3. Kat
Elektronik Ağ: stratejigelistime17@meb.gov.tr

Ayrıntılı bilgi için: Özgür AYDIN
Tel: 0286 217 11 35-117

EK 6: Anket Gerçek Uygulama İzni

T.C.
ÇANAKKALE VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : 60305806-44-E.6279371
Konu: Anket Çalışması

04.05.2017

MİLLÎ EĞİTİM MÜDÜRLÜĞÜNE
ÇANAKKALE

İlgi : Dokuz Eylül Mart Üniversitesi Rektörlüğü (Öğrenci İşleri Daire Başkanlığı)'nın 21/04/2017 tarihli ve 581 sayılı yazısı.

Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Öğretmenliği yüksek lisans öğrencisi Anıl DİNÇER tarafından "6. Sınıf Öğrencilerine Scratch ve Kodu Game Lab Programlama Dillerinin Öğretiminde Öğrencilerin Tutum, Öz Yeterlilik ve Akademik Başarılarının Karşılaştırılması" konulu tez çalışması kapsamında kullanılmak üzere 15 Mayıs-09 Haziran 2017 tarihleri arasında Biga İlçesi Yeniceköy Ortaokulunda öğrenim gören 6.sınıf öğrencilerine yönelik uygulama yapılma isteği ilgi yazı ile teklif edilmektedir.

Söz konusu çalışmalar Müdürlüğümüz Anket-Araştırma İnceleme Komisyonunca incelenerek uygun görülmüştür.

Makamlarınıza da uygun görülmesi halinde Olur'larınıza arz ederim.

Erdal DOĞANCI
Müdür Yardımcısı

OLUR
04.05.2017

Osman ÖZKAN
Millî Eğitim Müdürü

Ek: Komisyon Raporu (1 sayfa)

Güvenli Elektronik İmza

Aşlı İle Aydın

04.05.2017

Çanakkale İl Millî Eğitim Müdürlüğü
Strateji Geliştirme Bölümü Merkez/ÇANAKKALE
e-posta: argel17@meb.gov.tr

Ayrıntılı bilgi için: Türkan YÜCEKAYA Şef
Tel: (0 286) 217 11 35 -117
Fax: (0 286) 217 29 72

Bu evrak güvenli elektronik imza ile imzalanmıştır. <http://evraksorgu.meb.gov.tr> adresinden 0e03-8507-33a3-b6b6-4a8e kodu ile teyit edilebilir.

