

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



GUTÜD ÖĞRETİM MODELİ ETKİLİLİĞİNİN ÇEŞİTLİ
DEĞİŞKENLER AÇISINDAN İNCELENMESİ

YÜKSEK LİSANS TEZİ

ERHAN DEMİRTAŞ

TEZ DANIŞMANI
Prof.Dr. İbrahim ÇETİN

BOLU, TEMMUZ - 2023

KABUL VE ONAY SAYFASI

Erhan DEMİRTAŞ tarafından hazırlanan “GUTÜD ÖĞRETİM MODELİ ETKİLİLİĞİNİN ÇEŞİTLİ DEĞİŞKENLER AÇISINDAN İNCELENMESİ” adlı tez çalışması jürimiz tarafından Bilgisayar ve Öğretim Teknolojileri Anabilim Dalı Bilgisayar ve Öğretim Teknolojileri Bilim Dalı’nda Yüksek Lisans tezi olarak oy birliği ile kabul edilmiştir. 26/07/2023

Jüri Üyeleri

Danışman
Prof. Dr. İbrahim ÇETİN
Bolu Abant İzzet Baysal Üniversitesi

İmza

.....

Üye
Prof. Dr. Erkan TEKİNARSLAN
Bolu Abant İzzet Baysal Üniversitesi

.....

Üye
Doç. Dr. Polat ŞENDURUR
Ondokuz Mayıs Üniversitesi

.....

Lisansüstü Eğitim Enstitüsü Onayı

Prof. Dr. İbrahim KÜRTÜL
Lisansüstü Eğitim Enstitüsü Müdürü

ETİK BEYAN

Bolu Abant İzzet Baysal Üniversitesi, Lisansüstü Eğitim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

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

aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Teze ilişkin Turnitin adlı programında enstitü müdürlüğünce belirlenen filtrelemeler uygulanarak alınmış olan benzerlik raporuna göre, tezin benzerlik oranı %30'u geçmemektedir.

Bu çalışma için Bolu Abant İzzet Baysal Üniversitesi Sosyal Bilimlerde İnsan Araştırmalar Etik Kurulundan 2021/316 sayısı ile etik izin alınmıştır.

ERHAN DEMİRTAŞ

ÖZET

GUTÜD ÖĞRETİM MODELİ ETKİLİLİĞİNİN ÇEŞİTLİ DEĞİŞKENLER AÇISINDAN İNCELENMESİ

YÜKSEK LİSANS TEZİ

ERHAN DEMİRTAŞ

BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM
DALI

BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ
BİLİM DALI

(TEZ DANIŞMANI: PROF.DR. İBRAHİM ÇETİN)

BOLU, TEMMUZ - 2023

XIII + 336

Bu araştırmanın amacı blok tabanlı bir öğretim ortamı olan Scrtach ile yapılmış olan, aynı tür etkinlikler ve içerikler bulunduran; GUTÜD öğretim modeline göre, İnşacılık modeline göre ve doğrudan öğretim modeline göre öğretim yapılarak, bu öğretim modellerinin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algılarına, kodlama başarılarına, kodlamaya yönelik tutumlarına, bilgi işlemsel düşünme becerilerine ne derece etki ettiğini belirleyebilmektir. Bu kapsam da araştırmanın çalışma grubu Bolu ilinde bulunan bir ortaokuldaki kura ile seçilmiş üç farklı şubedeki 99 öğrenciden oluşmaktadır. Rastgele seçilen sınıflara GUTÜD modeli, doğrudan öğretim modeli ve inşacılık modeli atanmıştır. Bu çalışmada yarı deneysel desen kullanılmıştır. Seçilen sınıflara ön ve son test olarak bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısı ölçeği, scratch ortamı için kodlama başarı testi, kodlamaya yönelik tutumları ölçeği, bilgi işlemsel düşünme testi uygulanmıştır. Ön testlerin uygulanması ile öğrencilere sekiz hafta boyunca ikişer saat araştırmacı tarafından hazırlanan ders planları ve etkinlikler uzman öğretmen tarafından uygulanmıştır. Toplanan veriler SPSS 26 programında analiz edilmiştir. GUTÜD, doğrudan öğretim, inşacılık modelinin; bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısı ölçeğine, bilgi işlemsel düşünme testine, ortaokul öğrencilerinin kodlamaya yönelik tutumlarına, scratch ortamı için kodlama başarı testine yapılan analizlerin sonucunda bütün grupların son test puanları ile ön test puanlarına göre daha yüksek çıkmıştır. GUTÜD, doğrudan öğretim ve inşacılık modellerinin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısına ve bilgi işlemsel düşünme testine göre gruplar arasında anlamlı bir fark bulunamamıştır. Ortaokul öğrencilerinin kodlamaya yönelik tutumları ve scratch ortamı için kodlama başarı testine göre gruplar arasında anlamlı bir fark bulunmuştur.

ANAHTAR KELİMELE: GUTÜD, İnşacılık, Doğrudan öğretim, Scratch

ABSTRACT

ANALYSIS OF THE EFFECTIVENESS OF GUTÜD INSTRUCTIONAL MODEL IN TERMS OF CERTAIN VARIABLES

MA THESIS

ERHAN DEMİRTAŞ

BOLU ABANT İZZET BAYSAL UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

COMPUTER EDUCATION AND INSTRUCTIONAL TECHNOLOGY

FIELD

**DEPARTMENT OF COMPUTER EDUCATION AND INSTRUCTIONAL
TECHNOLOGY**

(SUPERVISOR: PROF. DR. İBRAHİM ÇETİN)

BOLU, JULY 2023

Xiii+ 336

The aim of this research is to teach using Scratch, a block based education environment, containing same activities and content, and determine to what extent GUTÜD teaching model, constructivism model and direct teaching model affect self-efficacy perceptions of computational thinking skills, coding success, attitudes towards coding, and computational thinking skills. In this context, the study group of this research consists of ninety-nine students from three different classes selected randomly from a secondary school in Bolu province. GUTÜD model, direct instruction model and constructivism model were assigned to those randomly selected classes. A quasi-experimental pattern was used in this research. A self-efficacy perception scale for computational thinking skills, a coding achievement test for Scratch environment, an attitude scale towards coding and a computational thinking test were applied to the selected classes as pre and post tests. The lesson plans and activities were applied to the students by the expert teacher with the application of the pre tests two hours a week for eight weeks. Collected data were analyzed in SPSS 26 program. As a result of the analyses made on GUTÜD, direct instruction and constructivism models using the self-efficacy perception scale for computational thinking skills, the computational thinking test, the attitudes of secondary school students towards coding, and the coding achievement test for the scratch environment, the post-test scores of all groups were higher than the pre-test scores. GUTÜD, direct instruction and constructivism models have no significant difference amongst groups in terms of self-efficacy perception for computational thinking skills and computational thinking test. A significant difference was found between the secondary school students' attitudes towards coding and the coding achievement test for scratch environment.

KEYWORDS: GUTÜD, Constructionism, Direct instruction, Scratch

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL VE ONAY SAYFASI.....	iii
ETİK BEYAN.....	iv
ÖZET.....	v
ABSTRACT	vi
İÇİNDEKİLER.....	vii
ŞEKİL LİSTESİ.....	x
TABLO LİSTESİ.....	xi
KISALTMA VE SEMBOLLER LİSTESİ.....	xii
TEŞEKKÜR.....	xiii
1. GİRİŞ.....	1
1.1 Araştırmanın Problem Durumu	6
1.2 Çalışmanın Önemi	8
1.3 Çalışmanın Amacı	9
1.4 Araştırma Problemi.....	9
1.4.1 Araştırma Soruları	9
1.4.2 Hipotezler	10
1.5 Sayıtlar.....	10
1.6 Sınırlılıklar	11
1.7 Tanımlar.....	11
1.7.1 Kodlama.....	11
1.7.2 Algoritmik Düşünme	11
1.7.3 Bilgi İşlemsel Düşünme.....	11
1.7.4 Tutum.....	11
1.7.5 Öz Yeterlik.....	11
1.7.6 İnşacılık.....	11
1.7.7 GUTÜD	12
1.7.8 Doğrudan Öğretim	12
2. ARAŞTIRMANIN KURAMSAL TEMELİ VE İLGİLİ	
ARAŞTIRMALAR.....	13
2.1 Bilgi İşlemsel Düşünme.....	13
2.1.1 Bilgi işlemsel Düşünme: Yapı ve İçerikler.....	14
2.1.2 Bilgi İşlemsel Düşünmenin Bileşenleri	16
2.1.2.1 Soyutlama.....	17
2.1.2.2 Bileşenlerine Ayırma.....	17
2.1.2.3 Algoritmik Düşünme.....	17
2.1.2.4 Mantıksal Düşünme.....	18
2.1.2.5 Hata Ayıklama.....	18
2.1.2.6 Örüntü Tanıma	18

2.1.2.7 Değerlendirme	18
2.1.3 Bilgi İşlemsel Düşünme Ne Değildir	18
2.1.4 Bilgi İşlemsel Düşünmenin Önemi.....	19
2.1.5 Bilgi İşlemsel Düşünme Nasıl Öğretilir	19
2.1.6 Bilgi İşlemsel Düşünmenin Eğitici Oyunlarda Kullanımı	21
2.1.6.1 SCRATCH	21
2.2 Kodlama Eğitimi.....	24
2.3 Tutum.....	26
2.3.1 Tutumun Oluşumunda Etkili Faktörler.....	27
2.3.2 Tutumu Oluşturan Bileşenler.....	28
2.3.3 Programlamaya Yönelik Tutum	29
2.4 Öz Yeterlik.....	30
2.4.1 Akademik Öz Yeterlik	31
2.4.2 Sosyal Öz Yeterlik	32
2.4.3 Entelektüel Öz Yeterlik	33
2.4.4 Özyeterlik İnancının Kaynakları.....	34
2.4.5 Eğitimde Öz Yeterlik	36
2.4.6 Bilgi İşlemsel Düşünme ve Öz Yeterlilik	37
2.5 Yapılandırmacılık	37
2.5.1 Yapılandırmacı Öğrenme	41
2.5.2 Yapılandırmacılık Temel Fikirler	42
2.5.3 Yapılandırmacı Öğrenme ve Öğretmenin Özellikleri.....	42
2.6 Çalışmada Kullanılacak Öğretim Yaklaşımları	45
2.6.1 İnşacılık.....	45
2.6.2 Doğrudan Öğretim Yöntemi	48
2.6.3 GUTÜD	53
2.6.3.1 Eşli Programlama	56
3. YÖNTEM.....	59
3.1 Araştırmanın Modeli.....	59
3.2 Çalışma Grubu	60
3.3 Veri Toplama Araçları	62
3.3.1 Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeği	62
3.3.2 Scratch Ortamı İçin Kodlama Başarı Testi	62
3.3.3 Kodlamaya Yönelik Tutumları Ölçeği	62
3.3.4 Bilgi İşlemsel Düşünme Testi.....	63
3.4 Verilerin Toplanması	63
3.5 Verilerin Analiz Edilmesi	63
3.6 Öğretim Ortamları	64
4. BULGULAR	68
4.1 Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeğine Yönelik Bulgular.....	68
4.2 Bilgi İşlemsel Düşünme Testine Yönelik Bulgular	73
4.3 Ortaokul Öğrencilerinin Kodlamaya Yönelik Tutumları Ölçeğine Yönelik Bulgular	78
4.4 Scratch Ortamı İçin Kodlama Başarısı Testine Yönelik Bulgular.....	83
5. TARTIŞMA.....	89
5.1.1 Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeğine Yönelik Sonuçlar	89

5.1.2 Bilgi İşlemsel Düşünme Testine Yönelik Sonuçlar	91
5.1.3 Ortaokul Öğrencilerinin Kodlamaya Yönelik Tutumları Ölçeğine Yönelik Sonuçlar	92
5.1.4 Kodlama Başarısı Testine Yönelik Sonuçlar	94
6. SONUÇ VE ÖNERİLER.....	97
6.1.1 Araştırmacılar İçin Öneriler	97
6.1.2 Eğitimciler İçin Öneriler	97
7. KAYNAKLAR.....	99
8. EKLER.....	115



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 Scratch programı web giriş sayfası	23
Şekil 2.2 Scratch program oluşturma sayfası.....	23
Şekil 2.3 Öğrenme döngüsü	54



