

T.C.

BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ

ANABİLİM DALI

**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ BİLİM
DALI**

**KODLAMA ORTAMLARININ ORTAOKUL
ÖĞRENCİLERİNİN BAŞARI, TUTUM VE BİLGİ
İŞLEMSEL DÜŞÜNME BECERİSİNE ETKİSİ**

Yüksek Lisans Tezi

Hazırlayan

Tarık OTU

BOLU 2020

Tarık OTU

**KODLAMA ORTAMLARININ ORTAOKUL ÖĞRENCİLERİNİN
BAŞARI, TUTUM VE BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİNE ETKİSİ**

YÜKSEK LİSANS.2020

T.C.
BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ ANABİLİM DALI
BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ BİLİM DALI

KODLAMA ORTAMLARININ ORTAOKUL ÖĞRENCİLERİNİN
BAŞARI, TUTUM VE BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİNE
ETKİSİ

Yüksek Lisans Tezi

Hazırlayan

Tarık OTU

Danışman

Doç.Dr. İbrahim ÇETİN

BOLU AĞUSTOS 2020

YÜKSEK LİSANS TEZİ ONAY FORMU

Tarık OTU tarafından hazırlanan “Kodlama Ortamlarının Ortaokul Öğrencilerinin Başarı, Tutum ve Bilgi İşlemsel Düşünme Becerisine Etkisi” başlıklı çalışma jüri tarafından Bilgisayar ve Öğretim Teknolojileri Ana Bilim Dalı Bilgisayar ve Öğretim Teknolojileri Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir. 05/08/2020

Jüri Üyeleri

Akademik Unvan, Adı ve SOYADI

İmza

Üye(Tez Danışmanı): Doç. Dr. İbrahim ÇETİN

Üye Doç. Dr. Ercan TOP

Üye Dr. Öğretim Üyesi Polat ŞENDURUR

Lisansüstü Eğitim Enstitüsü'nün Onayı

Prof. Dr. Osman GÖRÜR

Lisansüstü Eğitim Enstitüsü Müdürü

ETİK İLKELERE UYULDUĞUNA İLİŞKİN BEYAN

Yüksek Lisans Tezi olarak sunduğum, “Kodlama Ortamlarının Ortaokul Öğrencilerinin Başarı, Tutum ve Bilgi İşlemsel Düşünme Becerisine Etkisi” başlıklı çalışmanın yazılmasında bilimsel ve etik kurallara uyduğumu, başkalarının eserlerinden yararlanılması durumunda atıfta bulunduğumu, kullanılan verilerde herhangi bir tahrifat yapmadığımı, tezin tamamının ya da bir kısmının bu üniversitede veya başka bir üniversitede bir tez çalışması olarak sunulmadığını beyan ederim. 16 Haziran 2020



Tarık OTU

TEŞEKKÜR

Tez çalışmam boyunca bütün bilgi, donanım ve birikimini aktaran, mesleki anlamda benim donanımlı bir öğretmen olmamı sağlayan, çalışmam süresinde ümitsizliğe kapılmamı engelleyen, sürekli ince esprileri ile güldüren, bütün sabrıyla bana yol gösteren, tam donanımlı bir akademisyen, gerektiğinde hoca ve arkadaş olan, anlatmakla kelimelerin yetmeyeceği danışman hocam Doç. Dr. İbrahim ÇETİN'e sevgilerimi, saygılarımı ve yetmez ama teşekkürlerimi iletiyorum.

Çalışmamda tıkanıdığım dönemlerde sorularıma cevap veren, moral veren, sürekli iletişim kuran Doç. Dr. Pınar ÇETİN ve Doç. Dr. Ercan TOP'a teşekkür ediyorum.

Yüksek Lisans dönemim boyunca ders aldığım, saygıdeğer öğretmenlerim Prof.Dr. Erkan TEKİNARSLAN'a, Dr. Öğr. Üyesi Melih Derya GÜRER'e, Dr. Öğr. Üyesi Nuh YAVUZALP'e bilgi ve birikimlerini benimle paylaşmaları ve beni akademik anlamda söz sahibi yapacak bilgilerle donattıkları için teşekkürlerimi sunarım.

Çalışmam süresince her zaman yanımda olan Erhan TURGUT, Fatih KAHRUMAN, Yavuz AYDOĞDU, Cevdet ÖZDEMİR, Tarık AKSOY arkadaşlarıma, Ersin AYNAL'a yardımlarından ve desteklerinden dolayı teşekkür ediyorum.

Yüksek lisansa başlamamda cesaret veren, sürekli sabrıyla beni destekleyen, çalışmam süresince ayrı bir hassasiyet gösteren, çocuklarım konusunda fedakârlık yapan, sevgili eşim, annemiz Sanem OTU'ya gönülden sonsuz sevgi ve minnetlerimi sunuyorum.

İÇİNDEKİLER

YÜKSEK LİSANS TEZ ONAY FORMU	i
ETİK İLKELERE UYULDUĞUNA İLİŞKİN BEYAN.....	ii
TEŞEKKÜR.....	iii
TABLolar DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
KISALTMALAR	x
ÖZET	xi
ABSTRACT.....	xiii
I.BÖLÜM.....	1
1.Giriş	1
1.1 Araştırmanın Problem Durumu.....	4
1.2 Çalışmanın Önemi.....	6
1.3 Çalışmanın Amacı	7
1.4 Araştırma Problemleri	8
1.5 Sayıtlar	8
1.6 Sınırlılıklar	8
1.7 Tanımlar	9
1.7.1 Programlama	9
1.7.2 Bilgi İşlemsel Düşünme.....	9
1.7.3 Problem Çözme.....	9
1.7.4 Tutum.....	10
1.7.5 Kodlama.....	10
II.BÖLÜM	11
2.Kuramsal Çerçeve.....	11
2.1 Bilgi İşlemsel Düşünme Tanımı.....	11
2.2 Farklı Yaklaşımlar.....	12
2.3 Kodlama Eğitimi	14
2.4 Ülkelerin Kodlama Eğitimleri.....	15
2.5 21. Yüzyıl Becerileri	21
2.6 Kavram Yanılgıları.....	22
2.6.1 Dil	23

2.6.2	Matematiksel gösterim.....	23
2.6.3	Eski programlama deneyimi	24
2.6.4	Örneklerden aşırı genelleme	24
2.6.5	Kuralların değiştirilmesi	24
2.6.6	Anoloji	24
2.7	Kodlama Eğitiminde Araç Tabanlı Öğretim Yaklaşımları	26
2.7.1	Bilgisayarsız bilgisayar bilimi	27
2.7.2	Blok temelli programlama	28
2.7.3	Metin tabanlı programlama dilleri	32
2.7.4	Robotik kodlama.....	34
2.8	Eşli Programlama	37
2.9	Öğretim Ortamları	42
2.9.1	Scratch offline editör	42
2.9.2	mBlock ve Mbot	44
2.9.3	Python	48
III.BÖLÜM.....		53
3.Yöntem.....		53
3.1	Araştırma Deseni.....	53
3.2	Çalışma Grubu	54
3.3	Veri Toplama Araçları	56
3.3.1	Bilgi işlemsel düşünme testi	56
3.3.2	Programlama tutum anketi.....	56
3.3.3	Kodlama başarı testi.....	57
3.4	Öğretim Ortamları	62
IV.BÖLÜM.....		67
4.Bulgular ve Yorum		67
4.1	Bilgi İşlemsel Düşünme Testine Yönelik Bulgular	67
4.2	Bilgisayar Programlamaya Karşı Tutum Ölçeğine Yönelik Bulgular	74
4.3	Kodlama Başarı Testine Yönelik Bulgular	81
V.BÖLÜM		85
5. Sonuç ve Öneriler		85
5.1. Sonuç.....		85
5.1.1	KBT ile ilgili Sonuçlar	85
5.1.2.	BPTÖ ile ilgili sonuçlar.....	89

5.1.3. BİDT ile ilgili sonuçlar.....	90
5.2 Öneriler	94
5.2.1 Araştırmacılar için öneriler.....	94
5.2.2 Eğitimciler için öneriler.....	95
KAYNAKÇA.....	96
EKLER.....	109
Ek 1 : mblock Ortamı İçin Kodlama Başarı Testi.....	110
Ek 2: Python Ortamı İçin Kodlama Başarı Testi	115
Ek 3: Scratch Ortamı İçin Kodlama Başarı Testi.....	120
Ek 4: mblock Ortamı İçin Hazırlanan Ders Planları	126
Ek 5: Python Ortamı İçin Hazırlanan Ders Planları.....	148
Ek 6: Scratch Ortamı İçin Hazırlanan Ders Planları	169
Ek 7:Etik Kurul İzin	192
Ek 8: Meb İzin Belgesi.....	193
ÖZGEÇMİŞ	194

TABLOLAR DİZİNİ

Tablo 2.1: Örnek olarak Python turtle modülünde yazılmış kodlar.....	48
Tablo 2.2: Tablo 2.1 de yazılan kodların ekran çıktısı.....	49
Tablo 2.3: Python Turtle modülü ile penup ve pendown kodlarının kullanımı.....	49
Tablo 2.4: Python Turtle modülü ile dikdörtgen çizimi.....	50
Tablo 2.5: Python Turtle modülünde kapalı şekilleri doldurmaya yarayan kodlar.....	51
Tablo 3.1: Uygulanan Yarı-Deneysel Desenin Özeti.....	54
Tablo 3.2: Araştırmaya katılanların demografi özellikleri.....	55
Tablo 3.3: Scratch ortamı için hazırlanan örnek soru.....	57
Tablo 3.4: Python ortamı için hazırlanmış örnek soru.....	57
Tablo 3.5: mBlock ortamı için hazırlanmış örnek soru.....	58
Tablo 3.6: Belirtke Tablosu.....	59
Tablo 3.7: Madde Analizi Sonuçları.....	60
Tablo 3.8: Atılan Madde Sonrası Madde Analizi Sonuçları.....	61
Tablo 4.1: BİDT'ye ait betimsel istatistikler.....	68
Tablo 4.2: Kolmogorov-Smirnov testi sonuçları.....	69
Tablo 4.3: Levene testi sonucu.....	69
Tablo 4.4: ANOVA sonuçları.....	70
Tablo 4.5: Kolmogorov-Smirnov testi sonuçları.....	71
Tablo 4.6: Levene testi sonucu.....	71
Tablo 4.7: Kovaryans analizi sonuçları.....	72
Tablo 4.8: Fark puanlarının basıklık ve çarpıklık değerleri.....	73
Tablo 4.9: Kolmogorov-Smirnov testi sonuçları.....	73

Tablo 4.10: İlişkili örneklem t-test sonuçları.....	74
Tablo 4.11: BPTÖ'ye ait betimsel istatistikler.....	75
Tablo 4.12: Kolmogorov-Smirnov testi sonuçları.....	76
Tablo 4.13: Levene testi sonucu.....	76
Tablo 4.14: ANOVA sonuçları.....	77
Tablo 4.15: Kolmogorov-Smirnov testi sonuçları.....	77
Tablo 4.16: Levene testi sonucu.....	78
Tablo 4.17: Kovaryans analizi sonuçları.....	79
Tablo 4.18: Fark puanlarının basıklık ve çarpıklık değerleri.....	80
Tablo 4.19: Kolmogorov-Smirnov testi sonuçları.....	80
Tablo 4.20: İlişkili örneklem t-test sonuçları.....	81
Tablo 4.21: KBT'ye ait betimsel istatistikler.....	82
Tablo 4.22: Kolmogorov-Smirnov testi sonuçları.....	82
Tablo 4.23: Levene testi sonucu.....	83
Tablo 4.24: ANOVA sonuçları.....	84

ŞEKİLLER DİZİNİ

Şekil 2.1: Scratch 3 programının Offline Editör olarak genel görünümü.....	43
Şekil 2.2: mBlock Programının Genel Görünümü.....	44
Şekil 2.3: mBlock programında Robotlar isimli blok başlığı görünümü.....	45
Şekil 2.4: mBot Robotun aksesuarları il genel görünümü.....	46
Şekil 2.5: Tablo 2.3 de yazılan kod bloğunun çıktısı.....	50
Şekil 2.6: Tablo 2.4 te görünen kod bloklarının ekran çıktısı.....	51
Şekil 2.7: Tablo 2.5 de yazılan kod bloklarının çıktısı.....	52
Şekil 2.8: Scratch kodlama ortamında kazanım bazlı örnek ders planı.....	66

KISALTMALAR

ISTE : Eğitimde Uluslararası Teknoloji Topluluğu

MEB: Milli Eğitim Bakanlığı



ÖZET

KODLAMA ORTAMLARININ ORTAOKUL ÖĞRENCİLERİNİN BAŞARI,TUTUM VE BİLGİ İŞLEMSEL DÜŞÜNME BECERİSİNE ETKİSİ

Otu,Tarık

Yüksek Lisans Tezi

Bilgisayar ve Öğretim Teknolojileri Eğitimi Bilim Dalı

Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı

Tez Danışmanı: Doç. Dr. İbrahim ÇETİN

Haziran 2020,xiv+194 Sayfa

Bu çalışmanın amacı farklı kodlama ortamlarında programlama öğretimi alan ortaokul öğrencilerinin programlamaya karşı akademik başarıları, tutumları ve öğrencilerin bilgi işlemsel düşünme becerilerine etkisini ortaya çıkarmaktır. Araştırmanın çalışma grubunu Bolu ili Merkez ilçesinde bir ortaokulda bulunan altı tane altıncı sınıf şubesinden rastgele seçilmiş üç sınıf oluşturmaktadır. Seçilen sınıflara rastgele olacak şekilde blok tabanlı görsel programlama ortamı olan Scratch, blok tabanlı görsel programlama ortamı olan mBlock ve metin tabanlı programlama ortamı olan Python ortamları atanmıştır.

Bu araştırmada yarı deneysel desen kullanılmıştır. Öncelikle seçilen sınıflara ön test olarak öğrencilerin bilgi işlemsel düşünme becerilerini ölçmek için bilgi işlemsel düşünme testi ve programlamaya karşı tutumlarını ölçmek için bilgisayar programlama tutum ölçeği uygulanmıştır. Ön testlerin uygulanmasının ardından sekiz hafta boyunca haftada iki saat olacak şekilde her bir gruba araştırmacı tarafından oluşturulan ders planları ve etkinlikler uygulanmıştır. 8 hafta sonunda öğrencilere son test olarak bilgi işlemsel düşünme becerilerine etkisini ölçmek için bilgi işlemsel düşünme testi ,

programlamaya karşı tutumlarını ölçmek için bilgisayar programlama tutum ölçeği ve akademik başarılarını ölçmek için araştırmacı tarafından geliştirilen kodlama başarı testi uygulanmıştır.

Verileri analiz etmek için temel olarak ilişkili örneklemeler t-test, ANOVA ve ANCOVA testleri uygulanmıştır. Öğrencilerin bilgi işlemsel düşünme testi puanları kovaryans analiz ile karşılaştırıldığında Scratch, mBlock ve Python gruplarının bilgi işlemsel düşünme testi ön test puanına göre düzeltilmiş bilgi işlemsel düşünme testi son test puanlarının ortalamaları arasında anlamlı bir fark bulunmamıştır. Öğrencilerin bilgisayar programlama tutum ölçeği puanları kovaryans analizi ile karşılaştırıldığında grupların programlama tutum ölçeği ön test puanına göre düzeltilmiş programlama tutum ölçeği son test puanlarının ortalamaları arasında anlamlı bir fark bulunmuştur. Çoklu karşılaştırma sonuçlarına göre mBlock grubunda bulunan öğrencilerin programlama tutum ölçeği puanları Scratch grubunda bulunan öğrencilerin programlama tutum ölçeği puanlarından ve Python grubunda bulunan öğrencilerin programlama tutum ölçeği puanlarından anlamlı derecede yüksektir fakat Scratch ve Python gruplarında bulunan öğrencilerin programlama tutum ölçeği puanları arasında anlamlı bir fark bulunmamıştır. Son olarak grupların kodlama başarı testi ortalamaları arasında anlamlı fark olup olmadığını belirlemek üzere ilişkisiz örneklemeler için tek yönlü varyans analizi kullanılmıştır. Scratch, mBlock ve Python grubunda bulunan öğrencilerin kodlama başarı testi ortalamasının en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir. Çoklu karşılaştırma testi sonucunda kodlama başarı testi puanlarına bakıldığında Scratch ile Python grubunda bulunan anlamlı farkın Scratch lehine ve mBlock ile Python grubunda bulunan anlamlı farkın mBlock lehine olduğu gözlemlenmiştir.

Anahtar Kelimeler: bilgi işlemsel düşünme, Kodlama, Python, Scratch, mBlock

ABSTRACT

THE EFFECT OF CODING ENVIRONMENTS ON MIDDLE SCHOOL STUDENTS ACHIEVEMENT, ATTITUDE AND COMPUTATIONAL THINKING SKILLS

Otu, Tarık

Master Thesis

Computer Education and Instructional Technology Field

Department of Computer Education and Instructional Technology

Supervisor Assoc. Prof. İbrahim ÇETİN

Haziran 2020, xiv+194 Pages

The aim of this study is to reveal the effect of middle school students' academic achievements and attitudes towards programming and their computational thinking skills in different coding environments. The study group of the study consists of three randomly selected classes from six sixth grade branches in a secondary school in the central district of Bolu province. Scratch, a block-based visual programming environment, mBlock, a block-based visual programming environment, and Python environments, a text-based programming environment, are assigned to randomly selected classes.

Quasi-experimental design was used in this study. First of all, the computational thinking test to measure the computational thinking skills of the students as a pre-test and the computer programming attitude scale to measure their attitudes towards programming were applied to the selected classes. Following the application of the pre-tests, the lesson plans and activities prepared by the researcher were applied to each group for two hours a week for eight weeks. At the end of 8 weeks, the students were applied the computational thinking test to measure the effect of their computing thinking skills, the computer programming attitude scale to measure their attitudes towards programming, and the coding achievement test developed by the researcher to measure their academic achievements.

Basically, correlated samples t-test, ANOVA and ANCOVA tests were applied to analyze the data. When the computational thinking test scores of the students were compared with the covariance analysis, no significant difference was found between the averages of the computational thinking test posttest scores corrected according to the computational thinking test pretest score of the Scratch, mBlock and Python groups. When the students' computer programming attitude scale scores were compared with the covariance analysis, a significant difference was found between the averages of the programming attitude scale posttest scores corrected according to the programming attitude scale pretest score of the groups. According to the multiple comparison results, the programming attitude scale scores of the students in the mBlock group were significantly higher than the programming attitude scale scores of the students in the Scratch group and the programming attitude scale scores of the students in the Python group, but no significant difference was found between the programming attitude scale scores of the students in the Scratch and Python groups. Finally, one-way analysis of variance was used for unrelated samples to determine whether there was a significant difference between the coding success test averages of the groups. A statistically significant difference was observed between at least two of the coding achievement test average of students in the Scratch, mBlock and Python group. As a result of the multiple comparison test, when the coding success test scores were examined, it was observed that the significant difference in the Scratch and Python group was in favor of Scratch and the significant difference in the mBlock and Python group was in favor of mBlock.

Keywords: computational thinking, Coding, Python, Scratch, mBlock

I.BÖLÜM

1. Giriş

21. yüzyılın gelişiyile beraber öğrencilerden problem çözme, bilgi işlemsel düşünme, eleştirisel düşünme, iş birliği yapabilme, analiz ve sentez gibi birçok beceriyi geliştirmesi beklenmektedir. (Yükseltürk ve Altıok,2015; CSTA, NSF, ve ISTE, 2011) Programlama kavramı bilgisayarın yapması gereken işleri bilgisayara belirli bir sırayla iletme işidir ve bu iş için programlamadaki sıralama, yineleme, özyineleme, koşul ifadeleri gibi kavramları içeren algoritmalar kullanılabilir. (Yükseltürk ve Altıok, 2015). Programlama eğitimi, öğrencilerin günlük hayatta karşılaştıkları problemleri çözme, sistemli düşünme, algoritmik düşünme, işbirliği yapma, mantığını kullanma gibi kavramları geliştirmek için önemli bir eğitimidir (CSTA ve diğerleri 2011).

Cevahir ve Özdemir, (2017) yaptıkları çalışmada öğrencilerin programlama eğitiminde karşılaştıkları zorlukların sebeplerini; programlamada ezber yapma, tekrar etme ve uygulama yapmama eksikliği, kavramları soyut hale getirememe, matematik-aritmetik ve analitik düşünme becerisindeki eksiklikler, döngü ve dizi kullanımında yaşanan problemler olarak söylemişlerdir. Özmen ve Altun, (2014) ise yaptıkları çalışmada programlama eğitimde yaşanan zorlukları; pratik yapamama, programlama konusunda bilgi sahibi olamama ve algoritma oluştururken zorlanma olarak belirtmiştir. Ayrıca çalışmada öğrencilerin de programlama öğrenirken yaşadığı zorlukları ise; programlamaya ait bilgi ve becerisinin eksik olması, programlama mantığını kavrayamaması ve hataları ayıklayamaması olarak söylemiştir.

Programlama eğitiminde karşılaşılan bir çok sorunu gidermek için, öğrencilerin programlama derslerine olan ilgisini motivasyonunu arttırmak, programlamada

kullanılan kavramlar konusunda kaygılarını azaltmak için görsel programlama dillleri kullanılabilir (Fesakis ve Serafeim, 2009; Ozoran, Çağıltay, ve Topalli, 2012; Demir, 2015). Günümüzde bir çok kodlama ortamının geliştirilmiş olmasını temele alarak, bu tez çalışmasında ortaokul öğrencilerinin farklı ortamlarda kodlama eğitimi almalarını sağladıktan sonra çeşitli değişkenler açısından karşılaştırılması yapılarak incelenecektir.

Programlama eğitiminde yaşanan zorluklar beraberinde öğrencilerin derse olan ilgisini azalatarak, herhangi bir şeyi başaramama hissine kapılmasını ve öğrencilerin derse olan tutumlarını da olumsuz yönde etkileyebilir. (Dinçer, 2018) Bu zorluklardan kurtulmak için Scratch görsel programlama ortamı gibi programlama öğretimini görece kolay hale getiren (Maloney, Resnick, Rusk, Silverman, ve Eastmont, 2010), öğrencilere daha eğlenceli bir öğrenme ortamı sunan (Lopez, Gonzalez,ve Cano, 2016) öğrencilerin derse olan tutumlarını (Gülbahar ve Kalelioğlu,2014; Yüksel, 2017; Oh, Lee, ve Kim, 2010) ve özyeterlik algılarını (Yükseltürk ve Altıok,2016) olumlu yönde etkileyen ortamlar kullanılabilir.

Kendi ülkelerinde yazılım üreten Microsoft, Google ve Facebook gibi firmalar, dünya ülkeleri arasında kendi ülkelerine kattıkları değer göz önüne alındığında programlama eğitiminin sınırlarının daha net bir şekilde çizilip 21. Yüzyılda programlama eğitiminin bireyler için zorunlu olacağını söylemişlerdir (Sayın ve Seferoğlu, 2016). Bu araştırmalar baz alınarak programlama eğitime ne zaman başlanacağı, programlama eğitimi için hangi ortamların kullanılacağı, hangi ön bilgi ve becerilerin olması gerektiği, programlama eğitiminin nasıl,kimler tarafından verilmesi gerektiği gibi birçok problemin ortaya çıktığı görülmüştür.

Günümüzde programlama eğitime bakıldığında dünyada da bir çok ülkede programlama eğitimi küçük yaştan itibaren ülkelerin müfredatlarındaki yerini almaktadır. Dünyadaki ülkelere bakıldığında İngiltere, 2014 yılından itibaren 5-14 yaşlarındaki öğrenciler için kodlama eğitimi müfredatlarına yerleştirmiş ve çocuklarla ilgili küçük yaşlardan itibaren temel algoritma mantığını kavramalarını, programlama

eğitim mantığının kavranması ve yazılım ile ilgili kullanılacak programları iyi kullanmalarını hedeflemiştir (Öndeş, 2016). Özellikle ABD, Avusturalya, Güney Kore ve Tayvan gibi ülkeler de müfredatlarına küçük yaştan itibaren kodlama ve programlama eğitimlerini takviye etmişlerdir (Sayın ve Seferoğlu, 2016; Demirer ve Sak, 2016).

Ülkemizde kodlama ve programlama dersine verilen önem, 2012 yılında geliştirilen BTY programı ile başlamıştır. Dijital vatandaş yetiştirmek, iş birliği içerisinde öğrenmeyi desteklemek, bilgi paylaşımı sağlayarak öğrenmeyi gerçekleştirmek amaçlanmıştır. Gelişen teknolojilere paralel olarak 2017 yılında taslak bir program oluşturulmuştur. Taslak program uzmanların görüşüne sunulmuş ve gelen dönütlerle yeniden düzenlenip 2018-2019 eğitim-öğretim yılında 5. ve 6. sınıflarda uygulanmak üzere yayımlanmıştır. Yenilenen programdaki amaçlar eskisine benzer olup eski programdan farklı olarak bilişim teknolojilerini etkili kullanma, amacına uygun kullanma, bilgisayar bilimi ile ilgili beceri geliştirme, dijital vatandaş geliştirmek gibi kavramları öğrencilere iletmeyi amaçlamıştır. Bu öğretim programında ünite temelli bir yaklaşım benimsenmiştir. Bilişim teknolojileri; iletişim, araştırma ve işbirliği, ürün oluşturma, problem çözme ve programlama adı altında beş ünite belirlenmiştir (Milli Eğitim Bakanlığı [MEB], 2018).

Anasınıflarından ortaöğretim 12. sınıfa kadar kodlama öğretiminde kullanılan Lego Minstorms, Mbot, Makey-Makey gibi birçok set bulunmaktadır ve bu kitlerle beraber hazır kod yayınlarına başlanmıştır. Yayınlanan bu hazır kodların öğrencilerde programlamaya karşı olumsuz bir durum yarattığı ve ezberciliğe yönlendirdiği söylenebilir. Programlamadaki amaç, programlama mantığını kavratmak ve ezber olan kavramalardan uzak durmaktır. (Cevahir ve Özdemir, 2017)

Sonuç olarak programlamanın dünyada ve ülkemizde önemli bir yere sahip olduğu söylenebilir. Dolayısıyla öğrencilerin kodlama ve programlama eğitimini hangi ortamlarda hangi seviyelerde alması gerektiği gibi birçok soruya cevap aranmaktadır. Literatürdeki bu veriler ışığında bu çalışma ile öğrencilerin görsel programlama, robotik kodlama ve metin tabanlı programlama ortamlarında verilen öğretimin 6. sınıf

öğrencilerinin programlamaya karşı tutumlarına, kodlama başarılarına ve bilgi işlemsel düşünme becerilerine etkisini araştırmak amaçlanmaktadır.

1.1 Araştırmanın Problem Durumu

Günümüzde teknolojinin gelişmesi ve hızla yaygınlaşması beraberinde öğrencilerin teknolojiye ayak uydurmaları için birtakım standartları yakalamaları gerekmektedir (ISTE, 2020).

Wing (2006) yaptığı çalışmasında bilgi işlemsel düşünme becerisinin 21. Yüzyıl becerilerinden olduğunu söylemektedir. Bundy (2007) araştırmasında bilgi işlemsel düşünme becerisinin bir devrim olduğundan bahsedip bu konuda araştırma yapanların farklı sorular üretip ve farklı cevaplar bulmalarını ve bunları kabullenmelerini sağladığını söylemiştir. “21. Yüzyılı anlamamız için önce bilgi işlemsel düşünmeyi anlamalısınız” ifadesiyle bilgi işlemsel düşünmenin 21. Yüzyıl becerileri için ne kadar önemli olduğunu vurgulamıştır. Pfeiffer (1962) yaptığı çalışmasında bilgisayarların insanların en büyük yardımcısı olduğunu ve insanların etrafında bilgisayarların olması bile düşünme biçimini değiştirmesi gerektiğini söylemiştir. Buradaki düşünme biçimi bilgi işlemsel düşünme becerisi olabilir.

Programama öğretimi ve bilgisayar bilimlerinin öğretilmesi için öğrencilere kazandırılması düşünülen kavramların başında algortimik düşünme, yaratıcılık ve problem çözme gelmektedir (Güneş ve Karabak, 2013). Programlama eğitiminde verilmesi gereken becerilerden biri de algoritmik düşünme becerisidir. Algoritma sonu olan işlemleri tanımlamak için ardışık olan adımları açık çık yazmayı içerir. Programlama eğitiminde en çok kullanılan kavramların başında algoritmalar gelir. Bu kavram öğrencilerin analiz-sentez ve tasarım yapmalarına da rehberlik eder (Gökoğlu, 2017).

Genel olarak ortaöğretim ve lisans öğrencilerine verilen programlama öğretimi, son yıllarda gelişmiş ve gelişen ülkelerde daha erken yaşlarda verilmeye başlanmıştır. Erken yaşlarda programlama eğitimi vermenin amacı; öğrencilere programlamayı sevdirmek, programlama becerilerini geliştirmek ve aynı zaman da erken yaşta eğitim vererek daha çok kitleye hitap etmek olarak söylenebilir (Balanskat ve Engelhardt, 2014)

Dünyada bir çok ülkenin müfredatına giren kodlama eğitimi adından oldukça fazla söz ettirmeye başlamış ve ülkelerin müfredatlarında seçmeli ve zorunlu ders olarak girmeyi başarmıştır (Demirer ve Sak, 2016; Sayın ve Seferoğlu, 2016). Kodlama eğitiminin dünyadaki bir çok ülkenin müfredatına zorunlu olarak girmesinin sebebi olarak; öğrencilere yaratıcı düşünme, bilgi işlemsel düşünme, algoritmik düşünme becerilerini kazandırmaya yönelik olası ortamları oluşturma olduğu söylenebilir (Shin, Park, ve Bae, 2013). Kodlama eğitiminde kullanılan kavram ve beceriler soyut ve öğrenilmesi zor olarak değerlendirilebilir. Bu sorunu aşmak için görsel programlama ortamları kullanılabilir. Görsel programlama ortamlarının kullanılması programlama öğretiminde kullanılan kavramları somutlaştırmaktadır (Ersoy, Madran, ve Gülbahar, 2011). Oyun temelli programlama yapmak öğrencilerin , hem aktif olarak sürecin içinde yer almasını sağlamakta, hem yazdıkları kodun somut olarak sonuçlarını gördükleri için daha kalıcı öğrenmelerin gerçekleşmesini sağlamakta, hem de öğrencilerin bir oyun hakkında seneryolar geliştirerek yaratıcı düşüncelerini sağladıkları görülmektedir (Erekmekçi ve Fidan , 2012).

Simith, Cypher, ve Tesler, (2010) çalışmalarında bilgisayarın düşünme biçimine uygun olan programlama dillerinin metin tabanlı programlama (C++,Java v.b.) dili olduğunu söylemişlerdir. Görsel programlama dilleri ise metin tabanlı programlama dillerine göre daha güçsüz olmalarına rağmen insanların konuştukları dile en yakın olan programlama ortamlarıdır (Lye ve Koh, 2014). K12 seviyesinde metin tabanlı programlama yerine görsel programlamanın kullanılması öğrenciler için kolaylık sağlayabilir. Çünkü görsel programlama dillerinde metin tabanlı programlamada karşılaşılan söz dizimi hataları yoktur. Öğrenciler metin tabanlı programlama dillerinde programlamanın yapısıyla daha çok uğraşmak zorunda olduklarından, görsel

programlama ortamında programlamanın mantığına daha çok odaklanabilir (Kelleher ve Pausch, 2005).

Alanyazın incelendiğinde öğrencilerin birden fazla kodlama ortamlarında programlama öğretimi yaptığını görmekteyiz. Öğrenciler blok tabanlı kodlama ortamlarında kod bloklarını belirli bir mantığa göre dizerek oluştururlar ve sonuçları animasyon veya oyun olarak ekranda görmektedirler. Robotik kodlama yapılan ortamlarda ise yine aynı şekilde blok tabanlı görsel programlama dillerini kullanarak kurmuş oldukları seneryolara göre çıktıkları ekranda değil robotlar üzerinde görmektedir. Bu çalışmada Scractch ve mBblock ve Python kodlama ortamları kullanılmıştır. Scratch, mBlock ve Python kodlama ortamlarında aynı kazanımlar ve etkinlikler kullanılmış olup aynı süre zarfında öğrencilerin bilgi işlemsel düşünme becerilerine, akademik başarılarına ve öğrencilerin programlamaya karşı olan tutumlarında değişiklik gösterip göstermediği çalışmamızda aranan soruların temellerini oluşturmaktadır.

Alanyazında görsel programlama ortamları ve robotik kodlama ortamlarında yapılan birçok araştırmanın olduğu görülmektedir. Ancak robotik kodlama, görsel programlama ve metin tabanlı programlama ortamlarında programlama öğretimi yapılan, üç ortamın aynı anda kullanıldığı çalışmaların olmadığını görmekteyiz. Bu çalışmanın hem alan yazına katkı sağlayacağını hem de ortaokul seviyesinde çalışan bilişim teknolojileri öğretmenlerinin derslerinde hangi kodlama ortamlarını kullanabilecekleri ve hangi etkinlikler eşliğinde kodlama eğitimi yapabilecekleri konusunda rehber olacağı düşünülmektedir.

1.2 Çalışmanın Önemi

Alanyazın tarandığında kodlama eğitiminin dünya çapında bir geçerlilik kazandığı ve küçük yaştan itibaren öğrencilere verilmeye başlandığı görülmektedir. Ayrıca son zamanlarda bu konular ile ilgili yapılan araştırmaların

sayısında ciddi bir artış bulunmaktadır. Fakat alanyazın incelendiğinde özellikle ortaokullarda metin tabanlı programlama ortamları kullanılarak yapılan çalışmaların oldukça sınırlı olduğu görülmektedir. Buna ek olarak kodlama eğitime yönelik ölçme araçlarının eksiklikleri de araştırmaları sınırlandırmaktadır.

Bu çalışmada Scratch, mBlock ve Python kodlama ortamlarında yapılan öğretimin ortaokul 6. sınıf öğrencilerinin programlama akademik başarılarına, programlamaya karşı tutumlarına ve bilgi işlemsel düşünme becerilerine etkisini araştırmak hedeflenmiştir. Programlama eğitiminin erken yaşta verilmesinin bilgi işlemsel düşünme, analitik düşünme, algoritmik düşünme, analiz yapabilme, sentez yapabilme, yaratıcılık, problem çözme gibi bir çok beceriyi geliştirdiği iddia edilmektedir. Bu iddiaların araştırmak için daha fazla çalışmaya ihtiyaç duyulduğu söylenebilir. Özellikle metin tabanlı programlama ortamlarının bu çalışmada kullanılmasının alanyazına katkı sağlayacağı düşünülmektedir. Bu anlamda kodlama ortamlarının öğrencilerin akademik başarılarına, programlamaya olan tutumlarına ve bilgi işlemsel düşünmelerine olan etkisini inceleyip araştırmak önemli olacaktır.

1.3 Çalışmanın Amacı

Araştırmanın amacı; programlama öğretiminin, yazılan kodların sonuçlarının bilgisayar ekranında görüldüğü blok tabanlı görsel programlama ortamı olan Scratch, Mbot robotlar üzerinde görüldüğü blok tabanlı görsel programlama ortamı olan Mblock ve bilgisayar ekranında görüldüğü Python metin tabanlı programlama ortamının benzer etkinlikler eşliğinde ve aynı sürede verilen öğretimin öğrencilerin akademik başarılarına, programlamaya karşı tutumlarına ve bilgi işlemsel düşünmelerine olan etkiyi incelemektir. Araştırma sonuçlarına göre verilere dayalı kanıtlar eşliğinde ortaokul öğrencilerinin hangi programlama ortamını kullanması gerektiği konusunda alanyazına katkı sunmak hedeflenmiştir. Ayrıca öğretmenler ve araştırmacılar bu tez kapsamında geliştirilen ders planlarını kendi bağlamlarında kullanabilirler.

1.4 Araştırma Problemleri

- Scratch, Mblock ve Python ortamlarında öğrenim gören altıncı sınıf öğrencilerinin programlamaya karşı tutumları arasında anlamlı bir fark var mıdır?
- Scratch, Mblock ve Python ortamlarında öğrenim gören altıncı sınıf öğrencilerinin programlama akademik başarıları arasında anlamlı bir fark var mıdır?
- Scratch, Mblock ve Python ortamlarında öğrenim gören altıncı sınıf öğrencilerinin bilgi işlemsel düşünme becerileri arasında anlamlı bir fark var mıdır?

1.5 Sayıtlar

- Ortaokul öğrencilerinin programlamaya karşı tutumlarını ve akademik başarılarını ölçmek için kullanılan veri toplama aracındaki sorulara tarafsız ve doğru cevap verdikleri kabul edilmiştir.
- Ortaokul öğrencilerinin araştırmaya katılan sınıfların akademik başarılarının birbirine eşit olduğu kabul edilmiştir.
- Ortaokul öğrencilerinden programlamaya karşı tutum anketine ve başarı testine katılmayan öğrenciler etkisiz olarak kabul edilmiştir.

1.6 Sınırlılıklar

- Araştırmanın çalışma grubu Bolu merkez Orhangazi ortaokulunda bulunan altı altıncı sınıf şubesinden rastgele seçilen üç sınıf ile sınırlıdır.

- Araştırma için toplanan veriler 2018-2019 eğitim-öğretim yılı bahar dönemi ile sınırlıdır.
- Araştırmada kullanılan değişkenler Scratch, Mblock ve Python ortamları ile sınırlıdır.
- Araştırma, geliştirilen veri toplama araçları ile toplanan verilerle sınırlıdır.

1.7 Tanımlar

1.7.1 PROGRAMLAMA

Bilgisayarların kullanıcılar ile iletişimini sağlayan yazılımların oluşturulduğu platformlardır (Yükseltürk ve Altıok; 2017).

1.7.2 BİLGİ İŞLEMSEL DÜŞÜNME

“Bilgisayar ile ilgili terimlerin kullanılarak problem çözme,şş biliminin temel kavramlarını kullanarak problem çözmeyi, sistem tasarlamayı ve insan davranışlarını anlamlandırmayı içerir” olarak tanımlanmıştır (Wing, 2006,s.33).

1.7.3 PROBLEM ÇÖZME

Karşılaşılan sorunların cevaplarını arama, planlama, verilen görevi başarıyla tamamlama, bir konuya ilgi gösterme veya bir imkan gösterme olarak tanımlanmıştır.

(Aslan, 2002). Problem çözme herhangi bir problemin tanımlanması, uygulama haline getirilmesi ve değerlendirilmesi olarak tanımlanmıştır. (Kuzgun, 1992).

1.7.4 TUTUM

Smith (1968)'e göre tutum “bir bireye atfedilen ve onun bir psikolojik obje ile ilgili düşünce, duygu ve davranışlarını düzenli bir biçimde oluşturan bir eğilimdir” şeklinde tanımlamıştır. Thurstone (1967)'ye göre “psikolojik bir objeye karşı oluşan olumlu ya da olumsuz bir yoğunluk sıralaması” olarak tanımlamıştır. Tezbaşaran(1996)'ya göre tutum ise kişilere, kavramlara, nesnelere ve durumlara karşı olumlu yada olumsuz olan davranışlar şeklinde tanımlamıştır.

1.7.5 KODLAMA

Programlama sürecindeki işlem den birisi olan kodlama herhangi bir problemin çözümün için belirlenen senaryonun bilgisayara anlayacağı dilden yazılmasıdır. Programlama ve kodlama kavramlarının genellikle karıştırıldığı ancak programlamada problemin çözümü için ilk önce analiz yapılması , daha sonra kodlama yapılmasını gerektirecek bir yapısı olduğu söylenebilir. (Erümit ve Berigel , 2018)

II.BÖLÜM

2. Kuramsal Çerçeve

2.1 BİLGİ İŞLEMSEL DÜŞÜNME TANIMI

Bilgi işlemsel düşünme becerisi son yıllarda artarak akademisyenlerin, eğitimcilerin ve politikacıların gündemine girmiştir. Wing (2006) bilgi işlemsel düşünmenin sadece bilgisayar bilimcileri değil herkes tarafından geliştirilmesi gereken bir beceri olduğunu vurgulamıştır. Aslında bilgi işlemsel düşünme terimi Papert (Papert ve Harel, 1991) tarafından daha önceden kullanılmıştır. Fakat Wing (2006)'nın bilgi işlemsel düşünme isimli makalesi ile birlikte bilgi işlemsel düşünme becerisi sadece bilgisayar bilimcilerinin değil toplumun geniş kesimlerinin ilgisini çeken bir alan olmuştur.

Bilgi işlemsel düşünmenin tam olarak bir tanımı bulunmamaktadır ve birçok tanımı beraberinde getirdiği görülmektedir. Wing (2006) bilgi işlemsel düşünmenin “bilgisayar biliminin temel kavramlarını kullanarak problem çözmeyi, sistem tasarlamayı ve insan davranışlarını anlamlandırmayı” içerdiğini belirtmiştir. Wing makalesinde zor ve karmaşık problemleri parçalara ayırarak, soyutlama yaparak, algoritmik düşünerek problem çözmeyi vurgulamıştır.

Wing Cuny ve Snyder ile birlikte (Wing, 2010) tanımını güncellemiştir ve bilgi işlemsel düşünmeyi” Çözümlerin bir bilgi işleme birimi tarafından etkili bir şekilde yerine getirilebilecek formda sunulması amacıyla problemleri ve çözümleri

formüllemeyi içeren düşünce süresi “olarak tanımlanmıştır. Bilgi işleme birimi sadece bilgisayar olmak zorunda değildir. Bir insan, robot veya DNA olabilir. Aslında bu süreç bir düşünme süreci olarak nitelendirilmiş olup, bu süreç içerisinde çözüm ve problemi formülleştirme esas alınmıştır. Ayrıca Wing (2006) bilgisayar bilimcilerine tavsiyelerde bulunarak günümüz bireylerinin bilgi işlemsel düşünme konusunda cesaretlendirilmelerini tavsiye etmiştir.

Brennan ve Resnick (2012) ise bilgi işlemsel düşünmeyi kendi yaptıkları Scratch uygulamasını göz önüne alarak tanımlamışlardır. Buna göre bilgi işlemsel düşünmenin bilgi işlemsel kavramlar (işlem adımları, döngüler, olaylar, koşul ifadeleri vb.), bilgi işlemsel pratikler (test etme, hata ayıklama) ve bilgi işlemsel perspektifler (kendini ifade etme, diğer insanlarla etkileşim içerisinde olma) olmak üzere üç boyutunun olduğunu söylemiştir. Kafai ve Burke (2014) bilgi işlemsel düşünmeyi takdir etmişler ve buna ek olarak bilgi işlemsel düşünmeyi bireysel bir etkinlik olarak görmeyi eksik olacağını belirtmiştir. Bilgi işlemsel düşünmenin bireysellikten çıkıp toplumsal olması gerektiğini öne sürmüşlerdir.

2.2 FARKLI YAKLAŞIMLAR

Wing(2006)’nın yaptığı çalışma da bilgi işlemsel düşünme isimli makalesi yaygın bir pozitif etkiye sahip olsa da bu düşünceleri eleştirenler de bulunmaktadır. Denning (2009), Wing’in düşüncelerinin bilgi işlemsel düşünmenin temelini oluşturmadığını ve aslında bu düşüncelerin daha eskiye dayandığını iddia etmiştir. Denning (2014) Wing’in bilgisayar biliminin tarihini görmezden geldiğini söyleyerek aslında 1980’li yıllardan beri bilgi işlemsel düşünmenin hayatımızda olduğunu söylemiştir ve hareketin o zaman başladığını savunmuştur.

Denning (2009) yapay bilgi süreçlerinin yanında doğal olarak gerçekleşen bilgi işleme süreçlerinin olduğunu da vurgulamıştır. Buna örnek olarak DNA Transkripsiyonu ve kuantum kriptografiyi vermiştir. Doğal ve yapay yapılan bilgi işlem Wing 'in tarif ettiği bilgi işlemsel düşünmenin temelini daha ötesinde olduğunu savunmuştur. Ayrıca bilgi işlemsel düşünme sadece bilgisayar bilimine ait olmadığını savunan Denning buradan yola çıkılarak bilgisayar biliminin tanımının yapılamayacağını söylemiştir.

Barr (2014), Papert ve Wing'in makalelerini karşılaştırarak farklı bir bakış açısı geliştirmiştir. Wing(2006) makalesinin problem çözme bakımından tek yönlü olduğunu ileri sürmüştür. Problemin ne olduğunun yanında problemi çözecek olan bilgi işleme biriminin nasıl modelleneceği de ifade edilmelidir. Barr (2014) Papert'ın yaklaşımını ise çift yönlü olarak ele almıştır. Problem çözerken farklı problem çözme yöntemlerini deneyerek yeni problem çözme yöntemi oluşturmaya çalıştığını söylemiştir.

Hemmendinger (2010), Wing'in tanımının sadece bilgi işlemsel düşünme becerisine ait olmadığını aslında birçok alanı ilgilendirdiğini savunmuştur. Bilgisayar biliminin farklı alanlarda kullanılması o alandaki çözümlerin bilgisayar bilimcilerine ait olduğu anlamına gelmez. Analiz, sentez, hata ayıklama, diyagram çizme ve bunun gibi alanların bilime ait olduğunu ve bilimin bilgi işlemsel düşünmenin bu ortak kavramlarını kullandığını söylemiştir. Bilgi işlemsel düşünmenin değerini sadece bilgisayar bilimi alanında yapılan çalışmalarla anlatmaktansa diğer alanlarda yapılan çalışmalarda bilgisayar biliminin kullanıldığını uygulamalara bakmanın daha doğru olacağını söylemiştir. Aynı zamanda bilim insanlarına bilgisayar bilimcisi gibi düşünme yerine onlara nasıl düşünülmesi gerektiği vurgulanmalıdır. Hemmendinger (2010) bilgi işlemsel düşünme becerisinin çok abartıldığını ve bilgi işlemsel düşünme becerisinin asıl amacının dışına çıktığını savunmuştur.