EK 7:Scratch Ders Planları

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.2. Algoritma ve Strateji Geliştirme (algoritma oluşturma mantığı, sözde kod, akış şemaları vb.)
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, Adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Tahta, reçel, tost ekmeği, plastik bıçak
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sınıfa bir robot gibi hareket ederek elimizde bir kavanoz reçel, tost ekmeği ve plastik bıçak getirilerek dikkat öğrencilerin dikkati çekilir. Öğrencilere ” daha önce ekmeğine reçel süren var mı ?” diye sorulur. Seçilen bir öğrenciye söz hakkı verilerek “peki bunu nasıl yaptın bize anlatır mısın?” diye sorulur. Öğrenciden yanıt gelince ” peki beni bir robot olarak düşün, bilirsin robotlar sen bir şey yap demeden yapmazlar, sadece senin dediğini yaparlar ona göre anlat ben de ekmeğime reçel sürüp yiyebileyim.” diye söylenir. Öğrenciden gelen yönergeler eksiksiz uyulur. Öğrencilerden doğru cevap gelene kadar bu etkinlik sürdürülür. Öğrencilere adım adım yapmaları gerektiği fark ettirilir. Bir öğrenci seçilir bu yaptığımız işlemleri sırayla tahtaya yazması istenir. Algoritmanın tanımı yapılır ve tahtada yazılan komutların bir algoritma olduğu ifade edilir. Her şeyin bir komutla başladığı ve bittiği ifade edilir. Daha sonra sınıfın kapısı işaret edilerek “ kapıyı kim nasıl açacağımızı anlatır” diye sorulur. Bir öğrenci seçilir ve uygulayarak anlatması istenir. Öğrenci uygularken adımlar tahtaya yazılır. Yaptığımız işlerin basit gibi görünse de aslında bazı aşamalardan geçtiği söylenir. Öğrencinin tahtaya yazdığı adımlardan bir kaç silinerek bu adımları tekrar etmesi istenir. Öğrenci hedefe ulaşamayınca algoritmanın önemi vurgulanır. Öğrencilerden ikişerli grup olmaları ayrı nasıl yapılır bunu grup arkadaşlarıyla birlikte algoritmasını yapmaları istenir. Daha sonra öğrencilerden yazdıkları algoritmayı diğer grupla değiştirmeleri istenir. Doğru cevap tahtaya yazılarak öğrencilerden kontrol etmeleri istenir. Sonuç olarak öğrencilere algoritmanın neden önemli olduğu, bir yerde yanlış yaparsak sadece o adımı düzeltmenin işimizi kolaylaştıracağı ifade edilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Her grup kendi grubunun ve diğer grupların algoritmalarını kontrol eder.
Açıklama	Birinci hafta ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.2. Algoritma ve Strateji Geliştirme (algoritma oluşturma mantığı, sözde kod, akış şemaları vb.)
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Akış şeması, Adım adım düşünme, Algoritma
Güvenlik Önlemleri (Varsa)	Kağıt makası tercih edilir
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma,
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Kağıt, Kalem, Karton, Makas
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sınıfa daha önce bir problemin çözümünün her bir adımı yazılı farklı renkle kartonlar sınıfa getirilerek öğrencilerin dikkati çekilir. Daha sonra “bir problemin adım adım düşünülmesi neden önemlidir “ diye sorulur. Öğrencilerden alınan cevaplardan sonra bir önceki ders hatırlatarak özet geçilir. Daha sonra karton sayısı kadar öğrenci tahtaya çıkarılır ve her birine bir karton verilir. Bir öğrenci seçilerek bu karışık olarak verilen kartonları bir problemin çözümü olacak şekilde sıralamasını istenir. Bu şekilde adım adım sıralamanın algoritma olduğu cevabını alana kadar öğrenci yönlendirilir. Algoritmanın görselleştirilmiş ve ilişkilendirilmiş halinin akış şeması olduğu ifade edilir. Kartonlardaki şekilleri göstererek akış şemasının elemanları olduğu ifade edilir. Daha sonra öğrencilerden ele ele tutuşmaları istenir ve ortadaki iki kişinin ellerini bırakması istenir. Akış şemasının bu şekilde birbiriyle ilişkili olması gerektiği kopuk kısımlarının akış şemasına dahil olmadığı ifade edilir. Geçen ders yapılan algoritmaların akış şemaları yapılarak öğrencilere gösterilir. Öğrencilerden üçerli grup oluşturmaları istenir. Gruplardaki öğrencilerden bir problem bulmaları ve buldukları problemin çözümünün akış şemasını boş kağıtlara çizmeleri söylenir. Daha sonra her bir öğrenciden dersin başındaki gibi akış şemasının elemanlarını kesmeleri ve algoritmaları bunlara yazmaları istenir. Daha sonra öğrenciler bitirdikten sonra kağıtları karıştırıp diğer gruba karışık şekilde vermeleri sağlanır. Daha sonra öğrencilerden karışık şekilde olan akış şemasının sıralanması ve problemin ne olduğunu bulmaları istenir. Etkinlik tamamlanınca ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Problem kavramını tanımları. • Problem çözmenin önemini ifade eder.
Açıklama	İkinci hafta ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Programlama, Programlama dili, Kod
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon,
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Scratch’ ta yapılmış olan Flappy Bird oyunu öğrencilere gösterilerek öğrencilerin dikkati çekilir. Daha sonra “siz de bu ve buna benzer oyun yapmak ister misiniz?” diye sorularak öğrencilerin güdülenmesi sağlanır. Bunun için öğrencilere bir programlama dili kullanmaları ve kod yazmaları gerektiği söylenir. Scratch programı açılır ve öğrencilerden de programı açmaları istenir. Daha sonra öğrencilere öncelikle sahne gösterilir yazdığımız kodların burada hayat bulduğu söylenir ve sahnenin nasıl farklı dekorlar alabileceği gösterilir. Sonra karakter tanıtılır ve farklı karakterler nasıl eklenir gösterilir. Öğrencilerden de yapılanları yapmaları istenir. Daha sonra Kod Menüsü öğrencilere gösterilir ve benzer kodların aynı menüler altında bulunduğu söylenir. Kod yazım alanı tanıtılarak kodların sürüklenip bu kısma bırakıldığı söylenir. “Algoritma her zaman ne ile başlar” sorusu sorularak öğrencilerden başla cevabı alınır. Scratch’ta buna karşılık olaylar menüsü olduğu ifade edilir. Öğrencilere “sahnedeki karakteri konuşturalım mı?” diye sorulur. Tıklandığında bloğu altına Hello diye konuş bloğu kullanılarak ekranda karakterin konuşması sağlanır. Farklı karakterleri konuşturmak için öncelikle onu seçerek içerisine kodları sürüklememiz gerektiği vurgulanır. Daha sonra öğrencilerden istedikleri iki karakter seçmeleri ve karşılıklı olarak bu karakterleri konuşturmaları istenir. Öğrenciler yaptıktan sonra programlarını saklamaları istenir. Bunun için nasıl kaydedileceği gösterilir. Daha sonra tekrar edilerek ders bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Sahneyi değiştirebilirim. • Karakter ve kostüm değiştirebilirim. • Ekranı yazı yazdırabilirim.
Açıklama	Üçüncü Hafta Scratch ders planı