TABLO LİSTESİ

Sayfa

Tablo 1.1 ISTE 2016 Öğrenci standartları.....	1
Tablo 2.1 Öz yeterlik algıları yüksek veya düşük olan bireylerin özellikleri...	31
Tablo 3.1 Uygulanan yarı deneysel desenin özeti	60
Tablo 3.2 Araştırmaya dahil olan öğrencilerin demografik özellikleri	61
Tablo 4.1 Bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısına ait betimsel istatistikler	69
Tablo 4.2 Kolmogorov-Smirnov testi sonuçları	69
Tablo 4.3 Levene testi sonucu	70
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	72
Tablo 4.9 İlişkili örneklem t-testi	73
Tablo 4.10 Bilgi işlemsel düşünmeye ait betimsel istatistikler	74
Tablo 4.11 Kolmogorov-Smirnov testi sonuçları	74
Tablo 4.12 Levene testi sonucu	75
Tablo 4.13 ANOVA sonuçları.....	75
Tablo 4.14 Kolmogorov-Smirnov testi sonuçları	76
Tablo 4.15 Levene testi sonucu	76
Tablo 4.16 Kovaryans analizi sonuçları	77
Tablo 4.17 Fark puanlarının basıklık ve çarpıklık değerleri	77
Tablo 4.18 İlişkili örneklem t-testi	78
Tablo 4.19 Ortaokul öğrencilerinin kodlamaya yönelik tutuma ait betimsel istatistikler	79
Tablo 4.20 Kolmogorov-Smirnov testi sonuçları	79
Tablo 4.21 Levene testi sonucu	80
Tablo 4.22 ANOVA sonuçları.....	80
Tablo 4.23 Kolmogorov-Smirnov testi sonuçları	81
Tablo 4.24 Levene testi sonucu	81
Tablo 4.25 Kovaryans analizi sonuçları	82
Tablo 4.26 Fark puanlarının basıklık ve çarpıklık değerleri	82
Tablo 4.27 İlişkili örneklem t-testi	83
Tablo 4.28 Scratch ortamı için kodlama başarısına ait betimsel istatistikler ...	84
Tablo 4.29 Kolmogorov-Smirnov testi sonuçları	84
Tablo 4.30 Levene testi sonucu	85
Tablo 4.31 ANOVA sonuçları.....	85
Tablo 4.32 Kolmogorov-Smirnov testi sonuçları	86
Tablo 4.33 Levene testi sonucu	86
Tablo 4.34 Kovaryans analizi sonuçları	87
Tablo 4.35 Fark puanlarının basıklık ve çarpıklık değerleri	87
Tablo 4.36 İlişkili örneklem t-testi	88

KISALTMA VE SEMBOLLER LİSTESİ

CSTA	: Bilgisayar Bilimleri Öğretmenleri Derneği
ISTE	: Uluslararası Eğitim Teknolojileri Topluluğu
MEB	: Milli Eğitim Bakanlığı
MIT	: Massachusetts Teknoloji Enstitüsü
OECD	: Ekonomik Kalkınma ve İşbirliği Örgütü
TTKB	: Talim ve Terbiye Kurulu Başkanlığı



TEŞEKKÜR

Tez çalışmamın ilk gününden bu zamana kadar değerli görüş ve önerileriyle her daim desteğini hissettiğim, umutsuzluğa kapılıp benden bu kadar bu iş bitmeyecek dediğim anda beni tekrardan motive edip çalışmama yeniden odaklamamı sağlayan, bilgi, birikim ve büyük tecrübeleriyle çalışmamda her daim yanımda hissettiğim saygıdeğer tez danışmanım Prof. Dr. İbrahim Çetin'e sonsuz teşekkürlerimi sunarım.

Yüksek Lisans dönemim boyunca ders aldığım ve deneyimlerine her daim yararlandığım, değerli hocalarım Prof. Dr. Erkan Tekinarslan'a, Prof. Dr. Ercan Top'a, Doç. Dr. Bahadır Altıntaş'a, Dr. Öğr. Üyesi Orhan Curaoglu'na teşekkürlerimi sunarım.

Çalışmalarımı inceleyen ve değerli katkılarını eksik etmeyen değerli hocalarım Doç.Dr. Ümit Dilekçi'ye, Doç.Dr. Hakan Tahiri Mutlu'ya, Dr. Bayram Erden'e, Sezgin Koçyiğit'e ve Tarık Otu'ya teşekkürlerimi sunarım.

Hayatım boyunca aldığım tüm kararlarda arkamda duran, maddi ve manevi yardımlarını esirgemeyen anneme, babama ve abime sonsuz sevgilerimi sunuyorum.

Tez çalışmam boyunca yorgun ve stresli anlarımda bana sürekli moral veren, sabırlı, anlayışlı ve motive edici tavırlarıyla her zaman yanımda olduğunu hissettiren hayat arkadaşım biricik eşim Burcu Demirtaş'a sonsuz sevgilerimi ve teşekkürlerimi sunarım.

Her yerde "baba ders çalışıyor" diye ufak şikayetlerde bulunan, yüzümü gülümseten biricik oğlum Yusuf Kayra'ya da sevgilerimi sunarım.

1. GİRİŞ

Elli yıl kadar kısa bir süre önce, insanların okuma, yazma ve basit aritmetik işlemlerini yapabilmeleri, çoğu işte çalışabilmeleri için yeterliyen günümüzde yeterli görülmemektedir (Vasquez, 2013). Bugünün gençleri, mezun olduklarında yirmi birinci yüzyılın yaşama şartlarına hazır bir biçimde eğitim görmelidirler ve burada kendilerinden öğrenciyken henüz icat edilmemiş teknolojilerle çalışmaları, vatandaşlar olarak büyük mühendislik projeleri hakkında bilinçli kararlar vermeleri, günlük yaşamlarında karşılaşmadıkları sorunları çözmeleri beklenmektedir (Vasquez, 2013).

21.yüzyıl öğrencilerinden istenen temel beceriler; eleştirel düşünebilmeleri, analiz ve sentez yapabilmeleri, problem çözebilmeleri, yenilikçi ve üretken olabilmeleri, işbirlikçi çalışabilmeleri, doğru ve güncel bilgiye kolay erişebilmek şeklinde oluşturulmuştur (Günüç vd., 2013). Bireylerin yaşamlarında karşılaşacakları problemleri çözmeleri açısından 21.yüzyıl becerileri olarak ifade edilen bu beceriler önemlidir (Yıldız vd., 2017). International Society for Technology in Education [ISTE]'ya (2016) bakıldığında öğrencilerin sahip olması gereken yedi beceri yayınlanmıştır. Tablo1.1 de bu yedi özellik belirtilmiştir.

Tablo 1.1 ISTE 2016 Öğrenci standartları

Kendine Yön Verebilen Öğrenci
Dijital Vatandaş
Bilgi Oluşturucu
Yenilikçi Tasarımcı
Bilgi İşlemsel Düşünebilen
Yaratıcı İlişkiler Kurabilen

Öğrencilerin sosyal hayata uyum sağlayabilmeleri için gerekli olan 21. yüzyıl becerileri, ancak onlara uygun öğrenme ortamları sağlanarak elde edilebilir. Öğretmenlerin bu yaşantıları kazandırmaları gerekmektedir (Yaşar, 2021). Eğitimin niteliğini etkileyen diğer bileşenlerle sürekli etkileşim hâlinde oldukları ve onları doğrudan etkiledikleri (Leigh ve Mead, 2005) göz önünde bulundurulduğunda, öğretmenlerin niteliklerinin eğitim sürecinde kilit öneme sahip olduğu ve öğretmenlerin bilgi ve becerilerinin öğrenci gelişimini destekleyen en önemli unsurlar olduğu söylenebilir. Öğretmenlerin, öğrencilerinde istenen 21. yüzyıl becerilerini geliştirebilmeleri için en önemli konulardan biri de bu becerileri öğretme konusunda ne kadar yetkin olduklarıdır. Personal Computer System [PCS] ve ISTE gibi birçok uluslararası kuruluş, tüm öğrencilerin 21. yüzyılın küreselleşen dünyasına uyum sağlamaları için gerekli olan becerileri kategorize etme konusunda detaylı çalışmalar yürütmektedir. Bu kuruluşlar, 21. yüzyılda yaşayan bireylerin etkin birer vatandaş olabilmeleri için gerekli olacağını öngördükleri becerileri "21. Yüzyıl Becerileri" olarak tanımlı yapılmış ve farklı sınıflar ile temel niteliklerini açıklamışlardır (Organisation for Economic Co-operation and Development [OECD], 2020). 21. yy. becerilerinin nasıl gruplandırılacağına ilişkin literatürde birçok farklı görüş olsa da en çok kabul gören gruplandırma "21. Yüzyıl Becerileri Ortaklığı Raporu'nda bulunan sınıflar gruplandırılmıştır. Bu gruplandırmada yaşam ve mesleki beceriler öğrenmede yenilik içeren beceriler bilgi, medya ve teknolojileri yaşam ve kariyer becerileri şeklinde gruplandırılmıştır. Birinci grupta bulunan yaşam ve mesleki beceriler kişilerden sosyal olmasını, yapacağı işlerde esnek olması ve gerektiğinde inisiyatif alması gerektiğini söyler. Bu grupta bireylerden aynı zamanda kültürler arasında etkileşimde bulunmalarını, liderlik özelliklerinin güçlü olmasına, üretkenlik ve öz yeterliliklerinin yüksek olması beklenmektedir. Öğrenme ve yenilik içeren becerilerde ise kişilerden iş birliği içinde çalışmayı, konu hakkında farklı düşünebilmeyi ve kişiler arası iletişiminin güçlü olması beklenmektedir. Bu kategorideki bireylerin aynı zamanda eleştirel düşünme

yönlerinin ve yaratıcılık yönlerini de geliştirilmesi konusuna değinilmiştir. Bilgi, medya ve teknoloji becerilerinde ise bilişim okuryazarı ve medya okuryazarı konularından bahsedilmektedir (OECD, 2016).

Günümüzün yeni dünyasını ve teknoloji destekli öğrenmeyi anlamak için, bilgisayar bilimlerini öğrencilere erken yaşta tanıtmanın önemli olduğu düşünülmektedir. Bu nedenle, küçük yaşta öğrencilere programlama öğretme gereksinimi önemli ölçüde artmış, teknoloji eğitimi artık çocukların üst düzey ve problem çözme becerilerinin geliştirilmesinde önemli bir unsur olarak erken yaşta programlamaya daha fazla odaklanmıştır (Dagdilelis vd., 2004). Brennan (2012) programlamayı, kodlar ile kişilerin bilgisayar işlemlerini kontrol etme işlemi olarak tanımlanmıştır. Eğitim teknolojisi alanında yapılan araştırmalarda kişilerin programlama yaparken problem çözme yöntemlerini ve bilgi işlemsel düşünme becerilerini kullanarak problemin içeriğini, hangi şekilde olacağını belirlemelerinde bilgilerini nasıl kullanması gerektiği incelenmektedir. Programlama bilgi ve beceri gerektirdiği için, öğretim yöntemleri, uzmanlık alan bilgisi ve belirli işlemler için kullanılan araçların seçimi açısından sınıflandırılmasıdır. Bu sınıflandırmaların yapılması programlamanın öğrenilmesini ve konunun kavranmasını kolaylaştırır ve yararlı hale getirir (Smith vd., 1997).

Araştırmacılar Frey ve Osborne (2013) “sanayi sektörünün bilişim teknolojilerinin üzerinde oluşturduğu etkiyi incelemişlerdir”. Amerika Birleşik Devletleri'ndeki 702 mesleğin (işletme, finans, bilgisayar, mühendislik, bilim, eğitim, hukuk, toplum hizmetleri, sanat, medya, tıp, müşteri hizmetleri, satış, ofis işleri, tarım, inşaat, bakım, üretim, ulaşım vb. %47'sinin önümüzdeki 210 yıl içinde teknolojinin hızlanmasından etkilenme riskinin yüksek olduğunu ve hatta yok olabileceğini tespit etmişlerdir. Sonuçlar, düşük sosyal zekâ gerektiren mesleklerin taşımacılık, lojistik, ofis ve idari destek personeli, üretim meslekleri, kasiyerler vb.) yerini bilgisayarların alacağını göstermektedir. Buna rağmen, farklı ve sosyal zekâ gerektiren işler daha az risklidir. Küresel firma Manpower Group, 2017 yılında Z kuşağının gelecekteki işlerinin %65'inin henüz mevcut olmayacağını açıklayarak bu iddiaları destekledi. Geleceğin teknolojileri, alışlagelmiş (bilişsel ve manuel) işlerin yerini geçebilir ve çalışanların alışlagelmiş olmayan işlere ve daha çok tatmin olacağı kariyerlere katılmasını sağlayabilir. Bu görevler grup çalışması, etkili iletişim, yaratıcılık, sunma

becerisi, esnek zekâ ve etkili öğrenme becerisi gerektirir. Dahası, otomatik öğretme geleceğin çalışanları için çok önemlidir çünkü dinamik hızlanma sayesinde iş artık kişinin halihazırda bildiklerine değil, neleri öğrenmeye istekli veya yetenekli olduğuna bağlıdır. Ayrıca dijital ürünlerin ticaretini yapacak veri analizi ve profesyonel hizmet temsilcileri gerektiren mesleklerin de artması beklenmektedir (Manela, 2017). Yapılan bir araştırmaya göre BT alanında %26, insan kaynaklarında %20 ve müşteri ilişkilerinde %15'lik bir artış beklenmektedir. Öğrencilerin yaşama ve öğrenme stillerine yeni düşünce yöntemleri ekleyerek eğitim müfredatı yeniden oluşturulmalıdır. Okulun mezunlarına yönelik eğitim temeli, Sanayi Devrimi sırasında var olan ihtiyaçlara uygundur. Bu işler için gerekli olan beceriler ve bilgi birikimi belli bir noktada tutulmuştur. 21. yüzyılda ise daha esnek ve farkındalık yaratan işler yapılması istenmektedir. Bu nedenle, eğitim sisteminin bugün bir tuşa basarak ya da arama motorlarından ulaşılabilen hafızaya dayalı bilgilere odaklanmayı bırakması gerekmektedir. Eğitim sisteminin amacı, değişen teknolojik dünyada derin düşünmeyi geliştirmeyi hedeflemelidir (Eisenberg vd., 2019).