2.3 KODLAMA EĞİTİMİ

Dünyada teknolojinin gelişmesiyle beraber insanların ihtiyaçları da değişmekte ve gelişmektedir. İnsanlar teknolojinin gelişmesiyle beraber bilgisayar yazılımlarını daha fazla kullanmaya başlamışlardır. İhtiyaçlarına göre ya hazır yazılımlar ya da kendi hazırladıkları yazılımları kullanmaya başlamışlardır. Yazılımlarının kendileri tarafından hazırlanmasının sebebi ise hazır olan yazılımların ihtiyaçlara cevap vermemesidir. (Eryılmaz, 2003) Ülkemizde de kodlama dersleri erken yaşta müfredata girmiştir. 2013 yılından itibaren 5. ve 6. sınıflarda Bilişim Teknolojileri ve Yazılım dersinde, ders içeriğinde kodlamaya ve bileşenlerine yer verilmiştir.

Kodlama dersleri ileri seviye zihinsel etkinlikleri desteklemede kullanılabilecek bir araç olabilir. Bu dersler öğrencilerin karşılaştıkları problemleri çözebildikleri, hatalarını düzeltebildikleri ve analiz ve sentez adımlarına ulaşabildikleri ortamları yaratmak için kullanılabilirler. Kodlama eğitimi içerisinde bulunan medya araçları ile çocukların ilgisini çekerek motive etmektedir. Kodlama ortamlarının çeşitliliği bu ilgiyi daha da arttırmaktadır. Kodlama eğitiminin bilgisayar başında medya araçları ve oyunlaştırılarak verilmesi de bu eğitime öğrencilerin daha motive olmasını sağlamaya yardımcı olabilir (Baz, 2018).

Papert eğitim kuramlarından olan inşacılık kuramını en basit haliyle “yaparak öğrenme”(learning by making) şeklinde tanımlamıştır. (Papert ve Harel, 1991) Papert inşacılık kuramında, yapılandırmacılık yaklaşımının yaparak öğrenme kavramının üzerinde durmuştur. (Papert ,1999) İnşacılık yapılandırmacı yaklaşımda olduğu gibi öğrencinin pasif değil tamamen aktif olduğu bir kuramdır. İnşacılık “harici nesnelere”(external artifact) ve “diğerleri ile paylaşma” konusunda yapılandırmacı yaklaşımdan farklı olarak bu kavramların üzerinde daha çok durmaktadır. (Üçgül, 2017)

Papert ve Harel, (1991) inşacılık kuramını tanımlarken kuramın İngilizce anlamı olan “Constructivism” ve “Constructionism” kelimelerinde V ve N harfleri arasında farklılık olduğunu vurgulayarak kuram içerisinde benzerlik ve farklılık olduğuna dikkat çekmiştir. Yapılandırmacılık kuramı bilgiyi yapılandırmanın bilgiyi oluşturma süreci olarak kabul eder, inşacılık kuramı ise yapılandırmacılığın öğrenmeye bakış açısını kabul eder. İnşacılık aynı zaman da öğrenmenin gerçek hayatla bağlantı kurularak ortaya bir ürün çıkararak oluştuğunu savunmaktadır. (Papert ve Harel, 1991)

Kaliteli insan yetiştirmenin birçok yolu vardır. Bu yollardan biri de çocukların erken yaşta programlama eğitimi alması olduğu düşünülmektedir. (Yükseltürk, Altıok, ve Üçgül, 2016) Teknolojinin gelişmesiyle beraber gelişmiş ülkelerde programlama eğitimi ve kodlama eğitimi erken yaşta verilmeye başlanmıştır. Kodlama eğitimini alan öğrencilerin yazılım alanında ilerleme katetmeseler bile günlük hayatta öğrendikleri birçok beceriyi kullanmaktadırlar. Sadece bilişim alanında değil de başka alanlarda da başarılarına etkileri olacaktır. Kodlama, programlama eğitimi alanların kazanmış olduğu bazı beceriler; hata ayıklama , fikir üretme, akran öğrenmesi , işbirliğidir. (Karabak ve Güneş, 2013)

Yapılan araştırmalar incelendiğinde ülkelerin programlama eğitimi vererek hedefledikleri beceriler problem çözme becerisi, kodlama eğitimi, mantıksal düşünme gibi becerileri geliştirmeye yönelik olduğu görülmektedir. (Sayın ve Seferoğlu, 2016)

2.4 ÜLKELERİN KODLAMA EĞİTİMLERİ

Her ülkenin kodlama konusunda eğitim anlayışı farklılık göstermektedir. Kodlama etkinliklerini müfredata uyumlu hale getiren ülkeler olduğu gibi, 5-16 yaş eğitim müfredatına koyan ülkeler de vardır (Balanskat ve Engelhardt, 2014). Ülkelerin kodlama üzerine yaptıkları çalışmalara bakıldığında Belçika’da “Bilgi İşlemsel Düşünme

ve Programlama” şeklinde yer aldığını ve Bulgaristan’da “Algoritmik Problem Çözme ve Programlama” olarak müfredata girdiğini görülmektedir (Balanskat ve Engelhardt, 2014).

Amerika’da kodlama eğitimi bütün eyaletler ve hemen hemen bütün şehirler A-12 öğretim programına entegre edilmiştir. 2000 civarında sınıfta programlama eğitimi verilmiş ve 50000 civarında öğretmen, kodlama atölyelerine giderek sisteme entegre olmuşlardır. Amerika’da Silikon vadisinde teknolojik firmaların birçoğu okullarda olan programlama eğitimini desteklemişlerdir. Code.org platformu kurularak ülke genelinde kullanılmaya başlanmıştır. Ülke genelinde kodlama saati adı altında etkinlikler düzenlenerek programlama eğitimi desteklenmektedir. Code.org ABD ‘de 495000 yeni öğretmeni A-12 seviyesinde programlama eğitimine hazırlamıştır. Programlama eğitiminde Scratch, Arduino, Lego Mindstorm, Swift Playgrounf ve Web Desgin for Kids kullanılmıştır(Şimşek, 2018). Son yıllarda ülke A-12 seviyesinde okullarda kodlama eğitimine daha fazla önem vermektedir. ABD hükümetinin 2017 yılında destek amaçlı FETEMM(Fen Bilimleri,Teknololi,Matematik,Mühendislik) için 10 milyon dolar harcamıştı (Şimşek, 2018).

Estonya’da öğrencileri teknoloji konusunda geliştirmek ve dijital okur-yazar yapmak için Proge-Tiger adlı bir çalışma düzenlenmiştir. (Şimşek, 2018) Bu çalışmada öğrenciler kodlama ve programlama materyallerini kullanmayı, cihazları kodlamayı ve teknolojinin okullarda kullanılmasını hedeflemektedir. Estonya ‘da eğitimin bütün kademelerinde bilgisayar bilimi eğitimi desteklenmektedir. (Şimşek, 2018)

İngiltere’de programlama eğitimine ilkökul seviyesinde başlanmış olup ilkökul-ortaokul seviyesi zorunlu lise seviyesinde ise seçmeli ders olarak verilmektedir. Ülkede programlama eğitimi ayrı bir ders değil de diğer derslerle iş birliği içerisinde verilmektedir. Örneğin Fen bilgisi dersinde Raspberry Pi kullanılıp devre yapılırken, Edebiyat dersinde hikâye yazmak için Scratch uygulamasından faydalanılabiliyor. (Allsop ve Sedman, 2005) 2014 yılında 5-6,7-11 ve 11-14 yaş grubu için 3 basamak

olmak üzere eğitime başlamıştır. Birinci basamakta 5-6 yaşlarındaki öğrencilere algoritmanın ne olduğu öğretilmektedir. Bu eğitim daha çok görsellerle ilerlemektedir. İkinci basamakta, 7-11 yaş grubuna karmaşık problemleri belirli amaca göre çözmeyi ve hata ayıklamayı öğretmektedirler. Öğrencileri mantıksal düşünmeye yönlendirip temel bilgisayar eğitimi verilmektedir. Üçüncü basamak olan 11-14 yaş arasındaki eğitimde ise öğrencilerin en az 2 tane programlama diline hâkim olmaları hedeflenmektedir. Okul ve öğrencilerin kendi yeteneklerine göre programlama dili seçip devam etmektedirler. Aynı zamanda programlamayla beraber mantıksal ve matematiksel yeterlilikler de verilmektedir. Tüm basamaklarda öğrencilerin temel bilgisayar ve güvenli internet konusunda da bilinçlendirilmesi hedeflenmektedir. Ayrıca öğretmen eğitimleri içinde devletin ciddi desteği söz konusudur (Öndeş, 2016).

Avusturalya’da 14 Ekim 2014 tarihinde Endüstri İnovasyonu ve Rekabet Gündem’ini başlatarak ülkenin rekabet gücünü artırmayı planlamıştır. (Şimşek, 2018) Avusturalya hükümeti Avusturalya Öğretim Programını Değerlendirme ve Raporlama (ACARA) kuruluyla ortaklaşa çalışmaktadır. Avusturalya’da öğrenciler 3 yaşında kodlama eğitimine başlamaktadır. MIT (Massachusetts Teknoloji Enstitüsü) tarafından geliştirilen blok tabanlı kodlama (Scratch) uygulamalarını göstermeye başlamışlardır. 3-6 yaşındaki öğrenciler teknolojinin hızla gelişmesiyle beraber daha karmaşık ve daha ayrıntılı problemleri çözmek için eğitilip yönlendirilmektedir. Avusturalya’da 7. sınıf öğrencileri metin tabanlı (Python, Javascript) programlama dilleri ile programlamanın farklı bir boyutunu öğrenmeye başlamaktadır. 7-10 yaşında kodlama eğitimi verilmiş çocuklardan daha soyut düşünceleri ve daha ayrıntılı, karmaşık problem çözmeleri beklenmektedir. 5-8 yaşında olan öğrencilere sıklıkla görsel programlama dillerinden olan Scratch gösterilmektedir. Avusturya’da yaşayan öğrencilerden 10 yaşında temel kodlama becerisini kazanmaları ve 12 yaşından sonra belirli bir seviyede bilgisayar programlama yapması beklenmektedir. 14-16 yaşında ise modüler programlama kullanarak kodlar yazmaları ve seçilmiş algoritmaları kullanmaları beklenmektedir. Öğrencilere daha çok ilkökul seviyesinde blok tabanlı ve pratik çözümlü, algoritmik düşünme becerisi kazandıracak görsel programlama dilleri gösterilirken ortaokul ve lise seviyesinde genel programlama dillerine geçilerek daha karmaşık problemler bilgisayar bilimi etkinlikleri ile çözmeleri beklenmektedir (Şimşek, 2018).

Kanada’da Ağustos 2017’den itibaren 6. sınıflarda kodlama öğretimini zorunlu hale gelmiştir. (Şimşek, 2018) Kanada Millî Eğitim Bakanlığı öğrencilere okul öncesinden lise düzeyine kadar kodlama ve programlama eğitimini planlamıştır. 1, 2 ve 3.sınıflarda programlama ve sıralamayı geliştirecek olan robotik kitlerden 4, 5 ve 6. sınıflarda ise yaratıcılığı geliştirecek robotik kitlerden faydalanılacağını belirtmiştir. 7-12 sınıfları arasında ise mühendislik ve programlama konularını içeren müfredatta, ders öğrencilere seçmeli olarak sunulmaktadır (Julia, 2017). Kanada’da ders dışında da birçok kodlama ve robotik alanda çalışmalar yapılmaktadır. Ülke çapında Hour of code, Makerspace ve robotik yarışmalar düzenlenerek öğrencilerin dikkatlerinin bu alana çekilmesi hedeflenmektedir. Bu etkinlikler için dijital yeterliliğe sahip ve programlama eğitiminin pedagojisinin nasıl verileceği konusunda uzman yetiştirmek için öğretmen eğitimleri yapmaktadır. Kanada hükümeti programlama uygulamalarını destekleyerek bu alanda istihdam yaratılmasını da hedeflemektedir (Julia, 2017). Buna ek olarak Kanada Eğitim Bakanlığı kodlamanın her alana yayılmasını hedefleyerek sistematik bir yaklaşım oluşturmak amaçlı ülkede Coding Quest adlı bir uygulama başlatmıştır. Bu uygulamada öğretmenlerin de kendilerini geliştirerek, çeşitli materyal ve öğretim yöntemi kurallı olmaksızın öğrencilerle etkileşime girerek onların yenilikçi yolları bulmasına yardımcı olmaktadır.

Finlandiya’da Eğitim Bakanlığı 2016 yılında Bilgi ve İletişim Teknolojileri dersini müfredata eklemiştir. Finlandiya’da Bilgi ve İletişim Teknolojileri dersi ilk olarak 1980 yılında müfredata girmiş ama 1982 yılında liselerde kullanılmaya başlanmıştır. Bilgi ve İletişim Teknolojileri dersini çok önemsediklerinden ders dışı etkinliklerde programlama dillerini öğretecek etkinlikler geliştirmişlerdir (Saarikoski, 2011). Finlandiya’nın eğitim-öğretim programları 2016 yılında revize edilmiş olup 21.Yüzyıl becerilerinin de arasında yer aldığı müfredatta bütün alanlar teknoloji ile entegrasyonlu bir şekilde çalışmaya odaklanmışlardır. Finlandiya’da programlamanın temel alındığı tüm dersler sınıfların kendi öğretmenleri tarafından verilmeye başlanmıştır (Şimşek, 2018). Bundan dolayı her öğretmen basit düzeyde programlama bilgisi öğrenerek sanal ortamda küçük programlar tasarlamaktadır hatta öğretmenler ve araştırmacılar tarafından Koodiaapinen (coding ABC) isimli ortak bir çevrimiçi platform oluşturulmuştur. Bu ortak platform öğretmenlere öğretim yöntem ve teknikleri, materyal ve programlama desteği

vermektedir. Finlandiya’da innokast adlı bir ortak ağ bulunmaktadır. Bu ortak ağ öğretmenlerin, öğrencilerin, vatandaşların kullanımına açılarak bu kişilere programlamayı öğretmektedir (Korhonen, Lavonen, Kukkonen, Sormunen, ve Juuti, 2014). Finlandiya’da 2016 yılından itibaren uygulanan öğretim programında programlama ve robotik alanı da yer almaktadır. 6 yaşından 16 yaşına kadar olan robotik derslerdeki temel amaç öğrencilerin girişimci özelliklerini ve eleştirisel düşünme becerilerini geliştirmektir. Sürekli olarak oyun ve robotik platformlarda yeni etkinlikler hazırlanmaktadır (Şimşek, 2018).

İsveç’te programlama eğitiminin amacı öğrencilerin yazılımcı olarak yetiştirmek değil öğrencilerdeki yaratıcı düşünmeyi, problem çözüme becerilerini geliştirmektir. İsveç’te öğrencilerin ilkokuldan itibaren mantıksal düşünme, test etme, problem çözme, analiz yapma ve sentez yapma gibi becerileri geliştirmek hedeflenmektedir. Okullarda önemsenen kodlama eğitimlerini, kodlama ortamlarını aynı zamanda dışarıda da özel olarak da bulmak mümkündür. İsveç’te fritid adı verilen kelime anlamı “boş zaman” olan aktiviteler bulunmaktadır. Coder-Dojo 7-12 yaşındaki öğrencilerin katıldığı bir fritiddir. Bu 2012 yılında İsveç’te çocuklara programlama eğitimi vermek için yapılan bir etkinliktir (Şimşek, 2018).

İspanya’da 2015-2016 eğitim-öğretim yılında çocuklar için kodlama ve robotik müfredata konulmuştur. İspanya’da 130.000 1. ve 3. sınıf öğrencisi kodlama ve robotik uygulamalar içeren dersler almıştır. İspanya hükümeti okullara robotik kitler ve 3D yazıcılar gibi malzemeler göndererek öğrencilerin bu alanda gelişmelerini desteklemiştir. Okullara da öğrencilerin cep telefonu uygulamaları geliştirerek 3D tasarım yapma, oyun geliştirme gibi becerileri kazanmalarını hedeflemiştir (Şimşek, 2018).

Estonya’da 2012 yılından itibaren 7 yaşından başlayarak öğrencilere programlama eğitimi verilemeye başlanmıştır. ProjeTiiger adı altında öğrencilere tablet dağıtılarak, kodlama ve programlama alanlarında ilerlemeleri desteklenmiştir. Bu proje

kapsamında ülke genelindeki ilkokul öğretmenlerine ve öğrencilere uzman kişiler tarafından programlama eğitimi verilmiştir (Şimşek, 2018).

Güney Kore’de eğitimi planlayan bakanlıklar ilkokuldan itibaren yazılım derslerinin zorunlu olacağını 2017 yılında ilkokullarda sonra liselerde kademeli olarak programlama eğitimine geçiş yapılacağını söylemiştir. (Şimşek, 2018) Şubat 2016 yılında Güney Kore Eğitim Bakanlığı 3-5 yaşındaki çocuklara İngilizce, 6 yaşından sonra ise programlama öğrenmeleri ve bu alanda kendilerini geliştirmeleri için belirli bir miktar para vererek destek olmuştur.

Hindistan’da öğrencilere ilkokul seviyesinde bilgisayar hakkında temel bilgiler, bilgi işlemsel düşünme ve basit döngülerden bahsedilmektedir. Ortaokul seviyesinde ise yine bilgisayar kontrolü, etkili bilgisayar kullanımı konularında eğitim verilmektedir. Lise düzeyinde sistem kurma, ağ kurma ve müfredatta verilen belirli başlı programlama dilleri hakkında eğitim verilmektedir (Şimşek, 2018).

Ülkemizde kodlama eğitimi ise 2016 yılında bilgisayar bilimi ve bilişim teknolojileri dersi adı altında bilgi işlemsel düşünme, dijital okuryazarlık gibi bölümleri dikkate alarak ders içerikleri yeniden güncellenmiştir.

Türkiye’de programlama eğitimine 1998 yılında ilk önce ilköğretim okullarında bilgisayar dersi adı altında seviyesi 1-5. Sınıfları kapsayacak şekilde başlamıştır. Daha sonra 2006 yılında seçmeli bilgisayar adı altında, seviyesi ise 1-8. Sınıfları kapsayacak şekilde güncellenmiştir. 2018 yılında yılın da ilkokul bilgi ve iletişim teknolojileri dersi olarak 1-4 seviyesinde güncellenmiş olup kazanımlar doğrultusunda programlama eğitimleri verilmektedir.

Ortaokul ve imam hatip ortaokullarında ise 2016 yılında bilgi ve iletişim teknolojileri dersi adı altında, seviyesi 5-6. Sınıfları kapsayacak şekilde başlayıp 2018 yılında ise bilgi ve iletişim teknolojileri dersi adı altında seviye 7-8. Sınıfları kapsayacak şekilde güncellenerek kazanımlar doğrultusunda programlama eğitimleri verilmektedir.

Lise seviyesinde ise 2015 yılında Bilgi ve iletişim teknolojileri adı altında, 2016 yılında ise bilgisayar bilimi adı altında kazanımlar hazırlanarak programlama eğitimleri verilmektedir.

2.5 21. YÜZYIL BECERİLERİ

İçerisinde bulunduğumuz yüzyılda teknolojinin gelişmesi bireylerin gelişim beklentilerini ve hedeflerini her geçen gün değiştirmektedir. Eğitimde kalitenin artması için, öğrencilerin günlük hayatta karşılaştıkları problemlere çözebilmeleri ya da çözüm önerisi getirecek becerilere sahip olması planlanmaktadır (Şahin, Adıgüzel, ve Ayar, 2014). Gelişen dünyada 21.Yüzyıl içerisinde teknolojiye devrimler yapılmaktadır. Teknolojinin gelişimi yeni gelişmeleri tetiklemekte, herkes teknolojik gelişmelere göre yerini belirlemektedir. Teknolojik gelişmelere ayak uydurması gereken kurumlardan bir tanesi de okullardır. Bu kurumlardaki işleyişler de teknolojik gelişmelere göre yön değiştirmektedir. Okullarda, eğitim kurumlarında verilen eğitimlerde kullanılan teknolojik cihazlarla öğrencilerin günlük hayatta karşılaşacakları problemleri çözmelerine yardımcı olacak beceriler kazanmaları, 21. Yüzyıl becerilerini kazanmaları hedeflenmektedir (Coşar, 2013). Bu bilgilere dayanarak 21.Yüzyıl becerilerini yaratıcı düşünme, problem çözme, yansıtıcı düşünme, eleştirel düşünme, üstbilişsel farkındalık, işbirlikli çalışabilme, bilgi işlemsel düşünme, algoritmik düşünme, medya okur-yazarlığı, analiz edebilme gibi başlıklara ayırabiliriz (Pakman, 2018).

Teknolojinin hızla gelişmesiyle beraber yeni olayların, durumların yeniden değerlendirilmesi ve yeni tanımların ortaya çıkması gereksinimi oluşmuştur. Yeni

tanımlardan biri de insanların teknolojiye uyum sağlaması ve uyum sağlayacak karaktere sahip olma becerileridir (Gününç, Odabaşı, ve Kuzu, 2013). İnsanların teknolojiye uyum sağlamasını daha basit hale getiren yaratıcılık, yenilenme, problem çözme gibi hayat boyu devam eden beceriler mevcuttur. Öğrenme üzerinde rol oynayan bilgi okuryazarlığı, teknoloji okuryazarlığı gibi bilgiye ulaşma becerisi kazandıran, proje üreten, sorumluluk sahibi olmasını sağlayan ve kendini yöneten bireylerin becerileri 21.Yüzyıl becerileri adı altında toplanmıştır (Anagün, Atalay, Kılıç, ve Yaşar, 2016).

Bilgi ve teknoloji çağına uyum sağlamayı kolaylaştıran beceriler her ne kadar 21.yüzyıl becerileri adı altında toplansa da herkesin hem fikir olduğu bir tanım ortaya çıkmamıştır. Hemen hemen hepsi Amerika ve Kanada’da olan birçok kurum ve kuruluş tarafından bu beceriler temel yeterlilikler, derin öğrenme, gerekli beceriler (problem çözme, algoritmik düşünme, analiz, sentez vb.), hayatta kalma becerileri, istihdam becerileri gibi terimler adı altında toplanmıştır. (Ekici, Abide, Canbolat, ve Öztürk, 2017). Ekici, Abide, Canbolat, ve Öztürk, (2017) yaptıkları araştırmada 19 kaynağın incelenmesiyle 63 beceriye ulaşılmıştır. Becerilerden en fazla kullanılanları, yaratıcılık, iletişim, problem çözme, iş birliği, eleştirel düşünme, karar verme, bilgi okuryazarlığı, medya okuryazarlığı, teknoloji okuryazarlığı, üretkenlik, sorumluluk, esneklik ve uyum, liderlik, girişimcilik adı altında toplanmıştır.

2.6 KAVRAM YANILGILARI

Öğrencilerin bilimsel olarak temel alınan kavramlardan farklı olarak öğrendikleri kavramlara kavram yanılgısı denir. (Çetin & Berigel, 2017) Öğrencilerin daha önceden öğrendikleri kavram yanılgıları, yeni öğrenmeleri zorlaştırabilir (Hewson ve Hewson, 1988).Kavram yanılgıları üzerine yapılan çalışmalar sonucunda kavram yanılgılarının birçok ortak özelliğinin olduğu görülmektedir. (Hewson ve Hewson, 1988; Taber, 2001)

- Kavram yanılgıları alan yazında olan ve kabul edilen kavramlarla çelişir.

- Kavram yanlışları bilim adamları tarafından kabul edilen kavramlardan kesinliği daha az, daha az tutarlı ve daha çok bağımlıdır.
- Her öğrenci için kavram yanlışları tutarlılığı farklıdır.
- Kavram yanlışları klasik öğretim yöntemlerine karşı her zaman bir zorlu gösterir.

Kavram yanlışları aşırı veya yetersiz genelleme, hatalı bilgi işlemsel model ve örtük kavram yanlışları olmak üzere üç kısımdan oluşur (Clany, 2004). Bu kavram yanlışları çeşitlerinden birincisi aşırı veya yetersiz genelleme bölümünde öğrencilerin kavramı başka bir alanda kullanmak üzere kavramın transfer sürecinde ya aşırı ya da yetersiz bir genelleme yaptığı görülür. (Çetin & Berigel, 2017) Başlıklar şöyledir;

2.6.1 Dil

Öğrencilerin günlük hayatta kullandıkları dili, programlamaya transfer etmeleridir. Örneğin if (eğer) ifadesinde koşulun sürekli test edilerek koşul gerçekleşmesi durumunda aradaki komutların sürekli çalışacağını düşünmesidir. (Çetin & Berigel, 2017)

2.6.2 Matematiksel gösterim

Öğrencilerin matematikte kullanılan operatörlerin programlama dillerinde de aynı işlemi yaptıklarını düşünmeleridir. Örneğin eşittir ifadesi programlama dillerinde atama olarak kullanılırken matematikte sonuç gösteren ifade olarak kullanılmaktadır. (Çetin & Berigel, 2017)

2.6.3 Eski programlama deneyimi

Öğrencilerin daha önce öğrendikleri programlama dillerindeki kavramları, yeni öğrenecekleri programlama diline transfer ederek çalıştırmaları olarak söylenebilir. Örnek olarak farklı programla dillerinde “==” ifadesi eşit olup olmadığını kontrol ederken C dilinde aynı işlemi metin işlemleri için de beklemeleri gösterilebilir. (Çetin & Berigel, 2017)

2.6.4 Örneklerden aşırı genelleme

Öğrencilerin farklı örnekleri genelleyerek elde ettikleri kavram yanılığıdır. Örnek olarak nokta işaretinin programlama dilinde fonksiyon çağırdığını bilen birinin sadece sınıf çağırmak için kullanılacağını düşünmesi verilebilir. (Çetin & Berigel, 2017)

2.6.5 Kuralların değiştirilmesi

Öğrenciler programlama dillerinin kurallarını yanlış anlayıp kendilerine göre kural oluşturmaları örnek olarak verilebilir. (Çetin & Berigel, 2017)

2.6.6 Analoji

Farklı iki alan üzerinde bütünsel olarak yapılan eşleştirmelerdir. Örneğin öğrenciye değişken kavramı bir kutuya benzer dediğimizde, öğrenciler kutu içerisinde birden fazla nesne taşınabilir diye düşünüp değişkenlerinde birden fazla değer aldığını düşünebilirler. (Çetin & Berigel, 2017)

İkinci tip kavram yanılığı olan bilgi işlemsel model bilgisayar donanımının zihinsel modellemesi ile ilgilidir. Öğrencilerin programlamada veya bilgisayar

bilimlerinde soyut bilgi işlemsel modeli anlamayıp farklı modeller elde etmesi olarak söylenebilir. Örneğin bilgisayar bilimini bilen bir insanın bilgisayar donanımındaki belleğin şişmesinden bahsetmesi, öğrencinin bunu anlaması için söylediği bir tanımdır. Ama öğrencilerin bunları anlaması zor olabilir.

Sonuncu kavram olan örtük kavram ise öğrencilerin kodlama yaparken sistematik bir şekilde hata yapmalarına denir. Clany (2004) öğrencilerine Pascal programlama dili öğretirken zorlanmaları diye öncesinde Karel göstermiştir. Karel’de değişkenler olmadığı için öğrencilerin Pascal dilinde değişkenleri anlamakta zorluk çektikleri gözlenmiştir.

Ma, Ferguson, Roper, Ross ve Wood, (2009) araştırmasında öğrencilerin kavram yanlışlarını bulup onlardan kurtulmalarını sağlamak için görselleştirme kullanarak dört adımlı bir model geliştirmiştir. Bu maddeler başlangıç, bilişsel çelişki, model oluşturma ve uygulama aşamalarıdır. Başlangıç aşamasında öğrencinin kavram yanlışlığı ortaya çıkarılarak saptanır. Bilişsel çelişki aşamasında öğrencinin zihnindeki kavramın değişmesine inanması için çelişki bulunup onun tersine tartışmalar yapıp öğrencinin zihnindeki kavramın değişmesi sağlanır. Model oluşturma bölümünde öğrencinin doğru kavramı öğrenmesi için görselleştirme kullanılır. Uygulama aşamasında da öğrenci belirlediği modeli kod yazarak uygulamaya başlar. Ma ve diğerleri (2009) bu modelin basit problemler için etkili olduğunu daha karmaşık problemlerde işlemeyeceğini söylemiştir.

Eylon ve Linn, (1988) öğrencilerdeki bilişsel çelişkinin ortaya çıkmasının kavram yanlışlarını çözmede başarılı olamayacağını savunmuştur. Neden olarak öğrencilerin kendi görüşlerini savunarak bu çelişkileri görmezden gelecek olmalarıdır. Clany, (2004) bilgilerin sosyal etkileşim olan ortamlarda bütünleşmesinin kavram yanlışlarının ortadan kaldırılmasında etkili olduğunu savunmuştur. Hazzan, Lapidot, ve Ragonis, (2011) ise kavram yanlışlarının çözülmesi için öğretmen-öğrenci arasındaki etkileşimin sınıf ortamlarında kullanılmasını savunmuştur.

2.7 KODLAMA EĞİTİMİNDE ARAÇ TABANLI ÖĞRETİM YAKLAŞIMLARI

Bilgisayarsız Bilgisayar Bilimi: Bilgisayar ve Programlama ile ilgili temel becerilerin, eğitim verildiği ortamda bilgisayar, tablet vb. gibi araçlar olmadan, etkinlikler vasıtasıyla verilmesidir. (Kert, 2018)

Blok Tabanlı Kodlama: Blok tabanlı görsel programlama, program bloklarının bir Lego parçası ya da puzzle gibi mantıksal bir şekilde birleştirilerek yazılım ve uygulamaların geliştirildiği bir programlama yaklaşımıdır. (Haladjian, Bredies, ve Brügge, 2016)

Metin Tabanlı Programlama: Günümüzde insanlar tarafından çok yaygın olarak kullanılan programlama ortamlarının büyük bölümünü metin tabanlı programlama dilleri oluşturmaktadır. Küçük yaşlardan itibaren programlama eğitimlerinde genel olarak blok tabanlı programlama dilleri kullanılmaktadır. Bunun sebebi, blok tabanlı programlama dillerinde söz dizimi hataları vb. hatalar bulunmamaktadır. Programlama dillerinin temellerini öğretmekte etkili bir şekilde kullanılan ortamlar, blok tabanlı programlama dilleridir ama dünya üzerindeki gerçek problemleri çözmede kullanılan ortamlar, metin tabanlı programlama dilleridir. Dünya üzerinde hala etkili bir şekilde kullanılmaktadır. (Kandemir, 2017)

Robotik Kodlama: Genel olarak fiziki ortamdaki bir nesne ile iletişim kurmak amacıyla o nesneye ulaşarak kontrol altına almak için yazılan kodlara robotik kodlama adı verilir. (Kert, 2018)

2.7.1 Bilgisayarsız bilgisayar bilimi

Yapılandırmacı yaklaşımla beraber öğrencilere bilgisayar olmayan ortamlarda çeşitli etkinlikler yoluyla bilgisayar bilimi etkinliklerinin yaptırılmasıdır (Bell, Curzon, Cutts, Dagiene ve Haberman, 2011). Bilgisayarsız bilgisayar bilimi etkinliklerin amaçları arasında sadece bilgisayar bilimi değil matematik becerilerinin de olduğu söylenmiştir (Taub, Armoni, ve Ben-Ari M. , 2012). Öğrencilere bilgisayarların olmadığı ortamlarda sadece bilgisayarsız kodlama eğitimlerinin yaptırılması yerine, bilgisayar başında programlamaya başlamadan önce başlangıç için bilgisayarsız programlama eğitimi de verilebilir.

Curzon, McOwan, Donohue, Wright, & Marsh W., (2018) bilgisayarsız programlama etkinliklerini

- Kinestetik Ekinlikler
- Rol Oynama Çalışmaları
- Yap-Boz Uygulamaları
- Sanatsal Etkinlikler
- Oyunlar
- Gizem katılan etkinlikler

Olarak sınıflandırmıştır. Bu etkinlik sınıflarından en çok öğrencilerin yoğun olarak katıldıkları etkinlik çeşitleri Kinestetik ve Rol oynama etkinlikleridir.

“Bilgisayar biliminde kullanılan yaklaşımlardan diğeri de bilgisayarsız bilgisayar bilimi(B³) öğretimidir” (Kalelioğlu,2017,s.184) B³ öğretim etkinliklerinde kullanılan çeşitli yöntemler; kartlar, bulmacalar, boya kalemleri, boyama etkinlikleri ,eğlenceli olacak birçok oyun etkinlikleri bilgisayar bilimi ve bilgi işlemsel düşünme kavramlarını ücretsiz olarak öğrenmemizi sağlayan kaynaklar bütünüdür. (Kalelioğlu, 2017) B³ etkinlikleri sadece programlama bilgisi öğretimi için değil aynı zaman da yapay zeka, insan bilgisayar etkileşimi, bilgisayar grafikleri daha geniş alanları kapsayan konuları da

içermektedir. (Bell, 2014) Ülkelerin birçoğunda kullanılan B³ etkinliklerin temel amaçları arasında problem çözme, grup çalışması ve yaratıcılık kavramları öne çıkmaktadır. (Cortina, 2015)

B³ etkinliklerine metin sıkıştırma örneğini verebiliriz. Bu etkinlikte öğrencilere tekrar eden harfleri bularak sıkıştırma işlemi yapabiliyorlar. Bu etkinlik sayesinde bilgisayardaki verilerin nasıl saklandığını fiziksel olarak da bilgi sahibi oluyorlar. Ayrıca öğrenciler etkinlik sırasında grup çalışması yapıp diğer gruplarla da zaman zaman yarış içerisine girebiliyorlar. (Kalelioğlu, 2017)

Dünya genelinde kullanılan başlıca B³ etkinliklerine örnek verecek olursak; Bilge Kunduz, Keşf@-Kodlamayı Keşfediyorum, Eğlence için Bilgisayar Bilimi (CS4FN), Bilgisayarsız code.org (Code.org unplugged), Bilgisayarsız Bilgisayar Bilimi (CS Unplugged) olarak gösterilebilir (Kalelioğlu, 2017). Etkinliklerin ortaya çıkmasındaki en büyük neden bilgisayar bilimine dikkat çekmek olarak görülmektedir. Bu etkinliklerin diğer özelliği ise bilgisayar alt yapısı olmayan merkezlerin bilgisayar bilimini öğrenmeleri sağlamaktır. (Kalelioğlu,2017) Alan yazı tarandığında yapılan B³ etkinlikleri (Nishida, Kanemune,Idosaka, Namiki,Bell ve Kuno2008;Mano, Allan, ve Cooley, 2010;Taub, Ben-Ari, ve Armoni,2009;Thies ve Vahrenhold, 2013) ile öğrencilerde hayal gücünün arttığı, yaratıcılık becerisinin geliştiği, düşünme becerilerinin geliştiği, programlamaya karşı olan tutumlarının olumlu yönde arttığı, bilgisayar bilimine olan motivasyonun , ilginin arttığı, bilgisayar ile ilgili kavramlara yatkınlığın arttığı, matematiğe olan ilginin arttığı görülmektedir.

2.7.2 Blok temelli programlama

Metin tabanlı programlama dillerinde yazım kuralları özellikle kodlama eğitimine yeni başlayanlar için ciddi bir bilişsel yük oluşturabilir (Kert, 2018). Bu sorunlar Blok tabanlı programlama dilleriyle çözümlenebilir (Kurihara, Sasaki , Wakita, ve Hosobe,

2015). Metin tabanlı programlama dillerinde yazım yanlışları, boşluklar, küçük-büyük harf duyarlılığı gibi konularda öğrencilerin yaşadığı zorluklar yeni başlayanlar açısından motivasyon azalmasına sebep olabilir. Bu gibi sorunları ortadan kaldırmak için hazır programlama blokları kullanılabilir. Özellikle kodlama eğitimine yeni başlayanlar metin tabanlı yazılımlarda bulunan değişkenler, döngüler, mantıksal ifadeler gibi programlama konularını anlayamamaktadırlar (Grover ve Basu 2017). Kodlama bloklarının kullanılması öğrencilere birçok avantaj sağlayabilir fakat bazı kısıtlamalardan dolayı öğrencilerin yaratıcı düşünme becerilerini olumsuz yönde etkileyebilir. Profesyonel yazılım programlarına yakın işlemler yapan blok tabanlı programlama dilleri çok daha fazla tercih edilmektedir (Kert, 2018).

Öğrencilerin teknolojinin geliştiği dünyada dijital okur-yazar olmaları için teknolojiyi tüketen değil üreten nesiller olmaları için erken yaşta bilgisayar bilimi eğitimi almaları gerekir (Wilson ve Guzdial, 2010). Çocukların erken yaşta bilgisayar bilimi öğrenmesi öğrencilerin bilgi-işlem öz yeterliliği ve ilgisini arttırmaktadır. (Akbulut ve Looney, 2009)

Blok tabanlı görsel programlama ortamlarında öğrenciler kullandıkları bloklar sayesinde ezberden öteye giderek programlama mantığını kavradıklarını, (İmal ve Eser, 2009; Bryne ve Lyons, 2001) söz diziminde yaşanan problemlerin ortadan kalktığını, (Kinnunen ve Malmi, 2008; Lahtinen, Ala-Mutka, ve Jarvinen, 2005), algoritma mantığını oluştururken zorluk yaşamadığı düşünülmektedir.

Blok tabanlı görsel programlama ortamlarının iki amaca hizmet etmektedir. Bunlardan birincisi programlama yapısını basitleştirerek, en temel seviye programlama becerilerine sahip insanların bile programlama yapmasını sağlamak. İkinci hizmet ise insanların ilgisini çekerek daha çok insanın programlamayla ilgilenmesini sağlamak. (Durak, Karaoğlu Yılmaz, Yılmaz ve Seferoğlu, 2017; Yükseltürk ve Altıok, 2016)

Mohamad,Patel, Latih, ve Qassim,(2011) blok tabanlı görsel programlama yaklaşımının avantajlarını maddeler halinde sıralamıştır:

- Uygulamalar tekrar kullanılabilir.
- Geliştirilen uygulama maliyetlerinde belirli bir azalma meydana gelir.
- Temel düzeyde programlama bilgisi yeterlidir.
- Geliştirilen uygulamalarda zaman konusunda tasarruf etmemizi sağlar.
- Uygulama geliştirirken API eklemek için onay gerektirmez.
- Kullanıcıların programlama konusundaki ihtiyaçlarına cevap vermektedir.

Blok tabanlı görsel programlama ortamları öğrencilerin programlamam alt yapısı olmadan programlamayı öğrenmelerini sağlamaktadır. Blok tabanlı görsel programlama ortamları, programlamanın öğrenmekte zorluk çekilen yapısını ve öğrenirsek karşılaşılan sorunların aza indirmesini sağlamaktadır. (Yükseltürk ve Altıok, 2015)

“Alanyazın incelendiğinde blok temelli görsel programlamanın temel özellikleri kolay ara yüz, blok kod yapısı, hata ayıklama yapısı, çoklu ortam desteği, tasarım odaklı yapı ve çevrimiçi paylaşım özellikleri olarak sıralanmaktadır.” (Erol, 2015; Maloney ve diğerleri 2010; Resnick Maloney, Monroy-Hernandez, Rusk , Eastmont, Brennan ,Kafai, 2009; Yükseltürk ve Altıok, 2017).

Maloney, Peppler, Kafai, ve Resnick, (2008) araştırmalarında Scratch, acemi programcıların çalışma yaptığı kolay medya düzenleyici, blok tabanlı görsel programlama dili olarak tanımlanmıştır. Araştırmalarında 18 ay boyunca 8-18 yaşın olan gençlerin dersten sonra bir kulüp merkezinde, herhangi bir profesyonel yardım almadan ve tecrübeli bir danışman olmadan scratch ile yaptıkları 536 etkinlik analiz edilmiştir. Projedeki amaç, katılımcıların temel programlama kavramlarını anlamak olarak belirlenmiştir. Araştırma süresince katılımcıların programlamanın temel kavramlarını kullanmaları gözlemlenip kulüp üyelerinin scratch blok temelli görsel programlama ortamları hakkındaki fikirleri ve algıları gözlemlenmiştir. Bu araştırmada katılımcıların herhangi bir formal eğitim planına bağlı kalmaksızın tamamen informal ortamda

profesyonel olmayan diğer birçok yazılım varken neden programlamayı seçtikleri araştırılmıştır. Katılımcılara “Bilgisayar programlaması senin için ne ifade ediyor?” sorusu sorulduğunda “Bilgisayar programlaması da nedir?” yanıtını almışlardır. Araştırmacılar ilk olarak katılımcıların scratch ile programlama arasında bir bağ kuramadıkları, scratch’yi bir oyun tasarlama bir medya düzenleme programı olarak görmüş olmaları, öğrencilerinde bu yüzden scratch ile çalışma yapmak istedikleri konusu dikkat çekmiştir. Değişkenler ve rasgele sayı üreten kod bloklarının yapılan etkinlikler içerisinde kullanımının çok az olduğu ama zamanla bu bloklarında etkinliklere dahil edildiği gözlemlenmiştir. Bunun sebebi de katılımcıların daha önce programlama konusunda bir eğitim almamış olmaları ve onlara danışmanlık yapan kişinin programlama konusunda bir deneyiminin olmaması olarak görülmüştür. Temel programlama konusundaki kavramları, deneyimli bir danışman eşliğinde öğrenmenin daha kolay olduğu araştırmada dikkat çeken diğer konulardan bir tanesidir.

Yükseltürk ve Altıok, (2016) yaptıkları araştırma da 2013-2014 yılında Kırıkkale üniversitesi Bilgisayar ve Öğretim Teknolojileri Öğretmenliği bölümünde eğitim gören 169 bilişim teknolojileri öğretmen adayının bir seçmeli ders kapsamında herhangi bir kısıtlama olmadan bir oyun konusu seçip bu seçtikleri oyunu “Scratch” programında tasarlamaları istenmiştir. Araştırma sonuçlarında oyunun konusu, hedef kitlesi, oyunda olması gereken kriterlere göre değerlendirildiğinde oyunların kalitesinin “orta” ve “iyi” olduğu görülmüştür. Araştırmada Scratch programının seçilmesinin nedeni araştırmaya katılan bilişim teknolojileri öğretmen adaylarının güncellenen programlamam eğitimindeki yöntemler hakkında bilgili olmaları ilk neden olarak gösterilmiştir. Başka bir neden olarak programlamada bir oyun tasarlamamanın kolay olmadığını, zor ve karmaşık olduğunu, scratch ile oyun tasarlamak ve oyun programlamamanın başlangıç seviyesi olmasına rağmen gayet kolay, eğlenceli, ilgi çekici, heyecan verici olması olarak gösterilmiştir. Araştırma sonucunda araştırmaya katılan öğretmen adayları kendilerini değerlendirdiklerinde kendilerini başarılı olarak görüp sebep olarak da kendilerine ait bir ürün ortaya koymanın çok güzel bir duygu olduğunu söylemişlerdir.

Blok tabanlı görsel programlama ortamlarına örnek olarak, Scratch, Kodu Game Lab, Code.Org, Alice, App Inventor, Lego Mindstrom Education, Lego WeDo verilebilir.

2.7.3 Metin tabanlı programlama dilleri

Programlamaya yeni başlayan öğrenciler, blok temelli programlama dillerini kullanmaktadırlar ancak blok temelli programlama dillerinin geleneksel yazılım geliştirmelere yanıt vermemesi ve ara katmana ihtiyaç duyması bir dezavantaj olarak görülmektedir (Kim, Choi, Jung, ve Choi, 2018). Metin tabanlı programlama dilleri programlama becerileri bakımından gelişmiş öğrencilerin tercih ettikleri programlama dilleridir. Metin Tabanlı programlama dillerinin özellikleri şöyle olmalıdır (Vujosevic-Janicic ve Tosic, 2008):

- Giriş-Çıkış işlemlerinin kolaylığı
- Okunabilir ve tutarlı öz dizimi
- Küçük ve dikey özellikler kümesi
- Diğer programlama yapılarından açıkça ayrılması

Bu maddeler profesyonel olan metin tabanlı dillerde olması gerekmektedir. Ayrıca programlama dilinin sürekli güncellenmesi de gerekmektedir. Metin tabanlı programlama dillerinde program yazanın ihtiyaçlarına cevap verecek bir dil olmalı, öğrencilerin özelliklerine uygun olmalı ve en önemlisi tercih edilen dilin öğrenci motivasyonun artırıp pedagojik hedeflere uygun olması gerekmektedir.

Erken yaşta programlama öğrenmek, programlamayı öğrenenlerin günlük hayattaki yaşamlarına yardımcı olabilmektedir. Erken yaşta programlama öğrenmek isteyen bireylerde; bilgi işleme, düşünme ve iletişim anlamında fayda sağlamaktadır. Ayrıca erken yaşta öğrenmek 21. yüzyıl becerilerini geliştirme ve destekleme açısından faydalıdır.

Weintrop ve Wilensky (2015), blok tabanlı görsel programlama ortamlarında programlama öğrenen insanların “gerçek programlama” olmaması, ifade gücü, daha ileri seviye ve karmaşık programlar yapma konusunda yetersiz oluşu blok tabanlı görsel programlama ortamlarının eksik yanlarıdır.