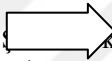
Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Ve, Veya, Değil, Büyüktür, Küçüktür, +, -, *, /, >, <
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Tahta parçası, Beyaz tahta
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse üç farklı uzunlukta tahta parçası getirilerek öğrencilerin dikkati çekilir. Daha sonra yazı tahtasına iki adet ‘>’ işareti çizilerek öğrencilerden bunları tahta parçalarını bu işaretlerin arasına koymaları istenir. Sonra öğrencilere niçin bu şekilde sıraladıkları sorulur ve ‘>’ işaretin büyüktür işareti olduğu vurgulanır. Bu işareten başka ‘<’ işareti olduğu da söylenir. Daha sonra bir öğrenci seçilerek üç tahtadan en uzun ve en kısa olanı seçmesi istenir. Başka bir öğrenci seçilerek ”en uzun veya en kısa tahta parçasını alırmısın?” denir. İlk öğrenciye ”niçin bir tane almadın” diye sorulur. İkinci öğrenciye de niçin bir tane aldın?” diye sorulur. Öğrencilere ‘ve’ , ‘veya’ kavramlarının ne olduğu aktarılır. Daha sonra başka bir öğrenci seçilerek “En uzun ve en kısa değilse al “ diyerek öğrencinin ortanca tahta parçasını alması beklenir. Burada da ‘değil’ kavramı açıklanır. Daha sonra “ve”, “veya”, “<”, “>”, “değil” operatörleri yazılarak bunların mantıksal operatörler olduğu vurgulanır. Daha sonra da “+”, “-”, “*”, “/” gibi sayılarla toplama, çıkarma, çarpma ve bölme gibi işlemler yapabilmek için kullanmamız gereken matematiksel operatörlerin de olduğu ifade edilir. Öğrencilere Scratch programını açmaları söylenir. Bu ifadelerin İşlemler kod menüsünde yer aldığı söylenir. Operatörlerin nasıl kullanıldığı uygulamalı olarak öğrencilere gösterilir. Öğrencilerin de gördüklerini uygulamaları istenir. Daha sonra öğrencilere grup arkadaşıyla beraber iki sayının toplamını, farkını, çarpımını ve bölümünü bularak ekrana yazdırmaları istenir. Daha sonra toplamını ve çarpımını ekrana yazdırmaları istenir. Ders özetlenerek sonlandırılır.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Operatör kavramını tanımlar. • Operatör kavramını kod bloklarında kullanır. • Hangi operatörü seçeceğinin farkına varır.
Açıklama	Dördüncü Hafta Scratch ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Tekrarla, Döngü, Sürekli tekrarlar
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Karton, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Döngü kavramını tanımlar. • Döngü kullanarak program oluşturur.
Açıklama	Beşinci Hafta Scratch ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Değişken, Sabit
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Top kek, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sınıfa üç farklı çeşit pop kek getirilerek öğrencilerin dikkati çekilir. Bu keklerin biri kakaolu, biri meyveli ve biri de top kek olduğu söylenir. Öğrencilere “annesi kek yaparken izleyen veya kekin nasıl yapıldığını bilen var mı?” diye sorulur. Evet diyen bir öğrenci seçilerek kek yapmak için hangi malzemelerin kullanıldığı sorulur. Tahta değişken ve sabit diye ikiye ayrılarak öğrencinin söylediği malzemeler uygun kısma yazılır. Diğer söz isteyen öğrencilere de söz verilerek uygun yerlere öğrencilerin söyledikleri malzemeler yazılır. Daha sonra öğrenciler üç farklı top kek yapmak için aynı olan malzemelerin hangileri olduğu sorulur. Ortak malzemelerin değişmediği yani sabit olduğu ve ne çeşit kek yapmak istersek bunları kullanmamız gerektiği vurgulanır. İsteğe bağlı olarak farklı çeşit kek olması için kullandığımız malzemelerin de değiştiği, bu yüzden değişken ismini aldığı ifade edilir. Daha sonra tahtaya iki kare çizilir ve bir kenarına “a” yazılır. Öğrencilerden “a” hangi değerleri alabilir diye sorulur. Öğrencilerin farklı değerler söylemesi sağlanır. Farklı değer alabildiği için değişken olduğu ifade edilir. Değişkenlerin sayıların yerini tuttuğu söylenir. Daha sonra Scratch programı açılarak nasıl değişken oluşturulacağı öğrencilere gösterilir ve öğrencilerin de değişken oluşturmaları istenir. Tahtadaki karenin çevresini hesaplamasını değişken kullanarak nasıl yapıldığı öğrencilere gösterilir. Öğrencilerin de karenin alanını Scratch programıyla bulup ekrana yazdıran programı yapmaları istenir. Son olarak öğrencilerin bilişim teknolojileri notlarının ortalamasını hesaplayan programı Scratch’ ta yapmaları istenir. Öğrenciler bitirdikten sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Değişken kavramını tanımlar. • Değişken kullanarak işlem yapar. • Değişken ve sabit kavramlarını ayırt eder.
Açıklama	Altıncı Hafta Scratch ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Koşul, Şart
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	“Yarın hava güzel olursa pikniğe gidelim, olmazsa sinemaya gidelim.” şeklinde sınıfa soru yöneltilerek öğrencilerin dikkati çekilir. Gelen yanıtlar doğrultusunda pikniğe gitmemizin belli bir şarta bağlı olduğu ifade edilir. Eğer yapacağımız seçim bir şarta bağlıysa bunun koşul kavramı olduğu söylenir. Daha sonra Scratch programı açılır ve öğrencilerden değişken oluşturup değişkene değer vermeleri istenir. Değer verdikten sonra ilk yazdığınız değişken daha büyükse ayağa kalkın denir. Öğrencilere bir koşula göre ayağa kalktıkları hatırlatılır. Sonra Scratch programından “Eğer ... ise” kod bloğu kullanılarak, ilk sayı büyükse ekrana “büyük” yazdıran program yazılır ve öğrencilere açıklanır. Öğrencilere gerekli süre verilir ve gerekli yerlerde destek verilir. Daha sonra öğrencilerden hava sıcaklığı 10 derecenin altındaysa “soğuk”, 10 ile 20 derece arasındaysa “ılık” 20 derecenin üzerindeyse ekrana “sıcak” yazdıran programı yazmaları istenir. Öğrenciler programı bitirdikten sonra kendilerinin bir koşul ifadesi oluşturmaları istenir. Öğrenciler koşul ifadesi içeren ifadeyi Scratch programında kod blokları halinde yazmaları ve programı çalıştırmaları istenir. Yardıma ihtiyacı olan öğrencileri ipuçları verilerek doğru cevaba ulaşmalarına yardımcı olunur. Herkes tamamladıktan sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Koşul kavramını tanımlar. • Koşul kavramı kullanarak program yazar.
Açıklama	Yedinci Hafta Scratch ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Koşul, Şart
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	Öğrencilere bir önceki ders işlenen koşul kavramı hatırlatılır. Hatırlatıldıktan sonra not diye bir değişken oluşturmaları istenir. Not değişkeni 1 ise ekrana Geliştirilmeli, 2 ise İyi 3 ise Peki yazdıran programın kodları yazmaları istenir. Öğrencilerin grup arkadaşlarıyla beraber programı yapmaları sağlanır. Öğrenciler tamamladıktan sonra “bazen bu koşullardan başka koşul da barındırabildiği söylenir.” Scratch programından “Eğer ... ise, değilse ” kod bloğu kullanılarak, eğer girilen sayı 1, 2 veya 3 değilse ekrana “Yanlış Giriş Yaptınız” yazdıran program yazılır ve öğrencilere açıklanır. Daha sonra öğrencilere iç içe koşul ifadelerinin de kullanılabileceği ifade edilir. Bununla ilgili eğer klavyeden girilen sayı ” 1 “ise rastgele 1-100 arası sayı üreten ve üretilen bu sayı da “50” den küçükse “Başarısız”, büyükse “Başarılı” yazan program öğrencilere anlatılarak yapılır ve öğrencilerin de aynı anda yapmaları istenir. Daha sonra öğrencilerden iç içe koşul ifadesi kullanarak bir senaryo oluşturmaları istenir. Oluşturamayan öğrencilere yardımda bulunulur. Oluşturdukları senaryoları Scratch’ ta uygulamaları istenir. Öğrenciler programı bitirdikten sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
- Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme - Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri - Koşul kavramı kullanarak program oluşturur. - İç içe koşul kavramı kullanarak program oluşturur.
Açıklama	Sekizinci Hafta Scratch ders planı