Yıllarca bilgisayar mühendisleri ve uzmanlar için bilgisayar programlama çok önemli bir araç haline gelmiştir. Programlama sadece mühendislik alanı ile sınırlı kalmayıp temel bir beceri olduğu kabul görmüştür. Programlama ile yapılan işlerin ve burada çalışacak olan kişilere ihtiyacın çok fazla artacağı öngörülmektedir (Balanskat ve Engelhard, 2015).

Bu sebepten dolayı kodlama eğitimi birçok ülkede zorunlu hale getirilmekte ve yaygınlaştırılmaktadır. European Schoolnet'in (2015) raporuna göre, 2014 yılı kodlamanın müfredata entegre edildiği yıl olmuştur ve İngiltere bu konuda ilk girişimde bulunmuştur. İlk olarak İngiltere'de ilkökul, ortaokul ve lise seviyelerinde kodlama eğitimi zorunlu hale getirmiştir. "Bilgisayar bilimi dersi adı altında altı yaşından on dört yaşına kadar olan öğrencilere eğitim programı oluşturan öncü ülkelerdendir. Öğrenciler için bakanlığın programına zorunlu ders olarak dahil eden ilk ülkedir. Rapor, İngiltere'nin yanı sıra eğitim bakanlıklarının kodlamayı müfredatlarına nasıl entegre ettiklerini de analiz etmiştir. Raporun sonucu Bulgaristan, Avusturya, Bulgaristan, Çek Cumhuriyeti, Danimarka, Estonya, Fransa, Macaristan, İrlanda, İsrail, Litvanya, Malta, İspanya, Polonya, Portekiz, Slovakya ve İngiltere olmak üzere 16 Avrupa ülkesinin kodlamayı müfredatlarına entegre ettiğini göstermiştir. Bunun yanı sıra "Finlandiya, Belçika,

ABD ve Güney Kore planlama yaparken, Hollanda ve Norveç'in" müfredatlarına dahil etme planları bulunmaktadır.

Kodlama eğitimi, programlama araçlarını kullanarak verilen bir soruna kod yazarak çözüm üretmeyi sağlar (Arabacıoğlu vd., 2007). Fakat programlama dillerinin öğrenilmesinde yaşanan sıkıntılar, öğrencilerin kod yazma ve öz yeterliliklerini kaybetmelerine neden olmaktadır (Byrne ve Lyons, 2001; Futschek, 2006). Kod yazarken öğrencilerin sürece metin tabanlı programlar ile başlaması kodlama sürecini olumsuz etkilemektedir (Özmen ve Altun, 2014). Kod yazma sürecinde yaşanan sıkıntılar algoritma mantığının kişilere aktarılmasında da engel oluşturmaktadır. İlk başta algoritmik düşünce yapısını öğretmek kodlama öğretmeyi de kolaylaştırmaktadır (Ala-Mutka, 2004). Bu sebepten dolayı iyi bir planlama yaparak sürece başlanılmalı ve kodlama eğitiminin öğretilmesini kolaylaştırmak için alt yapı yeterli hale getirilip, öğretici materyallerde daha çok görselliğe önem verilmelidir.

Birçok ülke kodlama eğitimini daha basit halede verebilmek için projeler yürütmektedir. Yapılan projeler sonucunda birçok ülke "code.org, Scratch, MBlock" gibi farklı programlama ortamlarının kullanılmasını ve ülke eğitim müfredatlarına koymuştur. Örneğin code.org üzerinde yapılan çalışmalarda öğrencilerin programlamayı rahatça öğrenebilmeleri için onlarca saatlik kodlama müfredatı oluşturulmuş ve geliştirilen kodlama eğitimi herkese ulaşabilmesi için 34 farklı dile çevrilmiştir (Code.org, 2022). Scratch, MBlock, code.org gibi programlama ortamlarının amacı öğrencilere programlamayı kolayca öğretmek ve öğrencilerin öğrenilen bilgilerle pratik yapmalarını sağlamaktır (Resnick vd., 2009). Bu sayede öğrenciler hikâye yazmada, animasyon ve oyun üretmede programlama dillerini ve algoritmayı kullanabilirler (Taylor vd., 2010).

Türkiye’de kodlamanın öneminin farkına varmış ve 2012-2013 eğitim-öğretim yılında kodlama eğitimini müfredatına dâhil etmiştir (Demirbaş, 2019). İlk olarak müfredat değişikliklerine gidilmiştir. Bilişim teknolojileri ve yazılım dersi öğretimi 2012 yılında hazırlanıp sunulmuştur ve 5. ve 6. sınıflar için zorunlu ders, 7. ve 8. sınıflarda ise seçmeli ders olarak okutulmaya başlanmıştır (Millî Eğitim Bakanlığı [MEB], 2015). Dersteki yetkinlikler “bilgi okuryazarlığı, iletişim kurmak, bilgi paylaşmak ve kendini ifade etmek için bilgi teknolojilerini kullanmak, araştırma yapmak, bilgiyi yapılandırmak ve iş birliği içinde çalışmak, problem çözebilmek, programlama ve orijinal ürün geliştirme” olarak

tanımlanmaktadır. Bu proje ile teknoloji ile barışık bir dijital vatandaş yetiştirmek, teknolojiyi devamlı olarak takip eden, iş birliği ve bilgi paylaşımı için öğretim teknolojisinin kullanımını sağlayan ve her öğrencinin kendi düzeyine uygun bir seviyede tasarım yapabilmesi için ulusal düzeyde temel yetkinlikleri belirleyen bir müfredatın oluşturulması amaçlamaktadır (Talim Terbiye Kurumu Başkanlığı [TTKB], 2015).

Türkiye'de Millî Eğitim Bakanlığı Yenilik ve Eğitim Teknolojileri Genel Müdürlüğü tarafından planlanan ve uygulanan Eğitim Bilişim Ağı adında bir eğitim konularını kapsayan içerik ağı üretilmiştir. İhtiyaç duyulduğunda teknolojiyi eğitime entegre etmeyi amaçlayan bu ağ, FATİH projesi kapsamında öğretmen ve öğrencilerin ders materyallerine çevrimiçi erişimini sağlamaktadır. “Bu kapsamda, 2023 eğitim vizyonu kapsamında öğrencilere kodlama becerisi kazandırmak amacıyla 3 EBA platformunda "MASKOD" adlı blok tabanlı bir kodlama aracı oluşturulmuştur” (MEB, 2015; OECD, 2020).

Bilgisayar programlama öğrenimini kolaylaştırmak için yıllar içinde birçok farklı yaklaşım önerilmiş (Bers vd. 2014; Marín vd. 2018) “ve programlama öğrenimini kolaylaştırmak için farklı araç yaklaşımlarını sınıflandırmak için bazı çabalar gösterilmiştir” (Kelleher ve Pausch 2005). İlk olarak, kodlama öğretimine yönelik metodolojilerin, klasik davranışçılık, bilişselcilik ve yapılandırmacılık teorileri de dâhil olmak üzere daha genel eğitim metodolojik yaklaşımları birçok öğretim programında kullanılmaya başlanmıştır (Grossman vd. 2009).

Bu araştırmalar sonucunda dünyada ve ülkemizde kodlama eğitiminin önemi ve kodlama eğitiminin nasıl verilmesi gerektiği konusunda sorulara cevaplar arandığı görülmektedir. Literatürdeki bu veriler üzerine çalışmamızda GUTÜD (Gözle Uygula Tasarla Üret Değerlendir), inşacılık ve doğrudan öğretim yöntemleri ile verilen öğretimin 6. sınıf öğrencilerinin bilgi işlemsel düşünme becerileri, bilgi işlemsel düşünme becerilerine yönelik öz yeterlik, kodlamaya yönelik tutum ve kodlama başarısına olan etkilerinin araştırılması amaçlanmaktadır.

1.1 Araştırmanın Problem Durumu

Toplumlarda köklü değişiklikler yapılarak, ülkelerde var olan sağlık, siyaset ve ekonomi bilgi teknolojileri ışığında düzenlemeye gidilmektedir. Bu sebeplerden dolayı öğretmenin ve öğrencilerin var olan ihtiyaçları ve onlardan

beklenen yetkinlikler deęiřmiřtir. Öğrencilere bu yetkinliklerin kazandırılması ve iş hayatında etkin bir rol alması ve aktif, sorumlu bir vatandaş olması için 21.yüzyıl hedefleri oluşturulmaktadır (OECD, 2020).

Yakın zamana kadar, bilgisayar bilimi veya bilgi işlemsel düşünme gibi alanlar tipik olarak ortaöğretim sonrası eğitim müfredatında veya altında, program geliřtirmeye odaklanarak izole edilmiřtir (Falkner, 2015). 21.yy. bilgisayar bilimi ve bilgi işlemsel düşünme biliřim okur yazarı olan bireylerin nasıl olması gerektiğini yeniden tanımlamaktadır (OECD, 2020). Bunun birçok nedeni bulunmaktadır. Dünya çapındaki arařtırmacılar, inovasyon, giriřimcilik, ekip çalıřması ve yaratıcı düşünme gerektiren bilgisayar bilimi ve bilgi işlemsel düşünmenin yeni bir zorunluluk hâline gelmesi gerektiğini belirtmektedir. Wing (2006) okuma, yazma ve aritmetiğin matbaanın ortaya çıkmasıyla öğrencilerin başarılı olmak için ihtiyaç duydukları temel beceriler olarak hizmet etmesi gibi, bilgi işlemsel düşünmenin de bilgisayarların yükseliři ve yaygınlařmasıyla ateřlenen temel bir beceri olarak görüldüğünü belirtmiřtir (Chen vd., 2017). Bilgi işlemsel düşünme, gençlerin güçlü fikirlere erişmelerine, yaratıcılıklarını serbest bırakmalarına, kendilerini heyecan verici yeni yollarla ifade etmelerine ve etraflarında hızla deęiřen dünyayı anlamalarına yardımcı olabilir (Yu ve Roque, 2018).

Günümüz biliřim çağında, programlar insan toplumunun bilgiyi işlemesini saęlamakta, bu da programlama eğitiminin küresel eğitim sistemlerinde giderek daha fazla ilgi görmesine neden olmaktadır (Zhang vd., 2019). Zihinsel gelişim sürecinde en kritik dönemlerden biri olan çocukluk sürecinde yapılan arařtırmalar göstermektedir ki programlama eğitimi verilen çocukların bilgi okuryazarlığı noktasında da gelişme göstermektedir (Grover ve Pea, 2013). Bu nedenle, "programlama yeteneęi" üzerine odaklanan deneysel arařtırmalarda, bilgi işlemsel düşünme, mantıksal muhakeme düşüncesi, uzamsal hayal gücü gibi birçok farklı görüş ortaya çıkmıřtır. (Unal ve Topu, 2021; PérezMarín vd., 2020). Programlama eğitiminin kalitesini artırmak için programlama araçlarının nasıl olması gerektięi konusunda birçok çalıřma yapılmaktadır. K-12 düzeyindeki bireylerin programlama konusunda tecrübe kazanmaları bu işin en önemli unsurlarını oluşturmaktadır (Sáez-López vd., 2016). Dersin içerięine uygun yöntem ve tekniklerin seçilmesi programlama öğretiminde öğrencilerin daha aktif ve derse daha ilgili olmalarını saęlamaktadır. Görsel programlama kullanılarak yapılan

alışmalar ilginin daha fazla olduğunu g stermektedir. Fakat yapılan bazı arařtırmalarda ise ocukların programlama eđitimi y ntemlerine dair konularda yeterli alışma olmadığı ve bu konular iin daha ok arařtırma yapılması gerektiđi s ylenmektedir. Programlama  đretimde farklı  đretim y ntem ve tekniklerinin kullanılması  đretime olan etkilerini de olumlu veya olumsuz y nleri incelenerek uygulanması gereklidir (Er mit, 2020). alışmalar, oyun tabanlı  đretimin,  đrencilere programlama etkinliklerine eđlenceli ve dinamik bir řekilde katılmalarını sađlayabilecek oyunlařtırılmıř bir ortam sađladığını g stermiřtir (Koupritzioti ve Xinogalos, 2020). Fakat, “programlama becerisi” ile ilgili yapılan alışmalar devam etmekle birlikte hali hazırda bulunan arařtırmalar hen z  đrencilerin programlama yeteneklerini aıklamada yeterli olmamaktadır (Grover ve Pea, 2013). “Farklı  đretim y ntem ve aralarının kullanılması da  đretim etkilerinin farklılařmasına neden olacaktır” (Scherer vd., 2020). ocukların bilgi ađının geliřimine bir an  nce uyum sađlamalarını teřvik etmek iin programlama  đretiminin ocukların d ř nme  zelliklerine uygun bir řekilde gerekleřtirilmesi gerekmektedir (Er mit, 2020). Bu sebepten dolayı, uygun olan  đretim y ntemini oluřturmak iin programlamanın  đrenilmesinde eđitim seviyelerini,  đretim modellerini, aralarını ve deđerlendirme kriterlerini ıkarmayı sađlamalıdır.