Problem çözme yaklaşımları açısından blok tabanlı görsel programlama dillerinde programlama öğrenen insanların problem çözme ve program yazma alışkanlıklarında bilgisayar bilimcilerinin karşı olduğu bazı davranışların ortaya çıktığı saptanmıştır. (Meerbaum-Salant, Armoni ve Ben-Ari, 2011)

İlkokul, ortaokul ve lise seviyelerinde programlama dersleri için programlama ortamları ve programlama dillerinde olması gereken özellikler aşağıdaki gibidir (Ristic, Milosevic, ve Urosevic, 2016).

- Kullanılacak programlama dili ve program yazılacak ortam temel düzeyde programlama kazanımlarını içerecek türde olmalıdır.
- Kullanılacak programlama dili yeni programlama dilleri ile uyumlu olmalı, program yazma becerilerini içermelidir.
- Eğitimde kullanılacak programlama dilleri günümüzde kullanılan yeni programlama dillerine uyumlu olmalıdır. Böylece programlama dilinde öğrendiği kurallarla yeni öğreneceği programlama dilini çok çabuk öğrenebilir.
- Programlama dilinin söz dizimi, daha önce yazılmış olan bir programı okuyup anlayabilecek basitlikte olmalıdır.
- Bellek yönetiminde çıkacak sorunlar, dinamik veri yapıları için önemli olduğundan üzerinde durulmalıdır.

Metin tabanlı programlama dillerine örnek olarak Codemonkey ve CodeCombat çevrimiçi programlama ortamları, PencilCode çevrimiçi Hibrit geliştirme ortamı, Logo Programlama Dili ve Kaplumbağa Grafikleri, Microsoft Small Basic programlama ortamı, Python Programlama Dili, Greenfoot programlama ortamı verilebilir.

2.7.4 Robotik kodlama

Öğrenme süreçlerinde öğrenmenin en iyi şekilde olması için içeriğin somutlaştırılması önemli bir adımdır. Programlama öğretimi için kodların yazılması aracılığıyla fiziksel ortam ile iletişim kurulması fiziksel programlama içerisine girmektedir. Fiziksel programlama genel olarak fiziksel ortamdaki donanımlar ve nesnelerle iletişim, etkileşim kurmak için kodların yazım sürecine/işlemine verilen addır. Programlama eğitiminde Arduino setleri, fiziksel programlama için kullanılabilir. Arduino setleri ile kodların yazıldıktan sonra fiziksel ortamda durumlarına, tepkilerine somut olarak bakılabilir (Kert, 2018).

Fiziksel programlama (Robotik kodlama) yazılım ve donanım kullanarak dışardaki dünya ile bir bağ kurarak veri alışverişi yapmak şeklinde tanımlanabilir. Endüstri 4.0 kavramında siber fiziksel sistemler robotik kodlama (fiziksel programlama) ile yapılmaktadır. Siber sistemlerdeki siber kısmı dışardan veriyi alıp aldığı veriyle üretim sürecine giren bilgisayar yazılımlarıdır. (Thramboulidis, 2015)

Programlama eğitimi alan öğrencilerde farklı düşünce yapılarına ihtiyaç olacağından programlamanın somutlaştırılması gerekir. (Ersoy ve diğerleri 2011) Programlama öğretiminde robotların kullanılması programlama eğitime pozitif yönde katkılar sağlayacaktır. (Papert, 1980)

Robotik kodlamada robotik kitlerle öğrenim gören öğrencilerin öğrenmeyi keşfederek öğrenmesi, inşa etmesi ve ayrıca kitlerle öğrenim sürecinde öğrenirken eğlenmeleri görülmüştür. (Golovinsky, Yim, Zhank, Eldershaw, ve Duff, 2004)

Programlama eğitiminde robotlar birer programlanabilir araç olarak kullanılabilir. Çocukların oyun geliştirirken yaptıkları kodlamayı robot geliştirirken yapmaları ve somut olarak kodlamanın sonuçlarını görmeleri ilgilerini ve motivasyonlarını olumlu yönde

destekleyebilir. Çocuklara genellikle robotik kodlama araçları içerisinde blok temelli kodlama biçimleri uygun görülmektedir. Lego Mindstorm EV3 bunlara örnek olarak verilebilecek blok tabanlı robotik kodlama aracıdır (Maason ve Cooper, 2013).

Fen ve Teknoloji dersinde yaptığı çalışmalarda Mindstorms NXT 2.0 robot setleri ile eğitim gören öğrencilerin akademik başarılarında olumlu yönde bir etki olduğu sonucunu elde etmiştir. (Özdoğru, 2013)

Programlama eğitimindeki en önemli kavramlardan olan problem çözme becerisinin kazandırılması için yapılan robotik kodlama projelerinde öğrencilerin hem problem çözme becerilerini geliştirdiği hem de teknolojik ve bilimsel kavramları öğrenme sürelerinin daha az olduğunu söylemiştir. (Barak ve Zadok, 2009)

Shim, Kwon ve Lee, (2017) fiziksel programlamayı üç bölümde incelemiştir: Öğrencilerin programlama ortamlarına zorlanmadan alışması ve ortamlara içgüdüsel olarak uyum sağlaması gerektiğini, programlama ortamlarının öğrencilerin kodlamaya olan ilgi ve motivasyonlarını olumlu yönde etkileyecek durumda olması, öğrencilerin program yazarken tam kendilerini o sürece dahil etmeleri gerektiğini söylemiştir.

Numanoğlu ve Keser (2017), Programlama öğretiminde robot kullanımı – Mbot örneği adlı çalışmalarında programlama öğretiminde kullanılan temel kavramlar olan döngüler, değişkenler, koşul ifadeleri, diziler, fonksiyonlar, listeler ile ilgili yaptıkları uygulamaları Mbot robot kitine yükleyip, robot kitleri ile programlama yapmanın eğlenceli ve basit olduğunu söylemişlerdir. Öğrencilerin birçok satırda yazdığı kodu mBlock robotik-kodlama platformunda tek satırlık kodla halledileceğini ve öğrencilerin kod yazarken daha hızlı yazdıklarını söylemiştir. Somut kavramları yaptıkları uygulamalar ile anında orada görmeleri öğrencilerin programlama eğitimine katkı sağladığını söylemiştir. Ayrıca Mbot- STEM Educational Robot Kit platformunun öğrenciler için etkili, verimli ve çekici bir öğrenme ortamı oluşturduğunu söylemiştir. Bu tarz robotik kodlama platformlarının eğitime katkılarının olacağını söylemiştir.

Kasalak, (2017) ortaokul öğrencileri üzerinde yaptığı çalışmada robotik-kodlamayla olan çalışmaların öğrenciler tarafından eğlenceli olarak görüldüğü yapılan istatistiki testlerle belirlemiştir. Robotik kodlama etkinliklerinin sıkıcı olduğunu söyleyen öğrenci gruplarının oldukça az olduklarını saptamıştır. Yapılan araştırmada robotik kodlama eğitimlerinin öğrencilerin kişisel gelişimlerine de faydasının olduğunu ve öğrencilerin robotik-kodlama etkinliklerinde soyut kavramları soyutlaştırarak daha etkili öğrenme sağladıklarını saptamıştır.

Sinap, (2017) yaptığı araştırmada, programlamada arduino setlerinin kullanımında öğrencilerin programlamaya karşı daha motive, ilgili ve bakış açılarının olumlu yönde değiştiğini söylemiştir. Programlama eğitimin Arduino robotik kitlerle verilmesinin soyut kavramları somutlaştırarak öğrencilerde daha kalıcı, eğlenceli bir öğrenme oluşturduğunu söylemiştir.

Martin-Ramos, Silva, Lopes, ve Silva, (2016) yaptıkları STEM eğitimleri kapsamında Portekizli lise öğrencilerini arduino etkinliklerinin olduğu bir yaz kampına almışlardır. Bu yaz kampında öğrenciler, Arduino robot kitleriyle birçok proje ve etkinlik yapmalarını sağlamışlardır. Kampın sonunda Arduino robot kitleri ile projeler üreten lise öğrencilerinin aralarındaki iletişim yeteneğinin arttığı ve akran koçluğu sayesinde öğrenmelerinin gerçekleştiği ortamlar olduğu görülmüştür. Daha sonra Arduino robot kitleri ile yapılan etkinlikler Fizik Mühendisliği birinci sınıf müfredatına girmiştir.

Karademir, (2018) yaptığı araştırmada 10 bilişim teknolojileri, 10 müzik öğretmeninin katılımı ile Arduino, Lego Mindstorm NXT ve Lego Mindstorm EV3 robotik kitlerini kullanarak müzik aletleri üretmişlerdir. Yaptıkları müzik aletlerini deneyen öğretmenlere sordukları sorulardan oluşan bir görüşme formu hazırlanmıştır. Robotların tasarımı ve kodlamasını müzik öğretmenleri ve bilişim teknolojileri öğretmenleri beraber yapmıştır. Yaptıkları robotlar bilişim teknolojileri ve yazılım dersi kazanımlarına göre hazırlanıp müzik dersinde kullanılmaya başlanmıştır. Bu robotların hazırlanması, tasarlanması, kodlanması ve bütün aşamalarında görev yapan öğrencilerde;

problem çözme becerisi, matematiksel düşünme becerisi, algoritmik düşünme, hata ayıklama, çok yönlü düşünme gibi birçok becerinin olumlu yönde geliştiğini söylemiştir. Uygulanan müzik dersindeki öğretmenler ise öğrencilerin daha gürültülü ve sesli müzikleri ayırt edebileceklerini, ritim sayesinde öğrendiklerini müziklere eşlik edebileceklerini, duyduğu sesleri ince ve kalınlık oranına göre ayırt edebileceklerini ve müzik dersi kazanımlarını daha iyi kazandıklarını söylemişlerdir. Ayrıca müzik dersinin robotlarla işlenmesinin öğrenciler açısından daha eğlenceli olduğu ve öğrencilerin derse olan ilgisinin arttığını söylemişlerdir.

Fiziksel programlamada çeşitlilik ne kadar artarsa öğrenciler ve nesneler arasındaki etkileşim de o kadar artar ve öğrencilerin motivasyon ve gelişimlerini olumlu yönde etkilenir. (Kert, 2018) Ama aynı zamanda robotik setlerdeki eksiklik, kaynak eksikliği ve fiziksel programlamanın pedagojik ve değerlendirme konularında yetersiz olması ve bunların müfredata entegre edilmesinin zor bir süreç olması fiziksel programlamanın olumsuz görünen kısımlarıdır. (Kert, 2018)

2.8 EŞLİ PROGRAMLAMA

Eşli programlama, aynı tasarım, aynı kod, aynı algoritma üzerinde iki kişinin birlikte çalıştığı bir projedir. (Beck, 2000) Eşli programlama yapılan uygulamalar araba yarışlarına benzetilebilir. Rallilerde olduğu gibi arabanın iki kişi tarafından kontrol edilmesine benzetilebilir. Rallilerde sürücü ve kılavuz olarak iki tane kontroller var; birisi arabayı kullanırken diğeri de kılavuzluk yaparak yön tayini, hızı, virajlardan haberdar eder. Eşli programlamada da sürücü rolündeki kişi sürekli olarak kod yazan, bilgisayarla ilgili işlemleri yapan, kodları test eden, tasarım yapan kişi olarak görevleri üstlenir. Sürücünün yanındaki kılavuz; sürücü, kod yazarken sürücüyü yönlendiren, hatalarını ileten, problemlere karşı fikir üreten ve bunları ifade eden, programı değerlendiren kişi olarak görevleri üstlenir. Eşli programlamanın rallilerden farkı sürekli olarak kişilerin belirli zaman dilimlerinde yer değiştirmesidir. (Çetin & Berigel, 2017)

Çetin & Berigel (2017), yazıkları makalede eşli programlama da sürecin nasıl işlediği kısmı 3 bölümde ele alınabilir;

- Bir bilgisayara iki tane programcı oturur.
- Oturan programcılardan bir tanesi bilgisayarı başında tasarım, kod yazarken diğer programcı kod yazanı değerlendirip hatalarını ve kendi fikirlerini söyler.
- Belirli bir süre sonra iki programcı kendi arasında yer değiştirir.

Eşli programlama eğitim alanında ve yazılım alanlarında kullanılmaktadır. Eşli programlamada kılavuz ve sürücü konumundaki programcılar aynı ortamda bulunmadan da eşli programlama yapabilirler. Ayrık eşli programlama olarak adlandırılan eşli programlama türünde programcılar aynı ortamda aynı bilgisayar başında olmazlar. Özel olarak geliştirilen yazılımlar sayesinde farklı ortamdaki ama aynı program üzerinde birlikte çalışabilirler. Farklı ortamlarda bulunan programcılar sunulan bu imkanla ayrık eşli programlama aracılığıyla eşli programlamanın yararlarını kullanabilirler. (Çetin & Berigel, 2017)

Programlama öğretimi verilen merkezlerde eşli programlama pedagojik bir araç olarak kabul edilip en çok kullanılan yöntemlerden birisi olmuştur (McDowel, Werner, Bullock, ve Fernald, 2006). Hanks, Fitzgerald, McCauley, Murphy, ve Zander, (2011) eşli programlama konusunda yaptıkları literatür taraması sonuçlarına göre eşli programlama temel ve orta seviye programlama eğitimi alan kişilerde başarıyı olumlu yönde değiştirmiştir. Programlama öğrenmeye, eşli programlama ile başlayıp sonradan bireysel olarak programlama eğitimi alan bireylerin bireysel eğitimde hiçbir güçlük çekmedikleri bulgusu ortaya çıkmıştır. Ayrıca yaptıkları çalışmanın diğer boyutu da öğrencilerin eşli programlama ile yaptıkları etkinliklerin büyük bir bölümünü eğlenceli olarak bulmuşlardır. Eşli programlama ile yazılan kodların kalite açısından son derece önemli olduğu bulgusuna da rastlanmıştır. Salleh, Mendes, ve Grundy, (2011) lisans ve lisansüstü öğrenciler ile eşli programlama üzerine yapılan çalışmaları incelediklerinde, eşli programlama üzerinde yapılan çalışmalardan sonuçları negatif yönde bulgular olmasına rağmen yapılan çalışmaların çok önemli bir bölümü akademik yönden başarı seviyesinin arttığı bulgusu yer almıştır. Yapılan istatistiki testlere göre eşli

programlamanın öğrencilerin girdiği final sınavları üzerinde herhangi olumlu ya olumsuz bir etkisinin olmadığı ama dönem süresince verilen ödevler üzerinde olumlu bir katkısının olduğu saptanmıştır. Ayrıca öğrencilerin memnuniyet açısından yaptıkları çalışmaya baktıklarında öğrencilerin çok büyük bir bölümü eşli programlama ile ilgili yaptıkları çalışmalardan son derece memnun oldukları görülmüştür. Öğrencilerin kod kalitesi açısından eşli programlama ile yaptıkları çalışmaların(yazılımların) daha kaliteli olduğu, içsel kod kalitesi açısından ise sonuçların şeffaf olmadığı gözlemlenmiştir. Ayrıca yazılımların yapılma süreleri göz önüne alındığında eşli programlama ile yapılan yazılımların sürelerinin daha kısa olduğu da dikkat çeken başka bir bulgudur.

McDowel, Werner, Bullock, ve Fernald, (2006) yaptıkları çalışmada öğrencilerin programlama eğitimine eşli programlama ile başlamasından sonra gelecek süreçte bireysel olarak yapılacak programlamaya uyum sürecinin kolay olduğunu, programlama ile ilgili yapılan tartışmalarda yardımcı olduğunu ve sosyal ilişkilerdeki düzeyi yukarıya çektiğini söylemiştir.

Hannay, Dyba, Arisholm, ve Sjoberg, (2009) yaptıkları çalışmada eşli programlamanın yazılımın kaliteli olması üzerine küçük ölçekli bir olumlu katkısının olduğunu söylemişlerdir. Bu çalışmayı daha ayrıntılı incelediklerinde verilen karmaşık problemlerin eşli programlama yöntemi kullanılarak yapılanların daha kaliteli kod kullandığını görmüşlerdir. Bu karmaşık problemlerin çözümünde eşli programlama yapan eşlerin normalden daha fazla çaba göstermeleri gerektiğini söylemişlerdir. Ayrıca eşli programlama ile yazılan yazılımların süreleri dikkate alındığında eşli programlama ile yapılan yazılımların sürelerinde orta seviyede bir olumlu etkisinin olduğunu söylemişlerdir.

Eşli programlamada, diğer önemli konulardan bir tanesi de eşlerin nasıl seçileceği konusudur. Hangi eşler bir araya gelmelidir? Eşleri seçerken hangi etmenler dikkate alınmalıdır? Eşli programlamada eşlerin seçiminden kaynaklanan olumsuzlukla yazılan yazılımların da sonuçlarını olumsuz yönde etkileyebilir. Mesela; programlama bilgi

seviyesi yukarıda olan bir öğrenci ile programlama bilgisi daha aşağı seviye de olan bir öğrenci beraber eş olduklarında, aralarından olacak alışveriş ve etkileşim sayesinde programlama bilgi seviyesi alt düzeyde olan öğrencinin kendisini sürekli geliştireceği düşünülebilir. Programlama bilgi seviyesi üst düzey olan öğrenci, bir süre sonra programlama bilgi seviyesi aşağıda olan öğrencinin fikirlerini kabullenmeyip kendi fikirlerini öne sürebilir ve programlama bilgi seviyesi aşağıda olan öğrenci de bu durumu kabullenip programlama ile ilgili çalışma ve ödevleri diğer arkadaşına bırakabilir. (Çetin & Berigel,2017) Alanyazın tarandığında sonuçların bu şekilde olmadığı görüşmüştür. Eşli programlama ile yapılan etkinlikler incelendiğinde eşler arasında çok yüksek oranda hiçbir uyum sorunun olmadığı görülmüştür. (Jacobson ve Schaefer, 2008)

Eşli programlama ilgili çalışmalar incelendiğinde iki ana faktörün göze çarptığını görüyoruz. Bunlardan bir tanesi eşli programlama yapacak eşlerin kişilikleri, diğer bir faktör ise bilgi ve becerileridir (Çetin & Berigel, 2017). Yapılan araştırmalardaki sonuçların büyük oranı eşli programlama yapacak eşlerin kişisel özelliklerinin birbirinden farklı olmasının eşli programlamada uyum sorununu ortaya çıkarmadığı gözlemlenmiştir (Hanks, Fitzgerald, McCauley, Murphy, ve Zander, 2011) .Yapılan bazı araştırmalara göre farklı kişisel özelliklere sahip kişilerin aynı kişisel özelliklerine sahip kişilere göre eşli programlamada daha performanslı olduğu görülmüştür (Choi, Deek, ve Im, 2008; Salleh, Mendes, ve Grundy, 2011). Eşlerin kişiliğinden ziyade eşli programlamada eşlerin bilgi ve beceri durumları daha önemli bir faktördür (Cliburn, 2003). Bilgi ve beceri seviyeleri eşli programlamadaki eşlerin algıladıkları seviye olarak gözetilmiştir. Araştırmacılar kendi yaptıkları araştırmalarda istatistiki verileri bu kişilerin algıladıkları bilgi ve becerileri olarak yapmıştır (Çetin & Berigel, 2017) . Bilgi ve beceri seviyesi farklı olan eşler, eşli programlama yaparken bilgi ve beceri seviyesi düşük olan bilgi ve beceri seviyesi yüksek olan eşin her dediğini kabul edebilir; bilgi ve beceri seviyesi yüksek olan ise bilgi ve beceri seviyesi yüksek olanın her dediğini kabul edebilir (Cao ve Xu, 2005).

Eğitimde eşli programlamada eşlerin birbiri ile iletişim kurması önemli bir faktördür. Eğitimci birbiri ile iletişim kuracak eşleri yan yana getirmelidir. Birbiri ile

iletişim problemi yaşayan eşlerin bir araya getirilmesi, eşli programlama performansının olumsuz yönde etkileyecektir (Çetin & Berigel, 2017).

Eşli programlama konusunda yapılan çalışmaların eşli programlama yönteminin faydalı olduğunu göstermektedir. Araştırmacılar, eğitimcilerin eğitim merkezlerinde uygulama sürecinde neler yapılması konusunda öneriler geliştirmiştir.

Bunlar:

- Eğitimciler, eşli programlama yapacakları ilk derslerde öğrencilerin birbirilerini tanımalarına olanak sağlayacak etkinlikler yaparak öğrencilerin birbiri ile uyum içerisinde çalışacağı eşler seçmelerine yardımcı olabilirler (Bevan, Werner, & McDowell, 2002).
- Programlama konusunda bilgi ve beceri seviyeleri birbirine yakın kişilerin eşli programlama yapması önemlidir. Bilgi ve beceri seviyeleri birbirine yakın olan öğrenciler seçilirse öğrenme daha hızlı gerçekleşir ve öğrencilerin yaptıkları projelerden sıkılma ihtimalleri ortadan kalkar (Bevan, Werner, ve McDowell, 2002).
- Eşli programlama yapmadan önce doğru eş bulmaları için eşli programlama yapacak adaylar bilinçlendirilmelidir (Williams , McCrickard, Layman, ve Hussein, 2008).
- Öğrenciler sürekli aynı kişilerle değil dönem dönem eşler değişmelidir (Williams ve diğerleri 2008).
- Eşli programlama da rollerin süreleri önemli bir faktördür. (Simon ve Hanks, 2008) Eşlerin birbirine güvenmesi verilen görevleri eşit olarak yapabilmesi ve beraber öğrenmesi için eğitimcinin eşli programlamada eşlerin rolleri için belirlenen sürelerin altını net olarak çizmelidir (Bevan ve diğerleri 2002).
- Öğrenciler programlama eğitimine başladıkları zaman ilk olarak programlama hakkında temel kavramları öğrenecekleri için, eşli programlama yapılacak zamanları eşit olarak iyi ayarlanmalıdır ve eşli

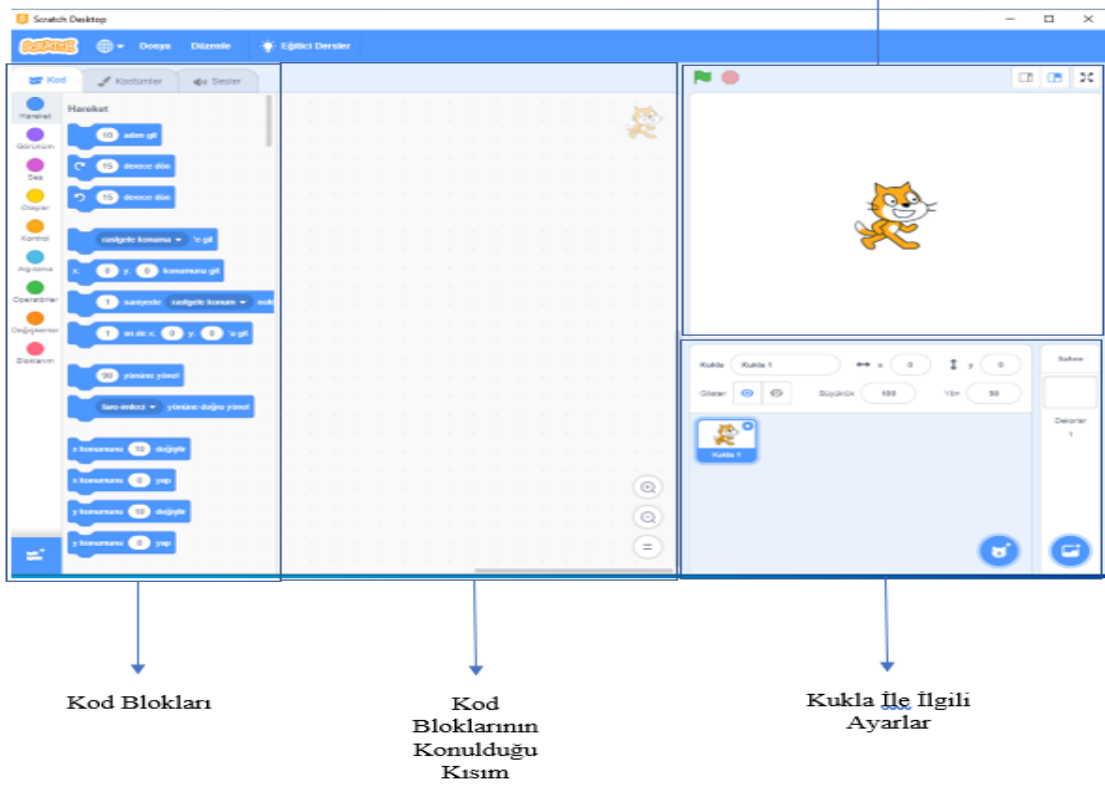
programlama yapacakları kadar süre verilmelidir (Bevan ve diğerleri 2002).

- Eğitimci eşli programlama yapacak eşlerin performanslarını gözetip, eğer bir sorun varsa devreye girmelidir (Williams ve diğerleri 2008).
- Eğitimci öğrencileri tek tek ya da eşli programlama yaptıkları eşlerle beraber ortak olarak değerlendirmelidir. (Williams ve diğerleri 2008).
- Eşli programlamada eşler farklı kod yapılarını kullanmak için direnç gösterebilirler. Eğitimcinin devreye girip bir kod standartı getirmesi gerekir. Bu standart, hem eşli programlama performansını arttıracak hem de eşler arasındaki ilişkiyi düzenleyecektir (Bevan ve diğerleri 2002).

2.9 ÖĞRETİM ORTAMLARI

2.9.1 Scratch offline editör

Scratch MIT Medya Laboratuvarında geliştirilen ücretsiz bir görsel programlama ortamıdır. 2003 yılında başlayan Scratch Projesi 2007 yılından itibaren offline yazılım ve web adresi üzerinden halkın kullanımına açılmıştır (Maloney ve diğerleri 2010). Scratch ile bloklar aracılığı ile program yazılır ve animasyon, oyun, simülasyon gibi birçok yazılım yapılabilir (Meerbaum-Salant, Armoni, ve Ben-Ari, 2010). Scratch 'in tek pencerelek, basit kullanıma sahip bir ara yüzü vardır. Kullanıcılar deneme-yanılma yoluyla daha önceden hazırlanmış projelerdeki kodları keşfetme yoluyla inceleyerek ya da kendileri program tasarlayarak programlamayı öğrenebilirler. Scratch programlama ortamı canlılık ve keşfedilebilirlik, kod kontrolünün sağlanma, yazım hatalarına izin vermeme, somutlaştırma sağlama, kod bloklarını en aza indirgeme gibi birçok avantaj sunmaktadır (Maloney ve diğerleri 2010). Şekil 2.1'de Scratch programlama ortamının ara yüzü sunulmuştur.



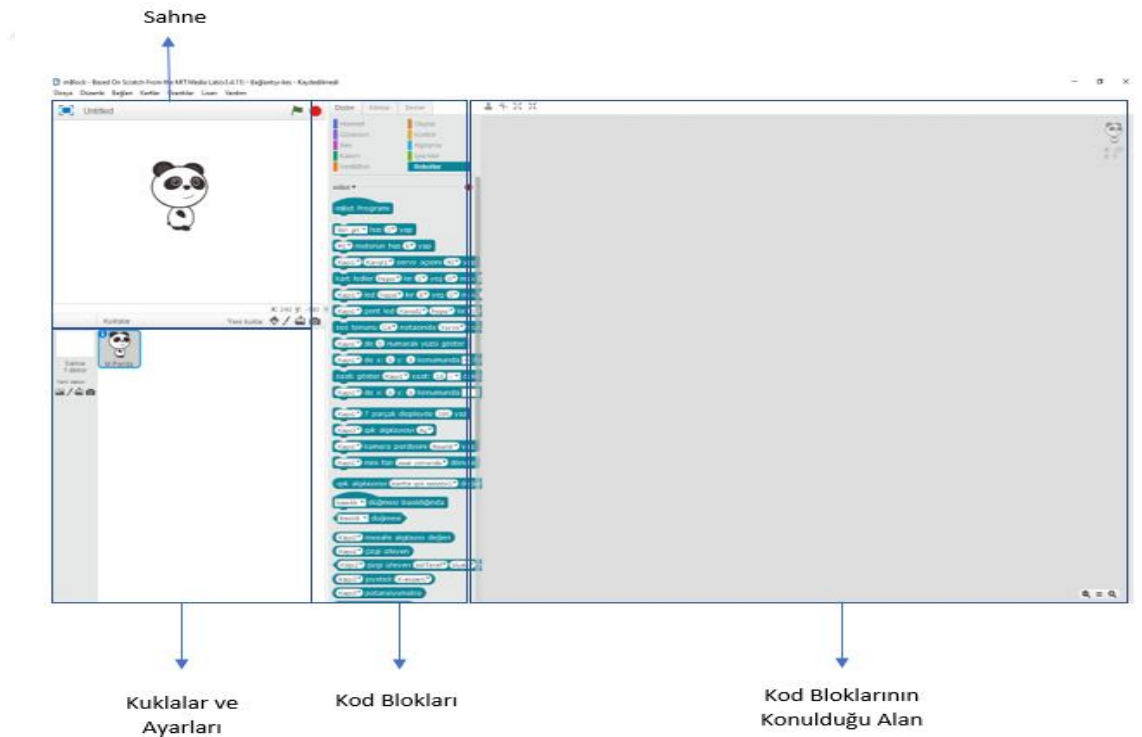
Şekil 2.1: Scratch 3 offline editör genel görünüm

Şekil 2.1’de sol kısımda bulunan kod blokları renklerine göre uygun olan blok başlıklarına yerleştirilmişlerdir. Scratch komutları Hareket, Görünüm, Ses, Olaylar, Algılama, Operatörler, Değişkenler ve Bloklarım olarak isimlendirilmiş her biri farklı renge sahip olan kod bloğu başlıklarında bulunmaktadır. Burada renklere göre ayrılan kodlar, kullanıcı tarafından oluşturulacak projeye göre kodlama alanına sürükle-bırak yapılar aktarılır. Kodların sürükle-bırak şeklinde olması, kullanıcının tarafından metin tabanlı bir kod parçası yazmak yerine hazır olan kod bloklarını sürükleyerek kolay bir şekilde proje yapmasını sağlayacaktır. Bu yüzden metin tabanlı programlama dilinde çıkan sözdizimi hataları olmayacaktır. Kodların unutulması, gözden kaçması, fiziksel olarak yanlış kullanılması mümkün değildir. Kodlar fiziksel olarak birbirine geçmeli bir şekilde tasarlanmıştır. Dolayısıyla bir kodu uygun olmayan başka bir kod ile birleştirmek mümkün değildir.

Diğer bir kısım olan Sahne’de ise kod blokları çalıştırıldıktan sonra kuklalarda ya da sahnede nelerin değiştiğini, nasıl bir projenin ortaya çıktığını yani yapılan projelerin sonuçlarının görüldüğü yerdir. Kukla Ayarları kısmında kuklanın ismi, kuklanın sahnede bulunduğu yerin koordinatları, kuklayı görünür-gizli yapma, kukla boyutu, kukla yönü gibi çeşitli ayarlar yapılabilir.

2.9.2 Mblock ve mbot

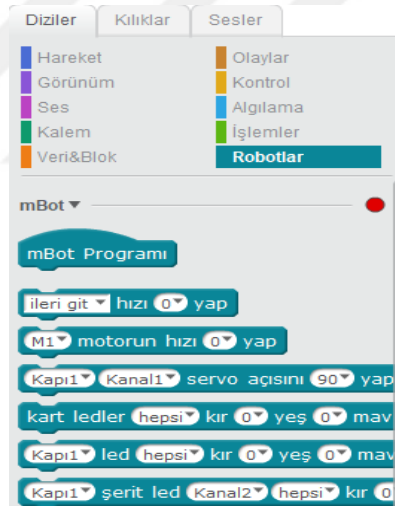
Bilişim öğrencilere gerçek dünyada hayal ettiklerini yapma ve geliştirme imkânı vermektedir (Papert, 1993). Eğitsel alanda kullanılan robotik kiti uygulamaya dönük etkinlikleri eğlenceli, öğrencilerin ilgi ve alakalarını arttıran bir eğitim ortamı sunduğu iddia edilmektedir (Eguchi, 2010). Günümüzde, Mbot robotik kiti mBlock programlama ortamıyla birlikte kullanılarak eğitsel robotik programlama için kullanılan ortamlardan biridir.



Şekil 2.2. mBlock programının genel görünümü

Şekil 2.2’de mBlock ortamının görsel ara yüzü sunulmuştur. mBlock programı 4 ana bölümden oluşur: Birinci bölüm olan “kuklalar ve ayarları” kısmından, oluşturulacak projelere uygun olan kuklaları eklenebilir, silinebilir ve kuklalar ile ilgili diğer bütün ayarlar (kukla ismi, koordinatları, yönü, gösterme-gizleme gibi) yapılabilir. Kuklanın üstünde olan ünlem işaretine tıkladığımız zaman kukla ile ayarların gözüktüğünü görülür.

İkinci bölümde kod blokları bulunmaktadır. mBlock programı Scratch tabanlı olduğu için genel anlamda kod bloklarının hepsi Scratch ile uyumludur. Scratch programlama ortamına ek olarak mBlock’ta Robotlar isimli bir kod bloğu başlığı bulunmaktadır. Şekil 2.3’te Robotlar başlığı sunulmuştur. Robot ile ilgili bütün kodlar bu başlıkta bulunmaktadır. Burada bulunan kodlar ile Mbot robotu kodlanabilir.



Şekil 2.3: mBlock programında robotlar isimli blok başlığı görünümü

Üçüncü bölümde kod bloklarının konulduğu alandır. Kod blokları bu alana sürüklenip mantıksal olarak sıralanır ve çalıştırılır. Kod blokları, Scratch ’de olduğu gibi paralel ya da doğrusal olarak çalıştırılabilir. Kullanılacak kod bloğu konusunda herhangi bir sınırlama yoktur. Hazırlayacağımız projeye uygun kodlar sürüklenip birleştirilir.

Dördüncü bölüm ise Sahne alanıdır. Hazırlanan projenin çıktısının görüldüğü ekrana sahne alanı denir. Yapılan projede kuklalarda ya da sahne üzerinde nasıl değişiklikler olduğu buradan gözlenir.

mBlock programıyla birlikte mBot robotu kullanılabilir. mBot robotu demonte bir kit halinde gelmektedir. Kullanıcılar gelen demonte kiti kendileri monte ederler. Robot kitiyle beraber uzaktan kumanda, çift taraflı kullanılabilen işlevsel tornavida, çizgi izleyen robot için tasarlanmış doküman, mBot robot kiti kullanım kılavuzu bulunmaktadır. Şekil 10 Mbot robotunun monte edilmiş hali görülmektedir.



Şekil 2.4. Mbot robot görünümü

Mbot robot kitinin içeriğinde iki tane DC motor, bir tane Ultrasonic Mesafe Sensörü, bir tane çizgi izleyen sensör (Siyah be beyaz renklerini algılayan) lityum pilin

olmadığı AA pil tutucu, kullanıcının isteğine göre bağlantı modüllerinden olan bluetooth modülü ya da 2.4 G uzaktan bağlantı modülü ve programlanabilir Mbot kartı olan mcore bulunmaktadır. Mbot robot kitinin üzerinde bulunan mcore kartı üzerinde RGB LED, Buzzer, kızıl ötesi alıcısı ve kızıl ötesi vericisi, ışık sensörü, sensörleri bağlamak için RJ25 portlar, buton, sağ ve sol motorların bağlandığı girişler bulunmaktadır.

2.9.2.1 Bağlantı şekilleri

2.9.2.1.1 2.4 G Kablosuz bağlantı

Bu yöntemle bir dongle bilgisayara takılıp mBlock üzerinden bağlantı ayarları yapıldıktan sonra anlık olarak veri alışverişi gerçekleştirilir. Mcore kart üzerinde 2.4 G kablosuz bağlantı modülünün olması gerekmektedir. Notebook ve masaüstü bilgisayarlar için uyumlu bir bağlantı şeklidir. Bağlantı yapıldıktan sonra sürükle-bırak ile oluşturulan kod bloklarının çıktıları robot üzerinde anında görülür.

2.9.2.1.2 Bluetooth bağlantı

Bu bağlantı şekli tabletler, bluetooth bağlantısı olan notebooklar için uyumlu olup masaüstü bilgisayarlara da dışarıdan takılacak Bluetooth cihazları ile bağlantı sağlanabilir. Bu bağlantı türünde de yazılan kodların çıktısı aynı anda robot kiti üzerinde görülür. Mcore kart üzerinde Bluetooth modülünün olması gerekmektedir. 2.4 G bağlantı şekline göre bağlantı hızı biraz yavaştır.

2.9.2.1.3 Kablo ile bağlantı

USB kablo ile bağlantının olduğu bir yöntemdir. Kablo takılıyken yaptığımız kodlamalar anında robot kiti üzerinde görülebilir ya da kablo üzerinde mBot robot kiti üzerinde bulunan karta gönderilerek yazdığımız kodların çıktılarını mBot üzerinde görebiliriz. Kablo ile bağlanmanın bir dezavantajı vardır. Yazılan kodların çıktılarını karta yüklemekten görmek istenildiğinde kablonun kısa olmaması gerekmektedir. Kısa olan kablolarda Mbot robot kitinin hareketleri de kısıtlı olacağından yapılan projeler amaçlarına ulaşamayabilir.

2.9.3 Python

Python 1991 yılında Guido Van Rossum tarafından geliştirilmiş, yüksek seviyeli, güçlü ve dinamik bir nesne tabanlı programlama dilidir (Harwani, 2012). Python platformdan bağımsız bir programlama dili olduğundan dolayı bütün işletim sistemlerinde sorunsuz olarak çalışabilmektedir. Python programlama dilinin söz dizimi basit olmasından dolayı hem program yazmak hem de başkalarının yazdığı yazılımları kodları okumak başka programlama dillerine göre daha kolaydır (Özgül, 2013).

Python için birçok kütüphane bulunmaktadır. Bu araştırmada Python Turtle kütüphanesi tanıtılacaktır. Python programlama dili birçok platformda kullanılmaktadır. Çevrimiçi ve çevrimdışı ile kod yazım işlemi gerçekleştirilmektedir. Burada Python programlama dilinde kod yazmak için **Idle** programlama ortamı kullanılmıştır.

Tablo 2.1. Örnek olarak python turtle modülünde yazılmış kodlar.

print ("Ahmet CAN")
print ("5A")
print ("Orhan Gazi Ortaokulu")

Yukarıda Idle’da yazılan birtakım kod satırları bulunmaktadır. Bu satırlarda bulunan “print” ekrana metin yazdırmayı sağlayan komuttur.

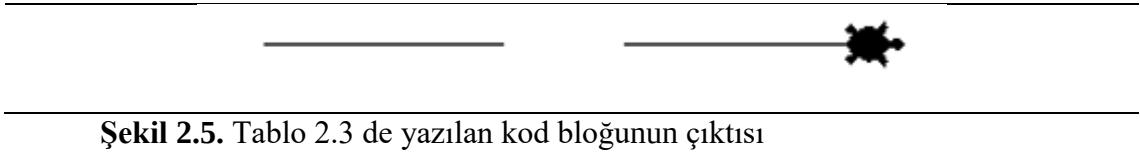
Tablo 2.2. Şekil 5 de yazılan kodların ekran çıktısı

Ahmet CAN
5A
Orhan Gazi Orhangazi
>>>

Python Turtle kütüphanesi koordinat düzlemi üzerinde küçük bir kaplumbağanın verilen koordinatlar ve kodlar sayesinde şekiller çizmesini sağlayan bir kütüphanedir. Kütüphanenin kullanılması için ilk önce içeri aktarılması gerekmektedir. “Import turtle” kodu yazılarak Turtle kütüphanesi kullanılır hale gelir. Aksi takdirde Turtle kütüphanesine ait olan kodları kullanmak mümkün değildir. Aşağıdaki örnekte Turtle kütüphanesi kullanılarak aralıklı iki çizgi çizilmiştir. Kaplumbağa çizim yaparken kalemi verilen ölçümler oranında yukarı kaldırıp (çizim olayını gerçekleştirmeden), verilen ölçümler oranında ise kalemi bastırıp (çizim olayını gerçekleştirerek) çizimler yapmayı sağlamaktadır. Tablo 2.3’te bu kodlar ile bir uygulama yapılmış olup ekran çıktısı şekil 2.5 de görülmektedir.

Tablo 2.3. Python turtle modülü ile penup ve pendown kodlarının kullanımı

import turtle
kalem=turtle. Turtle (shape=” turtle”)
kalem.forward (100)
kalem. penup ()
kalem.forward(50)
kalem.pendown()
kalem.forward(100)



Aşağıda Turtle kütüphanesi kullanılarak bir dikdörtgen çizdirilmiştir. Kodların yanında kullanım amaçları da yorum olarak eklenmiştir. Bu komutların ekran çıktısı Şekil 2.6’da görüldüğü gibi olur.

Tablo 2.4. Python Turtle modülü ile dikdörtgen çizimi.

import turtle	#Çizim yapmak için turtle modülünü çağırır.
kalem=turtle.Turtle(shape="turtle")	#Kaplumbağa ile bir çizim aracı oluşturur.
pencere=turtle.Screen()	#Çizim için bir pencere açar.
pencere.bgcolor("white")	# Pencerenin rengini beyaz olarak ayarlar.
pencere.title("Turtle ile Dikdörtgen")	#Pencerenin başlığını ayarlar.
kalem.speed(2)	#Kalemin hareket hızını 2 birim olarak ayarlar.
kalem.color("red")	#Kalemin rengini kırmızı olarak ayarlar.
kalem.pensize(4)	#Kalemin kalınlığını 4 birim olarak ayarlar.
kalem.forward(300)	#Kaplumbağanın 300 birim ilerlemesini sağlar.
kalem.color("yellow")	#Kalemin rengini sarı olarak ayarlar.
kalem.pensize(2)	#Kalemin kalınlığını 2 birim olarak ayarlar.
kalem.left(90)	#Kaplumbağanın 90 derece sola dönmesini sağlar.
kalem.forward(100)	#Kaplumbağanın 100 birim ilerlemesini sağlar.
kalem.color("red")	#Kalemin rengini kırmızı olarak ayarlar.
kalem.pensize(4)	#Kalemin kalınlığını 4 birim olarak ayarlar.
kalem.left(90)	#Kaplumbağanın 90 derece sola dönmesini sağlar.
kalem.forward(300)	#Kaplumbağanın 300 birim ilerlemesini sağlar.
kalem.color("yellow")	#Kalemin rengini sarı olarak ayarlar.
kalem.pensize(2)	#Kalemin kalınlığını 2 birim olarak ayarlar.
kalem.left(90)	#Kaplumbağanın 90 derece sola dönmesini sağlar.

<code>kalem.forward(100)</code>	<code>#Kaplumbağanın 100 birim ilerlemesini sağlar.</code>
---------------------------------	--

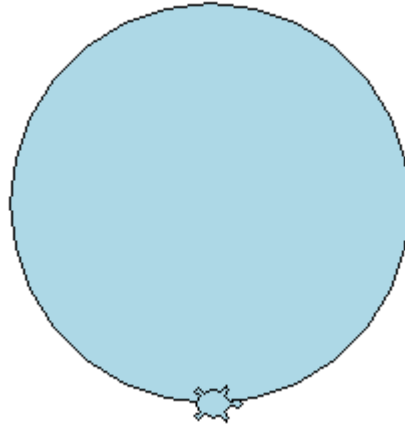


Şekil 2.6: Tablo 2.4’te görünen kod bloklarının ekran çıktısı.

Turtle kütüphanesi ile kütüphane içerisinde önceden tanımlanmış hazır şekiller de çizdirilebilir. Örneğin aşağıdaki kodlar bir daire çizer. Bu iş için “circle” fonksiyonu kullanılmıştır.

Tablo 2.5. Python Turtle modülünde kapalı şekillerin içini doldurmaya yarayan kodlar

<code>İmport turtle</code>	
<code>kalem=turtle.Turtle(shape="turtle")</code>	
<code>kalem.begin_fill()</code>	<code>#Doldurma işlemini başlatır.</code>
<code>kalem.fillcolor("light blue")</code>	<code>#Doldurulan nesnenin rengini açık mavi yapar</code>
<code>kalem.circle(100)</code>	<code>#100 birimlik daire çizmeye yarar.</code>
<code>kalem.end_fill()</code>	<code>#Doldurma işleminin bittiğini gösterir.</code>



Şekil 2.7. Tablo 2.5’te yazılan kod bloklarının çıktısı.

III.BÖLÜM

3. Yöntem

Bu bölümde araştırmanın deseni, araştırmaya katılan katılımcılar, araştırma yaparken kullanılan veri toplama araçları ve bu araçların geliştirilme süreci, verileri toplama süreci, toplanan verilerin yorumlanması ve çözümlenmesi ele alınacaktır.