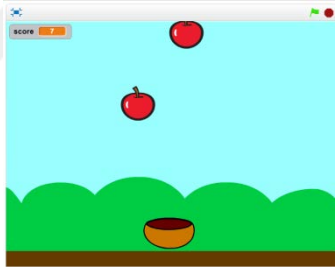
Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır. 4.3 Programın hatasız çalışarak sonuç üretmesini sağlar.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Oyun, Proje, x,y,koordinat
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	Elma toplama oyunu açılarak öğrencilerin dikkati çekilir. Öğrencilere iki hafta sonunda bu oyunu yapacakları söylenir. Öğrencilere “ gözlerinizi kapatın, bir gemide olduğunuzu hayal edin. Birden yakıtınız bitti telefonunuz var ama internetiniz yok yerinizi nasıl tarif edersiniz” diye sorulur. Öğrencilerden fikirler alınır. Sonra tahtanın tam ortasına bir nokta çizilir. Koordinat Sistemi tahtaya çizilerek orta noktanın merkez olduğu ve buradan ne kadar uzaklaşırsak yerimizi yatay ve düşey noktaların kesişiminden tespit edebileceği söylenir. Sahnemizdeki karakterlerin de x ve y değerleri olduğu vurgulanır. Ayrıca sahnenin sınırlarından da bahsedilir. Daha sonra Scratch programı açılarak karakterimizin konumu değiştirilir. Öğrencilerin de kendi karakterlerinin konumlarını aşağı ve yukarı olarak hareket ettirmeleri istenir. Daha sonra rastgele nasıl sayı üretecekleri ve konumlarının nasıl rastgele değiştirebilecekleri öğrencilere gösterilir ve öğrencilerini de aynı anda beraberce yapmaları istenir. Elma oyunu için gerekli görsel açılır ve öncelikle öğrencilerin bu görsellerin aynısını oluşturmaları beklenir. Daha sonra zilin çalmasına kadar öğrencilerin yaptıkları takip edilir. En son olarak ders özetlenir ve öğrencilerin gelecek ders devam etmeleri için projelerini kaydetmeleri istenir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Koordinat kavramını tanımlar. • Rastgele sayı üretir • Karakterin konumunu rastgele değiştirir.
Açıklama	Dokuzuncu Hafta Scratch ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır. 4.3 Programın hatasız çalışarak sonuç üretmesini sağlar.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Scratch, Oyun, Proje																							
Güvenlik Önlemleri (Varsa)																								
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap																							
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon																							
Öğretme – Öğrenme Etkinlikleri																								
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere geçen hafta kaydettikleri projelerini açmaları istenir. Projeksiyondan Elma Yakalama oyunu açılır ve öğrencilerin nasıl çalıştığını hatırlamaları sağlanır.  Öğrencilerin ders sonuna kadar tamamlamaları için öğrencilere rehberlik yapılır. Bütün gruplar dolaşarak süreç kontrol edilir. Öğrencilerden projeleri son halini alınca öğrencilerin kendi projelerini değerlendirmeleri istenir. Son beş dakika yaptıkları oyunları oynamaları sağlanır.																							
Ölçme - Değerlendirme																								
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Proje değerlendirme kontrol listesi <table border="1"><thead><tr><th rowspan="2">İşlem</th><th colspan="2">Puanlama</th></tr><tr><th>Evet</th><th>Hayır</th></tr></thead><tbody><tr><td>Değişken kullanılmıştır.</td><td></td><td></td></tr><tr><td>Döngü kullanılmıştır.</td><td></td><td></td></tr><tr><td>Koşul ifadesi kullanılmıştır.</td><td></td><td></td></tr><tr><td>Ekrana yazı yazdırılmıştır.</td><td></td><td></td></tr><tr><td>Görseller uygun yerlerde kullanılmıştır.</td><td></td><td></td></tr><tr><td>Program hatasız çalışmıştır.</td><td></td><td></td></tr></tbody></table>	İşlem	Puanlama		Evet	Hayır	Değişken kullanılmıştır.			Döngü kullanılmıştır.			Koşul ifadesi kullanılmıştır.			Ekrana yazı yazdırılmıştır.			Görseller uygun yerlerde kullanılmıştır.			Program hatasız çalışmıştır.		
İşlem	Puanlama																							
	Evet	Hayır																						
Değişken kullanılmıştır.																								
Döngü kullanılmıştır.																								
Koşul ifadesi kullanılmıştır.																								
Ekrana yazı yazdırılmıştır.																								
Görseller uygun yerlerde kullanılmıştır.																								
Program hatasız çalışmıştır.																								
Açıklama	Onuncu Hafta Scratch ders planı																							