Genel olarak, alan yazına bakıldığında birok  đretim programı ile yapılmıř alışmalar bulunmakta fakat GUT D, inřacılık ve dođrudan  đretim modeli ortamlarının kullanıldığı alışma bulunmamaktadır. Bu alışma ile literat re katkı sađlanacağı ve her program iin  retilen planların ve materyallerin ortaokul d zeyinde kodlama eđitimi yapacak  đreticilere de bir ıřık olacağı d ř n lmektedir.

1.2 alışmanın  nemi

Alan yazın incelendiđinde iinde bulunduđumuz ađda  đrenci rollerinin deđiřtiđi g r lmektedir.  đrencilerden  đrendiđi kavram, prosed r ve temel bilgileri bařka bađlamalara aktarması, bu bilgileri kendi amaları dođrultusunda yeniden inřa etmesi ve yeni bilgiler oluřturması beklenmektedir. Bu ama dođrultusunda  đrencilerin bilgi iřlemsel d ř nme ve problem  zme gibi  st d zey d ř nme becerileri kazandırmaya y nelik birok  đretim modeli bulunmaktadır. Bu konular  zerine yapılan alışmaların sayısında da ciddi bir artıř ve y nelim olduđu g r lmektedir. (etin vd., 2021) Fakat alan yazın incelendiđinde bu  st d zey becerilerin hangi  đrenme modeli ile daha iyi

verilebileceği noktasında tartışmaların ve eksikliklerin olduğu görülmektedir. Bu bağlamda çalışmanın konu aldığı inşacılık ve doğrudan öğretim modellerinin birçok yönden öğrencilere sağladığı faydalar bulunmaktadır. Fakat her iki yöntemin de bazı eksiklerini tamamlamak için GUTÜD modeli ortaya çıkarılmıştır. Bu model ile her iki yöntemde eksik kalan kısımların tamamlanması amaçlanmaktadır.

Bu çalışma ile GUTÜD, inşacılık ve doğrudan öğretim modellerinin ortaokul altıncı sınıf öğrencilerinin bilgi işlemsel düşünme becerilerine, bilgi işlemsel düşünme becerisi öz yeterliklerine, kodlamaya yönelik tutumlarına ve kodlama başarısına yönelik etkilerinin araştırılması hedeflenmiştir.

1.3 Çalışmanın Amacı

Araştırmanın amacı; blok tabanlı bir öğretim ortamı olan Scrtach ile ve aynı etkinlikler ve içerikler bulunduran; GUTÜD öğretim modeline göre, İnşacılık modeline göre ve doğrudan öğretim modeline göre öğretim yapılarak bu öğretimlerin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algılarına, kodlama başarılarına, kodlamaya yönelik tutumlarına, bilgi işlemsel düşünme becerilerine olan etkisini incelemektir. Araştırma sonuçlarına göre öğrencilere hangi öğretim modellerinin faydalı olabileceği yönünde alınyazına katkı sunulması amaçlanmaktadır. Ayrıca araştırmacıların bu tez kapsamında yapılan öğretim planlarını kendi istekleri yönünde kullanmaları sağlanacaktır.

1.4 Araştırma Problemi

Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin bilgi işlemsel düşünme becerisine yönelik öz yeterlik algılarına, kodlama başarılarına, kodlamaya yönelik tutumlarına, bilgi işlemsel düşünme becerilerine etkisi nedir?

1.4.1 Araştırma Soruları

1- Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD'ün, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin bilgi işlemsel düşünme becerisine yönelik öz yeterlik algılarına etkisi nedir?

2- Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD'ün, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin kodlama başarılarına etkisi nedir?

3- Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD'ün, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin kodlamaya yönelik tutumlarına etkisi nedir?

4- Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD'ün, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin bilgi işlemsel düşünme becerilerine etkisi nedir?

1.4.2 Hipotezler

H₀: Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin BİDBÖA (Bilgi İşlemsel Düşünme Becerisine Yönelik Öz yeterlilik Algısı) ön-test puanları sabit tutulduğunda BİDBÖA son-test puanları arasında fark yoktur.

H₁: Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin KBT (Kodlama Başarı Testi) ön-test puanları sabit tutulduğunda KBT son-test puanları arasında fark yoktur.

H₂: Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin KYTÖ (Kodlamaya Yönelik Tutumları Ölçeği) ön-test puanları sabit tutulduğunda KYTÖ son-test puanları arasında fark yoktur.

H₃: Bilişim teknolojileri ve yazılım dersinde Scratch ortamı bağlamında kullanılan GUTÜD, inşacılık ve doğrudan öğretim yaklaşımlarının altıncı sınıf öğrencilerinin BİDT (Bilgi İşlemsel Düşünme Testi) ön-test puanları sabit tutulduğunda BİDT son-test puanları arasında fark yoktur.

1.5 Sayıtlar

1-Araştırmaya katılan öğrencilerin yapılan testlerdeki sorulara tarafsız ve doğru cevap verdikleri kabul edilmiştir.

2-Araştırmaya katılan öğrencilerin yapılan testlere katılmayan öğrenciler etkisiz olarak kabul edilmiştir.

3-Araştırmaya katılan öğrencilerin, uygulama sürecinde doğal davrandıkları kabul edilmiştir.

4-Araştırmacının veri toplama araçlarını hazırlarken, uygularken ve verileri analiz ederken başvurduğu uzmanların tarafsız ve özenli davrandıkları varsayılmaktadır.

5- Uygulanan veri toplama araçlarının ve araştırma sırasında toplanan verilerin yeterli olduğu varsayılmaktadır.

1.6 Sınırlılıklar

- Araştırmanın çalışma grubu Bolu Orhangazi ortaokulunda bulunan rastgele seçilmiş üç altıncı sınıf şubesiyle sınırlıdır.
- Çalışma için toplanan veriler 2021-2022 eğitim-öğretim yılı bahar dönemi ile sınırlıdır.
- Araştırmada kullanılan değişkenler GUTÜD, inşacılık ve doğrudan öğretimin uygulandığı sınıf ortamları ile sınırlıdır.
- Araştırma, geliştirilen veri toplama araçları ile toplanan verilerle sınırlıdır.

1.7 Tanımlar

1.7.1 Kodlama

Bir eylemi gerçekleştirmek için bir bilgisayar sistemine bir dizi komut yazma işlemidir (Sayın ve Seferoğlu, 2016).

1.7.2 Algoritmik Düşünme

Algoritmik düşünme, bir problemin temel boyutlarını anlama becerisini, problem çözme becerilerini, çözümleri doğrulukları ve verimlilikleri açısından keşfetme yetkinliklerini ve adım adım problem çözme sürecini açıkça sunma becerilerini içeren bir dizi gelişmiş zihinsel işlemdir (Güler, 2021).

1.7.3 Bilgi İşlemsel Düşünme

Bilgisayarı üretken amaçlarla kullanmak ve yaşam problemlerini çözmek için gerekli bilgi, beceri ve tutumlara sahip olmaktır (Özden, 2015).

1.7.4 Tutum

İnsan davranışlarını etkileyen ve davranışların arka planında bulunan psikolojik bir etmendir (Anderson, 1988).

1.7.5 Öz Yeterlik

Bireylerin karşılarına çıkabilecek olası durumlarla baş edebilmek için gerekli olan davranışları ne kadar iyi yapabileceklerine ilişkin bireysel yargılarıdır (Bandura, 1986).

1.7.6 İnşacılık

Öğrenmenin bireylerin aktif bir şekilde kendi düşüncelerini ve anlam inşa etmeleri yoluyla gerçekleştiği bir öğrenme teorisi ve yaklaşımıdır. Bu yaklaşıma

göre, insanlar sadece bilgiyi pasif bir şekilde almakla kalmaz, aynı zamanda yeni bilgileri anlamak ve öğrenmek için önceki bilgileri kullanarak kendi anlamlarını inşa ederler (Harel ve Papert, 1990).

1.7.7 GUTÜD

GUTÜD “Gözle, Uygula, Tasarla, Üret ve Değerlendir” öğrenme döngüsünün baş harflerinden oluşmaktadır. GUTÜD modelinde temel kavramların oluşturulmasında etkinlik ve öğretmen temelli göster ve uygula adımları ön plana çıkarılırken, öğrencilerin verilen temel bilgiler ile ileri seviye zihinsel faaliyetlerde bulunması için Tasarla, Üret ve Değerlendir adımlarını kullanan bir öğretim modelidir (Çetin vd., 2021).

1.7.8 Doğrudan Öğretim

Öğretmenin dersi kontrol ettiği, öğrencilerin aktif bir şekilde katıldığı, öğretmen tarafından planlanan ve öğretmen tarafından sunulan öğrenme yöntemidir. Bu yöntemde, öğretmen öğrencilere konuyu anlatır, öğrenciler de bu konular hakkında ödevler yaparlar (Education Commission of the States, 1999).

2. ARAŞTIRMANIN KURAMSAL TEMELİ VE İLGİLİ ARAŞTIRMALAR

2.1 Bilgi İşlemsel Düşünme

“Düşünme, fikir ve kavram üretmekten veya bir durumu anlamaktan sorumlu olan, öğrenmeye ve önceki deneyime dayalı bilişsel bir süreçtir.” Kişiler yaşantıları süresince bu düşünce yapılarına geliştirerek farklı durumlarda oluşturabilirler. Bilgisayar bilimi de çok düşünce biçimlerini içermektedir (Frauenberger ve Purgathofer, 2019). Matematiksel düşünme, eleştirel düşünme, algoritmik düşünme, tasarım düşüncesi, bilgi işlemsel düşünme ve diğerleri, her biri birbirini dışlamayan ve verilen bir problemi çözmek için kullanılabilecek bilişsel süreçlere sahiptir (Wing, 2006).

Bu özelliklere dayanarak Aho (2012), “formüle edilmiş problemlere uygulanabilir yeni temel hesaplama modelleri oluşturmak için bir araştırma yöntemi olarak bilgi işlemsel yöntemi önerdi”. Başka bir yerde Denning ve Tedre (2019) programlamanın üniversite düzeyindeki herhangi bir sorunu anlamak ve çözmek için zihinsel bir araç olarak hizmet edebileceği fikrini destekledi; inşacı öğrenmenin öncülerinden Papert (1980), çocuklara öğretimde teknolojinin kullanımını tanıtırken, programlamanın bir problemi pratik bir şekilde anlamaya dayalı aktif bir öğrenme süreci olarak alınabileceğini savunarak, öğrencilere kendi temellerine dayalı bir düşünme öğrenim biçimi sağlamıştır.

Bilgi işlemsel düşünmenin herkes tarafından bilinmeye başlaması 21.yy. da olmuştur. Wing (2006), yalnızca belirli programlama dillerinde uzmanlaşmaya değil, öğrencilerin yeteneklerine odaklanarak bilgi işlemsel düşünme becerilerini geliştirmenin önemini vurgulamıştır. Sonraki aşamada Zapata-Ros (2015), Bilgi işlemsel düşünmenin gelişmesi için kişilerin erken yaşta eğitim almaları, gelişimlerinde etkili olduğunu Wing’in düşüncelerini baz alarak söylemiştir.

Bilgi işlemsel düşünme (computational thinking) çevirisi bazı kaynaklarda “hesaplamalı” (Özçınar ve Öztürk, 2018) olarak kullanılırken, bazılarında ise “bilgisayarca” (Korkmaz vd., 2015) olarak kullanılmaktadır. Resmî kurumların kullanımlarına bakıldığında ise “bilgi işlemsel “çevirisinin daha yaygın bir biçimde kullanıldığı görülmektedir (TTKB, 2015).

“Bilgi işlemsel düşünme ilk kez Papert (1980) tarafından kullanılmıştır”. Wing (2006) tarafından yapılan çalışmalar sonucunda popülerleşen bilgi işlemsel

düşünme kavramı uluslararası standartlarda ve öğretim programlarında yer almakta ve devamlı olarak bu çalışmaların sayısı artmaktadır.

Bilgi işlemsel düşünme, bilgisayar biliminden daha geniş bir alanın temel unsuru görülmekte ve tüm disiplinlerde kullanılabilecek bir yol olarak ifade edilmektedir (Computer-Supported Telecommunications Applications [CSTA], 2016). Bundy'e (2007) göre de fen ve sosyal bilimler alanlarındaki çalışmaları bilgi işlemsel düşünme etkilemektedir. Uluslararası Eğitim Teknolojileri Derneği 21. yüzyıl eğitimine rehber olacak standartlar geliştirmektedir.

Her yaştan insan ve özellikle çocukların animasyon, oyun ve hikâyelerini eğlenceli bir şekilde tasarlamalarını sağlayan blok tabanlı programlama dili Scratch, MIT Laboratuvarlarında geliştirilmiştir “scratch.mit.edu”. Brennan ve Resnick (2012) oluşturdukları çevrimiçi topluluk üyelerinin Scratch programlama dilini kullanarak bilgi-işlemsel düşünmeyi geliştirmelerini sağlamak istemektedir ve çalışmaların bilgi- işlemsel düşünme yönünden nasıl bir gelişim gösterdiğini incelemektedir.

Günümüzde bilgisayarın olmadığı ve bilgisayar kullanılmayan ortam neredeyse yoktur. Günlük kullanım çoğunlukla iletişim, internette gezinme veya ofis uygulamaları ile sınırlıdır. Bununla birlikte, bilgi işlemsel düşünme kavramları, olduğundan çok daha derine iner ve sıradan düşünmede değişiklikleri zorlar. Öyle ki 21. yüzyılı anlamak isteyen bir kişinin öncelikle bilgi işlemsel düşünmeyi anlaması gerektiği iddia edilmektedir (Bundy, 2007).