3.1 ARAŞTIRMA DESENİ

Bu araştırmada üç farklı kodlama ortamının (Scratch, Python ve mBlock) altıncı sınıf öğrencilerinin bilgi işlemsel düşünme beceri seviyelerine, başarılarına ve tutumlarına etkisinin araştırılması amaçlanmıştır. Araştırmada nicel araştırma yöntemlerinden deneysel yöntem kullanılmıştır. “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, Kılıç Çakmak, Akgün, Karadeniz, & Demirel, 2008,s.212). Bu araştırmada deneysel yöntemin bir çeşidi olan yarı deneysel araştırma yöntemi kullanılmıştır. “Yarı deneysel desen yansız atamanın kullanılmadığı durumlarda kullanılması uygun olan araştırma desenidir.” (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz, ve Demirel, 2008,s.216). Araştırma kapsamında altı farklı altıncı sınıf şubesi bulunan Bolu ili merkez ilçesinde bir ortaokulda üç farklı altıncı sınıf şubesi kullanılmıştır. Bu üç şube rastgele seçilmiştir. Şubelerden birisi Scratch tabanlı, diğeri Python tabanlı ve sonuncusu da mBlock tabanlı öğretim için rasgele atanmıştır. Çalışmanın başında bütün gruplara Çetin, Otu, Okaç (2020)’nin İspanyolcadan Türkçeye

uyarladığı Bilgi İşlemsel Düşünme Testi (BİDT) ve Gül, Çetin ve Özden (2019) tarafından geliştirilen Bilgisayar Programlama Tutum Anketi (BPTA) ön-test olarak uygulanmıştır. Bunun ardından üç farklı öğretim 8 hafta boyunca belirlenen şubelerde uygulanmıştır. Uygulanan öğretimden sonra BİDT, BPTA ve bu tez kapsamında geliştirilen Kodlama Başarı Testi (KBT) her üç gruba da uygulanmıştır. Bu çalışmada kullanılan yarı deneysel desenin özeti Tablo 3.1’de sunulmuştur.

Tablo 3.1: Uygulanan yarı-deneysel desenin özeti

Grup	Ön-Test	Programlama Dili	Son Test
Grup 1	BİDT BPTA	Scratch	BİDT
			BPTA
			KBT
Grup 2	BİDT BPTA	Python	BİDT
			BPTA
			KBT
Grup 3	BİDT BPTA	mBlock	BİDT
			BPTA
			KBT

- * Bilgi İşlemsel Düşünme Testi (BİDT)
- * Bilgisayar Programlama Tutum Anketi (BPTA)
- * Kodlama Başarı Testi (KBT)

3.2 ÇALIŞMA GRUBU

Bu tezde uygun örnekleme tekniği kullanılmıştır. “Uygun örnekleme yöntemi zaman, para ve iş gücü açısından var olan sınırlılıklar nedeni ile örneklemin kolay ulaşılabilir ve

uygulama yapılabilir birimlerden seçilmesidir” (Büyüköztürk ve diğerleri 2008,s.95). Yapılan araştırmalarda çalışmaya en uygun çalışma grubunun seçilmesi uygun örnekleme yöntemi olarak bilinir. (Büyüköztürk , Çakmak, Akgün, Karadeniz, ve Demirel, 2016) Bolu ili merkez ilçesinde bir ortaokulda hali hazırda bulunan 6 altıncı sınıf şubesinden rastgele 3 şube alınarak Tablo 3.1’de gösterildiği gibi öğretim yöntemleri rastgele olarak atanmıştır. Bu şubelerde bulunan öğrenciler 10-12 yaş aralığındadır. Çalışmaya toplamda 105 altıncı sınıf öğrencisi katılmıştır. Bu öğrencilerin 45’i kız, geriye kalan 60’ı ise erkek öğrencidir. Kız erkek dağılımları Tablo 3.2’de gösterilmiştir.

Tablo 3.2: Araştırmaya katılanların demografik özellikleri

Şube Adı	Kız	Erkek	Toplam
6/A	14	20	34
6/B	15	20	35
6/F	16	20	36
Toplam	45	60	105

Öğrencilerin cinsiyetinin sınıf bazlı dağılımı Tablo 3.2’de gösterilmiştir. Birinci grupta (Scratch tabanlı grup) 14 kız öğrenci, 20 erkek öğrenci bulunmaktadır. İkinci grupta (Python tabanlı) 15 kız öğrenci, 14 erkek öğrenci bulunmaktadır. Üçüncü grupta (mBlock tabanlı) 16 kız öğrenci ,20 erkek öğrenci toplamda 105 öğrenci bulunmaktadır.

12.09.2018 tarihli ve 123 sayılı Talim ve Terbiye Kurul Kararı eki İlköğretim Kurumları Haftalık Ders Çizelgesinin Açıklamalar bölümünde yer alan 7. Madde doğrultusunda ortaokul beşinci sınıfta Türkçe, matematik, fen bilimleri, sosyal bilgiler, din kültürü ve ahlak bilgisi dersleri (20 ders saati) ile en az dört (4) ders saatlik seçmeli dersi tamamlamak kaydıyla, imkân ve şartları uygun olan okullarda isteğe bağlı olarak onsekiz (18) ders saatine kadar yabancı dil dersi öğretimi yapılabilmektedir. (Millî Eğitim Bakanlığı [MEB], 2019) Bu kapsamda okulumuzda 5. sınıf öğrencileri İngilizce hazırlık gördüğünden bilişim teknolojileri ve yazılım dersi yapılamamaktadır. Dolayısıyla

araştırmaya katılan öğrencilerin bilişim teknolojileri ve yazılım dersini ilk defa 6. Sınıfta görmüşlerdir.

3.3 VERİ TOPLAMA ARAÇLARI

Bu araştırmada ön-test olarak BİDT ve BPTA kullanılmış olup son test olarak BİDT, BPTÖ ve KBT uygulanmıştır. Bu testler ile ilgili detaylar alt başlıklar halinde sunulmuştur.

3.3.1 Bilgi işlemsel düşünme testi

BİDT Roman-Gonzalez (2015) ve Román-González, Pérez-González ve Jiménez-Fernández (2017) tarafından İspanyolca olarak geliştirilmiş bir testtir. Test ile öğrencilerin bilgi işlemsel düşünme seviyelerinin ölçülmesi amaçlanmaktadır. BİDT 8 bölümden oluşmaktadır ve toplamda 28 çoktan seçmeli soru içermektedir. Her bir sorunun dört şıkkı bulunmaktadır. Soruların tamamı Blok tabanlı olarak hazırlanmıştır. BİDT Otu, Çetin ve Oktaç (2020) tarafından İspanyolcadan Türkçeye uyarlanmıştır (Çetin, Otu, & Oktaç, 2020). Türkçe uyarlaması sonucunda 24 maddeden oluşan bir test elde edilmiştir. Testin KR20 değeri 0,78 olarak bulunmuştur. Bu çalışmada ön test olarak KR20 değeri 0.76, son test olarak KR20 değeri 0,79 olarak hesaplanmıştır. 0.7 üzeri olan değerler genel olarak yüksek derecede kabul edilebilir. (Büyüköztürk Ş. , Çakmak, Akgün, Karadeniz, ve Demirel , 2017)

3.3.2 Programlama tutum anketi

Bilgisayar Programlamaya Karşı Tutum Anketi (BPTA) Gül, Çetin ve Özden (2019) tarafından geliştirilmiştir. (Gül, Çetin, ve Özden, 2019) BPTA'nın amacı ortaokul öğrencilerinin programlamaya karşı tutumlarını ölçmektir. Testte 13 madde bulunmaktadır. Maddeler 5' li likert tipinde hazırlanmıştır. Bunlar “Hiç Katılmıyorum,

Katılmıyorum, Kararsızım, Katılıyorum, Tamamen Katılıyorum” şeklindedir. Tek faktörden oluşan testin Cronbach Alfa güvenilirlik katsayısı 0.93 olarak bulunmuştur. Bu araştırma da BPTÖ ön test Cronbach Alfa güvenilirlik katsayısı 0.91 , BPTÖ son test olarak Cronbach Alfa güvenilirlik katsayısı 0.93 olarak hesaplanmıştır.

3.3.3 Kodlama başarı testi

Bu araştırma kapsamında geliştirilen KBT son test olarak kullanılmıştır. Kodlama-Başarı Testi 14 sorudan oluşan bir testtir. Bu testin üç sürümü bulunmaktadır. Bu sürümler verilen öğretime paralel olarak Scratch, Python ve mBlock ortamları için hazırlanmıştır. Her üç sürüm içerisinde bulunan sorular ortaktır. Fakat sorular programlama diline göre düzenlenmiştir. Aşağıda testte bulunan bir maddenin üç sürümünde de kullanılan örnek bir soru verilmiştir.

Tablo 3.3: Scratch ortamı için hazırlanan örnek soru

Soru : Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?	
a) 140 b) 270 c) 300 d) 410	

Tablo 3.4: Python ortamı için hazırlanmış örnek soru

Soru : Yandaki kod bloğu çalıştırıldığında kaplumbağa toplamda kaç birim ilerler?	
a) 140 b) 270 c) 300 d) 410	<pre>import turtle kalem=turtle.Turtle() kalem.forward(80) kalem.right(90) kalem.forward(40) kalem.right(90) kalem.forward(20) kalem.right(90)</pre>

Tablo 3.5: mBlock ortamı için hazırlanmış örnek soru

Soru 1. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- a) 140
- b) 270
- c) 300
- d) 410



KBT’de bulunan maddeler geliştirildikten sonra iki bilişim uzmanı ve iki Türkçe uzmanı tarafından kapsam, içerik ve dil açısından incelenmiştir. Uzmanlardan alınan dönütler neticesinde sorularda gerekli düzeltmeler yapılmıştır. Bu sürecin ardından test eşdeğerlik açısından iki alan uzmanı tarafından değerlendirilmiştir. Alan uzmanları testin her üç sürümde bulunan soruları sırayla değerlendirmiştir. Her bir soruya 0-10 arasında puan vermişlerdir. 0 puan “üç sürümdeki sorular tamamen birbirinden farklıdır”, 10 puan ise “üç sürümdeki sorular tamamen birbirlerinin aynısıdır” anlamına gelmektedir. Uzmanlar iki soru dışındaki bütün sorulara 10 puan vermiştir. Bir soru 9, son soru ise 8 puan almıştır. Uzmanlardan gelen dönütler doğrultusunda bu maddelerde gerekli düzeltmeler yapılmıştır. Düzeltmelerin ardından sorular alan uzmanlarına yeniden gönderilmiştir ve alan uzmanlarından soruları yeniden puanlamaları istenmiştir. Bu süreç sonucunda bütün sorular 10 puan almıştır. Bu çalışmada KBT KR20 değeri 0.74 olarak hesaplanmıştır. Tutarlılık seviyesi iyi durumdadır. (Büyüköztürk ve diğerleri 2017)

KBT bu tez kapsamında geliştirilmiştir. KBT için oluşturulan belirtke tablosunda Milli Eğitim Bakanlığının altıncı sınıf bilişim teknolojileri ve yazılım dersi için oluşturduğu kazanımlar kullanılmıştır (Gülbahar, Kalelioğlu, ve Kert, 2019). Belirtke tablosunda bulunan ilk dört kazanım basit düzeyli kazanımlardır. Diğer soruları yapabilmek için bu kazanımların elde edilmiş olması gerekmektedir. Bu yüzden bu kazanımlar için ayrı sorular yazılmamıştır. Soru numarası kısmına “bütün sorular” ibaresi yazılmıştır. Belirtke tablosundaki son kazanım ise bir proje oluşturulmasını istemektedir.

Bu kazanımı çoktan seçmeli bir sınav ile ölçmek mümkün olmadığı için bu kazanım için soru oluşturulmamıştır. Soru numarası yerine “uygulanamaz” ibaresi düşülmüştür.

Tablo 3.6. Belirtke tablosu

Kazanım	Soru No
Programlama aracının arayüzünü ve özelliklerini tanır.	Bütün Sorular
Programlama aracında sunulan bir programın işlevlerini açıklar.	Bütün Sorular
Programlama aracında sunulan bir programın hatalarını ayıklar.	Bütün Sorular
Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.	Bütün Sorular
Doğrusal mantık yapısını içeren programlar oluşturur ve hatalarını ayıklar.	1, 3, 5, 7
Karar yapısını içeren programlar oluşturur ve hatalarını ayıklar.	9, 12
Çoklu karar yapıları içeren programlar oluşturur ve hatalarını ayıklar.	6, 8, 12
Döngü yapısını içeren programlar oluşturur ve hatalarını ayıklar.	2, 4, 10, 11
Bir algoritmayı uyarlamak için en uygun karar yapılarını seçer.	6, 8, 9, 12
Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.	3, 10, 14
Tüm programlama yapılarını içeren özgün bir proje oluşturur.	Uygulanamaz

Belirtke tablosu oluşturulduktan sonra 25 adet soru hazırlanmıştır. Bu 25 sorunun 10 tanesi elenmiştir ve geriye kalan 15 soru ile birlikte belirtke tablosu gönderilerek uzman görüşü istenmiştir. Uzmanlar, maddelerin içerik, kazanımlara ve öğrenci seviyesine uygunluk ile kapsam geçerliği açısından maddeleri değerlendirmiştir. Buna ek olarak oluşturulan maddeler Türkçe dil uzmanına yazım kuralları açısından değerlendirilmek üzere gönderilmiştir. Uzmanlardan gelen dönütlere göre maddeler üzerinde gerekli değişiklikler yapılmıştır.

15 maddeden oluşan başarı testinin Scratch sürümü Bolu ili merkez ilçesinde yer alan iki farklı ortaokulda bulunan 169 yedinci sınıf öğrencisine (73 kız ve 96 erkek) uygulanmıştır. Bu öğrencilerin hepsi daha önce bilişim teknolojileri ve yazılım dersini almıştır. Scratch uygulamaları yapmıştır. Elde edilen yanıtlar TAP yazılımı kullanılarak incelenmiştir. Sonuçlar Tablo 3.7’de sunulmuştur.

Tablo 3.7: Madde analizi sonuçları

Madde	Madde Güçlüğü (p)	Madde Ayırt Ediciliği (r)	Üst Grup	Alt Grup
1	0,86	0,32	51	40
2	0,86	0,37	51	37
3	0,26	0,48	27	3
4	0,62	0,65	46	15
5	0,44	0,66	43	11
6	0,57	0,51	45	22
7	0,38	0,70	39	4
8	0,47	0,41	34	15
9	0,84	0,37	51	37
10	0,86	0,34	50	38
11	0,70	0,35	44	30
12	0,73	0,61	51	23
13	0,41	0,45	37	16
14	0,56	0,49	43	21
15	0,11	0,05	6	4

15. maddenin madde güçlüğü 0,11 ve madde ayırt ediciliği 0,05 olarak bulunmuştur. Bu maddenin indeks değerleri uygun görülmediği için (Büyüköztürk ve diğerleri 2017) testten çıkartılmıştır ve analizler yenilenmiştir. Elde edilen yeni değerler Tablo 3.8’de sunulmuştur.

Tablo 3.8: Atılan madde sonrası madde analizi sonuçları

Madde	Madde Güçlüğü (p)	Madde Ayırt Ediciliği (r)	Üst Grup	Alt Grup
1	0,86	0,31	48	42
2	0,86	0,36	48	39
3	0,26	0,49	26	3
4	0,62	0,65	44	16
5	0,44	0,68	42	12
6	0,57	0,50	43	24
7	0,38	0,73	38	4
8	0,47	0,43	33	16
9	0,84	0,36	48	39
10	0,86	0,34	47	39
11	0,70	0,38	42	30
12	0,73	0,62	48	23
13	0,41	0,45	35	17
14	0,56	0,47	40	22

14 maddeden oluşan başarı testinin madde güçlükleri 0,26 ve 0,86 arasında değişmektedir. Testin ortalama güçlüğü 0,61 olarak bulunmuştur. Testin farklı zorluk seviyelerinde soru bulunduran kolay bir test olduğu söylenebilir. (Büyüköztürk ve diğerleri 2017) Başarı testinin madde ayırt edicilik değerleri, 0,31 ile 0,73 arasında değişmektedir. Testin ortalama ayırt ediciliği 0,48 olarak bulunmuştur. Testin çok iyi seviyede ayırt ediciliğe sahip olduğu söylenebilir. (Büyüköztürk ve diğerleri 2017) Elde edilen sonuçlara göre KR20 güvenirlik değeri 0,74 olarak bulunmuştur ve iç tutarlılık seviyesi yeterlidir denilebilir. (Özdamar, 1999)

Testten atılan 15. madde “Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.” kazanımıyla ilgilidir. Bu madde çıkarıldıktan sonra bu kazanım ile ilgili üç madde kalmıştır. Atılan maddenin testin geçerliği üzerine etkisini araştırmak için iki alan uzmanına başvurulmuştur. Alan uzmanları testin atılması ile geçerliğin etkilenmediği görüşünü bildirmiştir. Sonuç olarak KBT’nin yeterli geçerlik ve güvenirliğe sahip olduğu söylenebilir. KBT’nin tamamına EK 1-2-3’de ulaşabilirsiniz.

3.4 ÖĞRETİM ORTAMLARI

Öğretim Ortamı olarak 35 civarı öğrencinin olduğu bilişim teknolojileri laboratuvarında işlendi. Öğrenciler, 20+1 bilgisayarın olduğu sınıfta bilgisayar başına ikişer kişi oturmuşlardır. Her üç grupta da dersler bu tezin araştırmacısı tarafından işlenmiştir. Bütün öğretim ortamları için aynı laboratuvar kullanılmış olup öğrenciler çalışmalar yaparken eşli programlama yöntemini kullanmışlardır. Eşli programlama, iki kişinin tek bilgisayar başında aynı program, aynı algoritma çözümü için uğraşmasıdır. (Beck, 2000) Bu uygulamada belirlenen kurallar rallilerde sürücü ve yanındaki yardımcı kişi ile benzerlik göstermektedir. Rallilerde iki kişinin tek bir arabayı yönettiği gibi burada da iki kişi tek bilgisayarı yönetmektedir. Birisi arabayı kullanırken diğeri de arabayı kullananın kılavuzu şeklindedir. Eşli programlamada arabayı kullanan kişi bilgisayarı aktif olarak kullanan, kodları yazan, kodların çalışmasını test eden, aynı zaman da kod tasarımı yapan kişidir. Sürücünün yanındaki kılavuz olan kişi ise kod yazanı yönlendiren, yazılan kodları değerlendiren, hatalar hakkında fikir veren kişidir. Eşli programlama uygulamasında araba yarışlarından farklı olarak kod yazan ve kılavuzluk eden kişinin belirli zaman dilimlerinde rolleri değişmeleridir. (Çetin ve Berigel, 2017). Araştırmada kullandığımız eşli programlamada öğrencilere araştırmaya başlamadan önce kuralları ile birlikte vererek öğrencilerin ders saatleri içerisinde belirli zaman aralıklarında bilgisayar başındaki rolleri değiştirmeleri sağlanmıştır.

Öğretim yöntemi olarak inşacılık yöntemi ele alınmıştır. İnşacılık yöntemi temel olarak Piaget kuramına dayanır ve yapılandırmacılığın bir parçası olarak tanımlanır. (Papert, 1990) Yapılandırmacılık yaklaşımına göre insanlar bilgiyi bir kaynaktan değil yaşadıkları deneyimler ile öğrenirler. Yapılandırmacı kuram, bilginin öğretmen tarafından öğrenciye iletileceği bir durumun olmadığı; öğrencinin bilgiyi kendi zihninde yapılandırarak oluşturduğu kuram olarak tanımlanır. Bu anlamda yapay zeka kuramları cinsiyet ve kişilik çalışmaları da Piaget'in bu kuramı oluştururken destek aldığı alanlardandır. (Harel, 1991) Bireylerde bilgi oluşurken bir süreçten geçilir. Nesneler;

dönüşür,değişir ve bu süreç birey tarafından anlamlı hale geldiğinde bilgi yapılandırılmış olur. (Piaget, 1964). Logonun dayandığı kuram olan inşacılık teknolojinin eğitimdeki önemini ve eğitsel yeniliklerin yönünü değiştiren kuramdır. (Kafai & Resnick, 1996) Derslerimiz öğretmenin genel anlamda pasif olduğu öğrencinin keşfederek öğrendiği bir süreçten geçmiştir. Öğretmen her derste daha çok rehber konumundadır. Dersin konusu hakkında bilgi verip öğrenciyi yeni bir problem ile karşılaştırıp ders içerisinde öğrenciye rehberlik etmiştir. Öğrenci,karşılaştığı problemi öğretmenin verdiği bilgilerin üzerine bazı yollar deneyerek bazen yanlış bazen doğru yaparak problemi çözmeye çalışmıştır. Öğrenciler, problemi çözerken öğretim ortamına göre değişik ve kalıcı bilgiler edinmiş olabilirler.

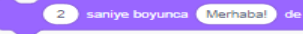
Bilişim teknolojileri laboratuvarında işlenen derslerde bir ders saati 40 dakikadan oluşmaktadır. Haftada 2 ders saati, toplamda 80 dakika olarak dersler işlenmiştir. Bütün dersler, araştırma süresince bilişim laboratuvarında işlenerek derslerde zaman ve yer konusuna riayet edilmiştir. Her ders saatinde bilişim laboratuvarı araştırma için seçilen kodlama ortamına uygun olarak düzenlenmiştir.

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

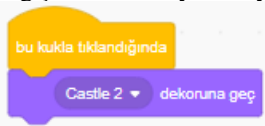
- Dikkat çekme
 - Güdüleme
 - Gözden geçirme
 - Derse geçiş
 - Bireysel öğrenme etkinlikleri
 - Grupla öğrenme etkinlikleri
 - Özet
- Ders başladıktan sonra öğrencilere geçen derste ilk defa bir kuklanın kılık değiştirerek hareket etmesini yani animasyon yapmayı öğrendikleri hatırlatılır,bugün ise bir tanışma animasyonunu yapacakları söylenir. Daha sonra iki tane hayvan resmi olan kukla eklemeleri ve iki kuklanın sahne üzerinde birbiri ile konuşacak şekilde birbirlerine dönmelerini sağlamaları istenir. Kuklaları ekledikten sonra konuşmak için hangi blokların kullanılabileceğini sorulur ve öğrencilerden cevap beklenir.
- Öğrencilerin  kod bloğu ve  kod bloklarını keşfetmeleri sağlanır. “Sizce ikisi arasındaki fark ne olabilir?” diye sorulur. Öğrencilerin kod bloklarını kuklalar üzerinde deneyerek birinci kod bloğunun 2 saniye sonra kaybolduğunu diğer kod bloğunun ise hiç ekrandan kaybolmadığını görmeleri için ipuçları verilir. Öğrencilere aynı zamanda kuklaların ses çıkarabildiklerini söylenir ve öğrencilerin  kod bloğunu keşfetmeleri sağlanır. Daha sonra eklenen hayvan kuklalarının birbiri ile tanışmaları için  yeşil bayrağa tıklanınca kod bloğu ile başlamaları ve kukları konuşturmaları gereken kod bloklarını kendi uydurdukları bir senaryo ile hem konuşarak hem kılık değiştirerek ve aynı zaman da ses çıkararak sağlayan uygulamayı yapmaları istenir.



Şekil 1

Şekil 2

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Yukarıda görülen Şekil 1 deki gibi her kukla için yazılan kod blokları çalıştırıldığında öğrenciler tarafından aynı anda görüldüğünü görmeleri sağlanır ve öğrencileri tanışma senaryosu için şekil 2 deki gibi kodları hazırlamalarını ve keşfetmelerini sağlayacak ipuçları verilir. Hikaye oluşturulduktan sonra “Bu animasyonu  kod bloğunu kullanmadan nasıl yapabiliriz?” diye sorulur ve gerekli zaman zarfında öğrencilerin düşünceleri sağlanır. Öğrencilere verilen ipuçları sayesinde  Blok 1  Blok 2  Blok 3 Yukarıda verilen kod bloklarını keşfetmeleri sağlanır. Daha sonra bu üç kod bloğunun ne işe yaradığı sorulur ve öğrenciler kod bloklarını denemeleri için yönlendirilir. Kod bloklarından Blok 1’in haber gönderip beklediğini, Blok 2’nin haberi sadece gönderdiğini, Blok 3’te haber geldikten sonra işlem yapacağını keşfettiklerinden emin olunur. Biraz önceki senaryonun bu kod blokları ile nasıl olacağını sorulur ve gerekli zamanı vererek öğrencilerin denemesi ve şekil 3’ teki gibi kod blokları ile başlamaları sağlanır.  Şekil 3 Daha sonra öğrencilere, “Kuklalara nasıl kod yazıyoruz, sahnelere de kod yazarak değiştirebilir miyiz?” diye sorulur ve birden fazla sahne eklemeleri söylenir.  kod bloklarını keşfetmeleri sağlanır.
---	---

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Etkinlik bittikten sonra öğrencilerden bu defa    bloklarını kullanarak iki insanın kılık değiştirerek ve sahne değiştirerek birbiri ile konuşmalarını sağlayan projeyi yapmaları istenir. Projenin yapım aşamasında gerekli yerler de öğrencilere destek olunur. Ders tekrar edildikten sonra bitirilir.
---	---

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	Dördüncü hafta ders planı

Şekil 2.8: Scratch kodlama ortamında kazanım bazlı örnek ders planı



IV.BÖLÜM

4. Bulgular ve Yorum

Bu bölümde toplanan verilerin analizine yer verilmiştir. Veriler SPSS programı kullanılarak analiz edilmiştir. Her bir araştırma problemine yönelik olarak, BİDT, BPTÖ ve KBT sonuçlarına ait betimsel istatistikler sunulacak ve ardından çıkarımsal istatistiklere yer verilecektir.

4.1 BİLGİ İŞLEMSEL DÜŞÜNME TESTİNE YÖNELİK BULGULAR

Scratch, mBlock ve Python grubunda bulunan öğrencilerin BİDT ön test ve son test düzeylerini belirlemek üzere betimsel istatistiklerden faydalanılmıştır. Tablo 4.1’de BİDT’ye yönelik betimsel istatistikler sunulmuştur. İlk olarak Scratch grubunda bulunan öğrencilerin ön test ortalama BİDT puanları, $\bar{X}$ = 15.17 olarak bulunmuştur. Scratch grubunda bulunan öğrencilerin son test ortalama BİDT puanları, $\bar{X}$ = 16.09 olarak bulunmuştur. İkinci olarak mBlock grubunda bulunan öğrencilerin ön test ortalama BİDT puanları, $\bar{X}$ = 13.66 bulunmuştur. mBlock grubunda bulunan öğrencilerin son test ortalama BİDT puanları, $\bar{X}$ = 15.97 olarak bulunmuştur. Son olarak Python grubunda bulunan öğrencilerin ön test ortalama BİDT puanları, $\bar{X}$ = 11.80 bulunmuştur. Python grubunda bulunan öğrencilerin son test ortalama BİDT puanları, $\bar{X}$ = 13.67 bulunmuştur. Bütün gruplarda son test BİDT puanları ön test BPTÖ puanlarına göre daha yüksektir. Uygulama öncesinde ve sonrasında elde edilen BİDT puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama Scratch grubuna ardından mBlock grubuna ve son olarak en düşük ortalama Python grubuna aittir.

Tablo 4.1: BİDT'ye ait betimsel istatistikler

Grup	Test	N	$\bar{X}$	SS	Ortanca	Mod	Mak	Min	Basıklık	Çarpıklık
Scratch	Ön test	35	15.17	3.82	16	18	23	7	-0.587	-0.059
	Son test	32	16.09	4.82	16	16	23	5	-0.347	-0.487
mBlock	Ön test	35	13.66	3.77	13	13	22	6	-0.365	0.406
	Son test	32	15.97	3.59	16	16	23	9	-0.803	0.000
Python	Ön test	30	11.80	4.80	12	9	21	5	-1.071	0.427
	Son test	30	13.67	4.33	14	14	23	5	-0.274	0.427

Scratch, mBlock ve Python gruplarında bulunan öğrencilerin BİDT ön test puanları arasında fark olup olmadığını sınamak üzere ilişkisiz örneklem için tek yönlü varyans analizi yapılmıştır. Asıl analiz yapılmadan önce ilişkisiz örneklem için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.1'de BİDT ön test puanlarının her bir grup içerisinde normal dağıldıklarını işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.1 incelendiğinde ortalama, ortanca ve mod değerlerinin birbirlerine çok yakın olduğu ve basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir. Tüm bu göstergelere ek olarak Kolmogorov-Smirnov testi uygulanmıştır. Tablo 4.2 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.2. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.138	35	0.200
mBlock	0.180	34	0.167
Python	0.153	30	0.069

Scratch, mBlock ve Python gruplarının varyansların eşit olup olmadığı Levene testi ile sınanmıştır. Levene testine göre $p > 0.05$ olduğu için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir. Levene testinin sonuçları Tablo 4.3'te sunulmuştur. Ayrıca öğrenciler BİDT'yi bireysel olarak tamamlamışlardır ve bu süreçte birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.3. Levene testi sonucu

Levene İstatistik	sd1	sd2	p
2.113	2	96	0.126

Grupların ortalamaları arasında anlamlı fark olup olmadığını ortaya koyan ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları Tablo 4.4'te sunulmuştur. Analiz sonucunda Scratch grubunda bulunan öğrencilerin ortalaması ($\bar{X} = 15.17$), mBlock grubunda bulunan öğrencilerin ortalaması ($\bar{X} = 13.65$) ve Python grubunda bulunan öğrencilerin ortalamasının ($\bar{X} = 11.80$) en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir [$F_{(2-96)} = 5.35$, $p < 0.05$]. Test sonucunda hesaplanan etki büyüklüğü ($\eta^2 = 0.10$) bu farkın orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Yapılan Tukey çoklu karşılaştırma testi sonucunda anlamlı farkın Scratch ve Python grubunda bulunan öğrencilerin puanları arasında olduğu görülmüştür.

Tablo 4.4. ANOVA sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	Anlamlı Fark
Gruplar arası	183.636	2	91.818			
Gruplar içi	1647.536	96	17.162	5.35	0.006	Scratch-Python
Toplam	1831.172	98				

Uygulanan üç farklı öğretim yönteminin öğrencilerin, BİDT ön test puanları kontrol edilerek, son test puanlarına etkisinin incelenmesi için kovaryans analizi yapılmıştır. Asıl teste geçmeden önce kovaryans analizinin yapılabilmesi için varsayımların sağlanıp sağlanmadığı kontrol edilmiştir. Öncelikle her bir öğrenci sadece bir grupta yer aldığı için ortalamaları kıyaslanacak grupların birbirlerinden bağımsız olduğu söylenebilir.

Tablo 4.1’de BİDT son test puanlarının her bir grup içerisinde normal dağıldıklarına işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.1 incelendiğinde ortalama, ortanca ve mod değerlerinin birbirlerine çok yakın olduğu ve basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir. Ek olarak normallik varsayımını test etmek üzere Kolmogorov-Smirnov testi uygulanmıştır. Tablo 4.5 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.5. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.135	32	0.145
mBlock	0.116	32	0.200

Python	0.131	30	0.198
--------	-------	----	-------

Gruplarda BİDT son test değişkeni açısından grup varyanslarının homojenliği Levene testi sonuçları ile belirlenmiştir. Levene testi sonuçları Tablo 4.6’da sunulmuştur. Levene testine göre $p>0.05$ olduğu için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir.

Tablo 4.6. Levene testi sonucu

Levene İstatistik	sd1	sd2	p
1.898	2	91	0.156

Öğrencilerin BİDT son test puanları ile ön test puanları arasında doğrusal bir ilişki bulunup bulunmadığını kontrol etmek için saçılma diyagramları oluşturulmuştur ve basit doğrusal regresyon analizi uygulanmıştır. Sonuçlar incelendiğinde Scratch grubu BİDT son test ve BİDT ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.32$) gözlenmiştir [$F_{(1-30)}=14.18$, $p<0.05$]; mBlock grubu BİDT son test puanları ile BİDT ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.28$) gözlenmiştir [$F_{(1-30)}=11.83$, $p<0.05$]; ve son olarak Python grubu BİDT son test puanları ile BİDT ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.57$) gözlenmiştir [$F_{(1-28)}=36.80$, $p<0.05$]

Kovaryans analizinin yapılması için son olarak gruptaki regresyon doğrularının eğimlerinin homojen olup olmadığını test edilmiştir. Yapılan analiz sonucunda $p>0.05$ ($p=0.514$) bulunmuştur. Bu sonuca göre regresyon doğrularının eğimleri homojendir denilebilir.

Kovaryans analizinin bütün varsayımlarının test edilmesinin ardından asıl teste geçilmiştir. Kovaryans analiz sonuçlarına göre grupların BİDT ön test puanına göre düzeltilmiş BİDT son test puanlarının ortalamaları arasında anlamlı bir fark

gözlemlenmemiştir [$F_{(2-90)}=0.668$, $p>0.05$]. Kovaryans analizinin sonuçları Tablo 4.7’de sunulmuştur.

Tablo 4.7. Kovaryans analizi sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p
BİDT ön test	636.267	1	636.267	55.917	0.000
Grup	15.204	2	7.602	0.668	0.515
Hata	1024.088	90	11.379		
Toplam (düzeltilmiş)	1774.809	93			

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan BİDT sonuçlarının ortalamaları arasında anlamlı bir fark olup olmadığını test etmek için ilişkili örneklemeler için t-testi yapılmıştır. Esas analizlere geçmeden önce ilişkili örneklemeler için t-testinin varsayımları kontrol edilmiştir. Öncelikle ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılmalıdır. Tablo 4.8’de BİDT son test - ön test fark puanlarının her bir grup içerisinde normal dağıldıklarını gösteren basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.8 incelendiğinde basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir.

Tablo 4.8. Fark puanlarının basıklık ve çarpıklık değerleri

Grup	N	Basıklık	Çarpıklık
Scratch	32	0.380	0.155
mBlock	32	0.530	1.87
Python	30	0.167	0.370

Normallik varsayımını test etmek için ayrıca Kolmogorov-Smirnov testi yapılmıştır. Tablo 4.9’da testin sonuçları sunulmuştur. Tablo 4.9 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir. Normal dağılım varsayımına ek olarak fark puanlarının birbirinden bağımsız olduğu varsayılmıştır.

Tablo 4.9. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.134	32	0.152
mBlock	0.145	32	0.083
Python	0.128	30	0.200

İlişkili örneklem t-test sonuçları Tablo 4.10’da sunulmuştur. İlişkili örneklem t-testleri sonucunda Python grubunun ön test ($\bar{X}=11.80$) ve son test ($\bar{X}=13.67$); ve mBlock grubunun ön test ($\bar{X}=13.84$) ve son test ($\bar{X}=15.97$) puanları arasında anlamlı bir fark bulunmuştur. Buna karşılık olarak Scratch grubunun ön test ($\bar{X}=15.00$) ve son test puanları ($\bar{X}=16.09$) arasında anlamlı bir fark bulunamamıştır. Test sonucunda hesaplanan etki büyüklükleri mBlock grubu için 0.59 ve Python grubu için 0.58 olarak bulunmuştur. Bu değerler orta büyüklükte olarak yorumlanabilir.

Tablo 4.10. İlişkili örneklem t-test sonuçları

BİDT	N	$\bar{X}$	SS	sd	t	p
Scratch Ön	32	15.00	3.66	31	1.52	0.139
Scratch Son	32	16.09	4.82			
mBlock Ön	32	13.84	3.86	31	3.33	0.002
mBlock Son	32	15.97	3.59			
Python Ön	30	11.80	4.80	29	3.16	0.004
Python Son	30	13.67	4.33			

4.2 BİLGİSAYAR PROGRAMLAMAYA KARŞI TUTUM ÖLÇEĞİNE YÖNELİK BULGULAR

Çalışmaya katılan öğrencilerin BPTÖ ön test ve son test düzeylerini belirlemek üzere betimsel istatistikler kullanılmıştır. Tablo 4.11’de BPTÖ’ye yönelik betimsel istatistikler sunulmuştur. Scratch grubunda bulunan öğrencilerin ön test ortalama BPTÖ puanları, $\bar{X}$ = 47.47 olarak bulunmuştur. Scratch grubunda bulunan öğrencilerin son test ortalama BPTÖ puanları, $\bar{X}$ = 45.47 olarak bulunmuştur. Scratch grubunda bulunan öğrencilere benzer bir şekilde Python grubunda bulunan öğrencilerin ön test ortalama BPTÖ puanları, $\bar{X}$ = 48.07 olarak bulunmuştur. Python grubunda bulunan öğrencilerin son test ortalama BPTÖ puanları, $\bar{X}$ = 45.52 olarak bulunmuştur. mBlock grubunda bulunan öğrencilerin ön test ortalama BPTÖ puanları, ($\bar{X}$ = 50.28) ile son test ortalama BPTÖ puanları, ($\bar{X}$ = 53.03) arasında, Scratch ve Python grubundan farklı olarak, küçük bir artış bulunmaktadır. BPTÖ’den alınabilecek minimum puan 13, maksimum puan 65’tir Bu veriler düşünüldüğünde uygulama öncesinde bütün grupların BPTÖ ön test puanlarının yüksek olduğu söylenebilir.

Tablo 4.11. BPTÖ'ye ait betimsel istatistikler

Grup	Test	N	$\bar{X}$	SS	Ortanca	Mod	Mak	Min	Basıklık	Çarpıklık
Scratch	Ön test	34	47.47	10.37	47.5	47	64	27	-0.660	-0.387
	Son test	34	45.47	10.78	45.0	39	65	21	0.070	-0.143
mBlock	Ön test	32	50.28	8.56	51.50	45	65	30	-0.465	-0.704
	Son test	32	53.03	9.02	53.0	65	65	35	-0.302	-0.372
Python	Ön test	30	48.07	9.53	47.5	55	65	28	-0.795	-0.489
	Son test	31	45.52	12.10	48.0	51	65	14	-0.027	-0.471

Scratch, mBlock ve Python gruplarında bulunan öğrencilerin BPTÖ ön test puanları arasında fark olup olmadığını sınamak üzere ilişkisiz örneklem için tek yönlü varyans analizi yapılmıştır. Asıl analiz yapılmadan önce ilişkisiz örneklem için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.11'de BPTÖ ön test puanlarının her bir grup içerisinde normal dağıldıklarına işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.11 incelendiğinde ortalama, ortanca ve mod değerlerinin birbirlerine çok yakın olduğu ve ayrıca basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir. Bunlara ek olarak Tablo 4.12 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.12. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.091	34	0.200
mBlock	0.138	32	0.128
Python	0.131	30	0.200

Scratch, mBlock ve Python gruplarının varyansların eşit olup olmadığı Levene testi ile sınanmıştır. Levene testine göre $p > 0.05$ olduğu için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir. Levene testinin sonucu tablo 4.13'te sunulmuştur. Ayrıca öğrenciler BPTÖ'yü bireysel olarak tamamlamışlardır ve bu süreçte öğrencilerin birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.13. Levene testi sonucu

Levene İstatistik	sd1	sd2	p
0.531	2	93	0.590

Grupların ön test BPTÖ puan ortalamaları arasında anlamlı fark olup olmadığını ortaya koymak üzere ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları tablo 4.14'te sunulmuştur. Analiz sonucunda Scratch grubunda bulunan öğrencilerin ortalaması, ($\bar{X} = 47.47$), MBlock grubunda bulunan öğrencilerin ortalaması, ($\bar{X} = 50.28$) ve Python grubunda bulunan öğrencilerin ortalamasının, ($\bar{X} = 48.07$) en az ikisi arasında istatistiksel olarak anlamlı bir fark gözlenmemiştir [$F_{(2-93)} = 0.783$, $p > 0.05$].

Tablo 4.14 ANOVA sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P
Gruplar arası	142.350	2	71.175		
Gruplar içi	8452.806	93	90.890	0.783	0.460
Toplam	8595.156	95			

Uygulanan üç farklı öğretimin öğrencilerin, BPTÖ ön test puanları kontrol edilerek, son test puanlarına etkisinin incelenmesi için kovaryans analizi yapılmıştır. Asıl teste geçmeden önce kovaryans analizinin yapılabilmesi için varsayımların sağlanıp sağlanmadığı kontrol edilmiştir. Öncelikle her bir öğrenci sadece bir grupta yer aldığı için ortalamaları kıyaslanacak grupların birbirlerinden bağımsız olduğu söylenebilir.

Tablo 4.11’de BPTÖ son test puanlarının her bir grup içerisinde normal dağıldıklarını gösteren ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.11 incelendiğinde ortalama, ortanca ve mod değerlerinin birbirlerine çok yakın olduğu ve ayrıca basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir. Buna ek olarak Tablo 4.15 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.15. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.098	34	0.200
mBlock	0.093	32	0.200
Python	0.126	31	0.200

Bütün gruplar için BPTÖ son test değişkeninin varyanslarının homojenliği Levene testi sonuçları ile belirlenmiştir. Levene testi sonuçları Tablo 4.16'da sunulmuştur. Levene testine göre $p>0.05$ olduğu için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir.

Tablo 4.16. Levene testi sonucu

Levene İstatistik	sd1	sd2	p
3.039	2	94	0.053

Öğrencilerin BPTÖ son test puanları ile ön test puanları arasında doğrusal bir ilişki bulunup bulunmadığını kontrol etmek için saçılma diyagramları oluşturulmuştur ve basit doğrusal regresyon analizi uygulanmıştır. Sonuçlar incelendiğinde Scratch grubu son test BPTÖ puanları ile ön test BPTÖ puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.43$) gözlenmiştir [$F_{(1-31)}=23.32$, $p<0.05$]; MBlock grubu son test BPTÖ puanları ile ön test BPTÖ puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.31$) gözlenmiştir [$F_{(1-30)}=13.31$, $p<0.05$]; ve son olarak Python grubu son test BPTÖ puanları ile ön test BPTÖ puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.21$) gözlenmiştir [$F_{(1-28)}=7.24$, $p<0.05$]

Kovaryans analizine geçmeden önce, regresyon doğrularının eğimlerinin homojen olup olmadığı test edilmelidir. Yapılan analiz sonucunda $p>0.05$ ($p=0.917$) bulunmuştur. Bu sonuca göre regresyon doğrularının eğimlerinin homojen olduğu söylenebilir.

Kovaryans analizinin varsayımlarının test edilmesinin ardından, kovaryans analizi elde edilen veriler üzerinde uygulanmıştır. Kovaryans analiz sonuçlarına göre grupların BPTÖ ön test puanına göre düzeltilmiş BPTÖ son test puanlarının ortalamaları arasında anlamlı bir fark gözlemlenmiştir [$F_{(2-91)}=4.703$, $p<0.05$]. Kovaryans analizinin sonuçları Tablo 4.17'de sunulmuştur. Test sonucunda hesaplanan etki büyüklüğü ($\eta^2=0.094$) bu farkın küçük düzeyde olduğunu göstermektedir.

Tablo 4.17. Kovaryans analizi sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p
BİDT ön test	3384.04	1	3384.04	39.755	0.000
Grup	800.586	2	400.293	4.703	0.011
Hata	7746.113	91	85.122		
Toplam (düzeltilmiş)	230239.0	95			

Grupların düzeltilmiş tutum puanları arasındaki farkları ortaya koymak için Bonferroni çoklu karşılaştırma testi uygulanmıştır. Bonferroni çoklu karşılaştırma sonuçlarına göre mBlock grubunda bulunan öğrencilerin BPTÖ puanları ($\bar{X}$ =51.95) Scratch grubunda bulunan öğrencilerin BPTÖ puanlarından ($\bar{X}$ =46.31) ve Python grubunda bulunan öğrencilerin BPTÖ puanlarından ($\bar{X}$ =45.28) anlamlı derecede yüksektir. Scratch ve Python gruplarında bulunan öğrencilerin BPTÖ puanları arasında anlamlı bir fark bulunmamıştır.

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan BPTÖ sonuçlarının ortalamaları arasında anlamlı bir fark olup olmadığını test etmek için ilişkili örneklem için t-testi yapılmıştır. Esas analizlere geçmeden önce ilişkili örneklem için t-testinin varsayımları kontrol edilmiştir.

Ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılmalıdır. Tablo 4.18’de BPTÖ son test - ön test fark puanlarının her bir grup içerisinde normal dağıldıklarını gösteren basıklık ve çarpıklık değerleri verilmiştir. Tablo 24 incelendiğinde basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir.

Tablo 4.18. Fark puanlarının basıklık ve çarpıklık değerleri

Grup	N	Basıklık	Çarpıklık
Scratch	33	-0.259	-0.540
mBlock	32	0.376	-0.395
Python	30	-0.691	-0.075

Normallik varsayımını test etmek için ayrıca Kolmogorov-Smirnov testi yapılmıştır. Tablo 4.19’da testin sonuçları sunulmuştur. Tablo 4.19 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir. Normallik varsayımına ek olarak fark puanlarının birbirinden bağımsız olduğu varsayılmıştır.

Tablo 4.19. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.107	33	0.200
mBlock	0.098	32	0.153
Python	0.141	30	0.131

İlişkili örneklem t-test sonuçları Tablo 4.20’de sunulmuştur. İlişkili örneklem t-testleri sonucunda Python grubunun ön test ($\bar{X}$ =48.07) ve son test ($\bar{X}$ =44.97); MBlock grubunun ön test ($\bar{X}$ =50.28) ve son test ($\bar{X}$ =53.03); ve Scratch grubunun ön test ($\bar{X}$ =47.36) ve son test ($\bar{X}$ =45.55) puanları arasında anlamlı bir fark bulunamamıştır.

Tablo 4.20. İlişkili örneklem t-test sonuçları

BPTÖ	N	$\bar{X}$	SS	sd	t	p
Scratch Ön	33	47.36	10.50	32	1.17	0.250
Scratch Son	33	45.55	10.93			
mBlock Ön	32	50.28	8.56	31	-1.87	0.071
mBlock Son	32	53.03	9.02			
Python Ön	30	48.07	9.53	29	1.41	0.169
Python Son	30	44.97	12.84			

4.3 KODLAMA BAŞARI TESTİNE YÖNELİK BULGULAR

Bu çalışmada öğrenciler ortaokul seviyesinde ilk defa programlamaya yönelik ders almışlardır. Öğrenciler hem programlama ortamına hem de programlama kavramlarına hâkim olmadıklarından KBT ön test olarak uygulanmamıştır. KBT sadece son test olarak uygulanmıştır. Her bir grupta bulunan öğrencilerin başarı düzeylerini incelemek için KBT'ye yönelik betimsel istatistiklerden faydalanılmıştır. Tablo 4.21'de KBT'ye yönelik betimsel istatistikler sunulmuştur. Tablo 4.21 incelendiğinde uygulama sonrasında Scratch grubunun KBT ortalaması $\bar{X}$ =54.57, mBlock grubunun KBT ortalaması $\bar{X}$ =53.09 ve Python grubunun KBT ortalaması 27.58 olarak bulunmuştur. Göreceli olarak değerlendirildiğinde Scratch ve mBlock gruplarının KBT ortalamaları birbirine yakın ve Python grubunun KBT ortalamasından oldukça yüksek olduğu gözlenmektedir.