EK 8: Kodu Game Lab Ders Planları

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.2. Algoritma ve Strateji Geliştirme (algoritma oluşturma mantığı, sözde kod, akış şemaları vb.)
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, Adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Tahta, reçel, tost ekmeği, plastik bıçak
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sınıfa bir robot gibi hareket ederek elimizde bir kavanoz reçel, tost ekmeği ve plastik bıçak getirilerek dikkat öğrencilerin dikkati çekilir. Öğrencilere ” daha önce ekmeğine reçel süren var mı ?” diye sorulur. Seçilen bir öğrenciye söz hakkı verilerek “peki bunu nasıl yaptın bize anlatır mısın?” diye sorulur. Öğrenciden yanıt gelince ” peki beni bir robot olarak düşün, bilirsin robotlar sen bir şey yap demeden yapmazlar, sadece senin dediğini yaparlar ona göre anlat ben de ekmeğime reçel sürüp yiyebileyim.” diye söylenir. Öğrenciden gelen yönergelere eksiksiz uyulur. Öğrencilerden doğru cevap gelene kadar bu etkinlik sürdürülür. Öğrencilere adım adım yapmaları gerektiği fark ettirilir. Bir öğrenci seçilir bu yaptığımız işlemleri sırayla tahtaya yazması istenir. Algoritmanın tanımı yapılır ve tahtada yazılan komutların bir algoritma olduğu ifade edilir. Her şeyin bir komutla başladığı ve bittiği ifade edilir. Daha sonra sınıfın kapısı işaret edilerek “ kapıyı kim nasıl açacağımızı anlatır” diye sorulur. Bir öğrenci seçilir ve uygulayarak anlatması istenir. Öğrenci uygularken adımlar tahtaya yazılır. Yaptığımız işlerin basit gibi görünse de aslında bazı aşamalardan geçtiği söylenir. Öğrencinin tahtaya yazdığı adımlardan bir kaç silinerek bu adımları tekrar etmesi istenir. Öğrenci hedefe ulaşamayınca algoritmanın önemi vurgulanır. Öğrencilerden ikiye bölünmüş grup olmaları ayrı ayrı yapılır bunu grup arkadaşlarıyla birlikte algoritmasını yapmaları istenir. Daha sonra öğrencilerden yazdıkları algoritmayı diğer grupla değiştirmeleri istenir. Doğru cevap tahtaya yazılarak öğrencilerden kontrol etmeleri istenir. Sonuç olarak öğrencilere algoritmanın neden önemli olduğu, bir yerde yanlış yaparsak sadece o adımı düzeltmenin işimizi kolaylaştıracağı ifade edilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Her grup kendi grubunun ve diğer grupların algoritmalarını kontrol eder.
Açıklama	Birinci hafta ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.2. Algoritma ve Strateji Geliştirme (algoritma oluşturma mantığı, sözde kod, akış şemaları vb.)
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Akış şeması, Adım adım düşünme, Algoritma
Güvenlik Önlemleri (Varsa)	Kağıt makası tercih edilir
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma,
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Kağıt, Kalem, Karton, Makas
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sınıfa daha önce bir problemin çözümünün her bir adımı yazılı farklı renkle kartonlar sınıfa getirilerek öğrencilerin dikkati çekilir. Daha sonra “bir problemin adım adım düşünülmesi neden önemlidir” diye sorulur. Öğrencilerden alınan cevaplardan sonra bir önceki ders hatırlatarak özet geçilir. Daha sonra karton sayısı kadar öğrenci tahtaya çıkarılır ve her birine bir karton verilir. Bir öğrenci seçilerek bu karışık olarak verilen kartonları bir problemi n çözümü olacak şekilde sıralamasını istenir. Bu şekilde adım adım sıralamanın algoritma olduğu cevabını alana kadar öğrenci yönlendirilir. Algoritmanın görselleştirilmiş ve ilişkilendirilmiş halinin akış şeması olduğu ifade edilir. Kartonlardaki şekilleri göstererek akış şemasının elemanları olduğu ifade edilir. Daha sonra öğrencilerden ele ele tutuşmaları istenir ve ortadaki iki kişinin ellerini bırakması istenir. Akış şemasının bu şekilde birbiriyle ilişkili olması gerektiği kopuk kısımlarının akış şemasına dahil olmadığı ifade edilir. Geçen ders yapılan algoritmaların akış şemaları yapılarak öğrencilere gösterilir. Öğrencilerden üçerli grup oluşturmaları istenir. Gruplardaki öğrencilerden bir problem bulmaları ve buldukları problemin çözümünün akış şemasını boş kağıtlara çizmeleri söylenir. Daha sonra her bir öğrenciden dersin başındaki gibi akış şemasının elemanlarını kesmeleri ve algoritmaları bunlara yazmaları istenir. Daha sonra öğrenciler bitirdikten sonrakâğıtları karıştırıp diğer gruba karışık şekilde vermeleri sağlanır. Daha sonra öğrencilerden karışık şekilde olan akış şemasının sıralanması ve problemin ne olduğunu bulmaları istenir. Etkinlik tamamlanınca ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Problem kavramını tanımlarım. • Problem çözmenin önemini ifade ederim.
Açıklama	İkinci hafta ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Programlama, Programlama dili, Kod
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon,