2.1.1 Bilgi işlemsel Düşünme: Yapı ve İçerikler

Soyut hesaplamanın ve bilgisayarların insan düşüncesinin evrimi üzerindeki etkisinin ön koşulları, ilk olarak programlama ve yapay zekâ alanındaki bilim adamlarının (Fink, 1966; Knuth vd., 2005; Papert, 1980) çalışmalarında ele alınmıştır. Bilgi işlemsel düşünme yakın bir zamanda ortaya atılmış bir fikir olarak kaynaklarda görülmektedir (Wing, 2006) ve tam olarak kabul edilmiş bir tanımlı bulunmamaktadır (Artym vd., 2017; Denning, 2017; National Research Council [NRC], 2010; Selby, 2015). “Bilgisayar bilimi, pedagoji, psikoloji ve düşünce felsefesi gibi farklı bilim alanlarından giderek daha fazla bilim insanı ve araştırmacıyı çekmektir”. Bilgi işlemsel düşünme süreçlerinin, belirli hesaplama modellerinin kullanıldığı alanda pratik deneyimi içermesi gerektiği varsayımı da vardır (Denning, 2017).

Wing (2006), her çocuğun analitik becerileri için okuma, yazma ve aritmetik süreçlerine bilgi işlemsel düşünmenin de eklenmesi gerektiğini vurgulamaktadır. Bilgi işlemsel düşünmenin tanımına yönelik farklı yaklaşımların varlığına rağmen, çoğu araştırmacı bilgi işlemsel düşünme modelinin sosyal ve beşerî bilimlere de dahil olmak üzere bilimin tüm alanlarına girdiğini ve bu fenomenlerin insanın bilişsel yeteneklerinin gelişimi üzerindeki etkisinin giderek arttığını belirtmektedir (Artym vd., 2017; Denning, 2017). Bundy'ye (2007) göre doğa bilimlerini ve beşerî bilimlerini bilgi işlemsel düşünme üzerine yapılan çalışmalar etkilemektedir.

Wing (2006), “bilgi işlemsel düşünmeyi” şu şekilde tanımlamaktadır; “Bilgisayar bilimi kavramlarından yararlanarak problem çözme, sistem tasarlama ve insan davranışlarını anlama” olarak belirtmiştir. “Uluslararası Eğitimde Teknoloji Topluluğu (ISTE) ve Bilgisayar Bilimi Öğretmenleri Derneği (CSTA)”, bilgi işlemsel düşünme için şöyle bir tanım söylemiştir. Bilgi işlemsel düşünme, bilgisayar ve diğer araçları kullanarak sorunu çözmek için veri toplama, veri analizi, veri sunumu, otomasyon çözümlerinin optimal kombinasyonunu geliştirmek için algoritma adımlarını ve kaynaklarını kullanma ve problem çözme süreci diğer alanlara genişletilmesidir (CSTA ve ISTE, 2016). 2016 yılında, Bilgisayar Bilimleri Öğretmenleri Derneği (CSTA) bilgi işlemsel düşünme tanımını güncelleştirir: Bilgi işlemsel düşünme, problem çözmek için bir tür metodolojidir, bu yöntem bilgisayar bilimleri alanındaki tüm disiplinlere genişletilebilir, hesaplama yöntemiyle çözülebilecek problemleri analiz etmek ve geliştirmek için benzersiz bir yol sağlamaktadır (CSTA, 2016). Bilgi işlemsel düşünme kavramı, “bilgisayarları üretim amaçlı yaşam problemlerinin çözümünde kullanabilmek için gerekli bilgi, beceri ve tutumlara sahip olmak” denilebilir (Özden, 2015).

Brennan ve Resnick (2012) önerdiği bilgi işlemsel kavramlar, bilgi işlemsel becerileri, olayları, anlamsal ve stratejik kavramlar ile programlamayı içermektedir. Araştırmacılar tarafından tanım yapmak ve bilgi işlemsel düşünmenin çeşitli yönlerini araştırmak amacıyla kullanılmaktadır.

2.1.2 Bilgi İşlemsel Düşünmenin Bileşenleri

Apostolellis vd. (2014) bilgi işlemsel düşünmenin alt bileşenlerini ISTE tarafından tanımlandığı şekliyle K-12 eğitim öğrenciler için bazı kategorileri karşılaması gerektiğini savunmuştur. Bunlar;

- Verileri mantıksak olarak düzenleyin ve analiz edin
- Algoritmik düşünme yoluyla çözümleri otomatikleştirin
- Adımların ve kaynakların en verimli ve etkili kombinasyonunu elde etmek amacıyla olası çözümleri belirleyin, analiz edin ve uygulayın.
- Karmaşıklıkla başa çıkma konusunda özgüven sağlayın
- Zor problemlerle çalışma konusunda sebat edilmeli
- Belirsizliğe karşı tolerans
- Açık uçlu problemlerle başa çıkma yeteneği
- Ortak bir hedefe veya çözüme ulaşmak için başkalarıyla iletişim kurma ve birlikte çalışma becerisi

Grover ve Pea (2013) ise müfredatın temeli olarak ifade ettiği kategorileri sıralamıştır. Bunlar;

- Soyutlamalar ve örüntü genellemeleri (modeller ve simülasyonlar dâhil)
- Bilginin sistematik olarak işlenmesi
- Sembol sistemleri ve gösterimleri
- Algoritmik kontrol akış kavramları
- Yapısal problem ayrıştırma (modülerleştirme)
- İteratif, özyinelemeli ve paralel düşünme
- Koşullu mantık
- Etkinlik ve performans kısıtlamaları
- Hata ayıklama ve sistematik hata tespiti

Zamansky vd. (2019) bilgi işlemsel düşünme çok tartışılan bir terim gibi gözüke de unsurları tüm eğitimciler için aynı olduğunu söyler ve alt basamakları oluşturan etmenleri şu şekilde sıralar;

- Bir problemi yönetilebilir alt problemlere bölmek
- Sorunu ele almak için bir dizi talimat tasarlamak
- İlgili ve temel ayrıntılara odaklanarak sorunun karmaşıklığını azaltmak (bilgisayar bilimcilerin “soyutlama” olarak ifade ettiği kısım)

-Sürecin bir makine tarafından tekrarlanıp tekrarlanamayacağının belirlenmesi (sorunun etkin bir şekilde "otomatikleştirilmesi").

Alan yazın incelendiğinde (Aho, 2012; Apostolellis vd., 2014; Fernández vd., 2018; Grover ve Pea, 2013; Zamansky vd., 2019) belirli kavramları ortak bileşen olarak söylemişlerdir.

2.1.2.1 Soyutlama

Bilgi işlemsel düşünmenin özelliklerinden biri olan soyutlama sorununun en önemli kısımlarına odaklanan özellik olarak tarif edilmektedir. Buradaki esas gaye hedefin dışına çıkılması durumunda sorun ortadan kaldırıcı çözüm yolları bulmaktır. Soyutlama her yerde bulunur. Bunun bir örneği, bilgisayarların aynı anda zor görevlerde çalışabilmesi ve böylece kullanıcıların aynı anda birçok komutu çalıştırabilmesidir (Wing, 2017). Fernandez vd.'e (2018) göre dikkat en önemli özelliklere odaklanır; buna soyutlama süreci denir. Amaç, temel olmayan özellikleri filtreleyerek ve en alakalı olanları koruyarak sorunun özünü yakalayan basitleştirilmiş bir temsil veya model oluşturmaktır.

2.1.2.2 Bileşenlerine Ayırma

Bileşenlere ayırma ortaya çıkan problemleri daha küçük öğelere ayırma işlemidir. Öncelikle soruna bütüncül bir bakış açısıyla bakılır, daha sonra daha küçük parçalara bölünür. “Üç adımda açıklanır: maddelerin ilişkisini tanımak, olguyu kısmen aşağı çekmek için bir yöntem seçmek ve yöntemin diğer problemlerdeki etkinliğini araştırmak” (Rich vd., 2019).

2.1.2.3 Algoritmik Düşünme

Denning ve Tedre (2019) algoritmaları, ifadelerin komutlar halinde bilgisayarlarda işlenebilecek biçime getirilmesi olarak ifade etmektedir. Beecher (2017) algoritma bir görev döngüsünü içeren dizi yapılarından oluşan gerekli işlemleri yapan faaliyetler bütünüdür. Curzon vd. (2009) algoritmik düşünce için gerçeğe ulaşmak için kullanılan bilişim teknolojilerinin merkezinde olan ve bilişim teknolojilerinin bilimsel düşünmesindeki farklı bir açı olduğunu vurgulamaktadır. Diğer bir literatürde ise algoritmik düşünceyi süreç içerisinde bir yöntem bulma aracı olarak kullanıldığını açıklamaktadır (Futschek, 2006). Bu faktörler dikkate alındığında algoritmik düşünme de dizilerin organize içinde çalışması için belirli bir içerik uyumu ve doğru çalışmasını sağlayan kurallar içermelidir.

2.1.2.4 Mantıksal Düşünme

Beecher (2017) mantığın, belirli bir süreç içerisinde ifadelerin kabullenilmesi olarak açıklamaktadır. Bilgisayarların, komutları düzgün bir biçimde yorumlaması için ifadelerin açık bir biçimde verilmesi gerekmektedir; istenmeyen bir sonuç çıkması durumunda yine bilgisayarın ne yapacağı ile ilgili bilgi sunulmalıdır. Özetle mantıksal düşünme, sisteminde şunları yap aksilik çıkması durumunda şu işlemleri yap denilebilir.

2.1.2.5 Hata Ayıklama

“Hata ayıklama”, programlama yaparken ortaya çıkan işlem hatalarının düzeltilme işlemi şeklinde açıklanabilir (McCauley vd., 2008).

2.1.2.6 Örüntü Tanıma

Çetin ve Uçar’a (2017) göre örüntü tanıma çevreyi makinaların nasıl gördüğü, örüntüleri ayırt etmeyi nasıl bildiği ve örüntünün sınıflara dayalı olarak doğru ve düzgün bilgiyi nasıl karar verdiğini inceler. Dasgupta ve Purzer (2016), örüntü tanımanın birçok bilgi işlemsel düşünme boyutunu yansıtmasına rağmen bir bilgi işlemsel düşünme boyutu olarak çok az çalışıldığını bildirmektedir. Grover ve Pea (2013) “örüntü tanımanın” var olan sorunun daha kolay çözülebilmesini sağlamakta ve bu sorunun çözümünde harcanan zamanı azalttığını ifade etmektedir.

2.1.2.7 Değerlendirme

Csizmadia vd.’e (2015) göre değerlendirme ister bir algoritma ister sistem veya süreç olsun, bir çözümün iyi olduğundan, yani amaca uygun olduğundan emin olma sürecidir. Çözümün farklı yönleri değerlendirilmelidir. Doğru mu? Yeterince hızlılar mı? Kaynakları tasarruflu kullanıyorlar mı? İnsanların kullanması kolay mı? Uygun bir deneyimi teşvik ediyorlar mı? Tüm durumlar için nadiren tek bir ideal çözüm bulunduğundan ödün verilmesi gerekir. Bilgi işlemsel düşünmeyi değerlendirme aşamasında ayrıntılara daha bir önem verilerek veriler düzenlenmektedir.

2.1.3 Bilgi İşlemsel Düşünme Ne Değildir

“Ulusal Araştırma Konseyinin 2010 tarihli raporunda” bilgi işlemsel düşünme hakkında bilgiler vererek aynı zamanda bilgi işlemsel düşünmenin ne olmadığı konusunda da ayrıntılı açıklamalara yer verilmiştir. Bilgi işlemsel düşünme için bilhassa oyunlar, sanal ortam, bilişim okuryazarlığı ya da

programlamaya yoğunlaşmadığını ifade etmektedir. (National Research Council, 2010). Denning (2009) bilişimsel düşünme ile ilgili söylenenlere zıt olarak bilgisayar bilimleri araştırmacıları tarafından kullanılan becerilerin yeniden adlandırılması olarak açıklamaktadır. Denning'e göre bilgisayar bilimleri ve programlama arasında geçmiş zamanda bir benzetim kurulmuştur fakat bunun geçerli olmadığını savunmaktadır (Denning, 2009). Denning (2009) bilgisayar bilimi ve bilgi işlemsel düşünme içinde aynı benzerliğin kullanılması durumunda bunun konuya zararlı olacağı görüşünü bildirmektedir. Bilgi işlemsel düşünmenin iyi bir tanımını yapmak için bilgi işlemsel düşünmenin ne olmadığını da iyi analiz edilmesi gerektiğini ifade etmektedir.