Tablo 4.21. KBT'ye ait betimsel istatistikler

Grup	N	$\bar{X}$	SS	Ortanca	Mod	Mak	Min	Basıklık	Çarpıklık
Scratch	35	54.57	9.58	55	55	70	35	-0.588	-0.430
mBlock	34	53.09	9.05	55	55	70	30	0.201	-0.496
Python	31	27.58	12.44	25	20	55	10	-0.746	0.403

Scratch, mBlock ve Python gruplarında bulunan öğrencilerin uygulama sonrasında uygulanan kodlama başarı test puanları arasında fark olup olmadığını sınamak için ilişkisiz örneklemeler için tek yönlü varyans analizi yapılmıştır. Asıl analiz yapılmadan önce ilişkisiz örneklemeler için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.21'de kodlama başarı test puanlarının her bir grup içerisinde normal dağıldıklarını gösteren ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.21 incelendiğinde ortalama, ortanca ve mod değerlerinin birbirlerine çok yakın olduğu ve ayrıca basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir. Bunlara ek olarak, normallik varsayımını test etmek için Kolmogorov-Smirnov testi uygulanmıştır. Kolmogorov-Smirnov testinin sonuçları Tablo 4.22'de sunulmuştur. Tablo 4.22 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.22. Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
Scratch	0.146	35	0.055
mBlock	0.143	34	0.078

Python	0.148	31	0.081
--------	-------	----	-------

Scratch, mBlock ve Python gruplarında bulunan öğrencilerin KBT puanları varyanslarının eşit olup olmadığını test etmek için Levene testi uygulanmıştır. Levene testine göre $p < 0.05$ bulunmuştur. Bu sonuca göre üç grubun varyanslarının eşit olduğu söylenemez. Levene testinin sonuçları tablo 4.23’de sunulmuştur. Ayrıca öğrenciler kodlama başarı testini bireysel olarak tamamlamışlardır ve bu süreçte birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.23. Levene testi sonucu

Levene İstatistik	sd1	sd2	p
3.312	2	97	0.041

Grupların ortalamaları arasında anlamlı fark olup olmadığını belirlemek üzere ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları tablo 4.24’te sunulmuştur. Analiz sonucunda Scratch grubunda bulunan öğrencilerin ortalaması, ($\bar{X} = 54.57$), mBlock grubunda bulunan öğrencilerin ortalaması, ($\bar{X} = 53.09$) ve Python grubunda bulunan öğrencilerin ortalamasının, ($\bar{X} = 27.58$) en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir [$F_{(2-97)} = 68.55$, $p < 0.05$]. Test sonucunda hesaplanan etki büyüklüğü, ($\eta^2 = 0.59$) bu farkın geniş düzeyde olduğunu göstermektedir. Yapılan Dunnett C çoklu karşılaştırma testi sonucunda anlamlı farkın Scratch ile Python grubunda ve mBlock ile Python grubunda bulunan öğrencilerin KBT puanları arasında olduğu görülmüştür.

Tablo 4.24. ANOVA sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	Anlamlı Fark
Gruplar arası	14788.145	2	7394.072	68.55	0.00	Scratch-
Gruplar içi	10462.855	97	107.864			Python;
Toplam	25251.000	99				MBlock-Python

V.BÖLÜM

5. Sonuç ve Öneriler

5.1. SONUÇ

Bu araştırma 2018-2019 Eğitim-Öğretim yılında Bolu ili merkez ilçesinde bulunan bir ortaokulda 3 farklı grupta bulunan 105 öğrenci (45 kız ve 60 erkek öğrenci) ile gerçekleştirilmiştir. Öğretim öncesinde öğrencilere ön test olarak BİDT ve BPTÖ uygulanmıştır. Bolu ili merkez ilçesinde bulunan ortaokulda altı ayrı 6. sınıf içerisinde üçü rastgele seçilmiştir. Bu üç sınıfa rasgele Scratch, mBlock ve Python grupları olarak atanmıştır. Her grup için daha önce tasarlanan öğretim programı 8 hafta boyunca haftada 2 saat olmak üzere uygulanmıştır. Dersler BT sınıflarında yapılmıştır ve iki öğrencinin bir bilgisayarda oturması sağlanmıştır. Öğrenciler BT sınıfında eşli programlama yaparak belirli süreler sonunda rolleri değiştirmişlerdir. 8 haftanın sonunda son test olarak BİDT, BPTÖ ve KBT öğrencilere uygulanmıştır.

5.1.1 KBT ile ilgili sonuçlar

Öğrencilerin son test KBT puanları karşılaştırıldığında Scratch ve mBlock gruplarının benzer sonuçlar ürettiği ve bu sonuçların Python grubu sonuçlarından farklı olduğu anlaşılmıştır. Scratch ve mBlock grubu KBT puanları arasında anlamlı bir fark bulunamazken. Scratch ve Python grubu KBT puanları arasında Scratch grubu lehine anlamlı bir fark bulunmuştur. Benzer şekilde mBlock ve Python grubu KBT puanları arasında mBlock grubu lehine anlamlı bir fark bulunmuştur. Bu sonuçlara göre Scratch ve mBlock gruplarında verilen öğretimin öğrencilerin KBT puanlarını geliştirmede Python

grubunda verilen öğretime göre daha başarılı olduğu ve Scratch ve mBlock gruplarının bu açıdan anlamlı olarak farklılaşmadığı söylenebilir.

Bu araştırmada elde edilen sonuçlara benzer sonuçları olan birçok araştırma bulunmaktadır. Ozoran, Çağıltay, ve Topalli, (2012) yaptıkları araştırmada mühendislik öğrencileri ile programlama dersi kapsamında Scratch ve C dillerini beraber işlemişlerdir. Araştırmanın sonucunda öğrencilerin C diline göre Scratch blok tabanlı programlama dilinin programlamaya girişteki temeli öğretmede daha etkili olduğunu, Scratch ile çalışmanın eğlenceli ve keyifli olduğunu ve oynanılan oyunların çalışma mantığını anlamada yol gösterdiği sonuçlarına varılmıştır. Meerbaum-Salant, Armoni, ve Ben-Ari, (2013) yaptıkları araştırmada Scratch programını kullanarak lise öğrencilerinin bilgisayar bilimleri ile alakalı kavramları öğretmeyi incelemişlerdir. Çalışmalarının sonucunda öğrencileri Scratch ile ilgili programların zevkli ve öğrenmeye motive ettiğini ifade etmişlerdir. Görsel programlama dillerinde birçok medya araçlarını kullanarak tasarlanan oyunlardan dolayı öğrenme ve tasarım beraber işleyen bir süreç haline gelip görsel tasarım odaklı bir yaklaşım kullanılabilmektedir (Yükseltürk ve Altıok, 2017). Yine benzer bir çalışma Maloney ve diğerleri (2010) yaptıkları araştırma da 8-16 öğrencilerinin Scratch kullanarak animasyonlu hikayeler yapması, programlama ve algoritma geliştirmede olumlu etkiye sahip olduğunu söylemişlerdir. Ozorana ve diğerleri (2012) yaptıkları araştırmada üniversite öğrencilerine programlama başlangıcı eğitimi için Scratch ile öğretim vermiş ve öğretimden sonra Scratch programlama dilinin programlama eğitiminde olumlu katkılarının olduğunu söylemişlerdir. Ancak literatürde bu araştırmanın aksine olan çalışmalar da mevcuttur. Mıhçı (2014) yaptığı çalışmadan görsel programlama ortamı olan App Inventor ile öğrencilere blok tabanlı programlama eğitimi vermiş ama değerlendirmeler sonucunda öğrencilerin başarısına olumlu bir katkısının olmadığını söylemiştir. App Inventor uygulamasında çıkan olası teknik sorunlar öğrencilerin motivasyonlarını düşürmüş olabilir. Öğrencilere uygulanan etkinliklerden ve etkinlik sürelerinden kaynaklanan bir motivasyon etksikliği olabilir. Alanyazın tarandığında Scratch blok tabanlı programlama dilinin benzer şekilde sonuçlarının göze çarptığı görülmektedir. Scratch blok tabanlı prgramlama dilinin programlama öğretiminde başarılı olmasının sebepleri arasında kod bloklarının hazır olması ve hızlı bir şekilde hikaye, animasyon, oyun oluşturmanın mümkün olması

söylenebilir. Öğrencilerin scratch programlama dilinde somut olan kodlarla yaptıkları somut animasyon, oyun vb. etkinlikler derse karşı daha motive edici olabilir.

Oppennikova, Ershow, ve Iljin, (2015) üç yıl boyunca yaptıkları araştırmada robotik kodlama yapan öğrencilerin, teknik bilgi ve beceri bakımından geliştikleri, teknolojik yeterliliğe ulaştıkları görülmüştür. Robotik, öğrencilerin bilgisayar bilimleri ile alakalı konuları somut olarak test etmesine yardımcı olur. Robotik , öğrencilerin motor becerileri, uygulama ve problem çözme becerilerinin gelişmesine katkıda bulunabilir (Bers, Flannery, Kazakoff, ve Sullivan, 2014). Metin tabanlı programlama ortamları soyut ifadelerden oluştuğu için küçük yaşta öğrenilmesi zordur. Ancak somut ortamlar ile programlamayı öğretmek daha kolay olabilir (Wyeth, 2008). Erken yaştaki bireylere programlama eğitimi için daha somut kavram olan robotik kodlama gösterilmelidir (Kuzu ve Türk, 2018). Öğretimde kullanılan robotların öğrencilerin ilgisini çekmesinden dolayı programlama eğitimine daha motive oldukları söylenebilir (Alvarez ve Lavranega, 2016; Çankaya, Durak ve Yünkül, 2017; Ersoy ve diğerleri 2011). Korkmaz (2016) yaptığı araştırmada içerisinde robot tasarımı faaliyetleri içeren uygulamaların , blok tabanlı uygulamalara göre öğrencilerin programlamaya yönelik başarılarının arttığını söylemektedir. Sonuç olarak alanyazında bulunan araştırmaların geneli bu çalışmanın sonuçlarına paralellik göstermektedir. Buradan yola çıkarak öğrencilerin robotik kodlama ile daha başarılı olmalarının sebepleri aşağıdaki maddeler ile ifade edilebilir:

- Öğrencilerin yazdıkları kodların sonuçlarını somut olarak görmeleri ve robotlara dokunmaları,
- Öğrencilerin robotları günlük hayattaki problemlere transfer ederek motive olmaları,
- Öğretmenin aktif olduğu klasik bir dersten, öğrencinin aktif olduğu bir derse geçmenin öğrencilerin başarılarında etkili olduğu söylenebilir.

Berry, (2014) erken yaşta programlama eğitimi gören öğrencilerin kullanacakları uygulamaların; Logo,Small Basic,Python dilleri olabileceğini belirtmiştir. Fakat küçük

yaşta programlama eğitimi alan öğrencilerin yaşadığı sorunların birkaç nedeni vardır. Bunlar;

Söz Dizimi: Program yazarken alanda uzman olmak kısa sürede başarılacak bir durum değildir. Öğretmenlerin bir çoğu derslerinin uzun zamanını öğrencilerin söz dizimi hatalarını gidermekle harcamaktadırlar.

İlerleme: Öğrencilerin programlama eğitiminin devamı olmayabilir. Öğrenciler ilerleyen zamanlarda programlama eğitiminde kullanılacak başka dillere geçiş yapamayabilir.

Wang, Mei, Lin, Chiu, ve Lin, (2009) yaptıkları araştırmada Alice ve C++ programlama ortamlarında öğrencilerin başarılarını ölçmüş, yapılan değerlendirmeler sonucunda Alice ortamında eğitim görenlerin daha başarılı olduğunu söylemiştir. Alanyazında tarama yapıldığında bizim araştırmamızla paralel sonuçların olduğu görülmektedir. Python grubunun KBT sonuçlarının Scratch ve mBlock gruplarına göre oldukça düşük çıktığı görülmektedir. Öğrencilerin ilk defa metin tabanlı bir programlama dili görmesinin ve alışma sürecinin uzun sürmesinin öğrenci başarısını olumsuz yönde etkilemiş olabilir. Metin tabanlı programlamada, blok tabanlı programlama diline göre kodları yazmak öğrenciler için daha zor ve kodların yazımında hata yapma olasılığı fazla olacağından öğrencilerin programlama başarısının olumsuz olarak etkilendiği söylenebilir. Metin tabanlı programlama dilinde söz dizim hataları, kodların tamamını yazmadaki zaman alımı, öğrencilerin ve öğretmenin dersin zamanının çoğu kısmını metin tabanlı programlama dilindeki hataları ayıklamakla geçmesi öğrencilerin derse olan motivasyon ve başarısını azaltmış olabilir.

Python gurubunda bulunan öğrencilerin BİDT puanları artmış olmasına rağmen KBT puan ortalamaları anlamlı olarak Scratch ve mBlock gurubundan düşük çıkmıştır. Öğrencilerin Python derslerinde yaşadıkları sıkıntıları çözmeleri onların problem çözme becerilerini ve sonuç olarak BİD becerilerini geliştirmiş olabilir. Fakat 8 hafta gibi kısa süreli yapılan derste öğrencilerin yaşadığı sıkıntılar onların başarılarına negatif yönde etkide bulunmuş olabilir.

5.1.2. BPTÖ ile ilgili sonuçlar

Yapılan kovaryans analizi sonucunda Scratch, mBlock ve Python gruplarının en az ikisinde bulunan öğrencilerin BPTÖ puanları arasında anlamlı bir fark bulunmuştur. MBlock grubunda bulunan öğrencilerin BPTÖ puanları Scratch ve Python grubunda bulunan öğrencilerin BPTÖ puanlarına göre mBlock grubu lehine olacak şekilde anlamlı farklılık göstermiştir. Scratch ve Python grupları arasında BPTÖ puanları açısından anlamlı bir fark bulunmamaktadır. Öğrencilerin BPTÖ puanları her grup içerisinde ön test ve son test açısından kendi içlerinde karşılaştırıldığında hiçbir grubun ön test ve son test puanları arasında anlamlı bir fark bulunamamıştır. Bu sonuçlara dayalı olarak hiçbir grupta verilen eğitimin BPTÖ puanlarını geliştirme açısından anlamlı bir fark yaratmadığı ve BPTÖ puanları açısından mBlock grubunda verilen öğretimin Scratch ve Python grubuna göre daha faydalı olduğu söylenebilir. Alanyazında bu bulgunun aksine sonuçlar bulunmaktadır. Gülbahar ve Kalelioğlu, (2014) Scratch görsel programlama dili kullanmanın ve o dili kolayca öğrenmenin öğrencilerin programlama dillerine olan ilgisini olumlu yönde değiştirdiğini söylemişlerdir. Bu çalışmaya paralel olarak Kopsiripat (2015), yaptığı çalışmanın sonuçları da paralellik göstermektedir. Yapılan çalışmaya paralel olarak öğretmen adaylarına yönelik yapılan araştırmada programlama derslerinde Scratch kullanımının derse olan tutumu olumlu yönde etkilediği sonucuna ulaşmışlardır. Alanyazında öğrencilerin tutumunun olumlu yönde değişmesinin sebepleri, kullanılan etkinlikler olabilir . (Fesakis ve Serafeim, 2009).

Korkmaz (2016) yaptığı çalışmada Scratch görsel programlama dilinin öğrencilerin programlamaya karşı olan tutumlarını değiştirmedini belirtmiştir. Gülbahar ve Kalelioğlu, (2014) yaptıkları çalışmadan Scrath programlama dilinin öğrencilerin programlamaya olan motivasyonlarına yönelik herhangi bir etkiye sahip olmadıkları sonucuna varmışlardır. Hutchinson, Moskal, Daan, ve Cooper, (2008) yaptıkları araştırmada bir başka görsel programlama dili olan Alice kullanmışlardır. Yalnız Alice kullanımının öğrencilerin programlamaya karşı tutumlarına etkisinin olmadığını söylemişlerdir. McGill (2012) robotların öğrencilerin programlamaya karşı

tutumlarını arttırdığını ama ilgi, güven ve memnuniyet değişkenleri üzerinde herhangi bir etkisinin olmadığını söylemiştir.

Bu araştırmada elde edilen BPTÖ sonuçlarına göre öğrencilerin çalışma öncesinde programlamaya karşı tutumlarının yüksek olması dikkat çekmektedir. Halihazırda yüksek olan bir tutum değerini daha yüksek bir seviyeye çıkarmak zor olabilir. Bunun yanında öğretimin süresi çalışmanın tutum açısından sonuçlarını etkileyen bir faktör olabilir. Scratch, Python ve mBlock ortamlarında verilen eğitimler sonunda mBlock lehine sonuçların biraz daha iyi çıkmasının nedeni öğrencilerin robotlarla çalışması, yaptıkları kodlamanın sonuçlarını somut olarak görmesi olabilir.

Scratch ve Python gruplarında öğrencilerin BPTÖ ortalama puanlarında istatistiksel olarak anlamlı fark olmasa da bir düşüş bulunmaktadır. Bu düşüşün sebebi, öğrencilerin karşılaştıkları öğretimin bekledikleri öğretimden farklı olması olabilir. Öğrenciler, dersi almadan önce kodlama eğitimine yönelik olarak pozitif bir algıya sahip olmalarına karşın öğretimin nasıl gerçekleşeceğine yönelik somut algıları bulunmayabilir. Öğrencilerde ders öncesinde bulunan pozitif tutum, yüksek olan beklentilerinin tam olarak karşılanmamasından düşmüş olabilir.

5.1.3. BİDT ile ilgili sonuçlar

Bu çalışmada kovaryans analizi sonucunda Scratch, mBlock ve Python gruplarının BİDT puanları arasında anlamlı bir fark bulunamamıştır. Tasarlanan farklı öğretim ortamları (Scratch, mBlock ve Python) öğrencilerin BİDT puanları açısından anlamlı bir fark yaratamamıştır. Fakat her bir grubun öntest BİDT ve son test BİDT puanları kendi içlerinde karşılaştırıldığında Python ve mBlock gruplarında öntest ve son test BİDT puanları arasında anlamlı bir fark olduğu halde Scratch grubunun öntest ve son test BİDT puanları arasında anlamlı bir fark oluşmadığı görülmüştür. BİDT temel alındığında Python ve mBlock öğretim ortamlarının anlamlı bir fark yaratacak kadar

gelişme sağladığı fakat Scratch öğretim ortamının anlamlı bir fark yaratacak kadar gelişme sağlamadığı söylenebilir.

Bilgi işlemsel düşünme temel alınarak alanyazın incelendiğinde farklı çalışmalardan farklı sonuçlar elde edildiği görülmüştür. Bu çalışmada elde edilen bulguların aksine Moreno-Leon, Robles ve Roman-Gonzalez (2015) Scratch programı ile bilgi işlemsel düşünme becerisini geliştirmeyi amaçladıkları çalışmada öğrencilere geri dönüt vererek çalışmanın öğrencilerde kendiliğinden öğrenme ve problemler karşısında algoritma geliştirmeyi sağladığı sonucu ortaya çıkmıştır. Aynı şekilde Feldhausen, Weese ve Bean, (2018) yaptığı çalışmada programlama eğitimi verilerek BİD becerilerine yönelik öz yeterlik algısına olumlu katkı sağladığı söylenmiştir. Benzer şekilde Djambong ve Freiman, (2016) yaptıkları çalışmada blok tabanlı kodlama yapan öğrenciler ile robotik kodlama yapan öğrenciler arasında BİD becerisi açısından hiçbir fark gözlenmediği sonucuna varmışlardır. Oluk, Korkmaz, ve Oluk, (2018) yaptıkları çalışmada Scratch ile çalışılan grubun bilgi işlemsel düşünme becerisini geliştirdiğini söylemişlerdir. Brennan ve Resnick, (2012) yaptıkları çalışma da online ortamdaki gruplar tarafından hazırlanan Scratch projelerine bakarak öğrencilerin bilgi işlemsel düşünme becerilerini geliştirdiklerini söylemişlerdir.

Scrath görsel programlama dilinin bilgi işlemsel düşünme üzerinde olumlu ya da olumsuz bir etkisinin olmadığı çalışmalar da mevcuttur. Yapılan çalışmada ilköğretim 5. sınıf öğrencilerinin problem çözme becerileri araştırılmış olup açıklayıcı ardışık karma yöntem kullanılarak 49 tane 5. sınıf öğrencisiyle yapılan çalışmada öğrencilerin scratch kullanarak problem çözme becerilerinde herhangi bir değişiklik gözlenmemiştir(Gülbahar ve Kalelioğlu, 2014). Yapılan diğer bir çalışmada görsel programlama dili kullanılarak ortaokul öğrencilerin bilgi işlemsel düşünme becerileri üzerinde bir değişim gözlenip gözlenmediği araştırılmış olup yapılan istatistiki sonuçlar doğrultusunda herhangi bir değişimin olmadığı söylenmiştir. (Uslu, Mumcu ve Eğin, 2018). Bir başka çalışmada 6.sınıf öğrencileri ile yapılan çalışmada kullanılan görsel programlama dilinin öğrencilerin mantıksal sorgulama becerileri üzerinde herhangi bir değişimin olmadığı söylenmiştir. (Lai ve Yang, 2011)

Bu çalışmanın sonuçları Scratch grubunda ön test ve son test BİDT puanları arasında bir fark oluşmadığını göstermektedir. Bunun sebebi bilgi işlemsel düşünme becerisinin geliştirilmesi için sekiz haftalık bir öğretimin yeterli olmaması gösterilebilir. Bilgi işlemsel düşünme bir beceridir ve geliştirilmesi uzun sürebilir. Buna ek olarak Scratch ortamındaki tasarım ile ilgili öğelerin fazlalığı öğrencilerin BİDT puanlarının gelişmemesine sebep olarak gösterilebilir. Scratch ortamında mBlock ve Python ortamlarına göre daha fazla tasarım ögesi bulunmaktadır. Öğrenciler bir oyun veya animasyon tasarlarlarken bu oyunun veya animasyonun bilgi işlemsel düşünme özellikleri ile ilgilenmektense Scratch’de bulunan grafik tasarım ile ilgili özelliklerle daha fazla ilgilenmiş olabilir. Bu da öğrencilerin bilgi işlemsel düşünme becerilerini geliştirmede bir engel oluşturmuş olabilir.

Bu çalışmada elde edilen bulgulara paralel bir şekilde 2016 yılında 44 üniversite öğrencisi ile yapılan robotik kodlama çalışmalarının bilgi işlemsel düşünme becerisini olumlu yönde etkilediği görülmektedir (Vieira, Penmetch, Magana, ve Matson , 2016). Benzer şekilde Durak-Yıldız ve Saritepeci, (2019) bilgi işlemsel düşünme becerisini geliştirmeye yönelik yaptıkları çalışmada öğrencilerin robotik kodlama yaparken problemleri günlük hayatla ilişkilendirmeleri, öğrencilerin robotik kodlama çalışmalarına daha motive oldukları, problemleri çözmek için daha çok düşündükleri ve çaba gösterdikleri sonucuna ulaşmışlardır.

Bu araştırmada elde edilen sonuçlara benzer şekilde Yolcu (2018), yaptığı araştırmada bir grup öğrencinin blok tabanlı programlama dili öğrendiğini, bir grup öğrencinin robotik kodlama yaptığı çalışmada gruplar arasında BİD becerisi ve öz yeterlik bakımından anlamlı bir farklılık gözlenmediğini söylemiştir. Djambong & Freiman (2016), yaptıkları araştırmada blok tabanlı programlama yapan öğrencilerle robotik kodlama yapan öğrenciler arasında BİD beceri seviyeleri incelendiğinde anlamlı bir farkın olmadığı ortaya çıkmıştır. Fakat bu araştırmadan elde edilen sonuçların tersi yönde çalışmalar da bulunmaktadır. Robotik kodlama ve blok tabanlı öğretim alan 6. sınıf öğrencileri arasında bilgi işlemsel düşünme becerisindeki değişimin robotik kodlama yapan öğrencilerde olumlu yönde daha fazla olduğu belirlenmiştir (Karaahmetoğlu ve

Korkmaz, 2019). 9. sınıf öğrencileri arasından blok tabanlı kodlama yapanlar ile blok tabanlı robotik kodlama yapan öğrencilerin BİD becerilerindeki öz yeterlilik düzeyleri karşılaştırıldığında robotik kodlama yapan öğrencilerin BİD becerilerindeki öz yeterlilik seviyesinin daha fazla olduğu saptanmıştır. (Saritepeci ve Durak, 2017). Bu araştırmada mBlock ortamında BİDT temel alındığında ön ve son test arasında oluşan anlamlı farkın sebebi olarak öğrencilerin kodlama yaparken yaptıkları kodlamanın sonucunu somut olarak görmesi gösterilebilir. mBlock ortamındaki öğrencilerin ön test ve son test BİDT puanları arasında anlamlı bir fark çıkmasına rağmen kovaryans analizi gruplar arasında bir fark oluşmadığını göstermektedir. Bu da sekiz haftalık sürenin bilgi işlemsel düşünmeyi istenilen kadar geliştirmek için yeterli olmadığı sonucuna bir destek olarak görülebilir.

Python ortamının da mBlock ortamı gibi bilgi işlemsel düşünme becerisinin gelişmesinde olumlu sonuçları olduğu söylenebilir. Alanyazın tarandığında metin tabanlı programlama diliyle BİD üzerine yapılan çalışmaların sınırlı olduğu gözlenmiştir (Özel, 2019). Bilgi işlemsel düşünme becerisini insanlara öğretmek için uygun ortam ve uygun bir programlama dilinin seçilmesi ve seviye seviye ilerlenmesi gerekmektedir. Bilgi işlemsel düşünme becerisini geliştirmek için problemin anlaşılması, birden fazla olan çözüm yolundan bir tanesine karar vermesi, paralel olarak algoritma geliştirmesi, uygun yazılım ortamında kodlaması, test aşamasına geçerek hatalarını bularak ayıklaması, sonuçlarını görerek yorumlaması beceriyi kazanmak adına oldukça önemlidir (Kalelioğlu ve Keskinlik, 2017). Benzer şekilde Feldhausen, Weese, ve Bean, (2018) yaptığı çalışma da bir grup öğrenciye blok tabanlı programlama platformu olan Scratch yazılımını gösterdikten sonra aynı gruba fiziksel programlama platformu olan Arduino'nun metin tabanlı yazılımını göstermiştir. BİD becerilerin yönelik öz yeterlik algısının olumlu olduğunu gözlemlemiştir. Bu çalışmanın sonuçlarına göre Python Turtle modülü öğrencilerin BİD becerisini geliştirme açısından faydalı bir araç olarak görülebilir. Öğrencilerin metin tabanlı bir dilde komutlar yazıp sonucunu hızlı bir şekilde somut olarak görmeleri ve metin tabanlı programlama dilinde program yazarken oluşan hatalara karşı önlemler alması öğrencilerin BİD'lerine katkı sağlamış olabilir.

Alanyazın tarandığında birçok çalışmada bilgi işlemsel düşünme testinin ölçülmesinin zor ve uzun bir süreç olduğu söylenebilir. Kazakoff, Sullivan, ve Bers, (2013) çalışmalarında bilgi işlemsel düşünme becerilerinde olumlu sonuçlar almak için etkinliklerde sürecin çok uzun tutulması gerektiğini söylemişlerdir. Buss ve Gamboa, (2017) yaptıkları araştırmada bilgi işlemsel düşünmenin olumlu yönde artması için çalışmaların uzun soluklu olmasının önemli olduğunu vurgulamıştır. Bu çalışma 8 haftalık bir uygulamayı içermektedir. 8 hafta BİD becerisini geliştirmek için tam olarak yeterli olmayabilir. Bu yüzden bu çalışmanın BİD ile ilgili sonuçları dikkatlice ele alınmalıdır. Verilen öğretimin süresinin uzatılması çalışmanın sonuçlarını etkileyebilir.

5.2 ÖNERİLER

Araştırmamızdaki bulgulara göre eğitimcilere ve araştırmacılara birtakım önerilerde bulunulmuştur.

5.2.1 Araştırmacılar için öneriler

- Bu çalışmada araştırılan problemler farklı bağlamlarda araştırıldığında farklı sonuçlar elde edilebilir. Yapılacak yeni çalışmalar ile sonuçların farklılaşıp farklılaşmayacağı test edilebilir.
- Öğrencilere hem metin tabanlı hem de blok tabanlı karma ortamlarda eğitim verilerek farklı bir araştırma yapılabilir.
- Araştırmacılara özellikle bilgi işlemsel düşünme becerilerini geliştirmede daha uzun sürelerde etkinlik yapılarak araştırma yapılması önerilebilir.
- Python programlama dilinin başarı alanındaki sonuçlarına bakıldığında 6. sınıf öğrencilerinin seviyesine uygun olmadığı söylenebilir. Fakat Python ortamı bilgi işlemsel düşünme becerisi açısından olumlu sonuçlar üretmiştir. Python ortamının

etkililiği çeşitli değişkenler açısından 8. sınıf öğrencileri ile yeniden değerlendirilebilir.

- Örneklem sayısı daha büyük olan gruplarla çalışma tekrarlanabilir.
- Daha uzun süreli yapılacak olan Python derslerinde öğrencilerin başarıları bu çalışmada ortaya konulduğu kadar düşük olmayabilir. Öğrenciler kodlama ortamlarına alıştıklarında başarıları BİD becerilerinde olduğu gibi artabilir.

5.2.2 Eğitimciler için öneriler

- Yapılan araştırmaya göre öğrencilerin bilgi işlemsel düşünme, tutum ve özellikle başarı kısımlarında mBlock robotik kodlama ortamının olumlu yönde verileri olduğundan 6. sınıflar için kodlama dersinde robotik kodlama yapılarak öğrencilere kodlama öğretiler.
- Robotik kodlama bir maliyet gerektirdiğinden, robotik kodlamaya erişimin olmadığı durumlarda Scratch blok tabanlı görsel programlama dili önerilebilir.
- Python programlama dilinin yapılan araştırmadaki sonuçları göz önünde tutulduğunda 6. sınıf öğrencileri için başarı anlamında yetersiz olduğu dikkat çekmektedir. Dolayısıyla Python metin tabanlı programlama dilinin 6. sınıf öğrencilerine kodlama dersinde gösterilmesi önerilmemektedir.

KAYNAKÇA

- Akbulut, A., Looney, C. (2009). Improving IS student enrollments: Understanding the effects of IT sophistication in introductory IS courses. *Journal of Information Technology Education*, 8(1), 87-100
- Allsop, Y., Sedman, B. (2005). Primary Computing in Action. *Cat Educational Ltd*.
- Alvarez, A., Lavranega, M. (2016). Experiences incorporating lego mindstorms robots in the basic programming syllabus: lessons learned. *Journal of Intelligent & Robotic Systems*, 81(1), 117-129.
- Anagün, Ş., Atalay, N., Kılıç, Z. ve Yaşar, S. (2016). Öğretmen adaylarına yönelik 21.Yüzyıl Beceleri yeterlilik algıları ölçeğinin geliştirilmesi: geçerlik ve güvenilirlik çalışması. *PAU Eğitim Fakültesi Dergisi*, 40(40), 160-175.
- Aslan, A. (2002). Yaratıcı problem çözme. *Örgütte Kişisel Gelişim*, 325-370.
- Balanskat, A., & Engelhardt, K. (2014). Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe. European Schoolnet.
- Barak, Y., Zadok, M. (2009). Robotics projects and learning concepts in science, technology and problem solving. *International Journal of Technology and Design Education*, 19(3), 289-307.
- Baz, F. (2018). Çocuklar için kodlama yazılımları üzerine karşılaştırmalı bir inceleme. *Curr Res Educ*, 4(1), 36-47.
- Beck, K. (2000). Extreme programming explained. Embrace change. Reading, MA; Addison Wesley.
- Bell, T. (2014). Ubiquity Symposium: The science in computer science: unplugging computer science to find the science. *Ubiquity*, 3. DOI:10.1145/2590528.2590531.
- Bell, T., Curzon, P., Cutts, Q., Dagiene, V. ve Haberman, B. (2011). Overcoming obstacles to CS education by using none-programming outreach programmes. *In international conference on informatics in schools: Situation, Evolution, and Perspective*, 71-81. Springer, Berlin, Heidelberg.
- Berry, M. (2014). Text based programming in primary schools? 6 Haziran 2018 tarihinde <http://milesberry.net/2014/02/text-based-programming-in-primaryschools/> sitesinden alınmıştır.
- Bers, M., Flannery, L., Kazakoff, E. ve Sullivan, A. (2014). Computational Thinking and Tinkering: Exploration Of an Early Childhood Robotics Curriculum. *Computers Education* (72), 145-157.
- Bevan, J., Werner, L., & McDowell, C. (2002). Guidelines for the use of pair programming in a freshman programming class. *In Software Engineering Education*

- and Training, 2002. (CSEE&T 2002), *Proceedings .15th Conference on (pp. 100-1007) IEEE.*, 100-107.
- Brennan, K., ve Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, 1(25). Vancouver, Canada.
- Byrne, P., ve Lyons, G. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin*. 33(3), 49-52. ACM. DOI: 10.1145/507758.377467.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- Buss, A., & Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. *In Emerging Research, Practice, and Policy on Computational Thinking* , 189-203. Springer, Cham.
- Büyüköztürk, Ş., Çakmak, E., Akgün, Ö., Karadeniz, Ş., & Demirel, F. (2017). *Bilimsel Araştırma Yöntemleri*. Ankara: Pegem Yayıncılık.
- Büyüköztürk, Ş., Çakmak, E., Akgün, Ö., Karadeniz, Ş., & Demirel, F. (2016). *Bilimsel Araştırma Yöntemleri* . Ankara: Pegem Akademi.
- Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö., Karadeniz, Ş., & Demirel, F. (2008). *Bilimsel araştırma yöntemleri*. Ankara: Pegem Yayınları .
- Büyüköztürk, Ş., Kılıç, E., Akgün, Ö., Karadeniz, Ş., & Demirel, F. (2009). *Bilimsel Araştırma Yöntemleri (4. Basım)*. Ankara: Pegem A. Yayıncılık.
- Cao, L., ve Xu, P. (2005). Activity patterns of pair programming. *In System Sciences 2005. HICSS'05. Proceeding s of the 38th Annual Hawai International Conference on (pp.88a-88a) IEEE*.
- Cevahir, H., & Özdemir, M. (2017). Programlama öğretiminde karşılaşılan zorluklara yönelik öğretmen görüşleri ve çözüm önerileri. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu* , İnönü Üniversitesi, Malatya.
- Choi, K., Deek, F. ve Im, I. (2008). Exploring the underlying aspects of pair programming :The impact of personality. *Information and Software Technology*, 50(11), 1114-1126.
- Clany, M. (2004). Misconceptions and attitudes that interfere with learning to program. *Computer science education research.*, 85-100.
- Cliburn, D. (2003). Experiences with pair programming at a small college. *Cournal of Computing Sciences in Colleges*, 19(1), 20-29.

- Cortina, T. (2015). Reaching a broader population of students through unplugged activities. *Communications of the ACM* 58(3), 25-27.
- Coşar, M. (2013). *Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri*,Yayımlanmamış Doktora Tezi, Gazi Üniversitesi, Ankara.
- CSTA, NSF, ve ISTE. (2011). Operational Definition of Computational Thinking for K-12 Education. *Computer Science Teaching Association, National Science Foundation, International Society for Technology in Education*.
- Cuny, J., Snyder, L. ve Wing, J. (2010). Demystifying computational thinking for non-computer scientists.14 Haziran tarihinde <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf> adresinden alınmıştır.
- Curzon, P., McOwan, P.W., Donohue, J., Wright, S., Marsh, W. (2018). Teaching of Concepts. *Computer Science Education: Perspectives on Teaching and Learning in School*, 91.
- Çankaya, S., Durak, G., ve Yünkül, E. (2017). Robotlarla programlama eğitimi: öğrencilerin deneyimlerinin ve görüşlerinin incelenmesi. *Turkish Online Journal of Qualitative Inquiry*, 8(4), 428-445.
- Çetin, İ.,Berigel, M. (2017). Bilgisayar Bilimi Eğitiminde Kavram ve Kuramlar. *Bilgi İşlemsel Düşünmeden Programlamaya* ,118-124, Ankara: Pegem.
- Çetin, İ.,Özden, M. (2014). Development of computer programming attitude scale for university students. *Computer Applications in Engineering Education*,23(5), 667-672.
- Çetin, İ., Otu, T. ve Oktaç, A. (2020). Bilgi İşlemsel Düşünme Testi'nin Türkçeye Uyarlanması. *Türk Bilgisayar ve Matematik Eğitim Dergisi*, Elektronik ön baskı. doi: 10.16949/turkbilmat.643709.
- 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.
- Dinçer, A. (2018). *6.sınıf öğrencilerine scratch ve kodu game lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarılarının karşılaştırılması*.Yayımlanmamış Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, Eğitim Bilimleri Enstitüsü, İzmir).
- Djambong, T., & Freiman, V. (2016). Task-Based assessment of students' computational thinking skills developed through visual programming or tangible coding

- environments. *13th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2016)*, 41-51. Mannheim, Germany.
- Durak, H., Karaoğlu Yılmaz, F., Yılmaz, R., ve Seferoğlu, S. (2017). Erken Yaşta Programlama Eğitimi: Araştırmalardaki Güncel eğitimlerle ilgili bir inceleme. *Eğitim Teknolojileri Okumaları*, 205-236.
- Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers Education*, 116, 191-202.
- Eguchi, A. (2010, March). What is educational robotics? Theories behind it and practical implementation. *In Society for information technology & teacher education international conference .Association for the Advancement of Computing in Education (AACE)*. 4006-4014, Mart 2010, California ABD.
- Ekici, G., Abide, Ö., Canbolat, Y. ve Öztürk, A. (2017). 21.yüzyıl becerilerine ait veri kaynaklarının analizi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 6(1), 124-134.
- Erekmekçi, M., Fidan, Ş. (2012). Oyunun tasarım platformları: Oyunun eğitim ve kültüre etkisi. *Yaşam Bilimleri Dergisi 1(1)*, 851-861.
- Erol, O. (2015). *Scratch ile programlama öğretiminin bilişim teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisi*. Yayınlanmamış Doktora Tezi Anadolu Üniversitesi Eğitim Bilimleri Enstitüsü.
- Ersoy, H., Madran, R., ve Gülbahar, Y. (2011). Programlama dilleri öğretiminde bir model önerisi . *Robot programlama. 13. Akademik Bilişim Konferansı Bildirileri*, 731-736. İnönü Üniversitesi, Malatya
- Erümit, A., ve Berigel, M. (2018). Programlama Dillerinin Tarihi ve Programlama Öğretimi. *Kuramdan Uygulamaya Programlama Öğretimi* (s. 2-3). Ankara: Pegem Akademi.
- Eryılmaz, S. (2003). Algoritma Tasarlama ve Programlamaya Giriş. *Ankara Detay Yayıncılık*.
- Eylon, B., ve Linn, M. (1988). Leerin and instruction: An examination of four research perspective in science education. *Review of educational research*, 58(3), 251-301.
- Feldhausen, R., Weese, J. L. ve Bean, N. H. (2018, February). Increasing student self-efficacy in computational thinking via STEM outreach programs. *In Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 302-307, 21-24 Şubat, Baltimore, MD, ABD .
- Fesakis, G., ve Serafeim, K. (2009). nfluence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin*, 41(3), , 258-262.
- Golovinsky, A., Yim, M., Zhank, Y., Eldershaw, C., ve Duff, D. (2004). Polybot and polykinetic/spl trade/system: a modular robotic platform for education. *In IEEE*

- International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, 1381-1366. 26 Nisan-1 Mayıs , New Orleans, LA, ABD.*
- Gökoğlu, S. (2017). Programlama Eğitiminde Algoritma Algısı: Bir Metafor Analizi. *Cumhuriyet International Journal of Education*, 6(1),1-14
- Green, S., Salkind, N. (2005). Using SPSS for Windows and Macintosh: Analyzing and understanding data (dördüncü baskı). *New Jersey: Pearson.*
- Grover, S., Basu, S. (2017). Mesuring student learning in introductory blok-based programming: Examining misconception of loops, variables, and boolean logic. In *Proceedings of the 2017 ACM SIGCSE. Technical Symposium on computer science Educational*, 267-272, 8-11 Mart, Seattle, Washington, ABD
- Grover, S., Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Gül, D., Çetin, İ., ve Özden, M. (2019). Bir Ölçek Çalışması: Ortaokul Öğrencilerinin Programlamaya Yönelik Tutumlarını. *7. Uluslararası Öğretim Teknolojileri ve Öğretmen Eğitimi Sempozyumunda sunulan bildiri*, 30 Ekim - 1 Kasım 2019, Antalya, Türkiye,
- Gülbahar, Y., Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*, 13(1), 33-50.
- Gülbahar, Y., Kalelioğlu, F., Kert, S., İliş, E.B., Demirhan, E.K., Yurdakök, E.A. ve Karaosmanoğlu, G. (2019). *6.Sınıf Bilişim Teknolojileri ve Yazılım Dersi Öğretmen Rehberi*. Devlet Kitapları Birinci Baskı, ISBN 978-975-11-4780-6
- Güneş, A., Karabak, D. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2 (3), 175-181.
- Gününç, S., Odabaşı, H. ve Kuzu, A. (2013). 21. yüzyıl öğrenci özelliklerinin öğretmen adayları tarafından tanımlanması, *Eğitimde Kuram ve Uygulama*, 9(4), 436-455.
- Haladjian, J., Bredies, K. ve Brügge, B. (2016). Interactex: an integrated development environment for small textlies, *In Proceedings of the 2016 ACM international Symposium on Wearable Computers.*, 8-15, Heidelberg, Germany.
- Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L. ve Zander, C. (2011). Pair Programming in education: a literature review. *Computer Science Education*, 21(2), 135-173.
- Hannay, J., Dyba, T., Arisholm, E. ve Sjöberg, D. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*, 51(7), 1110-1122.
- Harel, I. (1991). *Children Designes*. Ablex Pub. Corp. New Jersey: Ablex Publishing, 24-27.

- Harwani, B. (2012). Introduction to Python. *Programming and Developing GUI Applications with PyQt*, Course Technology, Boston. .
- Hazzan, O., Lapidot, T. ve Ragonis, N. (2011). Guide to teaching computer science: An activity-based approach, *New York:Springer*.
- Hewson, P. ve Hewson, M. (1988). An appropriate conception of teaching science. *Science Education*,72(5), 597-614.
- Hutchinson, A., Moskal, B., Daan, W. ve Cooper, S. (2008). Impact of the Alice curriculum on community college students' attitudes and learning with respect to computer science. . *In Paper appears in the proceedings of the annual meeting of the American Society for Engineering Education*, 24 Harizan,Pittsburgh, PA ABD.
- İmal, N. , Eser, M. (2009, Kasım). *Programlama dili öğrenmedeki zorluklar ve çözüm yaklaşımları*.Elektrik Elektronik Bilgisayar Biyomedikal Mühendislikleri Eğitimi IV. Ulusal Sempozyumu, Sözlü bildiri Kasım 2009, Eskişehir Osmangazi Üniversitesi Kongre ve Kültür Merkezi, Eskişehir, Türkiye.
- Jacobson, N., Schaefer, S. (2008). Pair programming in CSI:overcoming objections to its adoption. *ACM SIGCSE Bulletin*, 40(2), 93-96.
- Julia, A. (2017). Teaching coding in Canadian schools.How do the provinces measure up: Teaching coding in Canadian schools.How do the provinces measure up: <https://globalnews.ca/news/3693932/teaching-coding-in-canadian-schools-how-do-the-provinces-measure-up/> adresinden alındı.
- Kafai, Y., Resnick, M. (1996). Constructionism in practice:Designing,thinking,and learning in a digital world, *New Jersey:Lawrence Erlbaum*.
- Kalelioğlu, F. (2017). Bilgisayarsız bilgisayar bilimi (B3) öğretimi. *Y.Gülbahar(Ed) Bilgi İşlemsel Düşünmeden Programlamaya*, 183-206 Ankara:Pegem Akademi
- Kalelioğlu, F., Gülbahar, Y. (2014). The effect of teaching programming via scratch on problem sloving skills: A discussion from learners perspective. *informatics in Education* ,13(1), 33-50.
- Kalelioğlu, F., Kesinkılıç, F. (2017). Bilgisayar Bilimi Eğitimi İçin Öğretim Yöntemleri. Y. Gülbahar içinde, *Bilgi İşlemsel Düşünmeden Programlamaya* (s. 155-178). Ankara: Pegem Akademi.
- Kandemir, C. (2017). Metin Tabanlı Programlama. Y. Gülbahar içinde, *Bilgi İşlemsel Düşünmeden Programlamaya* , 267-292, Ankara: Pegem Akademi.
- Karaahmetoğlu, K., Korkmaz, Ö. (2019). The effect of project-based Arduino educational robot applications on students' computational thinking skills and their perception of Basic Stem skill levels. ,*Participatory Educational Research, Online Submission*, 6(2), 1-14.