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Kodu Game Lab’ ta yapılmış olan motosiklet yarışı oyunu öğrencilere gösterilerek öğrencilerin dikkati çekilir. Daha sonra “siz de bu ve buna benzer oyun yapmak ister misiniz?” diye sorularak öğrencilerin güdülenmesi sağlanır. Bunun için öğrencilere bir programlama dili kullanmaları ve kod yazmaları gerektiği söylenir. Kodu Game Lab programı açılır ve öğrencilerden de programı açmaları istenir. Daha sonra öğrencilere öncelikle Kodu Dünyası gösterilir, yazdığımız kodların burada hayat bulduğu söylenir ve sahnenin nasıl farklı dekorlar alabileceği gösterilir. Sonra karakterler tanıtılır ve farklı karakterler nasıl ekleneceği gösterilir. Öğrencilerden de yapılanları yapmaları istenir. Daha sonra Programlama Menüsü öğrencilere gösterilir ve benzer kodların aynı menüler altında bulunduğu söylenir. Kod oluşturma alanı tanıtılarak kodların sürüklenip bu kısma bırakıldığı söylenir. “Algoritma her zaman ne ile başlar” sorusu sorularak öğrencilerden başla cevabı alınır. Kodu Game Lab’ ta buna karşılık simgesi olduğu ifade edilir. Öğrencilere “sahnedeki karakteri konuşturalım mı?” diye sorulur. Karakterdeki programlama menüsü altında Söyle seçeneği kullanılarak ekranda karakterin konuşma sağlanır. Farklı karakterleri konuşturmak için öncelikle onu seçerek içerisine kodları seçmemiz gerektiği vurgulanır. Daha sonra öğrencilerden istedikleri iki karakter seçmeleri ve karşılıklı olarak bu karakterleri konuşturmaları istenir. Öğrenciler yaptıktan sonra programlarını saklamaları istenir. Bunun için nasıl kaydedileceği gösterilir. Daha sonra tekrar edilerek ders bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Sahneyi değiştirebilir. • Karakter değiştirebilir. • Ekranı yazı yazdırabilir.
Açıklama	Üçüncü Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanıtır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Ve, Veya, Değil, Büyüktür, Küçüktür, +, -, *, /, >, <
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Tahta parçası, Beyaz tahta
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse üç farklı uzunlukta tahta parçası getirilerek öğrencilerin dikkati çekilir. Daha sonra yazı tahtasına iki adet ‘>’ işareti çizilerek öğrencilerden bunları tahta parçalarını bu işaretlerin arasına koymaları istenir. Sonra öğrencilere niçin bu şekilde sıraladıkları sorulur ve ‘>’ işaretin büyüktür işareti olduğu vurgulanır. Bu işareten başka ‘<’ işareti olduğu da söylenir. Daha sonra bir öğrenci seçilerek üç tahtadan en uzun ve en kısa olanı seçmesi istenir. Başka bir öğrenci seçilerek “en uzun veya en kısa tahta parçasını alır mısın?” denir. İlk öğrenciye “niçin bir tane almadın” diye sorulur. İkinci öğrenciye de niçin bir tane aldın?” diye sorulur. Öğrencilere ‘ve’ , ‘veya’ kavramlarının ne olduğu aktarılır. Daha sonra başka bir öğrenci seçilerek “En uzun ve en kısa değilse al “ diyerek öğrencinin ortanca tahta parçasını alması beklenir. Burada da ‘değil’ kavramı açıklanır. Daha sonra “ve”, “veya”, “<”, “>”, “değil” operatörleri yazılarak bunların mantıksal operatörler olduğu vurgulanır. Daha sonra da “+”, “-”, “*”, “/” gibi sayılarla toplama, çıkarma, çarpma ve bölme gibi işlemler yapabilmek için kullanmamız gereken matematiksel operatörlerin de olduğu ifade edilir. Öğrencilere Kodu Game Lab programını açmaları söylenir. Bu ifadelerin Eğer menüsü altındaki menülerde yer aldığı söylenir. Operatörlerin nasıl kullanıldığı uygulamalı olarak öğrencilere gösterilir. Öğrencilerin de gördüklerini uygulamaları istenir. Daha sonra öğrencilere grup arkadaşlarıyla beraber sağlık değeri ne kadar yüksekse hızını ona göre ayarlamaları istenir. Daha sonra sağlık değerini ekrana yazdırmaları istenir. Ders özetlenerek sonlandırılır.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Operatör kavramını tanımlar. • Operatör kavramını kod bloklarında kullanır. • Hangi operatörü seçeceğinin farkına varır.
Açıklama	Dördüncü Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Tekrarla, Döngü, Sürekli tekrarla
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Karton, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	 Şeklinde kesilmiş on iki kartonla sınıfa girilerek öğrencilerin dikkati çekilir. Daha sonra kartonlar yere konularak düz bir sırayla yere konularak ve bunların her biri bir adımı temsil ediyor diyerek öğrencilere aktarılır. Daha sonra bir öğrenci seçilerek bu adımların algoritmasını yapmaları istenir. Algoritmasını yaptıktan sonra öğrencilerin Kodu Game Lab programını açmaları istenir. Yapılan algoritmanın kod blokları Kodu Game Lab programında öğrencilere göstererek yapılır. Sonra aynı olan kodları söylemeleri istenir. Aynı kodları tekrar yazmak yerine döngü kullanılabileceği anlatılır. Döngünün tanımı yapılır. Daha sonra öğrencilere bu kodları daha kısa şekilde yazabileceği döngü kullanarak gösterilir. Öğrencilerden on iki kartonla kare oluşturmaları istenir. Daha sonra bir öğrenci seçilir ve kartonların üzerinden yürümesi söylenir. Başka bir öğrenci seçerek arkadaşının yaptıklarını tahtaya yazması istenir. Kodu Game Lab programında tahtaya yazılan adımların patika oluşturarak kodlarını yazmaları beklenerek gerekli zaman verilir. Öğrenciler kodları yazdıktan sonra öğrencilere karakterin sürekli olarak kare oluşturacak şekilde hareket etmesi için gerekli kod bloğu gösterilir. Son olarak öğrencilerden döngü kullanarak dikdörtgen çizimleri istenir. Herkes bitirdikten sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Döngü kavramını tanımlar. • Döngü kullanarak program oluşturur.
Açıklama	Beşinci Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Değişken, Sabit