2.1.4 Bilgi İşlemsel Düşünmenin Önemi

Bilgi işlemsel düşünme becerilerinin 21. yüzyıl toplumunu anlamak için bir gereklilik olduğu ve bu becerilerin öğretilmesi gerektiği konusunda bir fikir birliği vardır. Alışveriş sitelerinde, banka kartı kullanımlarında ve genetik kod haritalarında dahi örüntüler kullanılmaktadır. Bundy (2007) Bilgi işlemsel düşünme fizik, biyoloji, tıp, felsefe, mimarlık ve eğitime kadar pek çok alana yeni araştırma güçleri getirebileceğini belirtmiştir. Ulusal Araştırma Konseyi (2010) bilgi işlemsel düşünme hakkında "... sadece bilgisayar bilimcilerinin değil, herkesin sorunları çözmek, sistemleri tasarlamak ve insan davranışlarını anlamak için kullanabileceği temel bir analitik beceri". Bu, Wing'in (2006) bilgi işlemsel düşünmenin 21. yüzyıl toplumunda var olabilmek için herkesin bilmesi gereken temel bir beceri olduğu yönündeki orijinal fikrini yansıtmaktadır. Şayet toplumdaki kişiler bir beceriyi öğrenmek istiyor ise o zaman bu beceriyi öğrenmeleri için bir sistem oluşturulması ve kişilere sunulması gerekmektedir. Wing (2017) daha yakın tarihli bir çalışmasında, lisans ve lisansüstü eğitim veren kurumların bilgi işlemsel düşünmeyi müfredatlarına dâhil etmeye başladıklarını varsaymaktadır. Wing, bilgi işlemsel düşünme becerilerini lise ve ilköğretim de bulunan öğrencilerde öğretilmesini savunmaktadır. Isbell (2010) hedefleri arasına ilköğretim ve lisede de bilgi işlemsel düşünme becerilerinin öğretilmesini de eklemiştir.

2.1.5 Bilgi İşlemsel Düşünme Nasıl Öğretilir

Hangi kavramların öğretileceği noktasında bir fikir bütünlüğü olmadığı gibi bilgi işlemsel düşünmenin de nasıl öğrencilere aktarılacağı ve bunun programlama ile nasıl birleştirileceği hakkında da bir görüş birliği yoktur. Ayrıca

bedensel hareketlerin bilgi işlemsel düşünme becerilerini ve sınıfın derse olan ilgisini nasıl artırabileceği yönüne önem verilmektedir. Curzon vd. (2009) göre motivasyonel öğrenme en iyi hobi olarak yapıyormuş modunda gerçekleşir. Yapılacak olan aktiviteler sanatsal, bilimsel ve bilgisayar bilimlerini bir arada olacak şekilde öğrenmeyi daha eğlenceli bir hale getirebilir. Bazıları ise bedensel hareketleri ile öğrenmeyi daha da artırabilir (Curzon vd., 2009). Bilgi işlemsel düşünme becerilerinin matematik, fen bilimleri, sosyal bilimler gibi derslerde öğretilebileceği CSTA (2011) ve Fletcher (2009) tarafından dile getirilmiştir. Yine Guzdial (2008) bilgisayar biliminin bilgi işlemsel düşünme becerileri kazandıracak şekilde nasıl öğretileceğine dair daha fazla araştırmaya ihtiyaç olduğunu belirtmektedir. Burada ele alınması gereken bir konuda, programlamanın bilgi işlemsel düşünmenin bir parçası olarak öğretilip öğretilmeyeceğiyle ilgilidir. Hiç şüphesiz, bilgisayar bilimlerinin bir dalı olarak bilgi işlemsel düşünme programlamayı öğretmek içinde kullanılabilir.

Lu ve Fletcher (2009) daha da ileri giderek "... matematik için kanıt oluşturma ve İngilizce için edebi analiz neyse, programlama da bilgisayar bilimleri için odur" demektedir. Stanford Üniversitesi'nden Eric Roberts, bilgisayar bilimi alan ilköğretim ve lise öğrencilerinin bu süreç içerisinde programlama eğitimini de almalarının kaçınılmaz olduğunu belirtmiştir. Programlama eğitimi verilmemesi sonucunda bilgisayar bilimlerinin öneminin yok olacağını düşünmektedir (Curzon vd. 2009). Dahası, bilgi işlemsel düşünmenin öğretilmesine programlama dahil edilmediğinde yazılım süreci yaratıcı olmaktan çıkarak sıradanlaşabilir. Bedensel aktiviteler kullanarak öğrenmeye eğlence katmak övgüye değer olsa da bu aktivitelerin örtük ya da açık bir şekilde bilgi işlemsel düşünme becerilerini kazandırma kabiliyeti henüz belirlenmemiştir. Bilgi işlemsel düşünme, insanlara bilgisayar süreçlerinin nasıl işlediğinin öğretilmesinden ziyade, nasıl benzer biçimde düşüneceklerinin öğretildiği bir insanî süreçtir. Sınıfta yapılan faaliyetlerin önemsiz hale gelmemesine itina göstermeliyiz. Guzdial'ın (2008) bilgi işlemsel düşünme becerilerinin güçlenmesi için bilgisayar bilimlerinin daha iyi nasıl olması gerektiği noktasından daha fazla inceleme yapılması gerektiğini söylemiştir. Aynı zamanda Eric Roberts'ın programlamayı bilgi işlemsel düşünmenin gerekli bir parçası olarak görmesiyle de uyumludur, ancak önemi bir araç olarak görülmektedir (Curzon vd., 2009).

2.1.6 Bilgi İşlemsel Düşünmenin Eğitici Oyunlarda Kullanımı

Bilgisayar programlama ve bilgi işlemsel düşünmenin birbirleriyle örtüşen yönleri olduğundan, birçok öğretim ortamı öğrencilere bilgi işlemsel düşünme becerileri öğretmek için kodlamayı kullanır (Grover ve Pea, 2013). Programlamanın öğrenilmesindeki zorluğu gidermek için birçok platform blok tabanlı ortamı kullanarak kişilere oyunlaştırarak programlamayı öğretmeyi amaçlamaktadır. Bu ortamlara örnek gösterecek olursak scratch (Resnick vd., 2009; Zhang ve Nouri, 2019), ve blockly uygulamalarıdır. Araştırmalar, bu ortamlarla öğrencilerin programlama ve bilgi işlemsel düşünmedeki motivasyonunu geliştirmede bir miktar başarı göstermiş olsa da daha derin öğrenmeyi teşvik etme söz konusu olduğunda yetersiz kalmaktadır (Brennan ve Resnick, 2012; Meerbaum-Salant vd., 2011).

Eğitici oyunlar ile kişilerin sorunları eğlenceli bir biçimde çözerek, bilgi işlemsel düşünme becerilerine de katkı sağlanması amaçlanmaktadır. Eğitsel oyunların blok tabanlı ve görsel programlama yapan diğer ortamlardan farkı belirli bir amaç ile öğrenmenin gerçekleşmesini sağlamaktır (Land, 2000).

Bilgi işlemsel düşünme becerilerini geliştirmek için Zhao ve Shute (2019), öğrencilerin özellikle bilgi işlemsel düşünme becerilerini geliştirmek için bir video oyunu geliştirmiştir. Penguin Go adlı oyunu 2 saatten daha kısa bir süre oynamanın öğrencilerin bilgi işlemsel düşünme becerilerini önemli ölçüde geliştirdiği, ancak bilgi işlemsel düşünmeye yönelik tutumlarında herhangi bir etki bulunmadığı ortaya konmuştur. Ayrıca, blok tabanlı programlama dillerine dayalı ve dolaylı olarak bilgi işlemsel düşünmeyi teşvik etmeyi hedefleyen başka eğitici oyunlar da vardır, örneğin, RoboBuilder (Weintrop ve Wilensky, 2012), LightBot (Gouws vd., 2013) ve Code.org.

2.1.6.1 SCRATCH

Programlama öğrenmeye başlayan birçok öğrenci programlamayı çok ileri düzeyde eğitim almış kişilerin başarabileceği bir iş olarak görmektedir (Genç ve Karakuş, 2011). Buna neden olarak soyutlama, eleştirisel düşünme gibi birçok becerinin programlama öğreniminde kullanılması gerekliliği gösterilebilir (Gomes ve Mendes, 2007). Programlama öğrenmek için görselliği ön planda tutan Scratch, Microsoft Small Basic gibi programlama yapma araçları bulunmaktadır (Çatlak vd., 2015). Öğrencilere algoritma mantığını ve algoritmik düşünme yapısını

öğretmek için araç olarak Scratch kullanıldığı bilinmektedir (Armoni vd., 2015; Maloney vd., 2010).

Scratch, Mayıs 2007'de indirilebilir bir masaüstü uygulaması olarak başlatılmıştır. Kişiler scratch ortamını kullanarak kendi hikayelerini, içeriklerini ve animasyonlarını yapabilirler. Scratch web sitesine yükleyerek çevrimiçi toplulukla paylaşabilirler. Scratch 2.0 yayınlanması ve 2013 yılından itibaren Scratch'ın web ortamında da olmasından dolayı scratch'a olan ilgi daha da artmıştır. Scratch'ın sürekli büyümesini desteklemek ve projenin ölçeklenmeye devam etmesine yardımcı olmak için bir yola ihtiyaç vardı, bu nedenle Scratch Vakfı 2013 yılında bağımsız bir 501 (c) (3) kâr amacı gütmeyen kuruluş olarak kurulmuştur. Bu vakfın ilk kurucu üyeleri Mitchel Resnick ve David Siegel dir. Mitch ve David, Scratch'ın ücretsiz bir şekilde tüm dünyadaki kişilerin yaratıcılıklarını kodlama ile ortaya çıkarma vizyonu ile bu vakfı kurmuşlardır (Scratchfoundation, 2022).

Scratch, çocukların kullanabileceği dünyadaki en büyük kodlama topluluklarından biridir. Gençlerinde basit görsel bir arayüz vasıtasıyla oyunlar, dijital hikâyeler ve animasyonlar geliştirebileceği bir kodlama dilidir. Scratch, bilgi işlemsel düşünme ve problem çözme becerilerini desteklemekle beraber yaratıcı öğrenme, kendini ifade etme gibi kavramlarında gelişmesini desteklemektedir. Scratch ücretsiz olmakla beraber 70'ten fazla dili desteklemektedir. Scratch 200'den fazla ülke ve bölgede kullanıma açıktır. Scratch özellikle 8-16 yaş arasındaki bireyler için tasarlanmış olup her yaşta insan tarafından da etkin olarak kullanılmaktadır (Scratch About, 2022). Scratch, Millî Eğitim Bakanlığı tarafından 5 ve 6. sınıflarda bilişim teknolojileri dersi programlama konusunda; geniş bir kullanıcı kitlesine hitap ettiği ve çok sayıda dil desteği içerdiği, etkileşimli arayüz grafiğine sahip olması açısından derslerde kullanılmaktadır (Scratch About, 2022).

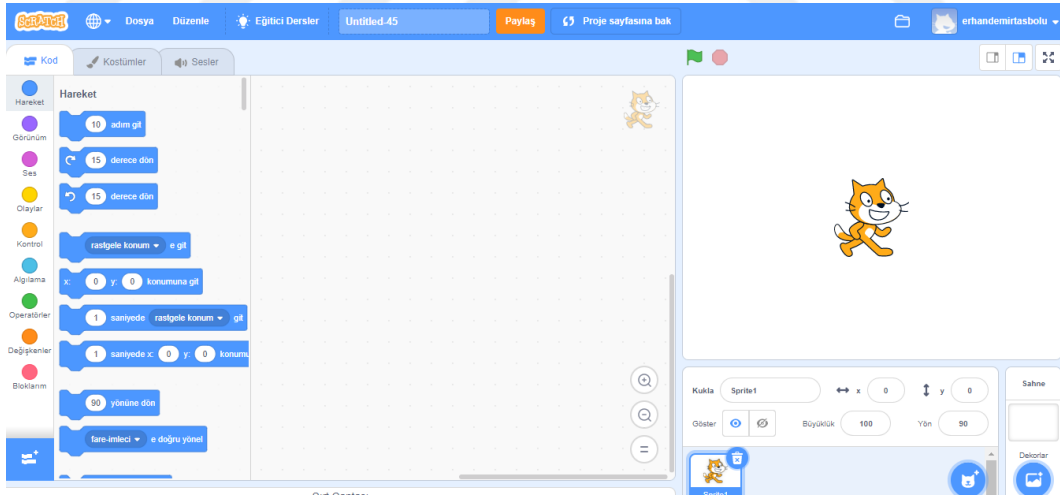
Scratch, sahip olduğu özellikler ile öğrencilerin tüm enerjilerini ve dikkatlerini hayalindeki projeye verebilmekte ürünlerini sergilemelerini sağlamaktadır. Scratch ortamında sürükle-bırak yöntemi ile programlar oluşturulmaktadır. Birbiri ile alakalı olmayan kod blokları birbiri ile birleşmesinin önü kesilmiş ve yanlışlığa mahal vermemektedir ve ezberleme de bu sayede ortadan kaldırılmıştır (Karabak ve Güneş, 2013; Maloney vd., 2004).

Bu çalışmada Scratch web versiyonu kullanılmıştır. <https://scratch.mit.edu/> adresinden ulaşım sağlanmaktadır.



Şekil 2.1 Scratch programı web giriş sayfası

Şekil 2.1 de görüldüğü üzere Scratch sayfasına girildiğinde giriş, scratch katıl, oluştur, keşfet fikirler ve hakkında sekmeleri bulunmaktadır. Scratch katıl denilerek programa üye olunmakta ve yapılmış olan programların kayıt altında bulunması sağlanmaktadır.