- Karabak, D., Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Journal of Research in Education and Teaching*, 2(3), 163-169.
- Karademir, T. (2018). Teknolojik Ritimler: Müzik Eğitiminde Robotik Uygulamaların Kullanımı. *İlköğretim Online*, 717-737.
- Karasar, N. (2005). *Bilimsel Araştırma Yöntemleri*. Ankara: Nobel Yayıncılık.
- Kasalak, İ. (2017). *Robotik kodlama etkinliklerinin ortaokul öğrencilerinin kodlamaya ilişkin özyeterlik algılarına etkisi ve etkinliklere ilişkin öğrenci yaşantıları*. Yayımlanmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara.
- Kazakoff, E., Sullivan, A. ve Bers, M. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 1-11.
- Kelleher, C., Pausch, R. (2005). Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.* 37(2), 83-137.
- Kert, S. (2018). Programlama Öğretimi İçin Pedagojik Yaklaşımlar. Y. Gülbahar, ve H. Karal içinde, *Kuramdan uygulamaya Programlama Öğretimi*, 112-113, Ankara: Pegem Akademi.
- Kim, D., Choi, J., Jung, I., ve Choi, C. (2018). Pedagogy, of programming Education for Higher Education Using Blok Based Programming Environment, *In international conference of design, User Experience, and Usability*, 39-50, Springer, Cham.
- Kinnunen, P., Malmi, L. (2008). CS minors in a CS1 course. *Proceedings of the Fourth international Workshop on Computing Education Research*, 79-90.
- Kopsiripat, 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.
- Korhonen, T., Lavonen, J., Kukkonen, M., Sormunen, K., ve Juuti, K. (2014). The Innovative scholls as an environment for the desing of educational innovations. *Finnish Innovations and Technologies in Scholls*, 99-113, Brill, Sense
- Korkmaz, Ö. (2016). Prospective CITE Teachers' Self-efficacy Perceptions on Programming. *Procedia-Social and Behavioral Sciences*, 83, 639-643.
- Korkmaz, Ö. (2016). The Effect of Lego Mindstorms Ev3 Based Design Activities on Students' Attitudes towards Learning Computer Programming, Self-Efficacy Beliefs and Levels of Academic Achievement. *Online Submission*, 4(4), 994-1007.
- Kurihara, A., Sasaki, A., Wakita, K. ve Hosobe, H. (2015). A Programming environment for visual-Block-Based Domain-Specific Languages. *Procedia Computer Science*, 287-296.

- Kuzgun, Y. (1992). *Rehberlik ve Psikolojik Danışma*. Ankara: ÖSYM Eğitim Yayınları.
- Kuzu, A., Türk, M. (2018). Fiziksel Programlama. Y. Gülbahar, & H. Karal içinde, *Kuramdan Uygulamaya Programlama Öğretimi*, 339-388, Ankara: Pegem Akademi.
- Lahtinen, E., Ala-Mutka, K. ve Jarvinen, H. (2005). A study of the difficulties of novice programmers. *Acm Sigcse Bulletin*, 37(3), 14-18.
- Lai, A., Yang, S. (2011). The learning effect of visualized programming learning on sixth graders' problem solving and logical reasoning abilities, *In: International conference on electrical and control engineering (ICECE)*, 6940-6944, 29-31 Mart, Bangkok, Thailand.
- Lopez, J., Gonzalez, M. ve Cano, E. (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.
- Lye, S., Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
- Lykke, M., Coto, M., Mora, S., Vandel, N. ve Jantzen, C. (2014). Motivating programming students by problem based learning and LEGO robots. *In 2014 IEEE Global Engineering Education Conference (EDUCON)*, 544-555, 4-6 Nisan, Amman, Jordan.
- Ma, L., Ferguson, J., Roper, M., Ross, I. ve Wood, M. (2009). Improving the mental models held by novice programmers using cognitive conflict and jeliot visualisations. *In Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education*, 166-170, Temmuz 2009, Paris, Fransa.
- Maason, R., Cooper, G. (2013). Mindstorms robots and the application of cognitive load theory in introductory programming. *Computer Science Education*, 23(4), 296-314.
- Maloney, J., Peppler, K., Kafai, Y. ve Resnick, M. (2008). Programming by choice: Urban youth learning programming with Scratch. *SIGCSE*, 367-371, 12-15 Mart, Portland, Oregon USA.
- Maloney, J., Resnick, M., Rusk, N., Silverman, M. ve Eastmont, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1-15.
- Mano, C., Allan, V. ve Cooley, D. (2010). Effective in-class activities for middle school outreach programs. *. Frontiers in Education Conference (FIE)*, F2E-1, 27-30 Ekim, Marriott Crystal Gateway, Arlington, Virginia

- Martin-Ramos, P., Silva, M., Lopes, M. ve Silva, M. (2016). Student2student: Arduino project-based learning. *In Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 79-84, Kasım, Salamanca, Spain.
- McDowel, C., Werner, L., Bullock, H. ve Fernald, J. (2006). Pair Programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8), 90-95.
- McGill, M. (2012). Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education (TOCE)*, 12(1), 1-32.
- MEB.(2019).MEB Temel Eğitim Genel Müdürlüğü: 19 Kasım 2019 tarihinde <http://tegm.meb.gov.tr/www/yabanci-dil-agirlikli-egitime-yonelik-5-ve-6-sinif-ingilizce-dersi-ogretim-programlari-ile-destek-materyalleri-hazirlandi/icerik/534> sitesinden alınmıştır.
- Meerbaum-Salant, O., Armoni, M. ve Ben-Ari, M. (2010). Learning computer science concepts with scratch. *In Proceedings of the Sixth International Workshop on Computing Education Research (ICER '10)*, 69-76. New York, NY: ACM
- Meerbaum-Salant, O., Armoni, M. ve Ben-Ari, M. (2011). Habits of programming in scratch. *Paper presented at the Proceedings of the 16 th annual joint conference on innovation and technology in computer science education*, 168-172, Haziran, Darmstadt Germany .
- Meerbaum-Salant, O., Armoni, M. ve Ben-Ari, M. (2013). Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 239-264.
- Mıhçı, C. (2014). *Programlama eğitiminde görsel blok programlama ve mobil uygulamaya geliştirme araçlarının karşılaştırılması*. Yayımlanmamış Yüksek Lisans Tezi. Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Mohamad, S., Patel, A., Latih, R. ve Qassim, Q. (2011). Principles and dynamics of block-based programming approach. *In Computer Informatics (ISCI), 2011 IEEE Symposium on*, 340-345, Mart 2020, Kuala Lumpur, Malaysia
- Moreno-Leon, J., Robles, G. ve Roman-Gonzalez, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, (46), 1-23.
- Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T. ve Kuno, Y. (2009). A CS unplugged design pattern. *ACM SIGCSE Bulletin*, 41(1), 231-235.
- Numanoğlu, M., Keser, H. (2017). Programlama öğretiminde robot kullanımı – Mbot örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*, 6(2), 497-515.
- Nam, D., Kim, Y., & Lee, T. (2010, November). The effects of scaffolding-based courseware for the Scratch programming learning on student problem solving skill.

- In Proceedings of the 18th International Conference on Computers in Education* , 723-727, Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.
- Oluk, A., Korkmaz, Ö., Oluk, H. (2018). Scratch'ın 5. Sınıf Öğrencilerinin Algoritma Geliştirme ve Bilgi-İşlemsel Düşünme Becerilerine Etkisi . *Turkish Journal of Computer and Mathematics Education Vol.9 No.1 (2018)*, 54-71.
- Oppennikova, E., Ershow, M. ve Iljin, I. (2015). Educational Robotics as an İnovative Educational Technology. *Procedia-Social and Bhavioral Sciences* 214, 18-26.
- Ozoran, D., Çağıltay, N. ve Topalli, D. (2012). Using Scratch in introduction to programming course for engineering students. *Proceedings of 2nd İnternational Engineering Education Conference(İEEEC2012)*, (Vol:2), 125-132, 31 Ocak-03 Şubat 2012, Antalya, Türkiye.
- Öndeş, Ö. (2016). İngiltere ve ABD de Kodlama Eğitimi. 29 Şubat 2019 tarihinde <http://www.hurriyet.com.tr/egitim/ingiltere-ve-abdde-kodlama-egitimi-40061604> adresinden alınmıştır.
- Özdamar, K. (1999). Paket Programlar ile İstatistik Veri Analizi 1. *Eskişehir:Kaan Kitabevi*.
- Özdoğan, E. (2013). *Fiziksel olaylar öğrenme alanı için Lego program tabanlı fen ve teknoloji eğitiminin öğrencilerin akademik başarılarına, bilimsel süreç becerilerine ve fen ve teknoloji dersine yönelik tutumlarına etkisi*. Yayımlanmamış Yüksek Lisans Tezi. Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü, İzmir.
- Özel, O. (2019). *Programlama Yöntemlerinin Ortaokul öğrencilerinin Bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısına programlama başarısına etkisi*. Yayımlanmamış Yüksek Lisans Tezi, Marmara Üniversitesi Eğitim Bilimleri Enstitüsü, İstanbul.
- Özgül, F. (2013). *Her Yönüyle Python*. İstanbul: 2.Baskı,Kodlab.
- Özmen, B., Altun, A. (2014). Undergraduate students' experiences in programming: difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 1-27.
- Pakman, N. (2018). *8-10 Yaş Grubu Öğrencilerne Uygulanan Temel Düzey Kodlama, Robotik, 3d Tasarım Ve Oyun Tasarımı Eğitiminin Problem Çözme Ve Yansıtıcı Düşünme Becerilerine Etkisi*. Yayımlanmamış Yüksek Lisans Tezi, Bahçeşehir Üniversitesi Eğitim Bilimleri Enstitüsü , İstanbul.
- Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36(2), 1-11.
- Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. . New York: Inc., Publishers., 414-423.
- Papert, S. (1990). Introduction. İn I. Harel(Ed), Constructionist learning. Cambridge, MA; MIT Media Laboratory. 1-17.

- Papert, S. (1993). *The Children's Machine*. New York: Basic Books.
- Papert, S. (1999). What is Logo? Who needs it. Logo Philosophy and implementation. *Logo Computer System inc.*, 4-16.
- Pfeiffer, J. (1962). The thinking machine. *Philadephia, PA: Lippincott*.
- Piaget, J. (1964). Cognitive development in children: Development and Learning. *Journal of Research in Science Teaching*, 2(3), 176-186.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmont, E., Brennan, K. ve Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM* 52(11), 60-67.
- Ristic, O., Milosevic, D. ve Urosevic, V. (2016). The Importance of programming languages in education. *Paper presented at the 6th international Conference Technics and informatics in Education, Cacak, Serbia*.
- Saarikoski, P. (2011). Computer courses in Finnish schools during 1980-1995. In *History of Nordic Computing HiNC3, 3rd IFIP WG9. 7 Working Conference on History of Nordic Computing*. 18-20 Ekim, Stockholm, İsveç.
- Salleh, N., Mendes, E. ve Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering*, 37(4), 509-525.
- Sarıtepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational research and practice*, 490-501.
- Saygıner, Ş. (2017). *Blok Tabanlı Görsel Ve Metin Tabanlı Programlama Öğretimlerinin Erişi, Mantıksal Düşünme Ve Motivasyona Etkileri*. Yayımlanmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara.
- Sayın, Z., Seferoğlu, S. (2016). Yeni bir 21. Yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim*, 3-5. 30 Ocak-5 Şubat 2016, Adnan Menderes Üniversitesi, Aydın.
- Shim, J., Kwon, D. ve Lee, W. (2017). The Effects of a robot game Environment on computer programming education for elementary Schols students. *IEEE Transactions on Education*, 60(2), 164-172.
- Shin, S., Park, P. ve Bae, Y. (2013). The Effects of an information-technology gifted program on friendship using Scratch programming language and clutter. *International Journal of Computer and Communication Engineering*, 2(3), 246-249.
- Simith, D., Cypher, A. ve Tesler, L. (2010). Novice programming comes of age. *Communications of the ACM*, 43(3), 75-81.
- Simon, B., Hanks, B. (2008). First-year students' impression of pair programming in CSI. *Journal of Educational Resources in Computing*, 7(4), 1-28.

- Sinap, V. (2017). *Programlama eğitiminde probleme dayalı öğrenmeye yönelik Arduino etkinliklerinin kullanılması*. Yayınlanmamış Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara
- Smith, H. (1968). Anti-microbial drugs in animal feeds. *Nature*, 218(5143),728-731.
- Şahin, A., Adıgüzel, T. ve Ayar, M. (2014). Fen, Teknoloji, Mühendislik ve Matematik Çerikli Okul Sonrası Etkinlikler ve Öğrenciler Üzerindeki Etkileri. *Kuram ve Uygulamada Eğitim Bilimleri*,14(1), 297-322.
- Şimşek, İ. (2018). Dünyada Programlama Öğretimi. Y. Gülbahar, & H. Karal içinde, *Kuramdan Uygulamaya Programlama Öğretimi* ,42-43, Ankara: Pegem.
- Taub, R., Armoni, M. ve Ben-Ari M. (2012). Cs unplugged and middle-school student's views,attitudes,and intentions regarding CS. . *ACM Transactions on Computing Education* .(TOCE),12(2),1-29
- Taub, R., Ben-Ari, M. ve Armoni, M. (2009). The effect of CS unplugged on middleschool students' views of CS. *ACM SIGCSE Bulletin*, 41(3), 99-113.
- Tezbaşaran, A. (1996). Likert tipi ölçek geliştirme kılavuzu. *Ankara: Türk Psikologlar Derneği*.
- Thies, R.,Vahrenhold, J. (2013). On plugging unplugged into CS classes. *Proceeding of the 44th ACM technical symposium on Computer science education*, 365-370, Mart 2013, Denver Colorado, ABD.
- Thramboulidis, K. (2015). A cyper-physical system-based approach for industrial automation systems. *Computer in Industry* ,72, 92-102.
- Thurstone, L. (1967). The measurement of social attitudes. *Readings in attitude theory and measurement (14-25)*. New York: John Wiley Sons.
- Uslu, N., Mumcu, F. ve Eğin, F. (2018). Görsel Programlama Etkinliklerinin Ortaokul Öğrencilerinin Bilgi-İşlemsel Düşünme Becerilerine Etkisi. *Ege Eğitim Teknolojileri Dergisi*, 2(1),, 19-31.
- Üçgül, M. (2017). Eğitsel Robotlar ve Bilgi İşlemsel Düşünme. Y. Gülbahar içinde, *Bilgi İşlemsel Düşünmeden Programlamaya* ,297-298, Ankara: Pegem Akademi.
- Vieira, C., Penmetch, M., Magana, A. ve Matson , E. (2016). Computational Thinking as a Practice of Representation: A Proposed Learning and Assessment Framework. *Journal of Computational Science Education*, (Vol:7) 21-30.
- Vujosevic-Janicic, M. ve Tomic, D. (2008). The role of programming paradigms in the first programming courses. *The teaching of mathematics*, (21), 63-83.
- Wang, T. C., Mei, W. H., Lin, S. L., Chiu, S. K. ve Lin, J. M. C. (2009). Teaching programming concepts to high school students with alice. *In 2009 39th IEEE Frontiers in Education Conference IEEE*, 1-6, 18-21 Ocak , San Antonio, Texas ABD.

- Weintrop, D., Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. Paper presented at the Proceedings of the 14 th. *International Conference on Interaction Design and Children*, 199-208, Boston, Massachusetts, ABD.
- Williams, L., McCrickard, D., Layman, L. ve Hussein, K. (2008). Eleven guidelines for implementing pair programming in the classroom. *In Agile 2008 Conference, IEEE*, 445-452. 4-8 Ağustos, Toronto, Kanada.
- Wilson, C., Guzdial, M. (2010). How to make progress in computing education. *Communications of the ACM*, 53(5), 35-37.
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. *Journal of the Learning Sciences*, 17(4), 517-550.
- Yolcu, V. (2018). *Programlama eğitiminde robotik kullanımının akademik başarı, bilişimsel düşünme becerisi ve öğrenme transferine etkisi*. Yayınlanmamış Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Eğitim Bilimleri Enstitüsü, Isparta.
- Yüksel, S. (2017). *Scratch programı öğretiminde ayrılıp birleşme tekniği kullanımının öğrencilerin derse yönelik tutumuna, akademik başarısına ve kalıcılığa etkisi*. Yayınlanmamış Yüksek Lisans Tezi Adnan Menderes Üniversitesi Sosyal Bilimler Enstitüsü, Aydın.
- Yükseltürk, E., Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65.
- Yükseltürk, E., Altıok, S. (2016). An Investigation of the effects of programming with scratch. *British Journal of Educational Technology*, (Vol:47) 789-801.
- Yükseltürk, E., Altıok, S. (2017). Blok Tabanlı Programlama. Y. Gülbahar içinde, *Bilgi İşlemsel Düşünmeden Programlamaya*, 241-263, Ankara: Pegem Akademi.
- Yükseltürk, E., Altıok, S. ve Üçgül, M. (2016). Oyun programlamanın ilköğretim öğrencilerinin problem çözme becerilerine etkileri: Bir yaz kampı deneyimleri. *In 4th International Instructional Technologies, Teacher Education Symposium*, 06-08 Ocak, Elazığ, Türkiye.
- Yünkül, E., Durak, G., Çankaya, S. ve Mısırlı, Z. (2017). The effects of Scratch software on students' computational thinking skills. *Necatibey Eğitim Fakültesi Elektronik 125 Fen ve Matematik Eğitimi Dergisi*, 11(2), 502-517.

EKLER

EK 1: Mblok Ortamı için Kodlama Başarı Testi

EK 2: Python Ortamı için Kodlama Başarı Testi

EK 3: Scratch Ortamı için Kodlama Başarı Testi

EK 4: Mblok Ortamı için hazırlanan ders planları

EK 5: Python Ortamı için hazırlanan ders planları

EK 6: Scratch Ortamı için hazırlanan ders planları

EK 7: Etik Kurulu İzin Belgesi

EK 8: MEB izin belgesi

Ek 1 : Mblock Ortamı İçin Kodlama Başarı Testi

Soru 1. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- e) 140
- f) 270
- g) 300
- h) 410



Soru 2. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- a) 40
- b) 100
- c) 150
- d) 300



Soru 3. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- a) 20
- b) 60
- c) 14
- d) 50



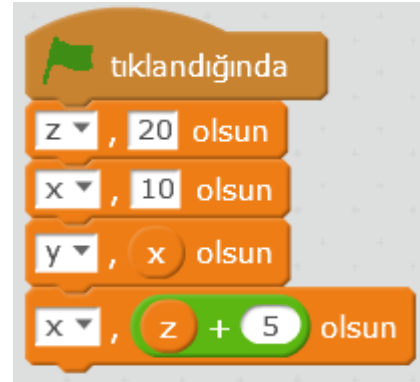
Soru 4. Yandaki Kod bloğu çalıştırıldığında ekrana kaç defa "TÜRKİYE" yazısı yazar?

- a) 12
- b) 13
- c) 14
- d) 15



Soru 5. Yandaki kod bloğu çalıştırıldığında x, y, z değişkenlerinin alacağı en son değer ne olur?

- | | x | y | z |
|----|----|----|----|
| a. | 25 | 10 | 20 |
| b. | 25 | 25 | 20 |
| c. | 10 | 10 | 5 |
| d. | 10 | 10 | 20 |



Soru 6. Yandaki kod bloğu çalıştırıldığında ekrana yazdırılacak sayı aşağıdakilerden hangisidir

- a) 5
b) 13
c) 11
d) 18



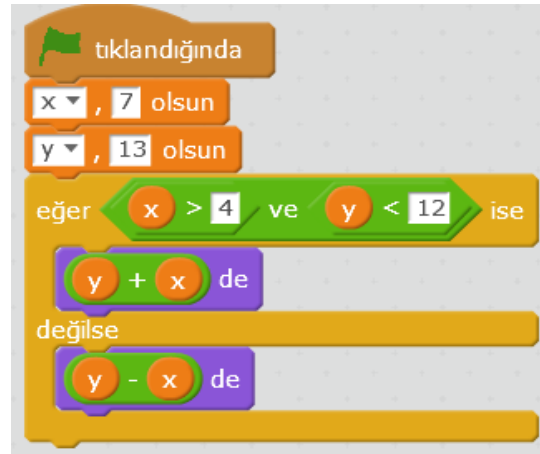
Soru 7. Yandaki kod bloğu çalıştırıldığında y' nin değeri ne olur?

- a) 6
b) 14
c) 20
d) Hiçbiri



Soru 8. Yandaki kod bloğu çalıştırıldığında ekrana ne yazar?

- a) 20
- b) 6
- c) 8
- d) 16



Soru 9. Yandaki kod bloğu çalıştırıldığında ekrana kaç defa SELAM yazar?

- a) 6
- b) 8
- c) 10
- d) 16



<u>Kutucuk 1</u>	<u>Kutucuk 2</u>	<u>Kutucuk 3</u>
<u>Kutucuk 4</u>	<u>Kutucuk 5</u>	<u>Kutucuk 6</u>
<u>Kutucuk 7</u>	<u>Kutucuk 8</u>	<u>Kutucuk 9</u>

Aşağıdaki 11., 12. ve 13. soruları yukarıdaki tabloya göre cevaplandırınız.

Soru 10. Yukarıdaki tabloya göre kaç numaralı kutucuktakiler tekrarlı işlem yapmak için kullanılabilir?

- a) 2-3
- b) 4-5
- c) 5-9
- d) 2-9

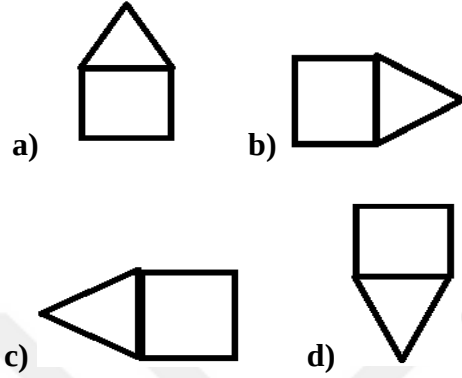
Soru 11. Yukarıdaki kutucuklardan hangi ikisi birleştirilerek kulanın sürekli Merhaba demesi sağlanabilir?

- a) 2-7
- b) 1-5
- c) 7-9
- d) 2-9

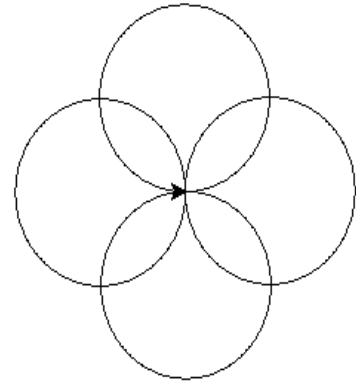
Soru 12.Yukarıdaki tabloya göre kaç numaralı kutucuktakiler koşul ile ilgili durumlarda kullanılabilir?

- a) 2-3
- b) 4-5
- c) 6-7
- d) 8-9

Soru 13. Yandaki kod bloğu çalıştırıldığında aşağıdaki şekillerden hangisi çizilir (**Dikkat:** Kukla başlangıç pozisyonunda ve sağa bakar şekilde çizime başlamaktadır)?



Soru 14. Yandaki şekli elde etmek için aşağıdaki kod bloklarından hangisini kullanılmalıdır (**Dikkat:** Kukla başlangıç pozisyonunda ve sağa bakar şekilde çizime başlamaktadır)?



Ek 2: Python Ortamı İçin Kodlama Başarı Testi

Soru 1. Yandaki kod bloğu çalıştırıldığında kaplumbağa toplamda kaç birim ilerler?

- a. 140
- b. 270
- c. 300
- d. 410

```
import turtle
kalem=turtle.Turtle()
kalem.forward(80)
kalem.right(90)
kalem.forward(40)
kalem.right(90)
kalem.forward(20)
kalem.right(90)
```

Soru 2. Yandaki kod bloğu çalıştırıldığında kamlumbağa toplamda kaç birim ilerler?

- a. 40
- b. 100
- c. 150
- d. 300

```
import turtle
kalem=turtle.Turtle()
for i in range(10):
    kalem.forward(5)
    kalem.right(15)
    kalem.forward(10)
```

Soru 3. Yandaki kod bloğu çalıştırıldığında kaplumbağa toplamda kaç birim ilerler?

- a. 20
- b. 60
- c. 14
- d. 50

```
import turtle
kalem=turtle.Turtle()
for j in range(5):
    kalem.forward(2)
    for i in range (5):
        kalem.forward(2)
```

Soru 4. Yandaki kod bloğu çalıştırıldığında ekrana kaç defa "TÜRKİYE" yazısı yazar?

- a. 12
- b. 13
- c. 14
- d. 15

```
print("TÜRKİYE")
for i in range(4):
    print("TÜRKİYE")
for i in range(8):
    print("TÜRKİYE")
```

Soru 5. Yandaki kod bloğu çalıştırıldığında x, y, z değişkenlerinin alacağı en son değer ne olur?

	x	y	z
a	25	10	20
b	25	25	20
c	10	10	5
d	10	10	20

```
z=20
x=10
y=x
x=z+5
```

Soru 6. Yandaki kod bloğu çalıştırıldığında ekrana yazdırılacak sayı aşağıdakilerden hangisidir

- a. 5
- b. 13
- c. 11
- d. 18

```
x=5
y=13
z=11
if x>y :
    z=x
else :
    z=y
print(z)
```

Soru 7. Yandaki kod bloğu çalıştırıldığında y' nin değeri ne olur?

- a. 6
- b. 14
- c. 20
- d. Hiçbiri

```
x=6
y=14
z=20
z=x
x=y
y=z
```

Soru 8. Yandaki kod bloğu çalıştırıldığında ekrana ne yazar?

- a. 20
- b. 6
- c. 8
- d. 16

```
x=7
y=13
if x>4 and y<12:
    print(y+x)
else:
    print(y-x)
```

Soru 9. Yandaki kod bloğu çalıştırıldığında ekrana kaç defa SELAM yazar?

- a. 6
- b. 8
- c. 10
- d. 16

```
import time
print("SELAM")
for i in range(8):
    print("SELAM")
    time.sleep(2)
```

<u>Kutucuk 1</u> <i>kalem.forward(5)</i>	<u>Kutucuk 2</u> <i>for i in range(10):</i>	<u>Kutucuk 3</u> <i>time.sleep(1)</i>
<u>Kutucuk 4</u> <i>if :</i> <i>.....</i> <i>else :</i> <i>.....</i>	<u>Kutucuk 5</u> <i>if :</i> <i>.....</i>	<u>Kutucuk 6</u> <i>kalem.right(90</i> <i>.</i>
<u>Kutucuk 7</u> <i>print("Merhaba</i> <i>...</i>	<u>Kutucuk 8</u> <i>kalem.setposition(70,31</i> <i>.</i>	<u>Kutucuk 9</u> <i>while True:</i> <i>.....</i>

Aşağıdaki 11., 12. ve 13. soruları yukarıdaki tabloya göre cevaplandırınız.

Soru 10. Yukarıdaki tabloya göre kaç numaralı kutucuktakiler tekrarlı işlem yapmak için kullanılabilir?

- a. 2-3
- b. 4-5
- c. 5-9
- d. 2-9

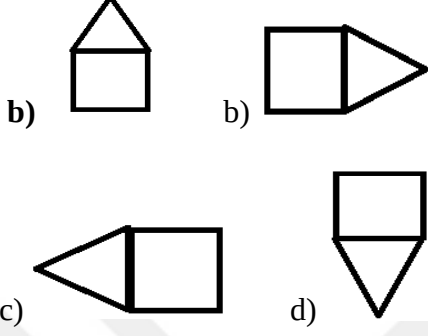
Soru 11. Yukarıdaki kutucuklardan hangi ikisi birleştirilerek ekranda sürekli Merhaba demesi sağlanabilir?

- a. 2-7
- b. 1-5
- c. 7-9
- d. 2-9

Soru 12. Yukarıdaki tabloya göre kaç numaralı kutucuktakiler koşul ile ilgili durumlarda kullanılabilir?

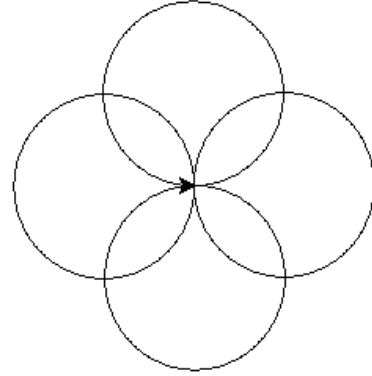
- a. 2-3
- b. 4-5
- c. 6-7
- d. 8-9

Soru 13. Yandaki kod bloğu çalıştırıldığında aşağıdaki şekillerden hangisi çizilir (**Dikkat:** Kukla başlangıç pozisyonunda ve sağa bakar şekilde çizime başlamaktadır)?



```
import turtle
kalem=turtle.Turtle()
kalem.forward(40)
kalem.right(90)
kalem.forward(40)
kalem.right(90)
kalem.forward(40)
kalem.right(90)
kalem.forward(40)
kalem.right(30)
kalem.forward(40)
kalem.right(120)
kalem.forward(40)
```

Soru 14. Yandaki şekli elde etmek için aşağıdaki kod bloklarından hangisini kullanılmalıdır (**Dikkat:** Kukla başlangıç pozisyonunda ve sağa bakar şekilde çizime başlamaktadır)?



- a)
- ```
import turtle
kalem=turtle.Turtle()
for j in range(4):
 for i in range(360):
 kalem.forward(1)
 kalem.right(1)
 kalem.right(90)
```
- b)
- ```
import turtle
kalem=turtle.Turtle()
for j in range(4):
    for i in range(180):
        kalem.forward(1)
        kalem.right(1)
    kalem.right(90)
```
- c)
- ```
import turtle
kalem=turtle.Turtle()
for j in range(8):
 for i in range(180):
 kalem.forward(1)
 kalem.right(1)
 kalem.right(90)
```
- d)
- ```
import turtle
kalem=turtle.Turtle()
for j in range(4):
    for i in range(180):
        kalem.forward(2)
        kalem.right(1)
    kalem.right(45)
```

Ek 3: Scratch Ortamı İçin Kodlama Başarı Testi

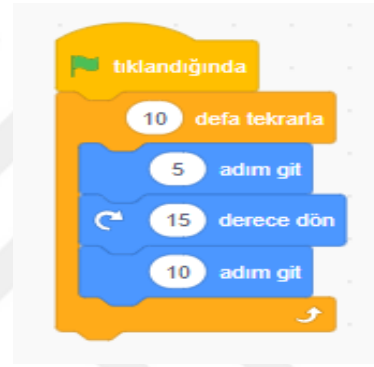
Soru 1. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- a. 140
- b. 270
- c. 300
- d. 410



Soru 2. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- a. 40
- b. 100
- c. 150
- d. 300



Soru 3. Yandaki kod bloğu çalıştırıldığında kukla toplamda kaç adım ilerler?

- a. 20
- b. 60
- c. 14
- d. 50



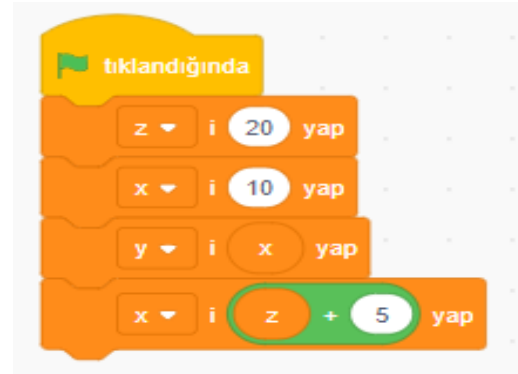
Soru 4. Yandaki Kod bloğu çalıştırıldığında ekrana kaç defa "TÜRKİYE" yazısı yazar?

- a. 12
- b. 13
- c. 14
- d. 15



Soru 5. Yandaki kod bloğu çalıştırıldığında x, y, z değişkenlerinin alacağı en son değer ne olur?

- | | x | y | z |
|----|----|----|----|
| a. | 25 | 10 | 20 |
| b. | 25 | 25 | 20 |
| c. | 10 | 10 | 5 |
| d. | 10 | 10 | 20 |



Soru 6. Yandaki kod bloğu çalıştırıldığında ekrana yazdırılacak sayı aşağıdakilerden hangisidir

- a. 5
b. 13
c. 11
d. 18



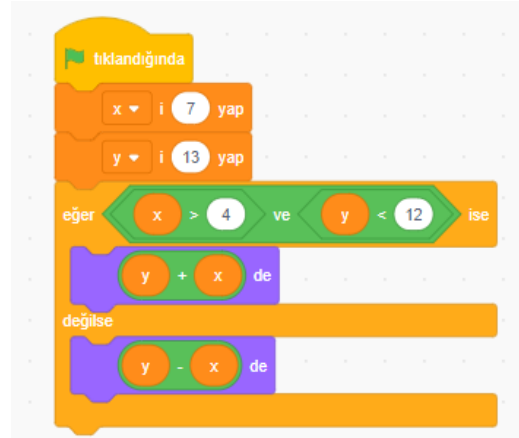
Soru 7. Yandaki kod bloğu çalıştırıldığında y' nin değeri ne olur?

- e) 6
f) 14
g) 20
h) Hiçbiri



Soru 8. Yandaki kod bloğu çalıştırıldığında ekrana ne yazar?

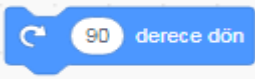
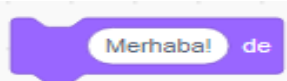
- e) 20
- f) 6
- g) 8
- h) 16



Soru 9. Yandaki kod bloğu çalıştırıldığında ekrana kaç defa SELAM yazar?

- e) 6
- f) 8
- g) 10
- h) 16



Kutucuk 1 	Kutucuk 2 	Kutucuk 3 
Kutucuk 4 	Kutucuk 5 	Kutucuk 6 
Kutucuk 7 	Kutucuk 8 	Kutucuk 9 

Aşağıdaki 11., 12. ve 13. soruları yukarıdaki tabloya göre cevaplandırınız.

Soru 10. Yukarıdaki tabloya göre kaç numaralı kutucuktakiler tekrarlı işlem yapmak için kullanılabilir?

- a. 2-3
- b. 4-5
- c. 5-9
- d. 2-9

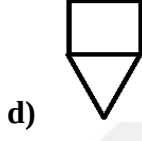
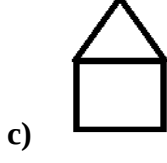
Soru 11. Yukarıdaki kutucuklardan hangi ikisi birleştirilerek kuklanın sürekli Merhaba demesi sağlanabilir?

- a. 2-7
- b. 1-5
- c. 7-9
- d. 2-9

Soru 12. Yukarıdaki tabloya göre kaç numaralı kutucuktakiler koşul ile ilgili durumlarda kullanılabilir?

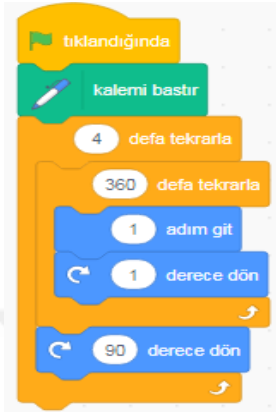
- a. 2-3
- b. 4-5
- c. 6-7
- d. 8-9

Soru 13. Yandaki kod bloğu çalıştırıldığında aşağıdaki şekillerden hangisi çizilir (**Dikkat:** Kukla başlangıç pozisyonunda ve sağa bakar şekilde çizime başlamaktadır)?



Soru 14. Yandaki şekli elde etmek için aşağıdaki kod bloklarından hangisini kullanılmalıdır (**Dikkat:** Kukla başlangıç pozisyonunda ve sağa bakar şekilde çizime başlamaktadır)?

a)



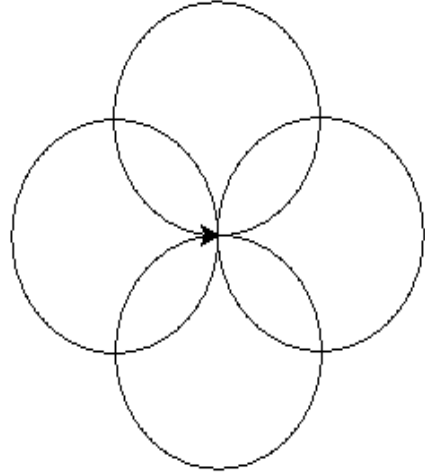
b)



c)



d)



Ek 4: Mblock Ortami İçin Hazirlanan Ders Planlari

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.1 Blok Tabanlı Programlamanın arayüzünü ve özelliklerini tanır.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse girdiğimizde öğrencilere “Blok Tabanlı Programlama kavramını daha önceden duyan var mı?” diye sorun ve öğrencilerinize konuşmaları için fırsat verin. Gelen cevaplara göre Blok Tabanlı Programlamanın ne olduğunu ve dünyadan en çok Blockly ve Scratch uygulamasının kullanıldığını söyleyin. “Scratch programını daha önce duyan var mı? Scratch programında neler yapılabilir? Sen neler yapmak isterdin?” diye sorun ve öğrencilerden gelen cevapları bekleyin. Tartıştıktan sonra “Oyunlar, hikayeler, animasyonlar yaratabiliriz.” dedikten sonra “Scratch arayüzünü iki türlü kullanabiliriz.” Deyin. “Birincisi online Scratch, ikincisi offline scratch olarak kullanılır.” diye açıklayın. Daha sonra online Sctach programını açarak öğrencilere gösterin

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

. İkinci olarak Scract offline editor indirilip bilgisayarlara kurulur. Sonra arayüz açılıp öğrencilere sahne, sahne değiştirme, sahne çizme, kuklaların olduğu ekran, kukla ekleme, kukla çizme, ses ekleme , kod blokları renklerine göre ve kodların mantıksal bir sıraya göre yapıştırıldığı bölüm gösterilir. Programın özellikleri öğrencilere anlatılır.

Sonra bir uygulama yapacağımız söylenerek eklenilen kuklanın “İlk önce hareketini sağlamalıyım.” deyip “Sizce hangi renk kod bloğundan seçeceğim, hangi kodla hareket ettirim?” diye sorulur. Öğrencilere verilen sürede, öğrencilerin mavi renkli kod bloğunun altında olan

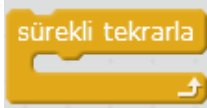
 10 adım git

kod bloğunu bulmaları ve kodları sürüklediğim yere sürüklemeleri söylenir. Kodlar bu alana sürüklendikten sonra kodu nasıl çalıştırabiliriz sorusunu sorarak öğrencilerin çift tıklama cevabını vermeleri sağlanır. Kodları çalıştırdıktan sonra “Çift tıklama dışında yeşil bayrak tıklanarak da kodlar çalıştırılır.” deyip öğrencilerin

 tıklandığında

kodunu bulmaları sağlanır. Daha sonra sizce bu kukla konuşur mu? Konuşursa hangi kod bloğu ile konuşur? şeklinde soru

sorup öğrencilerin mor blok altından  Merhaba! de 2 saniye bloğunu bulmalarını istenir. Kod bloklarını birleştirerek öğrencilerden kod bloklarını çalıştırmaları istenir. Öğrenciler kodları çalıştırdıktan sonra bu kodların bir defa çalıştığını öğrencilerin görmesini sağlayıp “Nasıl sürekli hale getiririz?” diye sorulur. Bu kod bloğunun hangi renkli blokta olduğunu bulmaları istenerek öğrencilerin

 sürekli tekrarla

kod bloğunu bulmaları beklenir. Kod bloğunu bulduktan sonra “Kodları nereye koymalıyız?” şeklinde bir soru sorarak öğrencilerin kodları bütün haline getirerek çalıştırmalarını söyleyiniz. Daha sonra adım sayısını ve saniyeyi gerekli yerlerden değiştireceklerini söyleyiniz. Konuları tekrar ederek dersi bitiriniz.

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

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.2 Blok Tabanlı Programlama aracında sunulan programın işlevini açıklar. 6.5.2.3 Blok Tabanlı Programlama aracında sunulan bir programın hatalarını ayıklar 6.5.2.4 Blok Tabanlı Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.
Süre	40 Dakika + 40 Dakika

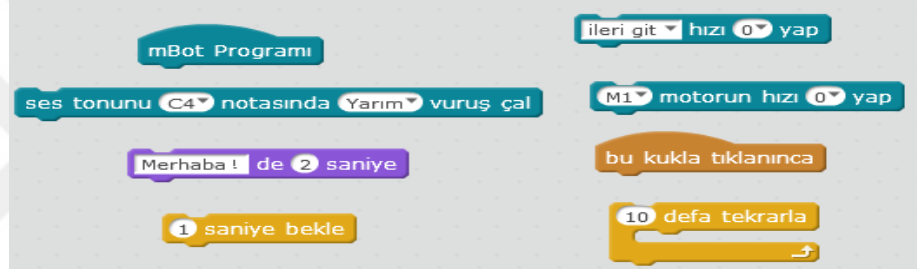
Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğretmen derse elinde mBot robotuyla gelir. Öğrencilere “Bunun ne olduğunu biliyor musunuz ve daha önce hiç robotlar ile çalışan var mı?” diye sorar. Öğrencilerden gelen yanıtlar doğrultusunda mBot robotunu öğrencilere vererek incelemelerini ister. Daha sonra neler olduğunu sorar. Öğrencilerin motorlar ve sensörleri keşfetmeleri için ipuçları verir. Sonra öğrencilerin mBot robot üzerinde iki sarı renkte olanların motor olduğunu ve bunların iki tane DC motor olduğunu ve üzerinde birden fazla sensor olduğunu söyleyerek ilerleyen derslerde sensörleri tanıtacağını söyler. Öğrencilere DC motorların 0 ile 255 değerinde güç aldığını söyler. Öğrencilere “Güç motorda ne işe yarar?” diye sorup onların motorun hızı olduğunu söylemelerini sağlar. Robotun nasıl açılacağını öğrencilere sorup mBot robotu açmalarını ister. Öğrencilere Mblock programındaki kodların bu robota nasıl gönderileceğini sorun ve cevapları bekleyin.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Öğrencilere birden fazla bağlantı şeklinin olduğunu söyleyin mBot robotların bilgisayara USB, Bluetooth ve 2.4 Wireless ile bağlantı şekillerinin olduğunu söyleyin. “Sizce bu bağlantı şekillerinin avantajları ve dezavantajları nelerdir?” diye sorun ve bekleyin. Öğrencilerin kapsama alanı gibi sorunları söylemeleri için ipuçları verin. Daha sonra mBot robotunun bağlantı şeklinin türüne göre bağlantı türlerini tek tek etkileşimli tahtada gösterin. Bağlantı şekilleri bittikten sonra bağlanan mBot robota robotun ileri gitmesi için kod gönderilerek küçük bir uygulama yapılır. Öğrencilerden ters tarafa giden mBot robot yapmaları isteyin. Öğrencilere dağıtılan mBot robotların bağlatılarını yapmalarını isteyin. Mblok programında bulunan kod blokları arasından 8 tanesini etkileşimli tahtaya yansıtarak sadece bu 8 tane kod bloğunun kullanarak 15-20 dakika süre içerisinde bir proje üretmelerini söyleyiniz. Öğrencileri başka kuklalar ve arka planları kullanmaya teşvik ederek onlara ipuçlarını verir.

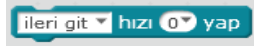


Daha sonra öğrencilerden gelen dönütler sonrasında onlara “Kısıtlı sayıda kod bloklarıyla proje üretmenin zor yanları nelerdir?” diye sorup onlarla konuşulmalıdır. Öğrencilerden gelen cevapları tartıştıktan sonra onlara “Kısıtlı sayıda kod bloklarıyla proje üretmenin kolay yanları nelerdir?” diye sorup onlarla konuşulmalıdır. Öğrencilerin kod bloklarını keşfetmeleri sağlanmalıdır. Öğretmen “Geçen hafta mBot robot ile mBlock programına başladık ve bu hafta beraber programı keşfetmeye çalışıyoruz. Zaman ilerledikçe birlikte projeler yapıp bunları kişisel bir hale getireceğiz. mBot robot ile mBlock programında proje hazırlarken kendinizi bir tiyatro yönetmeni gibi hissedebilirsiniz. Tiyatroda olduğu gibi burada da karakterler (kuklalar), kostümler(kılık), dekor, sahne ve projenin bir senaryosu vardır.” deyip öğrencilerle uygulamalara geçilir. Daha sonra

öğrencilere mBot robot ile bir uygulama için kodunu kullanarak robotun ileri götürülmesi sağlanır. Oradaki 100 değerinin motora verilecek güç olduğunu keşfetmelerini sağlayın. Bu değer 0-255 arasında değişeceğini ama 0-75 arasında motoru hareket ettirebilecek kadar uygun bir güç olmadığını söyleyin.



Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Sonra öğrencilere program durdurmak için  simgesine tıklamaları gerektiğini söyleyin. Daha sonra kod üzerine sürekli tıklayarak hareket ettirme dışında, klavyeden bir tuşa basılınca robotun hareket etmesi için  kod bloklarını kullanmalarını vereceğiniz ipuçları ile sağlayın. Öğrencilere İleri giden robotu nasıl başka bir tuşa basarak durdururuz?” sorusunu sorun.  kod bloğunu keşfetmelerini söyleyin ya da ipucu vererek kod bloğunu bulmalarını sağlayın. Öğrencilerden başka bir tuşa basarak mBot robotun durmasını ve geri gitmesini sağlayan projeyi hazırlamalarını isteyin. Onlara zaman verin. Öğrencilerden gelen dönütlerden sonra derste anlatılanları tekrar ederek dersi bitiriniz.
---	---

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	İkinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.3 Blok Tabanlı Programlama aracında sunulan bir programın hatalarını ayıklar 6.5.2.4 Blok Tabanlı Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler. 6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

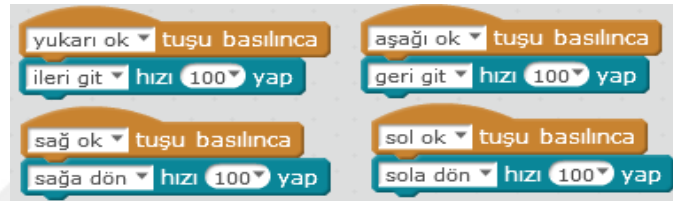
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğretmen derse başladıktan sonra, robotlar öğrenciler tarafından hazırlanmış olup bilgisayara bağlanmış şekilde beklenir. Bir önceki derste öğrenilen  kodu ve  kodlarından sonra öğrencilere “Sağa ve sola dönmek için ne yapabiliriz?” diye sorulur ve öğrencilere süre vererek onlardan robotlarını sağa ve sola çevirmeleri istenir. Öğrencilere verilecek ipuçları ile onların  kodu ve  kod bloklarını keşfetmelerini ister. Öğrencilerin kod bloklarını çalıştırdıktan sonra robotlarının istedikleri kadar dönmediklerini görmelerini sağlayın.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

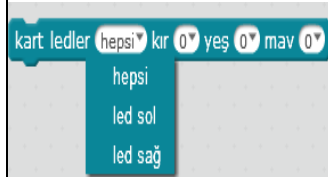
“Bunu çözmek için ne yapabiliriz?” diye sorun ve öğrencilerin **1 saniye bekle** komutu ile istedikleri kadar robotu döndürmelerini sağlayın. Öğrenciler artık robotu kod blokları ile sağa, sola, ileri ve geri hareket ettirebildiklerini görün. Daha sonra öğrencilere “Klavyenin yön tuşları ile hareket ettirebileceğimiz bir robot yapabilir miyiz?” diye sorun. Öğrencilere yeterli zaman vererek onların geçen hafta

öğrendikleri **boşluk tuşu basılınca** kod bloğunu bulup oradaki tuş ve motor hareketlerini değiştirerek robotu yön tuşları ile hareket ettirmelerini sağlayın. Öğrencilerin aşağıdaki kod bloklarına ulaşmaları için ipuçları veriniz.



Öğrencilerin bu kod bloklarını keşfettiklerinden emin olunuz. Daha sonra öğrencilerin sağ ok tuşuna basıp tuştan elini çektiğinde diğer herhangi bir tuşa basana kadar robotun sağa döndüğünü ve bunun bir problem olduğunu söyleyene kadar deneme yapmalarını sağlayın. İpuçları verin. Bunun bir sorun olduğunu ve nasıl çözebileceklerini sorun ve çözmeleri

için zaman veriniz. Bu zaman zarfında **any tuşu bırakılınca** kod bloklarına ulaşmalarını sağlayınız ve ipuçları vererek kod bloklarını keşfetmelerini sağlayınız. Artık klavyeden yön tuşları ile hareket ettirebileceğimiz robotumuzun hazır olduğunu söyleyiniz. Öğrencilere daha önce robot üzerinde birden fazla sensörün olduğunu ve bu sensörlerden ilkinin kart üzerinde bulunan ışık sensörü olduğunu söyleyiniz. Robotu ilk açtıklarından ışıkların yanıp söndüğü yerin ışık sensörlerine ait olduğunu söyleyin. Robot üzerinde ışık sensörlerini gösterdikten sonra “Hangi kod bloğu ile robot üzerindeki ışık sensörünü kontrol edebiliriz?” diye sorunuz ve öğrencilere



kod bloğunu keşfetmeleri için ipuçları veriniz. Öğrencilerin kod bloğunu keşfettikten sonra neden kırmızı, yeşil ve mavi renklerin olduğunu diğer renklerin olmadığını sorunuz. Öğrenciler ile konuyu tartıştıktan sonra bütün renklerin bu üç rengin karışımından oluştuğunu keşfetmelerini sağlayın. Oradaki rakamların renk şiddetleri olduğunu ve karışımlarda kullanıldığını ve klavyeden kullanıcı tarafından girildiğini söyleyin.

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Led sol ve led sağ komutlarının ne olduğunu sorunuz. Öğrencilerin kart üzerindeki sol ve sağ ledleri yaktığını keşfetmeleri için onlara ipucu veriniz. Öğrencilerden kart üzerindeki ledleri yakmalarını isteyiniz ve onlara yeterli zamanı veriniz. Kart ledlerini yaktıktan sonra öğrencilerden soldaki ledi yakıp sağdaki ledi söndüren, sonrasında sağdaki ledi yakıp soldaki ledi söndüren ve onlara gerekli zamanı vererek hepsinin birlikte sönüp birlikte yanmasını sağlayan robot projesini yapmalarını isteyin. Öğrencilerden klavye ile hareket eden robottaki ledlerin yön tuşlarına her basıldığında farklı renk ledlerin yanmasını sağlayan robotu yapmalarını isteyiniz ve gerekli zamanı veriniz. Konu tekrarı yaparak dersi bitiriniz.

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	Üçüncü hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğretmen derse başladıktan sonra, robotlar öğrenciler tarafından hazırlanmış olup bilgisayara bağlanmış şekilde beklenir. Geçen haftaki derste robotları klavyeden kontrol ederek kart üzerindeki ledleri yakmayı öğrenmiştik. Kart üzerindeki sensörün aynı zamanda ortamdaki ışık değerini ölçtüğünü ve bunu hangi kod bloğu ile yapabileceğini öğrencilere sorunuz ve ipuçları ile ışık algılayıcı kartta ışık sensörü değeri kod bloğunu keşfetmelerini sağlayınız. Bu hafta mBot robotunun ledlerinin birbiri ile iletişimini sağlayan proje yapacaklarını söyleyin. Daha sonra Mblok robotları arasında led ve ses sensörü ile iletişim kuracaklarını söyleyiniz.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Öğretmen derse başladıktan sonra, robotlar öğrenciler tarafından hazırlanmış olup bilgisayara bağlanmış şekilde beklenir. Geçen haftaki derste robotları klavyeden kontrol ederek kart üzerindeki ledleri yakmayı öğrenmiştik. Kart üzerindeki sensörün aynı zamanda ortamdaki ışık değerini ölçtüğünü ve bunu hangi kod bloğu ile yapabileceğini öğrencilere sorunuz ve ipuçları ile

ışık algılayıcı kartta ışık sensörü değeri

kod bloğunu keşfetmelerini sağlayınız. Bu hafta mBot robotunun ledlerinin birbiri ile iletişimini sağlayan proje yapacaklarını söyleyin. Daha sonra Mblok robotları arasında led ve ses sensörü ile iletişim kuracaklarını söyleyiniz. Kart üzerinde bir ses sensörü olduğunu ve “Sizce hangi kod bloğu ile kontrol edebiliriz?” diye sorun vereceğiniz ipuçları ile

ses tonunu C4 notasında Yarım vuruş çal

kod bloğunu keşfetmelerini sağlayınız. Daha sonra mBlokların

tıklandığında

kod bloğu ile başlayacağını ve iki mBlok arasında bir senkronizasyon olması gerektiğini söyleyin. Öğrencilere mBot robotları kendi belirleyecekleri ışık ve seslerle iletişim kurmalarını sağlayan bir uygulama yapmalarını söyleyin ve gerekli zaman zarfında yeterli ipuçları vererek etkinliği tamamlamalarını sağlayınız.



Şekil 1



Şekil 2

Öğrencilerin yaptıkları uygulamada kod bloklarının şekil 1’de olduğu zaman hepsinin aynı anda yandığını ve çaldığını söyleyerek uygulamanın şekil 2’deki gibi olması gerektiğini onların keşferecek öğrendiklerinden emin olunuz.

Daha sonra öğrencilere bu uygulamayı başka türlü nasıl yaparız diye sorun ve onlara ipuçları vererek

mesajı haberini sal ve bekle

mesajı haberini sal

mesajı haberi gelince

kod bloklarını keşfetmelerini ve kodları uygulayarak aralarındaki farkın birinci kod bloğunda haber gönderip beklediğini, ikinci blokta haberi gönderdiğini son blokta ise haber geldiğinde işlem yapacağını anlamalarını sağlayın.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Bu kod bloklarını yukarıdaki örneğe uygulamalı için zaman veriniz. Öğrencilerin aşağıdaki gibi başlamalarını sağlayınız.



Daha sonra öğrencilerin



kod bloklarını kullanarak kart üzerindeki ışık sensörü ve ledler arasında hem ışıklı hem de notalarla anlamlı müzik çalan etkinliği yapmalarını isteyiniz ve onlara gerekli zamanı veriniz.

Dersi tekrar ettikten sonra dersi bitiriniz.

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

Dördüncü hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilerle derse başladıktan sonra geçen haftaki konuları tekrar edip, bu hafta robot üzerinde yeni bir sensörü göreceğinizi söyleyin. Daha sonra robot üzerinde dışarıdaki ışık miktarını ölçen bir sensör olduğunu söyleyin. Robot üzerindeki sensörü gösteriniz. Robot üzerindeki ışık sensörünün hangi kodla control edileceğini sorun ve kısa bir zaman verin öğrencilerin mBlock program üzerinde  bulunan kodu keşfetmelerini sağlayınız . Işık algılayıcı sensörünün dışarıdaki ışık değerini 0-1023 arasında bir değer olarak hesapladığını ve ışığın en fazla olduğu ortamın 1023, en az olduğu ortamı ise 0 olarak ölçtüğünü söyleyiniz. Daha sonra öğrencilerden robottaki ışık sensörünü kullanarak ekrana ışık sensörü değerini yazdırmalarını ve ellerini sensörün üzerine kapattıklarında anlık olarak ölçtüğü değer değiştiğini görmelerini isteyin. Öğrencilere bunun için yeterli zaman verin . Öğrencilere aşağıdaki kod bloklarına ulaşmalarını sağlayan ipuçları veriniz.   Öğrenciler ışık algılayıcıdaki değeri ölçtükten sonra, onlara şartlara bağlı olarak kodların yazıldığını birden fazla şartın aynı projede olabileceğini söyleyin ve öğrencilere şartlara bağlı olan kod bloklarını bulmaları için zaman veriniz.  Öğrencilerin yandaki kod bloklarını keşfetmeleri için ipuçları veriniz. Öğrencilere if (eğer) kod bloğunun şartlara bağlı olarak olayları gerçekleştirdiğini ya da gerçekleştirmediğini söyleyiniz.
---	--

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere şarta bağlı olarak bir proje yapacaklarını söyleyiniz. Öğrencilere arabalardaki otomatik yanan farlara benzer bir proje yapacaklarını söyleyiniz. Ortamdaki ışık miktarına göre robotun lambalarını otomatik yakan ve kapatan projeyi yazmalarını söyleyiniz ve onlara gerekli zamanı veriniz. Bu süreçte öğrencilere rehberlik ederek ipuçları veriniz. Öğrencilerin kart üzerindeki ışık sensörüne göre kart üzerindeki ledleri yakmalarını isteyiniz. Dersin zamanına göre öğrencilere dışardaki ışık miktarını ölçüp ona göre hızlanan ya da yavaşlayan robot yapmalarını söyleyiniz ve yeterli zamanı vererek rehberlik ediniz. Dersi tekrar ederek bitiriniz.

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	Beşinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere bu hafta değinilecek olan değişkenler denen kavramın yapılan projelerde genelde skor ve puan yapımında kullanılan bir kavram olduğunu söyleyin. Daha sonra değişken denen bir kavramın olduğunu bunu bir kutuya benzetmelerini söyleyin. Kutu içerisine su dolduğumuzda kutudaki su miktarının arttığını veya kutudan su aldığımızda su miktarının azaldığını söyleyin. Kutunun içindeki değerin sürekli değişmesinden dolayı değişken olduğunu söyleyiniz. Öğrencilere değişkenler bloğunu keşfetmelerini söyleyiniz Bir Değişken Oluştur ☒ puan puan ▼ , 0 olsun puan ▼ 'i 1 arttır puan ▼ değişkenini göster puan ▼ değişkenini gizle Yandaki kod bloğunda değişkeni oluşturduktan sonra, başlangıç değerini belirleyeceğimiz şarta göre değişken değerini arttırıp azaltacağımızı söyleyiniz. Öğrencilerimize robot üzerinde robotun önündeki mesafeyi ölçen bir sensör olduğunu ve bunun adının “ultrasonik mesafe sensörü” olduğunu söyleyiniz ve robot üzerinde öğrencilere bu sensörü gösteriniz. Bu sensörün yarasalar gibi ses dalgaları ile önündeki mesafeyi ölçtüğünü anlatınız. Öğrencilere kod bloklarından hangisinin ultrasonik mesafe sensörünü kontrol ettiğini sorunuz ve öğrencilerin Kapı3 mesafe algılayıcı değeri kod bloğunu keşfetlerini sağlayınız. Daha sonra mesafe sensörünün çalıştığını kontrol etmek için ellerini yaklaştırıp uzakştıktıklarında mesafeyi anlık olarak ekranda gösteren programın kodlarını yazmalarını söyleyiniz. Ultrasonik mesafe sensörünün en fazla 400 cm ölçtüğünü söyleyiniz. 400 cm’den sonra değeri ölçemediği için programın stabil olarak çalışmadığını söyleyiniz.
---	--

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere mesafe sensörünü kullanarak engellere yaklaştığında çarpmayan sağa veya sola dönen ve ekrana eklenen kuklanın da mesafeye göre konuşmasını sağlayan robotun kodlarını yazmalarını söyleyiniz. Bu süreçte öğrencilere rehberlik ediniz ve ipuçları veriniz. Dersin zamanına göre öğrencilere ultrasonik mesafe sensörünün algıladığı değere göre hızlanan ya da yavaşlayan, aynı zamanda ekrandaki kuklanın da hareketlerinin arttığı ya da azaldığı programı yazmalarını söyleyiniz. Dersi tekrar ederek bitiriniz.

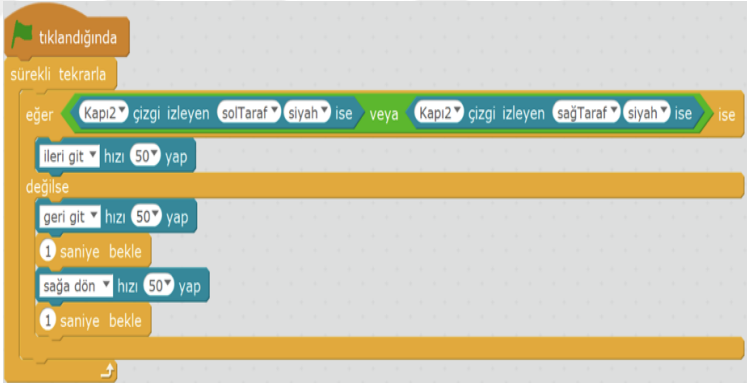
Ölçme – Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Her grup kendi grubunun ve diğer grupların algoritmalarını kontrol eder.
Açıklama	Altıncı hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

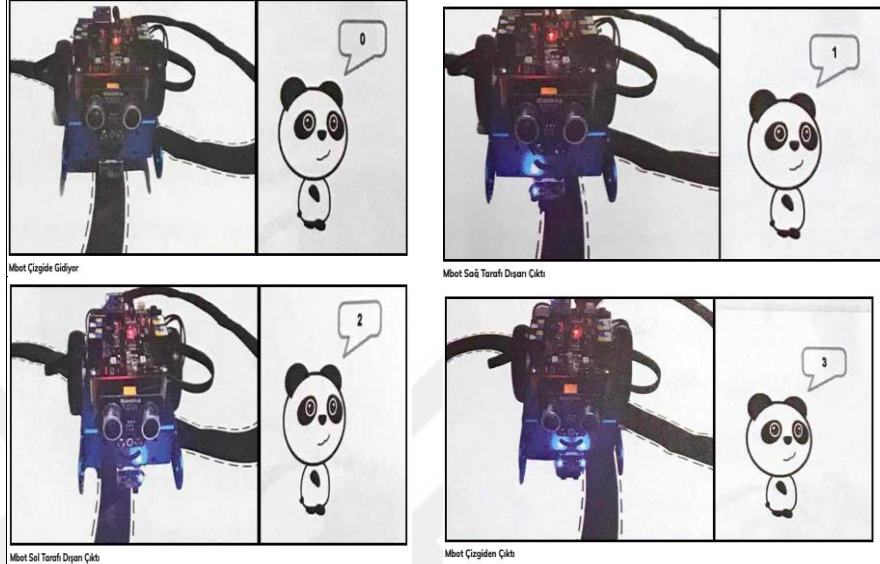
	Derse	başladıktan	sonra	öğrencilere
	ve	veya		kavramlarının neler olduğunu sorunuz ve öğrencilerden cevaplar alınız. Verdiğiniz ipuçları doğrultusunda öğrencilerin verilen şartların tamamının gerçekleşme ihtimaline
	ve			, verilen şartlardan sadece bir tanesinin gerçekleşme ihtimalinin ise
		veya		kod bloğu ile kontrol edileceğini keşfetmelerini sağlayınız.
				Daha sonra bugün robot üzerinde olan çizgi izleyen sensörünü tanıyacaklarını söyleyiniz ve sensörü robot üzerinde gösteriniz. Çizgi izleyen sensörünün aslında kısmen renk sensörü olduğunu ve iki tane gözünün olduğunu siyah ve beyaz renkleri algıladığını söyleyiniz. Öğrencilerin çizgi izleyen sensörünün hangi kodlarla kontrol edilebileceğini sorun ve
			Kapı2 çizgi izleyen	kod bloğunu keşfetmelerini sağlayınız
				Öğrencilere hemen çizgi izleyen sensörünü kullanarak masadan düşmeyen robot yapmak için gereken kodları yazmalarını söyleyiniz. Robotun masada algıladığı rengin siyah olduğu ipucunu veriniz.
				
				Öğrencilerin kendilerinin yukarıdaki kodlara ulaşarak projeyi tamamlaması için ipuçları vererek rehberlik ediniz.

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Öğrencilere daha sonra çizgi izleyen siyah ve beyaz renkleri algılama dışında bazı değerlerle kontrol edilebileceğini söyleyin ve öğrencilere beyaz kağıt üzerinde siyah çizgilerin olduğu düzenekleri dağıtın. Öğrencilerden çizgi izleyen sensörlerini kullanarak



yukarıdaki değerlere ulaşmalarını sağlayınız ve bu değerleri bulmaları için ipuçları veriniz.

Öğrencilere verdiğiniz mat üzerinde çizgi izleyen robotun kodlarını yazmalarını söyleyiniz.

Dersi tekrar ederek bitiriniz.

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

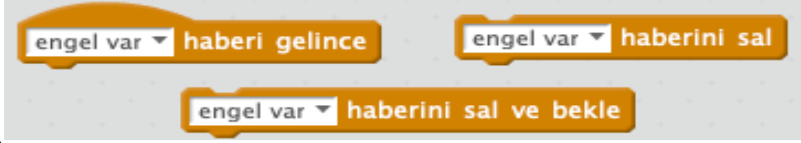
Yedinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse başladıktan sonra öğrencilere insanların telefon olmadan önce nelerle haberleştiğini sorunuz ve öğrencilerden mektup cevabını almayı sağlayınız. Öğrencilere kod bloklarını kendi aralarında haberleştiğini ve ona göre olayları gerçekleştirdiğini söyleyiniz. Öğrencilere kod blokları sizce hangi kodlarla haberleşir diye sorun ve aşağıdaki kodları keşfetmelerini sağlayınız.  Öğrencilere robotun engele geldiğinde haber salıp ses ve ışık çıkarmasını sağlayan robotun kodlarını yazmaları için zaman veriniz ve rehberlik ediniz. Öğrencilere şimdi bütün kodların kullanılacağı bir proje yapacaklarını söyleyiniz ve arabalarda kullanılan PARK SENSÖRÜ yapmalarını isteyiniz. Kurallar: • Engel 30 cm'nin altına düştüğünde ses başlayacak normal zamanda ses çıkmayacak. • Park sensörü mantığı ile çalışacak tek farkı engele iyice yaklaşıncaya robot duracak. • İleri, sağa, sola ve geriye gittiğinde farklı renkler yanacak. Bu sürede öğrencilere ipucu vererek rehberlik ediniz. Dersi tekrar ederek bitiriniz.
---	--

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	Sekizinci hafta ders planı

Ek 5: Python Ortami İçin Hazirlanan Ders Planlari

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.1 Blok Tabanlı Programlamanın arayüzünü ve özelliklerini tanır.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse girdiğimizde öğrencilere “Metin tabanlı programlama kavramını daha önceden duyan var mı?” diye sorulur ve öğrencilere konuşmaları için fırsat verilir. Gelen cevaplara göre metin tabanlı programlamanın ne olduğu ve dünyada bir çok metin tabanlı programın kullanıldığı; uygulamaların hepsinin metin tabanlı programlama sayesinde yapıldığı ve en çok kullanılan metin tabanlı programların Python,C##, Java olduğu söylenir. Python programını daha önce duyan var mı ? Python programında neler yapılabilir? Sen neler yapmak isterdin? diye sorulur ve öğrencilerden gelen cevaplar beklenir. Tartıştıktan sonra oyunlar,uygulamalar, çizimler yaratılabilir. Python yazılımının kütüphaneler ile çalıştığı ve birçok kütüphanesinin içinde bulunduğunu dışarıdan da kütüphane ekleneceği söylenir. Bizim dersimizde Python yazılımının Turtle kütüphanesi kullanılacağı ifade edilir. Pythonun birçok arayüzünün olduğu online ya da offline olarak birçok editörün bulunduğunu söylenir.

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Dersimizde İDLE PYTHON arayüzünün kullanacağı söylenir. Daha sonra İdle Python programı indirilerek bilgisayara kurulumu öğrencilerle beraber yapılır.. Sonra arayüz açılıp öğrenciler çift tıkladıktan sonra açılan pencereden kod yazma editörüne File menüsünden New File sekmesini kullanarak geçmeleri söylenir ve öğrencilerin kod editörünü açmaları sağlanır. Sonra bir uygulama yapılacağı, ilk kodumuzun yazılacağı söylenir. Kod editörü açıldıktan sonra Print(“Orhan Gazi Ortaokulu “) kodunu yazar ve bu kodu ekrana çıktı olarak Orhan Gazi Ortaokulu yazacağını söyler. Print komutunun içine yazılan yazıyı ekrana çıktı olarak verildiği söylenir. Daha sonra kodu çalıştırmak için Run menüsünden Run Modül ya da F5 tuşuna basmamız gerektiği söylenir. Gelen uyarıda program çalıştırmadan önce kaydetmemiz gerektiği vurgulanır. Programı istediğimiz yere kaydettikten sonra çıktısı öğrencilere gösterilir, ilk yazdığımız kodla ekrana mesaj vermeyi başardık denilir. Sonra öğrencilerden alt alta birkaç satır yazı yazmaları istenir. Öğrencilerin alt alta yazı yazmaları sağlanır. Konuları tekrar ederek dersi bitirilir.
---	---

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

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.2 Blok Tabanlı Programlama aracında sunulan programın işlevini açıklar. 6.5.2.3 Blok Tabanlı Programlama aracında sunulan bir programın hatalarını ayıklar 6.5.2.4 Blok Tabanlı Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.
Süre	40 Dakika + 40 Dakika

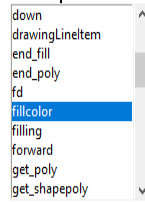
Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse girildikten sonra ilk önce Python yazılımının kütüphanelerle çalıştığı ve kütüphanelerin kullanacağımız editor içerisine aktarılması gerektiği söylenir. Bunun için İDLE Python programında kod yazma editörünün açılması istenilir ve öğrencilerle eş zamanlı olarak uygulamaya başlanır. İlk önce kütüphanenin içeri aktarılması için “import turtle()” kodunun yazılması gerektiği söylenir. Bu kütüphanenin editöre aktarılmadığı durumları da turtle kütüphanesine ait kodların kullanılamayacağı öğrencilere söylenir. Daha sonra ‘turtle’da yazılan kodların yazılış şekli ve kullanımının önemi hatırlatıldıktan sonra turtle oluşturmamız gerektiği söylenir. “kalem=turtle.Turtle” kodu ile kalem adında çizim yapacağımız bir turtle oluşturduğumuz söylenir ve öğrencilere uygulanır. Programı çalıştırmaları söylenir , Öğrencilere çalıştırdıkları pencerede üçgen şeklindeki çizim aracının şeklini değiştirebili miyiz?

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

kalem.fil



Gelen yanıtlardan sonra kalem yazıp nokta işareti koyduktan sonra bir kaç saniye beklediğimizde o araca ait bir çok özelliğin olduğu pencerenin çıktığı hatırlatılıp öğrencilerden turtle'nin rengini değiştirmeleri istenir ve süre verilir.

diye sorulur, ve öğrencilerden hangi özelliğin turtule'nin şeklini değiştireceğini bulmaları istenir. Öğrencilere verilen süre zarfında ipuçları verilerek “**kalem.shape(“turtle”)**” kodunu keşfetmeleri sağlanır. Kalem adında oluşturduğumuz turtle'mizin şeklini kodlarla kağılumbağa yaptık deneyin diye söylenir. Öğrencilerin çalıştırılan pencere turtlenin şeklinin kaplumbağa olduğu görülür ve sizce başka şekiller olur mu ? diye sorulur ve onlardan yanıtlar alınır. Öğrencilere turtle şekillerinin birden fazla olduğu ve klasik geometric şekiller olduğu söylenir. Kare, üçgen, vs. daha sonra öğrencilere turtle'nin şeklini kare ve üçgen yapmalarını söyler. Öğrencilerin yapmaları için ipuçları verir. Sonra kodun “**kalem.shape(“triangle”)**” olduğu etkileşimli tahtada gösterilir. Öğrencilere oluşturduğumuz turtlenin rengini değiştirebilir miyiz? diye sorulur.İpuçları verilerek öğrencilerin “**kalem.fillcolor(“blue”)**” kodunu keşfetmeleri sağlanır ve rengini değiştirmeleri sağlanır.

kalem.fillcolor(

(*args)
Return or set the fillcolor.

Öğrencilere bir aracın özelliğini doğru yazıp içerisine hangi değerleri gireceğimizi belirlemek için parantez açıp açılan açıklama bölü-

münde değer içeriğini görüp ona göre kodumuzu tamamlayabiliriz ipucunu verikten sonra öğrencilere sizce kaplumbağanın şeklinin boyutunu ayarlayabili rmiyiz? diye sorulup öğrencilerin “**kalem.shapesize(5,5)**” kodunu keşfetmeleri için ipuçları verilir ve kodu yazdıktan sonra uygulamaları sağlanarak kodların çalıştığı gösterilir.

Öğrencilere çalıştığımız pencere ile ilgili ayarlarda yapabiliriz denilir ve çizim aracında olduğu gibi “**pencere=turtle.Screen()**” pencere isminde kodların çalıştığı pencerenin özelliklerini değiştirebileceğimiz bir araç oluşturulur. “öğrencilere siz artık oluşturulan aracın özelliklerinin nasıl açıldığını biliyorsunuz” diyip pencerenin arka plan rengini değiştirmeleri istenir.

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere ipuçları verilerek “pencere.bgcolor(“yellow”)” kodunu keşfetmeleri sağlanır. “Pencerenin başlık çubuğunda yer alan yazıyı değiştirmek için nasıl bir kod yazarız” diyip öğrencilerin “pencere.title(“Bolu Merkez”)” kodunu keşfetmeleri sağlanır. Daha sonra öğrencilere Başlığı Orhan Gazi Ortaokulu olan , Pencerenin arka plan rengi mavi olan, Oluşturulan turtlenin rengi sarı olan, Turtle boyutu 10,10 olan, Turtle şekli kare olan, Bir uygulama yapmaları istenir ve süre verilir. Daha sonra gelen yanıtlar kontrol edildikten sonra tekrar yapılarak ders bitirilir.

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	İkinci hafta ders planı

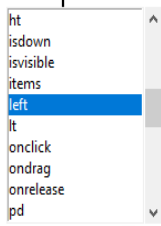
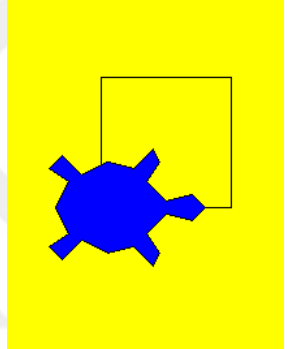
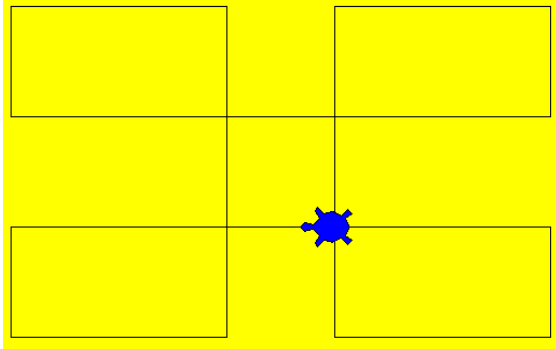
Orhan Gazi Ortaokulu
2018-2019 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar
Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.3 Blok Tabanlı Programlama aracında sunulan bir programın hatalarını ayıklar 6.5.2.4 Blok Tabanlı Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler. 6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse girdikten sonra turtle ile ilk defa çizim yapılacağı söylenir.. Çizim aracımızla çizim yapılacağı ifade edilir. Sizce ileri doğru çizim yapabilmek için çizim aracının özellikleri olan pencereden hangi kodu kullanmalıyız? diye sorulur ve öğrencilere “kalem.forward(100)” kodunu keşfetmeleri için ipuçları verilir. Buradaki 100 sayısının ne olabileceği sorulur. öğrencilerden bir önceki dersle bağlantısı olan piksel cevabını vermeleri için ipuçları verilir. Daha sonra bu kod öğrencilerle beraber eş zamanlı olarak uygulanır. ‘Turtle’ nin ileri çizim yaptığı gösterilir ve hemen geri nasıl gidebiliriz, diye sorulur.

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilerin “kalem.backward(100)” kodunu keşfetmeleri sağlanır. Öğrencilere “kalem.fd(100)” ve “kalem.back(100)” kodlarının da aynı işlemleri yaptığını yazım konusunda daha hızlı yazmak için kullanılabileceği söylenir. Öğrencilere sağa ve sola nasıl dönebiliriz? diye sorulur. kalem.lef  Öğrencilerden turtlenin özelliklerinden “kalem.left(90)” ve “kalem.right(90)” komutlarını bulmaları ve içerisine yazdıkları rakamın turtlenin dönme açısının olduğu hatırlatılır. Öğrencilerden bir kare çizmelerini isteyiniz. Karenin köşe açılarının matematik dersinden bilmeleri gerektiği hatırlatılır. <pre>import turtle kalem=turtle.Turtle() kalem.shape("turtle") kalem.fillcolor("blue") kalem.shapesize(5,5) pencere=turtle.Screen() pencere.bgcolor("yellow") kalem.fd(100) kalem.left(90) kalem.fd(100) kalem.left(90) kalem.fd(100) kalem.left(90) kalem.fd(100) kalem.left(90)</pre>  Öğrencilerin yukarıdaki kod bloklarını keşfetmeleri ve çıktıların çalıştırarak görmeleri sağlanır.  Öğrencilerden çıktısı yukarıdaki gibi olan turtlenin çizimin oluşturan kodların yazılması istenilir. Gerekli zaman verilir. Konu tekrar edilerek ders bitirilir.
---	---

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	Üçüncü hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

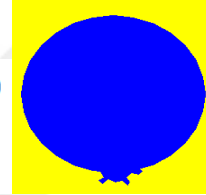
Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse başladıktan sonra geçen hafta ile ilgili özet yapılır. Daha sonra öğrencilere daire nasıl çizilir? diye sorulur. Öğrencilere zaman verilir sonra öğrencilere dairenin çevresi toplam kaç derecedir? diye sorulur ve 360 cevabını almak için ipuçları verilir. 360 cevabını aldıktan sonra öğrencilerin bir piksel ilerleyip bir piksel dönerek daire çizilebileceği cevabını alındıktan sonra bunun için çok uzun satırlar yazılması gerektiğini öğrencilerin fark etmesi sağlanır. Turtle modülünde bunu tek satır kod ile halledilebileceği söylenir. Öğrencilere özellikler penceresinden “circle” kodunu bulmaları sağlanır. Daha sonra öğrencilere ipucu verilerek “kalem.circle(100)” kodunu keşfetmeleri ve 100 piksel çapında bir daire çizildiğini göstermek için eş zamanlı olarak uygulanır.

Öğretme – Öğrenme Etkinlikleri

‘Öğrencilere turtlenin hızını ayarlayabilir miyiz?’ diye sorulur ve kodu bulmaları için zaman verilir. Daha sonra öğrencilerin “**kalem.speed(10)**” kodunu keşfetmeleri sağlanır ve bu kodu uygulamaları istenilir. Değer aralığının 0-? aralığında olduğu söylenir. Öğrencilere bu kapalı şekillerin içi doldurulur mu ? diye sorulur ve hangi kod olabileceği cevabı aranır, biraz zaman verilir. Gelecek cevaplara göre “**kalem.begin_fill()**” kodunun öğrenciler tarafından keşfedilmesi ya da tarafınızca söylenmesi sağlanır. Doldurmaya başlamak için “**kalem.begin_fill()**” komutunun kullanıldığı ifade edilir. Doldurmanın bittiği ise “**kalem.end_fill()**” komutuyla tamamlandığı öğrencilere söylenir ve öğrencilere kapalı bir daire çizen kodları yazmaları istenir.

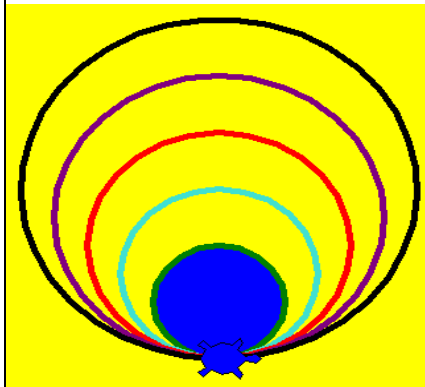
- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

```
kalem.begin_fill()
kalem.circle(100)
kalem.end_fill()
```



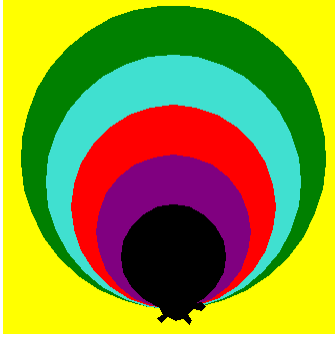
Yukarıda yazılan kod ve çıktısı görüldükten sonra öğrencilere ‘Şeklinin içinin ve dışının rengi değişik olur mu?’ diye sorulur ve öğrencilerin “**kalem.pencolor(“red”)**” komutunu keşfetmeleri ipuçları verilerek sağlanır. İçinin renginin ise daha önceki derste gördüğümüz “**kalem.color(“blue”)**” kodu ile olduğu hatırlatılır.

Daha sonra çıktısı aşağıdaki gibi olan uygulamanın kodlarını yazmalarını isteyin ve gerekli zamanı verin.



```
kalem.begin_fill()
kalem.pencolor("green")
kalem.circle(50)
kalem.end_fill()
kalem.pencolor("turquoise")
kalem.circle(75)
kalem.pencolor("red")
kalem.circle(100)
kalem.pencolor("purple")
kalem.circle(125)
kalem.pencolor("black")
kalem.circle(150)
```

Öğrencilerin yukarıdaki çıktı için yandaki kodları kendilerinin yazmaları sağlanır.

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	 Yandaki şeklin çıktısını alabileceğimiz kodlar öğrencilerden istenilir ve gerekli zaman verilir.. Ders tekrar edilerek bitirilir.

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	Dördüncü hafta ders planı

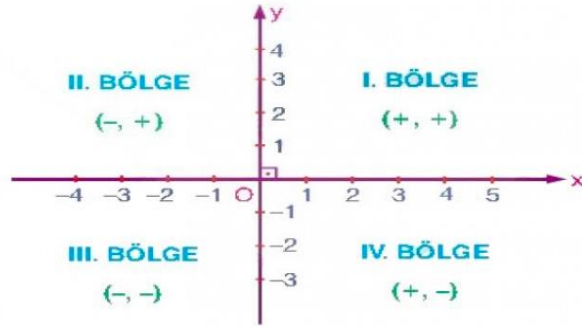
Orhan Gazi Ortaokulu
2018-2019 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar
Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

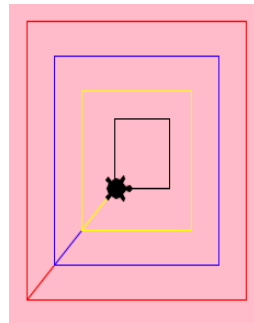
Derse başladıktan sonra öğrencilere geçen haftaki derste python turtle ilgili bir çok kod gördükleri hatırlatılır ve bu hafta yenilerini görecekleri söylenir. “Koordinat düzlemini biliyor musunuz?” diye sorulur ve cevaplar alınır.



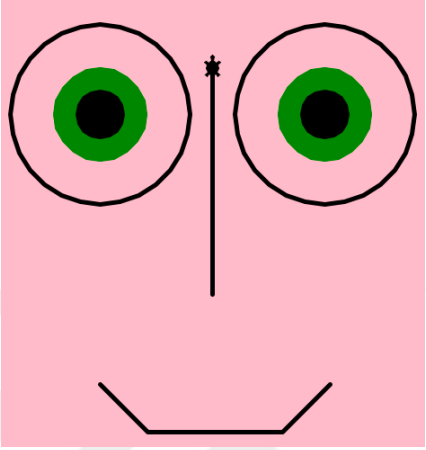
- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Koordinat düzleminde bölgelerin kavranması sağlanır. Turtlenin başlangıç konumunun (0,0) olduğu hatırlatılır ve turtlenin konumunu kooordinat düzlemine göre ayarlayacağımız söylenir. Özellikle önündeki işaretlerin kod yazmada ve turtlenin konumunda etkili olduğu belirtilir.

Sonra öğrencilere, “Ssizce turtlenin konumunu hangi kod değiştirebilir?” sorusu sorulur ve özellikler penceresinden verilecek ipuçları ile “**kalem.setposition(x,y)**” kodunu ve “**kalem.goto(x,y)**” kodlarını keşfetmeleri sağlanır.. Kodların kullanımını da keşfettikten sonra



ekran çıktısı yandaki gibi görünen uygulamanın kodlarını yazmaları istenir. Öğrencilere yandaki uygulamanın çıktısı için yeteri kadar zaman verilir. Öğrencilerin turtlenin konum değiştirirken yandaki çizginin görüntüsünü fark etmeleri ve bunu sormaları beklenir. Öğrenciler buradaki hatanın farkına vardıktan sonra, “Sizce bunu nasıl düzeltebiliriz? Diye sorulur ve öğrencilerden cevap beklenir.

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Kenardaki çizginin olmaması için kalemın bastırılması ve kaldırılması olayının olduğu ipuçları ile anlatılır. Öğrencilerin turtlenin özelliklerinden “kalem.penUp()” ve “kalem.penDown()” kodlarını keşfetmeleri sağlanır. Kullanımlarını anlattıktan sonra öğrencilerden çıktısı aşağıdaki gibi olan uygulamanın kodları istenir.  Öğrencilere yeteri kadar zaman verilip ve rehberlik edilerek uygulama yaptırılır. Daha sonra tekrar yapılarak ders bitirilir.

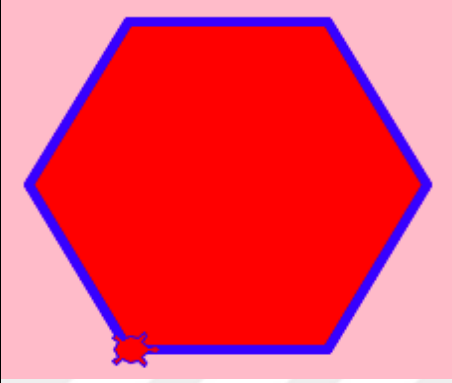
Ölçme – Değerlendirme	
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Her grup kendi grubunun ve diğer grupların algoritmalarını kontrol eder.
Açıklama	Beşinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar
Günlük Planı

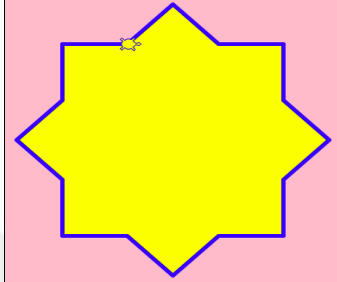
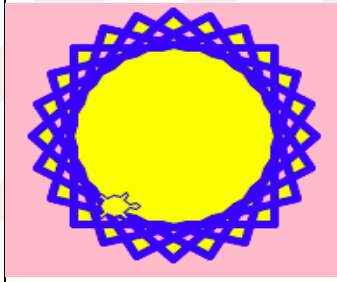
Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Dersin başında öğrencilere geçen hafta turtlenin konumunu, penup ve pendown komutları ile turtlenin istediğimiz konumlara yazmasını ya da yazmamasını öğrendiğimiz hatırlatılır. Öğrencilere şimdi bir altıgen çizen turtlenin kodlarını yazmaları gerektiği söylenip aşağıdaki çıktı ve kodu yazmaları için rehberlik edilir.  <pre>kalem.begin_fill() kalem.pencolor("blue") kalem.fd(100) kalem.left(60) kalem.fd(100) kalem.left(60) kalem.fd(100) kalem.left(60) kalem.fd(100) kalem.left(60) kalem.fd(100) kalem.left(60) kalem.end_fill()</pre> “ Yukarıdaki çıktısı verilmiş olan kodların arasında aynı kodlardan kaç tane vardır? “sorusu sorularak öğrencilere zaman verilir. <pre>kalem.fd(100) kalem.left(60)</pre> Verilecek ipuçları ile öğrencilerin kodlarının aynı olduğunu keşfetmeleri sağlandıktan sonra amacımızın yazılım yaparken en kısa yoldan problemleri çözmek olduğu ve yazılım yaparken ne kadar az kod yazılırsa o kadar hızlı çözüme ulaşıldığı söylenir. “Bu kodları nasıl kısaltabiliriz?” diye sorularak öğrencilerde merak uyandırılır.. Programlamada döngülerin olduğu, döngülerin içine yazılan kodların tekrarlandığı söylenir. Daha sonra ilk döngü olan for i in range(6): döngüsünü kullanacakları ve kullanımının gösterildiği şekilde olduğu anlatılır. Range(6) döngünün kaç defa döneceğini anlatan kod olduğu, i harfinin herhangi bir harf olduğu, istenilirse yerine başka harfte kullanılabileceği, for ve in turuncu yazmasının nedeni programın kendine ait kodlarının olduğu ve doğru olduğu, döngülerin sonunda mutlaka iki nokta üst üste kullanılması gerektiği ifade edilmelidir. Döngünün iki değer arasında dönmesinin de sağlanılabileceği, range(1,11) 1den başlayıp 11 e kadar dönmesini sağlayan kod olduğu ifade edilir. Burada döngü 1’ den başlayıp 11’ e kadar döner, 11 dahil değildir. Dolayısıyla döngünün 10 defa tekrar edeceği belirtilmelidir.
---	--

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Yukarıdaki altıgenin döngü ile yapılacağı söylenir ve kodlar öğrencilerle eş zamanlı olarak yazılır. <pre>kalem.begin_fill() kalem.pencolor("blue") for i in range(6): kalem.fd(100) kalem.left(60) kalem.end_fill()</pre> Yandaki kodların yukarıdaki altıgenin aynısını daha az kod kullanarak çizdiği ifade edilir. Öğrencilerden aşağıdaki çıktısı olan kodları yazmalarını isteyiniz.  Yandaki şekli çizmeleri için öğrencilere rehberlik edilir ve döngü kullanılması sağlanır.  Zaman durumuna göre yandaki şekli çizmeleri için öğrencilere rehberlik edilir ve döngü kullanılması sağlanır. Ders tekrar edilerek bitirilir.
---	--

Ölçme – Değerlendirme

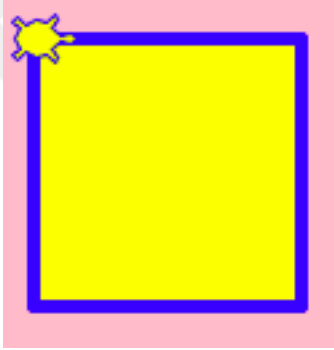
• Bireysel öğrenme etkinliklerine göre ölçme – değerlendirme • Grupla öğrenme etkinliklerine göre ölçme ve değerlendirme	Öz değerlendirme ölçütleri Her grup kendi grubunun ve diğer grupların algoritmalarını kontrol eder.
Açıklama	Altıncı hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
Süre	40 Dakika + 40 Dakika

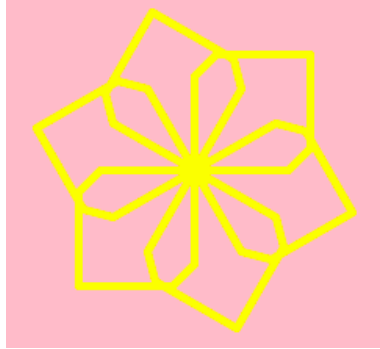
Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilere ders başladıktan sonra geçen hafta döngülere giriş yapıp çizeceğimiz şekillerin daha kolay çizildiğini görmüştük bu hafta başka bir döngü göreceğimiz, bu döngünün adının da While Döngüsü olduğu söylenir. While döngüsünün Türkçede “Olana Kadar Tekrarla” anlamındadır. Yazılım dilinde değişkenler denen bir ifadenin olduğu ve bunu somut olarak bir kutuya benzetmeleri söylenir. Kutu içerisindeki değer, içerisine koyacağınız maddenin değerine göre artacağı veya azalacağı öğrencilere söylenir. En çok oyunlardaki puan kısmında değişkenlerin kullanıldığı, “Python Turtle” modülünde değişken tanımlamanın ise aşağıdaki gibi olduğu belirtilir. i=0 Yandaki kodun Python Turtle’da değişken tanımlamak için kullanıldığı oradaki “i” değerinin değişkenin adını temsil ettiği ve istenilen harfin ya da kelimenin olabileceğini söylenir. Değişkenin başlangıç ve bitiş değerleri kendi yazacağımız programın amaçlarına göre belirlenmelidir. <pre> i=0 kalem.begin_fill() kalem.pencolor("blue") while (i<4): kalem.fd(100) kalem.right(90) i=i+1 kalem.end_fill() </pre>  Kare çizmek için While döngüsünün kullanımının bu şekilde olduğunu döngü içerisinde her defasında “i “ değerinin bir arttığını ve 4 değerinden küçük olduğu sürece devam edeceğini kavramaları sağlanır.
---	--

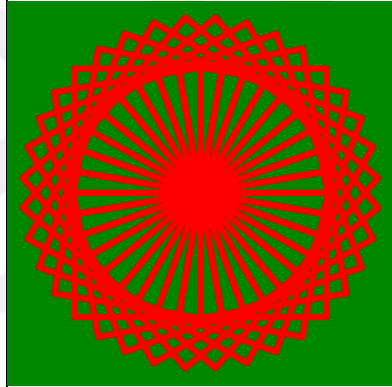
Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet



```
i=0
while (i<6):
    k=0
    while (k<2):
        kalem.right(90)
        kalem.fd(60)
        kalem.right(45)
        kalem.fd(20)
        kalem.right(45)
        kalem.fd(60)
        k=k+1
    kalem.right(60)
    i=i+1
```

Öğrencilerden çıktısı yukarıdaki gibi olan bir uygulama yapmaları istenir ve gerekli zaman verilerek iç içe while döngüsünü kullanmaları sağlanır.



```
i=0
while (i<36):
    k=0
    while (k<4):
        kalem.fd(100)
        kalem.left(90)
        k=k+1
    kalem.right(10)
    i=i+1
```

Öğrencilerden çıktısı yukarıdaki gibi olan uygulamayı yapmaları istenir. Gerekli zaman ve ipuçları verilerek doğru kodlara ulaşmaları sağlanır.