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Top kek, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sınıfa üç farklı çeşit pop kek getirilerek öğrencilerin dikkati çekilir. Bu keklerin biri kakaolu, biri meyveli ve biri de top kek olduğu söylenir. Öğrencilere “annesini kek yaparken izleyen veya kekin nasıl yapıldığını bilen var mı?” diye sorulur. Evet diyen bir öğrenci seçilerek kek yapmak için hangi malzemelerin kullanıldığı sorulur. Tahta değişken ve sabit diye ikiye ayrılarak öğrencinin söylediği malzemeler uygun kısma yazılır. Diğer söz isteyen öğrencilere de söz verilerek uygun yerlere öğrencilerin söyledikleri malzemeler yazılır. Daha sonra öğrenciler üç farklı top kek yapmak için aynı olan malzemelerin hangileri olduğu sorulur. Ortak malzemelerin değişmediği yani sabit olduğu ve ne çeşit kek yapmak istersek bunları kullanmamız gerektiği vurgulanır. İsteğe bağlı olarak farklı çeşit kek olması için kullandığımız malzemelerin de değiştiği, bu yüzden değişken ismini aldığı ifade edilir. Daha sonra tahtaya iki kare çizilir ve bir kenarına “a” yazılır. Öğrencilerden “a” hangi değerleri alabilir diye sorulur. Öğrencilerin farklı değerler söylemesi sağlanır. Farklı değer alabildiği için değişken olduğu ifade edilir. Değişkenlerin sayıların yerini tuttuğu söylenir. Daha sonra Kodu Game Lab programı açılarak nasıl değişken oluşturulacağı öğrencilere gösterilir ve öğrencilerin de değişken oluşturmaları istenir. Öğrencilere skor değişkeni oluşturmayı ve skor değişkeninin yukarı tuşuna basılınca artan, aşağı tuşuna basılınca azalan kodlarını yazmaları istenir. Öğrenciler yaptıktan sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Değişken kavramını tanımlar. • Değişken kullanarak işlem yapar. • Değişken ve sabit kavramlarını ayırt eder.
Açıklama	Altıncı Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Koşul, Şart
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	“Yarın hava güzel olursa pikniğe gidelim, olmazsa sinemaya gidelim.” şeklinde sınıfa soru yöneltilerek öğrencilerin dikkati çekilir. Gelen yanıtlar doğrultusunda pikniğe gitmemizin belli bir şarta bağlı olduğu ifade edilir. Eğer yapacağımız seçim bir şarta bağlıysa bunun koşul kavramı olduğu söylenir. Daha sonra Kodu Game Lab programı açılır ve öğrencilerden skor değişkeni oluşturup değişkene değer vermeleri istenir. Değer verdikten sonra ilk yazdığımız değişken daha büyükse ayağa kalkın denir. Öğrencilere bir koşula göre ayağa kalktıkları hatırlatılır. Sonra Kodu Game Lab programından “Eğer” kod menüsü kullanılarak, ilk sayı büyükse ekrana “büyük” yazdıran program yazılır ve öğrencilere açıklanır. Öğrencilere gerekli süre verilir ve gerekli yerlerde destek verilir. Daha sonra öğrencilerden “a” karakterin zıpladığı, “b” tuşuna basıldıysa elma fırlattığı programı yazmaları istenir. Öğrenciler programı bitirdikten sonra kendilerinin bir koşul ifadesi oluşturmaları istenir. Öğrenciler koşul ifadesi içeren ifadeyi Kodu Game Lab programında yazmaları ve programı çalıştırmaları istenir. Yardıma ihtiyacı olan öğrencileri ip uçları verilerek doğru cevaba ulaşmalarına yardımcı olunur. Herkes tamamladıktan sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Koşul kavramını tanımlar. • Koşul kavramı kullanarak program yazar.
Açıklama	Yedinci Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Koşul, Şart
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere bir önceki ders işlenen koşul kavramı hatırlatılır. Öğrencilerden yön tuşları kullanarak karakteri sahne üzerinde hareket ettirmeleri istenir. Öğrencilerin grup arkadaşıyla beraber programı yapmaları sağlanır. Öğrenciler tamamladıktan sonra “bazen bu koşullardan başka koşul da barındırabildiği söylenir.” Kodu Game Lab programından “değilse ” kodu kullanılarak, eğer basılan tuş “a” değilse kalp fırlat uygulaması yapılır. Öğrencilerden de adımları takip etmeleri istenir. Daha sonra öğrencilere iç içe koşul ifadelerinin de kullanılabileceği ifade edilir. Bununla ilgili eğer klavyeden “a” tuşuna basılırsa ve sahnedeki başka bir karaktere değerse ekrandan sağlığı 10 birim azalan ve sağlık 0 olunca da sahneden yok olan program öğrencilere anlatılarak yapılır ve öğrencilerin de aynı anda yapmaları istenir. Daha sonra öğrencilerden iç içe koşul ifadesi kullanarak bir senaryo oluşturmaları istenir. Oluşturamayan öğrencilere yardımda bulunulur. Oluşturdukları senaryoları Kodu Game Lab’ ta uygulamaları istenir. Öğrenciler programı bitirdikten sonra ders özetlenerek bitirilir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Koşul kavramı kullanarak program oluşturur. • İç içe koşul kavramı kullanarak program oluşturur.
Açıklama	Sekizinci Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Kodu Game Lab, Oyun, Proje
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, Gösterip Yaptırma, Soru Cevap
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Projeksiyon, Yazı tahtası
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Motosiklet oyunu açılarak öğrencilerin dikkati çekilir. Öğrencilere iki hafta sonunda bu oyunu yapacakları söylenir. Sonra oyunun ayrıntıları öğrencilere gösterilerek açıklanır. Daha sonra öğrencilere karakterlerin nasıl hareket ettikleri, hangi kavramları kullanmaları gerektiği sorulur. Kodu Game Lab programı açılarak karakterimiz hareket ettirilir, sahne değiştirilir. Diğer karakterlerin yaptıkları hareketler incelenerek öğrencilerden her karakterin ne yaptığının notunu tutmaları istenir. Motosiklet oyunu için gerekli görsel açılır ve öncelikle öğrencilerin bu görsellerin aynısını oluşturmaları beklenir. Daha sonra zilin çalmasına kadar öğrencilerin yaptıkları takip edilir. En son olarak ders özetlenir ve öğrencilerin gelecek ders devam etmeleri için projelerini kaydetmeleri istenir.
Ölçme - Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri • Senaryo oluşturabilir. • Senaryoya uygun davranışlar geliştirir.
Açıklama	Dokuzuncu Hafta Kodu Game Lab ders planı