Şekil 2.2 Scratch program oluşturma sayfası

Scratch programında oluştur sekmesine tıklanılarak web sayfasında sürükle-bırak yöntemi ile programlama yapabileceğiniz sayfa açılmaktadır. Scratch programını web olarak kullanıldığında yapılan çalışmalar bulut sistemde

kayıtlı olarak bulunmakta her nerede olursanız olun yaptığınız uygulamalara ulaşabilme imkânı sunmaktadır.

Eğitimciler için Scratch öğrenciler için sınıf oluşturulmasını sağlamak ve öğrenci takibini kolaylaştırmaktadır. Eğitim rehberleri yardımıyla Scratch sınıflarını ve atölyelerinin nasıl hazırlanacağı ve çalıştıracağını gösteren birçok uygulama sunmaktadır. Harvard'daki Scratch ekibinden gelen yaratıcı bilgi işlem, sınıfta yaratıcı bilgisayar kullanımı için planlar, aktiviteler ve stratejiler sunmaktadır (Scratch Educators, 2022).

2.2 Kodlama Eğitimi

Gelişen ve gelişmekte olan teknolojiler hayatımızın her alanını etkilemektedir. Teknoloji aynı zamanda eğitim alanını da etkilemeye başlamıştır. Teknoloji ve eğitimin bütünleşmesi ve 21.yy. becerilerinin kullanımı bakımından özellikle küçük yaşlardan itibaren bu gereksinimleri karşılayacak beceriler kazandırılması bir zorunluluk hâline gelmektedir. Eğitim alanında bu zorunluluktan dolayı yeni yazılımların ortaya çıkması sağlanmıştır (Eryılmaz, 2003). Bu çerçevede 21.yy becerileri göz önünde bulundurulduğunda kodlama eğitiminin önemine daha mantıklı cevaplar ile yaklaşılabilir (Sayın ve Seferoğlu, 2016). Kodlama mantığının bireylere erken yaşlarda kazandırılması bireylerin geleceğe hazırlanması ve diğer alanlarındaki başarıları da katkı sağlayacağı bir fırsat oluşacaktır (Karabak ve Güneş, 2013). Yapılan araştırmalara göre kodlama eğitimi alan bireyler ile kodlama eğitimi almayan bireyler arasında birçok farkın olduğu görülmektedir (Akpınar ve Altun, 2014).

Bir problemin çözümü için ilk önce problemin tanımı yapılmalı ve problemin çözümü için hangi basamakların kullanılacağını bilmesi gerekir. Bilgisayar tarafından işlenen veya kişinin problem çözme basamaklarında uyması gereken yöntemi, yolu, kuralları sıralı ve mantığa uygun ifade edilmelidir (Michael ve Omoloye, 2014). “Algoritma, karşılaşılan herhangi bir sorunun çözmeye yönelik atılan adımların gösterimi olarak tanımlanabilir” (Eker, 2011). Algoritma geliştirme sürecinde yapılan işlemler sadece bilgisayar bilimleri alanında değil hayatın her kısmında kullanılmaktadır (Tepgeç, 2017). Bu kapsamda tüm dünya ülkeleri kodlama eğitimini küçük yaşlardan itibaren çocuklara kazandırılması konusunda harekete geçmiştir. Örnek olarak İspanya’da “algoritma, programlama ve robotik”, Belçika’da “bilişimsel düşünce ve programlama”, Bulgaristan’da “algoritmik problem çözme ve programlama,

İngiltere’de “computing” kavramları kodlamanın karşılığı olarak sunulmaktadır (Balanskat ve Engelhardt, 2014).

Gove, İngiltere’de şu anda bulunan müfredatın ülkenin ekonomik olarak geleceğini zora soktuğunu ve altyapılarının yetersiz olduğunu söyleyerek bilgi ve iletişim teknolojileri dersi müfredatının değiştirilmesi gerektiğini söylemiştir (Demirer ve Sak, 2016).

Tüm dünyada programlama eğitimi genellikle lisans düzeyinde yapılmaktadır (Karabak ve Güneş, 2013). Yalnız farkındalık oluşturacak çalışmaların yapılması için farklı seviyelerde kodlama becerilerinin kazandırılması gündeme gelmiştir. Örnek olarak 2013 yılının Kasım ayında İngiltere’de ilköğretim seviyesinde de kodlama eğitimi başlamıştır. 2014 yılı tüm ülke genelinde “Kodlama Yılı” (Year of Code) olarak ilan edilmiştir.

Avrupa’da kodlama üzerine yapılan bu çabaların Amerika Birleşik Devletleri’nde de benzerleri görülmektedir. ABD başkanı öncülüğünde “Herkes kodlamayı öğrenebilir” iletişi ile “code.org” ve “kodlama saati” gibi çalışmalar sadece ABD’de öncülüğünde diğer ülkelerde de uygulamaya başlanmıştır (Code.org, 2022). 2016 yılında ise Finlandiya algoritmik düşünme konularını zorunlu dersler olarak okutulmasına karar vermiştir (Gülbahar ve Kalelioğlu, 2018).

Türkiye’de 2012 yılında alınan karar önceden “bilgisayar” olarak ifade edilen ders “bilgi teknolojileri ve yazılım” olarak yenilenmiştir. Yazılım ifadesi ile derse algoritma ve programlama ile ilgili konular eklenmiştir. 5 ve 6.sınıftan itibaren zorunlu 7 ve 8’ler de seçmeli ders olarak öğrencilere programlama eğitimi verilmeye başlanmıştır (MEB, 2012). Ayrıca programlama eğitimini yaygınlaştırmak amacıyla MEB, sivil toplum örgütleri, üniversiteler, dernekler, şirketlerde çeşitli projeler yapmaktadır.

Avrupa ülkelerinin kodlama eğitimine dikkat çekmek için 2013 yılından itibaren küçük yaşlardaki öğrencilere yönelik 6-21 Ekim tarihleri arasında CodeWeek (kodlama haftası) etkinlikleri yapmaktadır (codeweek about, 2022). Türkiye’de ise MEB protokolü ile 2018 yılında başlayarak kodlama haftası etkinlikleri yapılmaya başlanmıştır (Codeweek Türkiye, 2022). 2017 yılında geleceğin mühendislerini, yazılımcılarını yetiştirmek amacıyla kurulan Türkiye Teknoloji Takımı Vakfı tarafından deneyap atölyeleri faaliyete başlamıştır (Deneyap, 2022). Milli teknoloji hamlesi kapsamında Türkiye’nin gelişmesi ve

her alanda bağımsızlığının güçlendirilmesi hedefi ile birçok farklı kategoride yarışmalar düzenlenerek her düzeyden kesimin bu teknoloji hamlesine katılmasını amaçlanmaktadır (Teknofest, 2022). Kodlama ve robotik eğitimi veren atölye çalışmaları yapan; Eba-Mobil Kod, Habitat Derneği, Türkiye Bilişim Derneği, Maker Çocuk, STEM Çocuk gibi örgütlenmeler bulunmaktadır (Yıldırım, 2020).

2.3 Tutum

Allport'a (1935) göre çoğu soyut terim gibi, tutumun da birden fazla anlamı vardır. Latince aptus kelimesinden türetilmiş olup bir yandan "uygunluk" ya da "uyarlanmışlık" anlamını taşır ve yan biçimi olan aptitude gibi eylem için öznel ya da zihinsel bir hazırlık durumunu ifade eder. Bununla birlikte, sanat alanındaki kullanımıyla, terim oldukça bağımsız bir anlama sahip olmuştur; heykel veya resimdeki bir figürün dışa dönük veya görünür duruşuna (bedensel pozisyonuna) atıfta bulunmuştur. Allport (1935) "tutumu deneyim yoluyla organize edilen, bireyin tüm davranışlarına verdiği tepki üzerinde yönlendirici veya dinamik bir etki yaratan zihinsel ve sinirsel bir hazır olma durumudur" ilişkili olduğu nesneler ve durumlar şeklinde ifade etmektedir.

Triandis (1971) tutum ile ilgili tanım yapan birçok araştırmacıdan yola çıkarak şöyle bir tanım yapmaktadır. Onun tanımına göre, "tutum, belirli bir sosyal durum sınıfına yönelik bir eylem sınıfına yatkınlık kazandıran duygu yüklü bir fikirdir ve tutum teorisyenleri tarafından kullanılan birçok fikri içermektedir". Tanımda tutumun üç bileşeni olduğu öne sürülmüştür: "(a) Bilişsel bir bileşen, yani genellikle insanlar tarafından düşünmede kullanılan bir kategori olan fikir. Gruplar, farklı uyarıcılara gösterilen tepkilerin aynı biçimde olması ile ortaya çıkmaktadır. Örneğin arabalar grubu, insanların Ford'lara, Chevrolet'lere vb. ve ayırt edebildikleri diğer uyaranlara benzer tepkiler verdiğini belirleyerek çıkarılabilir. (b) Duygusal bileşen. Bir kişi bir grup hakkında düşünceleri iyimser veya kötümser bir duygu içindeyse bu grupta bulunan bileşenlere de olumlu veya olumsuz yönde bakmasına neden olmaktadır. Örnek olarak arabalar için birey iyimser düşünmektense arabalar hakkında da duygusal bir bileşeni vardır denilebilir. (c) "Davranışsal bir bileşen, yani araba kullanmak, satın almak veya arabalara hayranlık duymak gibi eyleme yönelik bir yatkınlık".

Son dönemlerde bir kısım araştırmacılar (Di Martino ve Zan, 2001; Ruffell vd., 1998) tutum hakkında net bir tanım olmadığını yapılan tanımların ise genellikle kabul edilebilir olduğu fakat daha fazla araştırmanın yapılması

gerektiğini söylemişlerdir. Bununla birlikte, günlük hayatta tutum terimi, bir kişinin tanıdık bir nesneye yönelik temel hoşlanma ya da hoşlanmama durumu olarak kullanılmaktadır. Mouly (1960) “tutumların, bir bireyi herhangi bir durumla karşılattığında belirli şekillerde davranmaya yönelten öğrenilmiş davranış kalıpları olduğunu iddia etmiştir”. Kerlinger vd. (2000) ise “tutumları, bilişsel bir hedefe yönelik düşünme, değer verme ve hareket etme yatkınlığını temsil eder”. Sherif ve Cantrel (1965) tutumları kişilerin olaylara karşı nasıl tepki koyacağını gösterdiğini söylemiştir. Son olarak, McLeod (1992) tutumun orta yoğunlukta ve makul düzeyde istikrarlı olumlu ve olumsuz duyguları içeren duygusal tepkiler olduğunu belirtmiştir. McLeod’un (1992) duyuşsal alan kavramlarına ilişkin sınıflandırmasında tutum kavramı, duygular ve inançlar kavramlarının ortasında yer almaktadır.

Fazio (1995) başka bir tanım sunarak nesne-değerlendirme ilişkileri olarak tutumlara değinmektedir. Bu tanım, örtük tutum ölçümlerini kullanan araştırmalar için en önemli çerçevelerden biri olan MODE Modelinin Fazio (1995) temel bileşenini temsil etmektedir. “Özellikle MODE modeli, tutumların bellekte nesne değerlendirme ilişkileri olarak temsil edildiğini ve bu ilişkilerin güçlerinin değişebileceğini savunur”. Bu çağrışımlar yeterince güçlüyse, nesneyle ilgili uyaranlarla karşılaşıldığında değerlendirmeler otomatik olarak etkinleştirilebilir. Bu otomatik olarak etkinleştirilen tutumun, tutum nesnesinin değerlendirici yargısını basitleştirip basitleştirmediği, insanların motivasyonuna ve tutum nesnesinin değerlendirme özelliklerini ayrıntılı olarak işleme fırsatına bağlıdır. Fazio (1995) tutum inşası ile kişilerin çevresindeki bireyleri doğrudan doğruya izleyemese bile o kişi hakkında olumlu veya olumsuz görüş bildirme sürecine değinmektedir. Bu kavramsallaştırma otomatik olarak harekete geçtiğini ve şekil aldığını söyler.

2.3.1 Tutumun Oluşumunda Etkili Faktörler

İnsanların tutumları doğuştan gelen özellikler değildir (Sherif, 1965). Tutumlar, yaşam ve deneyimin sonucudur. Kişinin hayatında çevre ve toplumun etkisi altında öğrenmesinin sonucudur. Tutumların oluşumu hızlı bir şekilde gerçekleşmez, tutumlar uzun vadeli birikimlerle yavaş yavaş oluşur. Düşünceler gibi geçici durumlar değildir. Tutum, bireyin davranışlarının oluşumunda tutarlı ve kararlı davranmasını sağlar. Tutumlar, öğrenme sürecinde çevre ile ilgili olarak kademeli olarak oluştuğundan, kullanımda alışılmış nesneye ve

çevreye uyarlanmış sabit tutumlar haline gelirler. Tutumlar edinilmiş olmakla birlikte, yoktan var edilmez öğrenilmezler, bir nesneye, bir olaya, değerlere ve yargılara ve öğrenme aşamasının kültürel koşullarına göre edinilirler. Kullanımdaki tutumlar, öğrenme aşamalarını etkileyen materyal kullanımından bağımsız olamaz. Yani tutumlar tarafsız değildir. Öğrendiklerinde etkileniyorlarsa kullandıklarında da etkileniyorlar (Sherif, 1965).