Tekrar yapıldıktan sonra ders bitirilir.

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

Yedinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Dersin başında bir önceki haftada işlenen konulara kısaca değinildikten sonra Python’da bu hafta Input komutunu göreceğimiz söylenir ve ne olabileceği öğrencilere sorulur. Öğrencilere verilen ipuçları sayesinde öğrencilerin veri girişi yapmak için kullanıldığını keşfetmeleri sağlanır. <pre>input("Bir sayı giriniz:")</pre> şeklinde kullanıldığı söylenir. Daha sonra bu komutla dışarıdan aldığımız veriyi kontrol etmek için bir değişkene atadığımız belirtilir. <pre>a=input("Bir sayı giriniz:")</pre> Bir sayı giriniz: şeklinde olduğu gösterilir ve çok kısa bir zaman verilerek öğrencilerin veri girişi yapmaları sağlanır. Daha sonra Python’da şartlara bağlı olarak durumların kontrol edildiği ve İF ELSE yapısını göreceğimiz söylenir. Öğrencilere, “Nasıl kullanılabilir?” diye sorulur ve tartışılır. <pre>if şart : şartın gerçekleştiği durum elif şart: şartın gerçekleştiği durum elif şart: şartın gerçekleştiği durum else: şartın gerçekleşmediği durum</pre> Bir şartın ya da birden fazla şartın olduğu durumlar gösterilir. Daha sonra öğrencilerden klavyeden girilen bir sayının negative ya da pozitif olduğunu söyleyen python programını yazmaları istenir ve ipuçları verilerek rehberlik edilir. Örnek yapıp bittikten sonra iki operator olduğu söylenir. Pythonda VE,VEYA yani AND,OR operatörleri olduğu ifade edildikten sonra bu kavramların ne anlama geldiği sorulur ve üzerinde tartışılır. Verilecek ipuçlarından hareketle öğrencilerin verilen şartlardan hepsinin gerçekleşme durumuna AND, herhangi bir şartın gerçekleşem durumuna OR denildiğini söylemeleri sağlanır. Daha sonra klavyeden girilen 3 sayıyı büyükten küçüğe doğru sıralayan program yazmaları istenir ve ipuçları verilerek rehberlik edilir. Ders tekrar edildikten sonra bitirilir.
---	---

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	Sekizinci hafta ders planı

Ek 6: Scratch Ortami İçin Hazirlanan Ders Planlari

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.1 Blok Tabanlı Programlamanın arayüzünü ve özelliklerini tanır.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse girildiğinde öğrencilere, “Blok tabanlı programlama kavramını daha önceden duyan var mı?” diye sorulur ve öğrencilere konuşmaları için fırsat verilir. Gelen cevaplara göre, blok tabanlı programlamanın ne olduğu ve dünyada en çok Blockly ve Scratch uygulamasının kullanıldığı söylenir. “ Scratch programını daha önce duyan var mı ? Scratch programında neler yapılabilir? Siz neler yapmak isterdiniz?”diye sorulur ve öğrencilerden gelen cevaplar beklenir. Cevaplar doğrultusunda tartışmalar yapıldıktan sonra, oyunlar,hikayeler, animasyonlar yaratılabileceği ifade edilir. Scratch arayüzünün online Scratch , offline scratch olarak iki türlü kullanılabileceği belirtilir. Daha sonra online sctach açılarak öğrencilere gösterilir. Scract offline editor indirilip bilgisayarlara kurulum yapılır.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Sonra arayüz açılıp öğrencilere sahne, sahne değiştirme, sahne çizme, kuklaların olduğu ekran, kukla ekleme, kukla çizme, ses ekleme, kod blokları, renklerine göre ve kodların mantıksal bir sıraya göre yapılandırıldığı bölüm gösterilir; öğrencilere anlatılır.

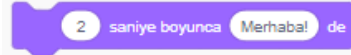
Sonra bir uygulama yapacağımız söylenir, eklenilen kuklanın ilk önce hareketinin sağlanması gerektiği ifade edilip, sizce hangi renk kod bloğundan seçeceğim kodla hareket ettiririm diye sorulur. Öğrencilere verilen zaman zarfında mavi renkli kod bloğunun altında olan



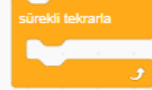
kod bloğunu bulup kodları sürüklediğim yere sürüklemeleri sağlandıktan sonra “Kodu nasıl çalıştırabiliriz?” sorusu sorulup, öğrencilerin çift tıklama cevabını vermelerini sağlanır. “Kodları çalıştırdıktan sonra çift tıklama dışında yeşil bayrağa



tıklanarak da kodlar çalıştırılır.” deyip öğrencilerin kodunu bulmaları sağlanır. Daha sonra “Sizce bu kukla konuşur mu? Konuşursa hangi kod bloğu ile konuşur?” şeklinde soru sorulup



öğrencilerin mor blok altından bloğunu bulmalarını istenir.. Kod bloklarını birleştirerek öğrencilerden kod bloklarını çalıştırmaları istenir. Öğrenciler kodları çalıştırdıktan sonra bu kodların bir defa çalıştığını öğrencilerin görmesini sağlayıp “Nasıl sürekli hale getiririz?” diye sorulur. Bu kod bloğunun hangi renkli



blokta olduğunu bulmaları istenerek öğrencilerin kod bloğunu bulmaları beklenir. “Kod bloğunu bulduktan sonra kodları nereye koymalıyız?” şeklinde bir soru sorularak, öğrencilerin kodları bütün haline getirerek çalıştırmaları söylenir. Daha sonra adım sayısını ve saniyeyi gerekli yerlerden değiştirecekleri söylenir. Hareket, olaylar, görünüm, kontrol, ses, algılama, işlemler, veri ve özel taşlar blokları açıklanarak konu sonlandırılır. Konu tekrar edilerek ders bitirilir.

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

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.2 Blok Tabanlı Programlama aracında sunulan programın işlevini açıklar. 6.5.2.3 Blok Tabanlı Programlama aracında sunulan bir programın hatalarını ayıklar 6.5.2.4 Blok Tabanlı Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.
Süre	40 Dakika + 40 Dakika

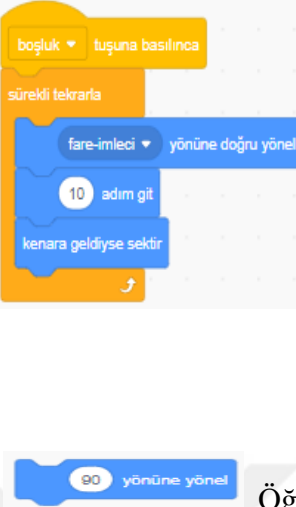
Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğretmen derse girdikten sonra Scratch da bulunan kod blokları arasından 10 tanesini etkileşimli tahtaya yansıtır ve sadece bu 10 tane kod bloğunu kullanarak 15-20 dk. içerisinde bir proje üretmelerini söyler. Onları başka kuklalar ve arka planları kullanmaya teşvik ederek ipuçlarını verir.

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Daha sonra öğrencilerden gelen dönütler sonrasında onlara “ Kısıtlı sayıda kod bloklarıyla proje üretmenin zor yanları nelerdir?”sorusu sorularak onlarla konuşulur. Öğrencilerden gelen cevaplar doğrultusunda, “ Kısıtlı sayıda kod bloklarıyla proje üretmenin kolay yanları nelerdir?” sorusu sorulup üzerinde tartışılır. Öğrencilerin kod bloklarını keşfetmeleri sağlanmalıdır. Öğretmen; “ İki haftadır Scratch programına başladık, beraber keşfetmeye çalışıyoruz. Zamanla beraber projeler yapıp bunları kişisel bir hale getireceğiz. Scratch Programında proje hazırlarken kendinizi bir tiyatro yönetmeni gibi hissedebilirsiniz. Tiyatroda olduğu gibi , karakterler(kuklalar) ,kostümler(kılık), dekor ,sahne ve projenin bir senaryosu vardır. “ diyerek öğrencilerle uygulamalara geçilir. İlk olarak öğrencilere sahneye ilk gelen kuklanın nasıl silindiğini ve daha sonra başka bir kuklanın nasıl eklenildiği gösterilir. Daha sonra eklenen kuklayı fare okuyla hareket ettirmek için kodlarını ekrana ekleyip çift tıkladığımızda neler oluyor ?diye sorulur. Öğrencilerin verdiği cevaplardan sonra şimdi kodlara çift tıklama yerine klavyeden bir tuşa bastığı mızda olması için kodunu alıp 172diğer kodlarla birleştirmesini sağlanıp boşluk tuşuna bastığında deneme yapması sağlanır. Daha sonra boşluk tuşuna bastığımızda daha çok hareket etmesi için yazılan kodların aynısını daha fazla yazarak daha uzun 172süre gitmesini sağlayabileceğimiz ifade edilir. Burada öğrencilere aynı kodları kod üzerinde sağ tıklayarak kopyasını çıkart komutuyla yapabileceği söylenir.“Bu kodları bu şekilde yazmak uzun sürebilir. Başka bir yolu var mı? diye sorulur.Öğrencilerin“Sürekli Tekrarla”komutunu keşfedip kod bloklarını birleştirmeleri sağlanır. Öğrencilere,” Fare imlecini daha hızlı nasıl takip edebiliriz?” diye sorularak adım sayısını değiştirmeyi keşfetmeleri sağlanır. Ardından,”Değeri negatif bir değer(-5) yaparsak ne olur?” diye sorulur ve denemeleri sağlanır. Programı durdurmak için kırmızı altıgen işaretine basmalarını söylenir. Fare okunun ekranın dışına çıktığında nasıl tepkiler verdiği sorulur ve “Ne yapabiliriz?” diye ipuçları verilir, öğrencilerin kod boluğunu bulmaları sağlanır..
---	---

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	 Daha sonra kod bloklarını birleştirerek öğrencilere kuklanın rahatça fare okuyla beraber hareket etmesini sağladıkları açıklanır. Projedeki kuklanın belirli bir kurala göre hareket etmesi ve öğrencilerin arka plana bir labirent çizmesi sağlanır. Daha sonra kuklanın sahne içerisinde istenilen yöne dönebileceği kod bloğunun hangisi olduğu sorulur. ve öğrencilere belirli bir süre verilip cevapları beklenir . Öğrencilerin kodu bulmalarını sağlandıktan sonra kuklanın yön tuşlarına basınca bastığımız yöne dönüp o yöne doğru hareket etmesini sağlayan projeyi yapmaları istenir ve ipuçları verilerek yapmaları sağlanır. Daha sonra öğrencilere, "Projeyi daha da zengin hale getirmek için hangi hamleleri kullanabiliriz?" diye sorulur ve öğrencilerin cevapları üzerine tartışılır. Tekrar yapılarak konu bitirilir.
---	---

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	İkinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.3 Blok Tabanlı Programlama aracında sunulan bir programın hatalarını ayıklar 6.5.2.4 Blok Tabanlı Programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler. 6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar.
Süre	40 Dakika + 40 Dakika

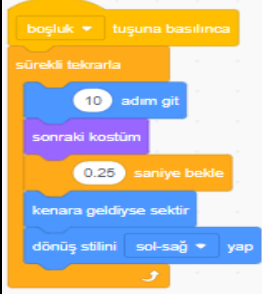
Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Derse başladıktan sonra öğrencilere, “Animasyon nedir, bildiğiniz animasyonlar var mıdır?” diye sorulur ve öğrencilerden yanıtlar alınır. Öğrenciler kendi aralarında tartıştıktan sonra aslında animasyon hareketsiz resimlerin farklı açıdan çekilen resimlerin bir karede birleştirilmesi olarak tanımlanabileceği ifade edilir.. Öğrencilerin çizgi filmlerin de bir animasyon olduğunu anlamaları sağlanır. Scratch programında da benzer animasyonların yapılabileceğın burada bulunan kuklaların farklı açıdan çekilmiş resimlerinin olduğu, bunlara da kılık adı verildiği söylenir. Bu kılıkları birleştirirsek animasyon olacağını belirtilir.

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Öğrencilerle eş zamanlı olarak bir yarasanın kanat çırparak uçacağını gösteren bir animasyon yapılacağı söylenir. İlk olarak ekrandaki kedinin silinip kukla kütüphanesinden yarasanın eklenmesi sağlanır. Daha sonra kostümler sekmesinde yarasaya ait olan kostümleri (kılıkları) göstererek hızlı bir şekilde kuklalar arası geçiş yapıldığında yarasanın nasıl hareketlendiğini görmeleri sağlanır. Bunları kod blokları ile yapacakları öğrencilere hatırlatılır. Kuklayı ekledikten sonra kuklanın bir sahnede uçuşması gerektiğini söyleyerek öğrencilere sahne kısmından bir arka plan seçmeleri söylenir ve seçtiklerinden emin olunur. Öğrencilere sahnelerini kendi isteklerine göre hayal edip dizayn etmeleri söylenir ve 5 dakika zaman verilir. Sahneye eklenecek kuklaların program üzerindeki boyutunun nasıl ayarlandığı ve nasıl çoğaltıldığı gösterilir. Öğrencilere bu üç bloğu vererek deneme yapmaları ,boşluk tuşuna bastığımızda yarasanın sürekli kanat çırpmasını ve ilerlemesini sağlayan kodları birleştirmeleri gerektiği söylenir. Ama burada sürekli boşluk tuşuna basmak gerekir. “Bunu sürekli yapmak için neler yapabiliriz?” sorusu sorularak öğrencilerin bloğunu keşfedip , vereceğiniz ipuçları ile kod bloklarını “sürekli tekrarla” bloğu içerisine koymaları sağlanır. Boşluk tuşuna bastıklarında yarasanın hareket ettiğini görmeleri ve yarasanın hareketlerinin çok hızlı olduğunu farketmeleri ve bunun bir sorun olduğunu size söylemeleri sağlanmalıdır. “ Ne yapabiliriz?” diye sorulur ve öğrencilere bloğunun bu işi yapabileceği söylenerek, bloğu birleştirilen kod blokları arasında nereye koymaları gerektiğini bulmaları ifade edilip, vereceğiniz ipuçları ile bloğu doğru yere yerleştirmeleri sağlanır.“Oradaki saniyenin değişmesi sizce nasıl bir etki yapar?” diye sorulur ve etkiyi görmeleri sağlanarak yarasayı istedikleri hızda uçurmaları sağlanır. Bir önceki hafta görmüş oldukları kod bloğunu ekleyerek kenarlardan çıkmaması ve sahnede kalması istenir ve öğrencilerin yaptığından emin olunur. Öğrencilerin yarasanın uçup kenara çarptıktan sonra ters dönerek uçmaya devam ettiğini görmeleri ve öğrenciler tarafından farkedilmesi sağlanır. Bunun için dönüş stilini seçmeleri gerektiği söylenir ve bloğu doğru yere yerleştirmeleri sağlanır.
---	---

Öğretme – Öğrenme Etkinlikleri

• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	 Öğrencilerin kod bloklarının son halini bu şekilde yapması sağlanır ve öğrencilere,”Farklı olarak neler yapabiliriz?”sorusu sorularak “ Kuklaları ve sahne arkaplanını değiştirerek başka animasyonlar yapabiliriz.”demeleri sağlanır. Sonra öğrencilerden bir akvaryum yapmaları ve bu akvaryumda köpek balıkları ve küçük balıkların olması, arka planı da buna uygun dizayn ederek animasyon yapmaları istenir. Zamanın durumuna göre sonra sahilde yürüyen bir insan animasyonu yapmaları istenebilir. Ders tekrar edilerek birilir.	Öğrencilerin kod bloklarının son halini bu şekilde yapması sağlanır ve öğrencilere,”Farklı olarak neler yapabiliriz?”sorusu sorularak “ Kuklaları ve sahne arkaplanını değiştirerek başka animasyonlar yapabiliriz.”demeleri sağlanır.
---	--	---

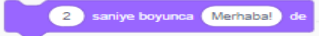
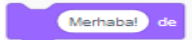
Ölçme – Değerlendirme

• 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	Üçüncü hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar.
Süre	40 Dakika + 40 Dakika

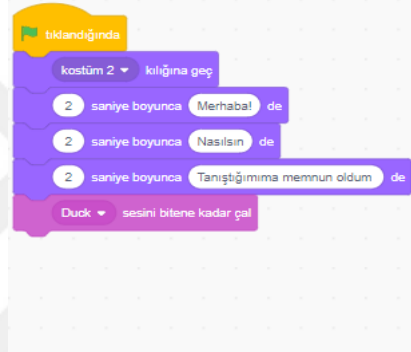
Ünite Kavramları ve Sembolleri / Davranış Örüntüsü	Algoritma oluşturma mantığı, adım adım düşünme
Güvenlik Önlemleri (Varsa)	
Öğretme – Öğrenme Yöntem ve Teknikleri	Anlatım, soru-cevap, tartışma, gösterip yaptırma
Kullanılan Eğitim Tek, Araç – Gereçler ve Kaynakça	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri	
• Dikkat çekme • Güdüleme • Gözden geçirme • Derse geçiş • Bireysel öğrenme etkinlikleri • Grupla öğrenme etkinlikleri • Özet	Ders başladıktan sonra öğrencilere geçen derste ilk defa bir kuklanın kılık değiştirerek hareket etmesini yani animasyon yapmayı öğrendikleri hatırlatılır,bugün ise bir tanışma animasyonunu yapacakları söylenir. Daha sonra iki tane hayvan resmi olan kukla eklemeleri ve iki kuklanın sahne üzerinde birbiri ile konuşacak şekilde birbirlerine dönmelerini sağlamaları istenir. Kuklaları ekledikten sonra konuşmak için hangi blokların kullanılabileceğini sorulur ve öğrencilerden cevap beklenir. Öğrencilerin  kod bloğu ve  kod bloklarını keşfetmeleri sağlanır. “Sizce ikisi arasındaki fark ne olabilir?” diye sorulur.

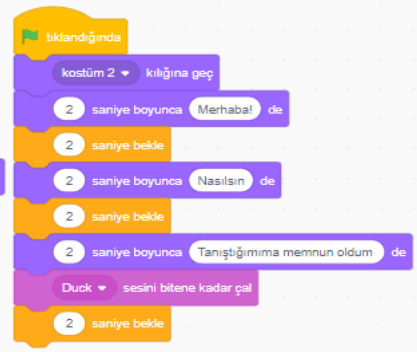
Öğretme – Öğrenme Etkinlikleri

Öğrencilerin kod bloklarını kuklalar üzerinde deneyerek birinci kod bloğunun 2 saniye sonra kaybolduğunu diğer kod bloğunun ise hiç ekrandan kaybolmadığını görmeleri için ipuçları verilir. Öğrencilere aynı zamanda kuklaların ses çıkarabildiklerini söylenir ve öğrencilerin  kod bloğunu keşfetmeleri sağlanır. Daha sonra eklenen hayvan kuklalarının birbiri ile tanışmaları için  yeşil bayrağa tıklanınca kod bloğu ile başlamaları ve kukları konuşturmaları gereken kod bloklarını kendi uydurdukları bir senaryo ile hem konuşarak hem kılık değiştirerek ve aynı zaman da ses çıkararak sağlayan uygulamayı yapmaları istenir.

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

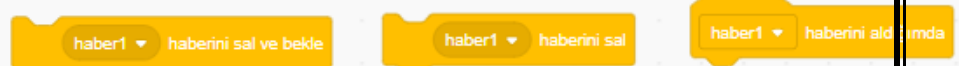


Şekil 1



Şekil 2

Yukarıda görülen Şekil 1 deki gibi her kukla için yazılan kod blokları çalıştırıldığında öğrenciler tarafından aynı anda görüldüğünü görmeleri sağlanır ve öğrencileri tanışma senaryosu için Şekil 2 deki gibi kodları hazırlamalarını ve keşfetmelerini sağlayacak ipuçları verilir. Hikaye oluşturulduktan sonra “Bu animasyonu kod bloğunu kullanmadan nasıl yapabiliriz?” diye sorulur ve gerekli zaman zarfında öğrencilerin düşünmeleri sağlanır. Öğrencilere verilen ipuçları sayesinde



Blok 1

Blok 2

Blok 3

Yukarıda verilen kod bloklarını keşfetmeleri sağlanır. Daha sonra bu üç kod bloğunun ne işe yaradığı sorulur ve öğrenciler kod bloklarını denemeleri için yönlendirilir. Kod bloklarından Blok 1’in haber gönderip beklediğini, Blok 2’nin haberi sadece gönderdiğini, Blok 3’te haber geldikten sonra işlem yapacağını keşfettiklerinden emin olunur.

Öğretme – Öğrenme Etkinlikleri

Biraz önceki senaryonun bu kod blokları ile nasıl olacağını sorulur ve gerekli zamanı vererek öğrencilerin denemesi ve şekil 3’ teki gibi kod blokları ile başlamaları sağlanır.



Şekil 3

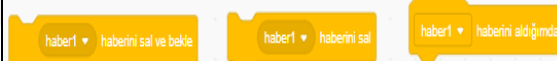
- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Daha sonra öğrencilere, “Kuklalara nasıl kod yazıyoruz, sahnelere de kod yazarak değiştirebilir miyiz?” diye sorulur ve birden fazla sahne eklemeleri söylenir.



kod bloklarını keşfetmeleri sağlanır.

Etkinlik bittikten sonra öğrencilerden bu defa



bloklarını kullanarak iki insanın kılık değiştirerek ve sahne değiştirerek birbiri ile konuşmalarını sağlayan projeyi yapmaları istenir. Projenin yapım aşamasında gerekli yerler de öğrencilere destek olunur.

Ders tekrar edildikten sonra bitirilir.

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

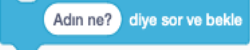
Dördüncü hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

Dersin başında öğrencilere, “Geçen hafta kuklaları konuşturmuştuk, bu hafta soru sormalarını sağlayabilir miyiz?” sorusu yöneltilerek soru sorma kodunu bulmaları için zaman verilir. Verilen ipuçları ile  kodunu keşfetmeleri sağlanır. Daha sonra kukla ve sahne ayarlamaları yapmaları istenir. “Verilen cevaplar sizce nerede tutulmaktadır?” sorusu sorularak ve öğrencilerin  kod bloğunu bulmaları sağlanır. Kukla ve sahne ayarlamaları bittikten sonra öğrencilerden kuklanın soru sorması ve cevap vermesi istenir. Öğrencilere zaman verilir ve ipucu vererek öğrencilerin aşağıdaki kod bloklarını keşfetmeleri sağlanır.

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet



Yandaki kod bloğuna ulaşan öğrencilere, “Eğer verilen yanıt ile ilgili bir şeyler yazacaksam ne yazabilirim?” diye sorulur.

Öğrencilere zaman ve ipucu verilerek  kod bloğuna ulaşmaları sağlanır. Öğrencilere soru sorulduktan sonra mesaj olarak “**Merhaba YANIT memnun oldum.**” mesajını vermeleri istenir ve aşağıdaki kod bloğuna ulaşmaları sağlanır.



“Öğrencilere sorduğumuz sorulara göre cevap verebilir miyiz ?” sorusu sorulur ve üzerinde tartışılır. Daha sonra verilen cevaplara göre öğrencilerin şarta bağlı olan kodları keşfetmeleri sağlanır.



Öğrencilerin şartlara bağlı olan kodları keşfetmesini sağlandıktan sonra şartları kolay yazmak için



bloğundaki kodları kullanmaları gerektiği hatırlatılır.

Öğrencilere “**sınavdan aldığın nota göre dersten geçtin ya da kaldın**” mesajı veren uygulamayı yapmaları için zaman verilir. Öğrencilerin aşağıdaki kod bloklarına ulaşması sağlanır.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet



Yandaki kod bloğuna ulaşan öğrencilere, “Eğer bloğunu başka türlü kullanabilir miyiz?” diye sorulur. Öğrencilerin iki şarta bağlı olan uygulamalarda ya iki tane eğer bloğu ya da eğer ise değilse bloğu kullanacaklarını anlamaları sağlanır.

Öğrencilere kod bloklarının anlamı sorulur. Verilecek ipuçları ile öğrencilerin verilen şartların tamamının gerçekleştirilme ihtimalinin kod bloğu ile, verilen şartlardan sadece bir tanesinin gerçekleştirilme ihtimalinin ise kod bloğu ile gerçekleşeceğini keşfetmeleri sağlanır. Daha sonra öğrencilere kuklaya dersten aldığı notu sorup buna göre;

0-45	Zayıf
45-55	Geçer
55-70	Orta
70-85	İyi
85-100	Çok iyi

Ders tekrar edilerek bitirilir.

Yandaki tabloya uygun olarak kuklanın sorduğu not sorusuna göre yandaki şartları sağlayan uygulamayı yapmaları istenir ve gerekli zaman verilir.

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

Beşinci hafta ders planı

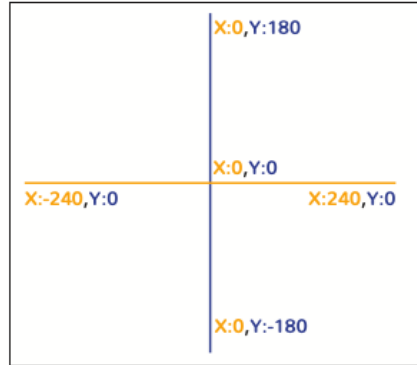
Orhan Gazi Ortaokulu
2018-2019 Eğitim-Öğretim Yılı 2. Dönem
Bilişim Teknolojileri ve Yazılım Dersi 6. Sınıflar
Günlük Planı

Dersin Adı	Bilişim Teknolojileri ve Yazılım
Sınıf	6. Sınıflar
Ünite Adı	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

Öğretmen dersin başında geçen haftanın konusunu tekrarladıktan sonra öğrencilere bu haftaki konunun kuklaların x ve y konumları olduğunu ve kodlamanın sadece bilgisayar olmadığını aynı zamanda matematik bilgisini de geliştirdiğini ve koordinat düzlemini bilmeleri gerektiğini söyler.



- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

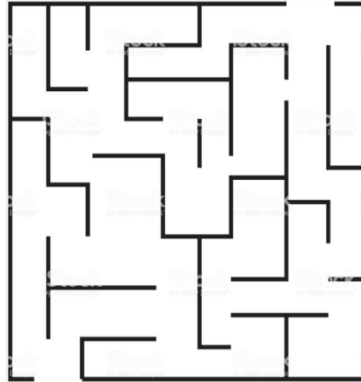
Yukarıda görülen koordinat düzleminin Scratch programına ait olduğu belirtilerek buradaki rakamların ne olduğu sorulur. Öğrencilerden kuklaların ekranda gidebileceği son mesafe olduğunu keşfedene kadar ipuçları verilir. Daha sonra rakamların önündeki matematiksel ifadelerin ne olduğunu sorulur ve öğrencilerin konum bölgelerine göre işaretlerin olduğunu ve kuklaların hareketlerine göre matematiksel ifadelerin değişeceğini keşfetmeleri sağlanır. Öğrencilere, “Kuklalar yatay olarak hareket ederse hangi eksen de hareket eder, dikey eksen de hareket ederse hangi eksen de hareket eder?” soruları yöneltilir ve kod blokları arasında bu değerlere karşı gelen kod bloklarını bulmaları istenir.



Kod bloklarını keşfederek öğrencilerin kuklaları hareket ettirmenin adım atmak dışında konum değiştirme ile de yapıldığını iyice kavramaları sağlanır. Belirli bir saniyede de konumlarını değiştirdiklerini oradaki saniyenin değiştirme hızı olduğunu kavradıktan sonra uygulama yapacakları söylenir.

Öğretme – Öğrenme Etkinlikleri

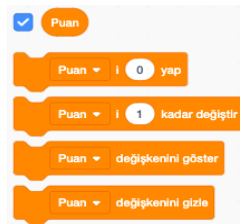
- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet



Öğrencilere kuklayı yön tuşları ile hareket ettirerek labirentin bir ucundan bir ucuna götürmeleri gerektiği, siyah çizgilere çarptığında kuklanın tekrar başladığı yere dönmesi, kuklanın hareket ederken kılık değişmesi ve labirentin sonuna geldiğinde “kazandın” mesajı vermelerini sağlayan uygulamayı yazmaları söylenir. Öğrencilere rehberlik ederek x ve y konumlarını değiştirerek kullanmaları sağlanır.



yandaki kod blokları ipuçları vererek hatılatılır. Daha sonra etkinliği tüm öğrenciler bitirdikten sonra öğrencilere, “Sizce bu oyuna puan ekleyebilir miyiz?” diye sorulur ve öğrenciler cevap vermeleri konusunda cesaretlendirilir. Daha sonra değişken denen bir kavramın olduğu bunu bir kutuya benzetmeleri gerektiği, kutu içerisine su doldurduğumuzda kutudaki su miktarının arttığı veya kutudan su aldığımızda su miktarının azaldığı söylenir. Kutunun içindeki değer sürekli değişmesinden dolayı değişken olduğunu ve bu oyuna puan eklemek için bir değişken tanımlamamız gerektiği söylenir.



Yandaki kod blokları gibi değişken oluşturulduğu ve değişkenin başlangıç değerini istediğimiz gibi değiştirebildiğimiz, bunun için yandaki kod bloklarını kullandığımız anlatılır. Sonra yukarıdaki labirent oyununa puan eklemek için öğrenciler teşvik edilir. “Değişkeni azaltmak için ne yapabiliriz?” diye sorulur ve öğrencilerin –(eksi) değeri söylemeleri sağlanır. Değişken değerinin labirentin sonuna geldiğinde arttığını ve siyah çizgiye değdiğinde azaltılması gerektiği söylenir ve tekrar yapıldıktan sonra ders bitirilir.

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

Altıncı hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

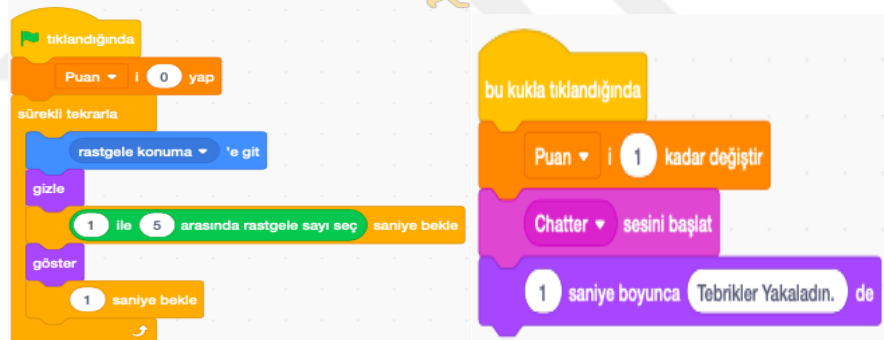
- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

Derse başladıktan sonra geçen haftaki konular tekrar edilip yeni konuya geçilir. Bugün eş zamanlı olarak bir oyun yapılacağı, daha sonra kendilerinin de bir oyun yapacakları söylenir.

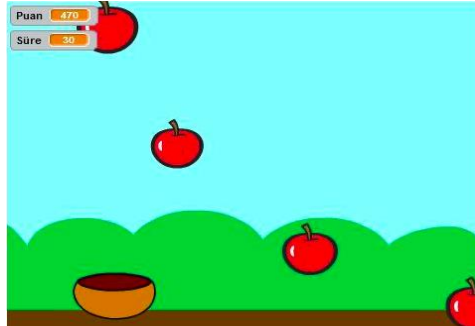
Oyunumuzun adının “**Beni Yakala**” olduğu ve bir karakterin sürekli olarak ekranda belirli aralıklarla gözükp gizlendiği ve biz karaktere tıkladığımız zaman puan kazanmamız gereken oyunu yazacakları söylenir.

Daha sonra sahneye herhangi bir karakter ekleyip ve arka plan eklemeleri gerektiği söylenir. Ekledikten sonra sahnedeki karakterin bizim belirlediğimiz aralıkta rastgele olarak gözükp gizlenmesini nasıl sağlayabiliriz, diye sorulur ve öğrencilere zaman verilir.

Öğrencilerin belirli bir zaman sonra kod bloğuna ulaşmaları için ipuçları verilir. Kod bloğuna ulaşıldıktan sonra kod bloğunun verilen aralıklarda rastgele olarak sayı üretmesini sağladığı konusunda hem fikir olunmalıdır. Oyunda puan olacağı için öğrencilere ne eklememiz gerekir, diye sorulur ve değişken cevabını vermeleri sağlanır. Öğrencilere belirli zaman ve ipuçları verilerek oyunu kodlamaları ve aşağıdaki kod bloklarına ulaşmaları sağlanır.



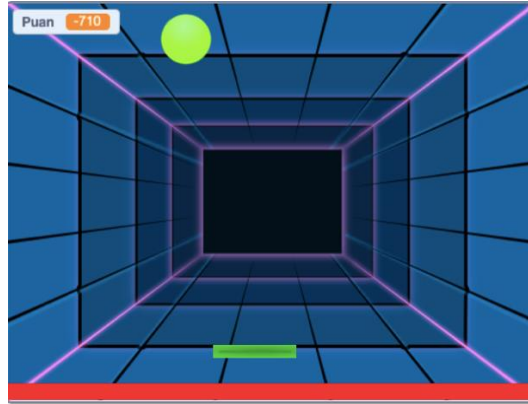
Daha sonra ekran çıktısının aşağıdaki gibi olması gereken elma toplama oyununu yapmaları istenir.



Yukarıda ekran çıktısı olan elma toplama oyununda belirli bir sürede sepete toplanan elma kadar puan alınmasını sağlayan bir oyun yapmaları söylenir ve öğrencilere gerekli zaman verilir. Bu zaman zarfında öğrencilere rehberlik edilerek oyunu yazmaları sağlanır ve zaman kalırsa oynamaları için fırsat verilir.

Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet



Dersteki zaman durumuna göre alternatif olarak Pong game oyunu da yaptırılabilir. Pong game oyununda top rastgele olarak ekran içerisinde gezerek kenarlara geldiğinde seker ve aşağıdaki kırmızı zemine değmemesi için yeşil renkli olan çubukla engellenmeye çalışılır. Yeşil renkli çubuğa çarptığında rastgele renk ve yön değiştirmesi sağlanır. Tekrar yapılarak ders bitirilir.

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

Yedinci hafta ders planı

Orhan Gazi Ortaokulu
2018-2019 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	6. ÜNİTE: Problem Çözme, Programlama ve Özgün Ürün Geliştirme
Konu	6.5.2.5 Doğrusal mantık yapısını içeren programlar oluşturur. 6.5.2.6 Doğrusal mantık yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.7 Karar yapısını içeren programlar oluşturur. 6.5.2.8 Karar yapısı içeren programları test ederek hatalarını ayıklar. 6.5.2.9.Çoklu karar yapılarını içeren programlar oluturur. 6.5.2.10.Çok karar yapılarını içeren programları test ederek hatalarını ayıklar. 6.5.2.11.Döngü yapısını içeren programlar oluşturur. 6.5.2.12.Döngü yapısını içeren programları test ederek hatalarını ayıklar. 6.5.2.13 Bir algoritmayı uyarlamak için en uygun karar yapısını seçer. 6.5.2.14 Farklı programlama yapılarını kullanarak karmaşık problemlere çözüm üretir.
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	Bilgisayar, Etkileşimli Tahta

Öğretme – Öğrenme Etkinlikleri

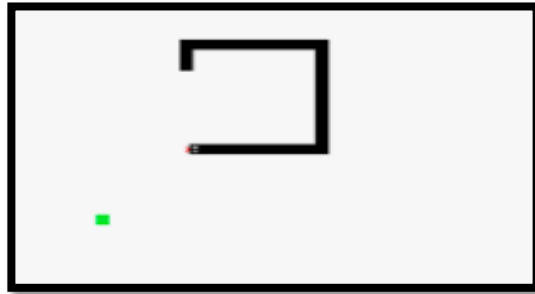
Dersin başında geçen hafta işlenen konular tekrar edildikten sonra yeni kod bloklarını göreceğimiz söylenir. Daha önce şartlara bağlı olarak bazı koşul ifadeleri kullandığımız söylenip yeni bir ifade kullanacağımız belirtilir. Daha sonra, “Önceden telefon kullanımları daha az iken insanlar nasıl haberleşirlerdi?” sorusu yöneltilir. Öğrencilerin mektupla demeleri için ipuçları verilir. Daha sonra “Kodlar arasında da mektuplaşma olabilir mi?” diye sorulur ve öğrencilerin kod bloklarını karıştırmaları için zaman verilir. Öğrencilerin zaman içerisinde



- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet

kodlarına ulaşmaları sağlanmalıdır. Burada yapılacak animasyonların olaylara göre haberleştikleri söylenerek öğrencilerin konuyu kavramaları sağlanır.

“**Haber Sal**” komutunu kullanarak eski telefonlarda oynadığımız yılın oyununu yapmaları istenir. Bu süreçte öğrencilere rehberlik edilerek ipuçları verilir.



Öğretme – Öğrenme Etkinlikleri

- Dikkat çekme
- Güdüleme
- Gözden geçirme
- Derse geçiş
- Bireysel öğrenme etkinlikleri
- Grupla öğrenme etkinlikleri
- Özet



Öğrencilerden çıktısı yukarıdaki gibi olan oyunu yapmaları istenir ve kurbağanın hiç bir araca çarpmadan yön tuşları ile kırmızı bayrağa gelmesini sağlamaları gerektiği ve oyunun tamamında üç tane can olduğu söylenir ve yeterli zaman verilerek öğrencilere rehberlik edilir.

Tekrar yapılarak ders bitirilir.

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

Sekizinci hafta ders planı

Ek 7: Etik Kurul İzin



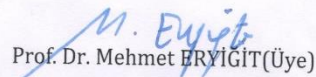
Bolu Abant İzzet Baysal Üniversitesi
Sosyal Bilimlerde İnsan Araştırmaları Etik Kurulu

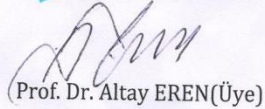
Doç. Dr. İbrahim ÇETİN¹
Tarık OTU²
 Bolu Abant İzzet Baysal Üniversitesi
 Eğitim Fakültesi
 Matematik ve Fen Bilimleri Eğitimi¹
 Bilgisayar ve Öğretim Teknolojileri Eğitimi²

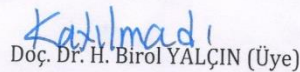
Sayın **Doç. Dr. İbrahim ÇETİN¹**
Tarık OTU²

"Kodlama Ortamlarının Ortaokul Öğrencilerinin Başarı ve Tutumlarına Etkisi" İnsan Araştırmaları Etik Kuruluna yapmış olduğunuz başvuru (Protokol NO. 2019/08) kurulumuzun 07.02.2019 tarihli ve 2019/01 toplantısında değerlendirilerek etik olarak uygun bulunmuştur. Bilgilerinize sunarız.

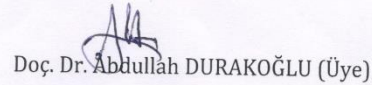

 Prof. Dr. Hamit COŞKUN (Başkan)

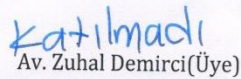

 Prof. Dr. Mehmet ERYİĞİT (Üye)


 Prof. Dr. Altay EREN (Üye)


 Doç. Dr. H. Birol YALÇIN (Üye)


 Doç. Dr. Seval ALKOY (Üye)


 Doç. Dr. Abdullah DURAKOĞLU (Üye)


 Av. Zuhale Demirci (Üye)

Ek 8: Meb İzin Belgesi

	T.C. BOLU VALİLİĞİ İl Millî Eğitim Müdürlüğü
Sayı : 39307281-605.01-E.7819668 Konu : Araştırma İzni (Tarık OTU)	17/04/2019
VALİLİK MAKAMINA	
İlgi : a) Abant İzzet Baysal Üniversitesinin 15.04.2019 tarih ve 7554769 sayılı yazısı. b) Milli Eğitim Bakanlığının 22.08.2017 tarih ve 35558626-10.06.01-E.12607291 (2017/25) sayılı genelgesi.	
Abant İzzet Baysal Üniversitesi Eğitim Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Anabilim Dalı, Yüksek Lisans Programı öğrencisi Tarık OTU'nun "Kodlama Ortamlarının Ortaokul Öğrencilerinin Başarı ve Tutumlarına Etkisi" konulu araştırma çalışmasına veri sağlamak amacıyla Müdürlüğümüze bağlı Orhangazi Ortaokulu 6. Sınıf öğrencilerine anket uygulama talebine ilişkin ilgi (a) yazı ve ekleri incelenmiştir.	
Söz konusu uygulamanın; Türkiye Cumhuriyeti Anayasası, Milli Eğitim Temel Kanunu ile Türk Milli Eğitiminin genel amaçlarına uygun olarak yürürlükte olan tüm yasal düzenlemelerde belirtilen ilke, esas ve amaçlara aykırılık teşkil etmeyecek şekilde, denetimi ilgili okul müdürlükleri tarafından gerçekleştirilmek üzere, eğitim öğretimin aksatılmaması kaydıyla ve ilgi (b) Genelge doğrultusunda uygulanması müdürlüğümüzce uygun görülmektedir.	
Makamlarınızca da uygun görülmesi halinde olurlarınıza arz ederim.	
Yasin TEPE Millî Eğitim Müdürü	
OLUR 17/04/2019	
Çağlayan KAYA Vali a. Vali Yardımcısı	
Adres: Aktaş mah. Şehit Güven Keskin Cad. No:20 Merkez/BOLU Elektronik Ağ: bolu.meb.gov.tr e-posta: stratejigelistirme14@meb.gov.tr	
Bilgi için: Tuba ÇETİN Tel: 0 (374) 280 14 43 Faks: 0 (374) 280 14 50	
Bu evrak güvenli elektronik imza ile imzalanmıştır. https://evraksorgu.meb.gov.tr adresinden e888-745f-3916-927c-b803 kodu ile teyit edilebilir.	

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Tarık OTU
 Doğum Yeri ve Tarihi : Arpaçay/KARS, 11.11.1984
 Uyruğu : T.C.
 Medeni Durum : Evli
 E-posta : tarikotu@gmail.com

EĞİTİM BİLGİLERİ

Derece	Yıl	Bölüm	Üniversite
Yüksek Lisans	2017-2020	Bilg. ve Öğr. Tekn. Eğit.	BAİBÜ
Lisans	2002-2006	Bilg. ve Öğr. Tekn. Eğit.	Sakarya Üniversitesi
Lise	1998-2002	Bilişim Teknolojileri	Ünye EML

İŞ DENEYİ

Yıl	İl/İlçe	Görev	Okul Adı
2006-2007	İst/Küçükçekmece	BTY Öğretmeni	Söğütlüçeşme İÖO
2007-2009	Adıyaman/Merkez	BTY Öğretmeni	Erdemir Lisesi
2009-2013	İstanbul/Şişli	BTY Öğretmeni	Halide Edip Adıvar İÖO
2013-2015	Bolu/Gerede	BTY Öğretmeni	S.M.Dayıoğlu Ortaokulu
2015-2018	Bolu/Merkez	BTY Öğretmeni	M.Akif Ersoy Ortaokulu
2018-.....	Bolu/Merkez	BTY Öğretmeni	Orhangazi Ortaokulu

AKADEMİK YAYINLAR

Çetin, İ., Otu, T. ve Oktaç, A. (2020). Bilgi İşlemsel Düşünme Testi'nin Türkçeye Uyarlanması. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*. Advanced online publication. doi: 10.16949/turkbilmat.643709