Yeniceköy Ortaokulu
2016-2017 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	4. ÜNİTE: Problem Çözme, Programlama ve Özgün ürün geliştirme
Konu	4.3 Animasyon ve yazılım geliştirme konusunda temel kavramları tanıtır. 4.3 Programı çalıştırmak için gerekli derleyiciyi kullanır. 4.3 Programın hatasız çalışarak sonuç üretmesini sağlar.
Süre	40 Dakika + 40 Dakika

Ünite Kavramları ve Sembolleri / Davranış Örüntüsü		Kodu Game Lab, Oyun, Proje																							
Öğretme – Öğrenme Yöntem ve Teknikleri		Anlatım, Gösterip Yaptırma, Soru Cevap																							
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça		Bilgisayar, Projeksiyon																							
Öğretme – Öğrenme Etkinlikleri																									
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere geçen hafta kaydettikleri projelerini açmaları istenir. Projeksiyondan Motosiklet oyunu açılır ve öğrencilerin nasıl çalıştığını hatırlamaları sağlanır.																								
	 Öğrencilerin ders sonuna kadar tamamlamaları için öğrencilere rehberlik yapılır. Bütün gruplar dolaşarak süreç kontrol edilir. Öğrencilerden projeleri son halini alınca öğrencilerin kendi projelerini değerlendirmeleri istenir. Son beş dakika yaptıkları oyunları oynamaları sağlanır.																								
Ölçme - Değerlendirme																									
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Proje değerlendirme kontrol listesi																								
	<table><tr><th rowspan="2">İşlem</th><th colspan="2">Puanlama</th></tr><tr><th>Evet</th><th>Hayır</th></tr><tr><td>Değişken kullanılmıştır.</td><td></td><td></td></tr><tr><td>Döngü kullanılmıştır.</td><td></td><td></td></tr><tr><td>Koşul ifadesi kullanılmıştır.</td><td></td><td></td></tr><tr><td>Ekrana yazı yazdırılmıştır.</td><td></td><td></td></tr><tr><td>Görseller uygun yerlerde kullanılmıştır.</td><td></td><td></td></tr><tr><td>Program hatasız çalışmıştır.</td><td></td><td></td></tr></table>		İşlem	Puanlama		Evet	Hayır	Değişken kullanılmıştır.			Döngü kullanılmıştır.			Koşul ifadesi kullanılmıştır.			Ekrana yazı yazdırılmıştır.			Görseller uygun yerlerde kullanılmıştır.			Program hatasız çalışmıştır.		
	İşlem	Puanlama																							
		Evet	Hayır																						
	Değişken kullanılmıştır.																								
	Döngü kullanılmıştır.																								
	Koşul ifadesi kullanılmıştır.																								
	Ekrana yazı yazdırılmıştır.																								
Görseller uygun yerlerde kullanılmıştır.																									
Program hatasız çalışmıştır.																									
Açıklama																									
Onuncu Hafta Kodu Game Lab ders planı																									

EK9. İntihal Raporu



DOKUZ EYLÜL ÜNİVERSİTESİ EĞİTİM BİLİMLERİ ENSTİTÜSÜ YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU



Tarih: 13/07/2018

Tez Başlığı:

6.sınıf öğrencilerine Scratch ve Kodu Game Lab programlama dillerinin öğretiminde öğrencilerin

Tutum, Öz-yeterlilik ve Akademik Başarılarının Karşılaştırılması

Yukarıda başlığı belirtilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam 175 sayfalık kısmına ilişkin, 13/07/2018 tarihinde tez danışmanım tarafından Dokuz Eylül Üniversitesi Kütüphane ve Dokümantasyon Daire Başkanlığı'nın sağladığı İntihal Tespit Programından (Turnitin-Tez İntihal Analiz Programı) aşağıda belirtilen filtreleme tiplerinden biri (uygun olanı işaretleyiniz) uygulanarak alınmış olan orijinallik raporuna göre, tezin benzerlik oranı % 13'dür.

- <http://www.kutuphane.deu.edu.tr/turnitin-tez-intihal-analiz-programi/> adresindeki Tez İntihal Analiz Programı

Kullanım Kılavuzunu okudum ☒

Filtreleme Tipi 1(Maksimum %15) ☒

Filtreleme Tipi 2(Maksimum %30) ☐

☒ Kabul/Onay ve Bildirim sayfaları hariç, ☒ Kaynakça hariç, ☒ Alıntılar dâhil, ☐ Altı (6) kelimeden daha az örtüşme içeren metin kısımları hariç.	☐ Kabul/Onay ve Bildirim sayfaları hariç, ☐ Kaynakça dâhil, ☐ Alıntılar dâhil.
EK 1- İntihal Tespit Programı Raporu İLK SAYFA Çıktısı. ☒ EK 2- İntihal Tespit Programı Raporu (Tümü) Cd İçinde. ☒	

Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Uygulama Esasları'nı inceledim ve yukarıda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Adı Soyadı : Anıl Dinçer
 Öğrenci No : 2013950261
 Anabilim Dalı : Bilgisayar ve Öğretim Teknolojileri Eğitimi
 Programı : Bilgisayar ve Öğretim Teknolojileri Öğretmenliği
 Statüsü : Yüksek Lisans ☒ Doktora ☐

ÖĞRENCİ

Anıl Dinçer, 13.07.2018

DANIŞMAN

Doç. Dr. Bahar BARAN, 13.07.2018

Açıklamalar

- 1: Bu formu teslim etmeden önce sizden istenen bilgileri uygun kutucuğu (☐) işaretleyerek doldurunuz.
Kullanıcı şifre vb. konusunda sorun yaşanması durumunda Üniversitemiz Merkez Kütüphanesinde bulunan Turnitin yetkilisine (Ali Taş Tel: +90 (232) 3018026 veya ali.tas@deu.edu.tr) başvurunuz.
- 2: Yüksek Lisans/Doktora Tez Çalışması Orijinallik Raporu" formu tezin ciltlenmiş ve elektronik nüshalarının içerisinde ekler kısmında yer alır.
- 3: Tez savunmasında düzeltme alınması durumunda bu form güncellenerek yeniden hazırlanır.
- 4: Turnitin-Tez İntihal Analiz Programına yükleme yapılırken Dosya Başlığı (document title) olarak tez başlığının tamamı, Yazar Adı (author's first name) olarak öğrencinin adı, Yazar Soyadı (author's last name) olarak öğrencinin soyadı bilgisini yazınız.