Tutum genellikle bir nesne veya nesne sınıfına karşı tutarlı bir değerlendirme tarzında tepki vermeye yönelik öğrenilmiş bir yatkınlık olarak kavramsallaştırılır (Allport, 1935; Rosenberg ve Hovland, 1960). “Tutarlı değerlendirici tutum” terimi, olumlu, olumlu ve olumsuz gibi farklı özelliklere sahip bir boyutu ifade etmektedir. Ortalama olumlu tutuma sahip olan kişiler sorulan sorulara genellikle olumlu bir biçimde yanıt vermektedir. Bazen bu kişiler ne olumlu ne olumsuz veya olumsuz yanıt verme olasılıkları da vardır. Bu yaklaşım belirli bir özellik veya konunun değerlendirilmesinde deki tek bir boyut üzerinden genellemeyi söyler. Değerlendirici yanıtlardaki bu değişkenlik iyi kurulmuş olsa da güncel teori ve araştırmaların çoğu, insanların verdikleri tutumsal yanıtlarda değerlendirici homojenliği korumaya çalıştıkları varsayımına dayanmaktadır. Bunlar, “tutumların bilişsel tutarlılık teorisinin” öngörüsüdür (McGuire, 1966).

Rosenberg ve Hovland (1960) ve onlardan önceki araştırmacılar (Allport, 1935; Harding vd., 1954) üç sınıf değerlendirici tepki tanımlamışlardır. Değerlendirici tepkiler tutumun kişilerin inançları, önyargıları, deneyimleri sonucu alabileceği kararların tümü şeklinde açıklanabilmektedir. Sempatik sinir tepkileri ve duygulanımla ilgili sözel ifadeler duyuşsal bileşende; açık eylemler ve davranışla ilgili sözel ifadeler davranışsal bileşende; algısal tepkiler ve inançlarla ilgili sözel ifadeler ise bilişsel bileşende yer alır. Bu ayırım, geçmişte araştırma sonuçlarının karşılaştırılmasını kolaylaştırmak için bir sınıflandırma sistemi olarak sıklıkla yararlı olmaktadır ve (Harding vd., 1954) daha yakın zamanda Hovland (1960) ve Schopler'in (1967) yapmış oldukları çalışmalarda dahada önemli hale gelmiştir.

2.3.2 Tutumu Oluşturan Bileşenler

Bireyde bulunduğu düşünülen tutumlar bilişsel, duygusal ve davranışsal olarak bulunmaktadır (Kağıtçıbaşı, 2010).

Tutumları duygusal bir faktör olarak gördüğümüzde, kişisel geçmiş deneyimler tutumların oluşumunda önemli bir rol oynar (İnceoğlu, 2010). Bireyin önceki yaşamındaki olumlu koşullar, olumlu tutumların oluşmasına yol açsa da birey benzer deneyimlerle karşılaştığında da olumlu duygular besleyecektir. Aynı şekilde olumsuz durumlar da benzer durumlarla karşılaşıldığında bireylerde olumsuz tutumlara ve olumsuz duygulara yol açmaktadır. Amaca uygun bir tutum bireye hizmet edecektir. Duygusal faktörler, bireylerin olumlu veya olumsuz tutumlara sahip olma konusundaki apriori eğilimleridir. Duygusal faktörlerin oluşturduğu tutumları değiştirmek diğer faktörlere göre daha zordur (İnceoğlu, 2010).

Tutumları bilişsel bir unsur olarak düşündüğümüzde, bireyin deneyimlerine dayanarak çevredeki uyaranlara tepki olarak oluşturduğu bilgi, yorum, anlayış ve inançlardır. Birey, deneyimlerini kendi içinde gruplara ayırır ve karşılaştığı durumlara göre bu grupları birbirine bağlar. Bu çağrışımlar aracılığıyla bireysel inanç sistemleri ortaya çıkar (İnceoğlu, 2010). Tutumları etkileyen bilişsel faktörler, bireyin çevresindekileri iyi-kötü, olumlu-olumsuz, yeterli-yetersiz şeklinde derecelendirmesini sağlar. Bireysel tutumların oluşmasında ve davranışa dönüşmesinde önemli bir etkiye sahiptir (Güney, 2008).

Tutumları davranışın unsurları olarak ele aldığımızda, bireyin bir durum hakkındaki bilgi ve inançları, olumlu ya da olumsuz davranışın şekillenmesinde etkilidir. Olumlu davranışsal öğeler içeren durumlarda yakın, ilişkilendirilebilir davranışlar, olumsuz davranışsal öğeler içeren durumlarda ise eleştirel, mesafeli ve mesafeli davranışlar gösterir. Davranışsal faktörler kişisel duygu ve düşüncelerden etkilenir. Bireyler duygularına ve düşünme eğilimlerine göre hareket etme eğilimindedirler (Konuk, 2011).

2.3.3 Programlamaya Yönelik Tutum

Tezbaşaran (1996) tutumu, konum, belirli bir biçimdeki nesne, ifade veya kişilere karşı yapılan olumlu veya olumsuz davranışlar olarak ifade etmektedir. Tutumlarımız nasıl bilgi elde edebileceğimizi ve ne tür davranışlar sergileyeceğimizi de etkilemektedir (Maio ve Haddock, 2009). Öğrencilerin herhangi bir derse karşı yönelik tutumları akademik başarılarını da etki altına almaktadır. Yapılan çalışmalarda görülmektedir ki akademik başarı ile derse olan tutum arasında pozitif bir ilişki bulunmaktadır (Baser ve Geban, 2007). Buradan

çıkarılacak sonuçla öğrencilerin programlamaya karşı gösterdikleri tutumlar programlama başarılarını da etkileyecektir (Erdoğan vd., 2008).

Son zamanlarda bilişim sektöründe yazılım geliştirme en yüksek payı almaktadır. Bilgisayar bilimleri endüstrisinde program yazabilme yeteneği önemli bir yer tutmaktadır. Bununla birlikte başlangıç seviyesindeki çoğu öğrenciler için zaman zaman programlama dersi zor olarak algılanmaktadır (Aşkar ve Daveport, 2009; Başer, 2013). Başka bir kaynakta ise bilgisayar programlamanın asosyal, sıkıcı ve zor olarak öğrenciler tarafından algılandığı ifade edilmektedir (Farkas ve Murthy, 2005). Bu durumlar öğrenciler için programlama dersinde başarısız olmalarını ve programlamaya karşı olumsuz tutum geliştirmelerine neden olmaktadır (Başer, 2013). Öğrencilerin derslere yönelik tutumları ile akademik performansları arasında pozitif bir ilişki bulunmaktadır (Başer ve Geban, 2007; Nisbet, 2006). Dolayısıyla öğrencilerin programlamaya yönelik olumlu ya da olumsuz tutumlarının programlama başarılarını da etkilediği söylenebilir (Başer, 2013; Erdoğan vd., 2008; Tai vd., 2003). Ayrıca, programlamaya yönelik olumsuz algılar ve düşük motivasyon gibi psikolojik faktörler de programlama öğrenme sürecini olumsuz etkileyebilir (Heersink ve Moskal, 2010).

2.4 Öz Yeterlik

Öz yeterlik terimi ilk olarak Albert Bandura tarafından sosyal öğrenme teorisinde önerildi. 1977 yılında “Self efficacy: Toward a unifying theory of behavioral change” adlı eserinde bu konuyu işlemiştir (Bandura, 1977). Bandura, *sosyal öğrenme kuramını ortaya* atarak kişilerin becerilerini etkin bir biçimde kullanmasına ön koşul olarak kendilerini özgüvende hissetmelerini söylemiştir. Bu kuramın en önemli kavramlarından biri de öz yeterlik olmuştur (Kurbanoğlu ve Takunyacı, 2012). Öz yeterlik, İngilizce olarak ifade edilen “self-efficacy” tercüme edilip yetkinlik beklentisi olarak çevrilmiştir (Yıldırım, 2010).

Alan yazın da öz yeterlik için çeşitli tanımlar bulunmakta ve genellikle ortak noktada birleşmektedirler. Bu ifadenin duayeni olan Bandura (1986) öz yeterlik için “*bireylerin karşılarına çıkabilecek olası durumlarla baş edebilmek için gerekli olan davranışları ne kadar iyi yapabileceklerine ilişkin bireysel yargıları*” şeklinde söylemektedir. Diğer araştırmacılar tarafından yapılan tanımlar; Bireyin kendi performansını sergileyerek yapabileceği davranışlar bütünü (Açıkgöz, 2005), bireylerin karşılaştığı farklı durumlarda yeteneklerini kullanarak başa çıkabilme inancı (Woolfolk, 1990) Bireylerin karşılaştıkları

durumlara göre bireysel kapasitesi hakkında yorum yapması (Meger, 1997) şeklinde tanımlar bulunmaktadır. Ekici (2006) öz yeterlik kavramı için bireyin içinde bulunduğu davranışların meydana çıkmasında önemlidir vurgusu yapmıştır. Literatürde öz yeterlik, öz yeterlik algısı, öz yeterlik yargısı ve öz yeterlik inancı olarak da ifade edilmektedir (Aşkar ve Umay, 2001).

Öz yeterlik teknik anlamda “algılanan öz yeterlik” olarak da geçmektedir (Senemoğlu, 2003). Senemoğlu’na (2003) göre öz yeterlik kişilerin gelecekte karşılaşabilecekleri problemler ile ilgili problemleri çözme yöntemlerini kullanmada ne derecede başarılı olduğu konusunda kendini yargılamasıdır. Gelecekte karşılaşılabilecek problemlere örnek olarak; yarışmalara katılma, sınavlar olma, bir okulda öğretmenlik yapma, herhangi bir topluluk önünde sunum yapma vb.

2.4.1 Akademik Öz Yeterlik

Bireyin akademik olarak istenen görevleri başarılı ve istendik düzeyde yapabileceğine olan inancına akademik öz yeterlik denilebilir (Gore, 2006). Kişilerin bireyselleşmiş vaziyet ve hünelerinden farklı olarak, akademik olarak verilen görevlerin yapılabilmesi, özgüveninin bireyde oluşmasını vurgular (Milburg, 2009).

Bandura’ya (1997) göre öz yeterlik inancı, bireylerin hayattaki amaçlarını, yaşam stillerini zorluklara karşı gösterdiği çabayı etkiler. Yeteri kadar öz yeterlik seviyesinde olmayanlar karşılaştıkları problemlerin olağandan daha zor olduğunu düşünürler ve sorunlarını çözemezler. Lakin öz yeterlikleri yüksek olan bireyler ise karşılaştıkları bu tip problemlerin üstesinden gelmekte oldukça başarılıdır (Kaptan ve Korkmaz, 2002).

Duman (2007), Öz yeterlik algıları yüksek veya düşük olan bireylerin özelliklerini tablo 2.1’de karşılaştırmıştır.

Tablo 2.1 Öz yeterlik algıları yüksek veya düşük olan bireylerin özellikleri

Öz Yeterlik Algısı Yüksek Bireyler	Öz Yeterlik Algısı Düşük Bireyler
Başa çıkılması zor gözüken işlerden kaçmak yerine onlarla mücadele edilmesi gerektiğini düşünürler.	Başa çıkılması zor gözüken işleri çaba göstermeden bırakırlar.
Verilen vazifelere karşı yapma	Verilen vazifelere karşı yapmakta çok

istekleri yüksektir.	az istekli davranırlar.
Bireyler kendileri için ilgi çekici ve yapmak istedikleri amaçlar edinirler.	Fazla uğraş gerektirmeyen ufak çaplı işler ile meşgul olurlar.
Zor işler ile karşılaştıklarında daha çok çaba gösterme isteği duyarlar.	Zor işleri gördüklerinde işin zorluk kısmına ve başarısızlık kısmına odaklanırlar.
Zor işler karşısında yapacakları işe odaklanırlar ve farklı yollardan çözme eğilimi gösterirler.	Zor işler karşısından başaramam duygusu ve çıkan terslikler üzerinde daha çok dururlar.
Başaramama durumlarında kendilerinin bilgi yoksunluğuna ve yeterince işlerinde çaba harcamadıklarını söylerler.	Başaramamalarının kendi yoksunlukları ile açıklarlar.
Başaramama durumunda öz yeterliklerini hızlı bir şekilde aktif hale getirirler.	Başaramama durumunda öz yeterliliklerini aktif hale getirmekte zorlanırlar.
Stres anlarında güven oluştururlar ve kontrolü sağlarlar.	Stres anlarında hemen pes ederler.
Başarılı neticeler almayı düşünürler.	İyi neticeler almayı düşünmezler, olumsuz düşündükleri için istedikleri seviyeye çıkamazlar.

2.4.2 Sosyal Öz Yeterlik

Kişiler kendi hareketlerinin sonucunu yorumlayarak ve başkalarının hareketlerinden de bazı dersler çıkararak öz yeterlik algılarını geliştirirler (Kurbanoğlu, 2004). Özyeterliği güçlü hâle getirmenin diğer yöntemi ise başkalarının deneyimlerini gözlemektir (Bandura, 1997). Alderman'a (1999) göre yeni bir maharet elde edilmesinde, başarılı kişileri gözlemlemenin yararı olmaktadır. Sosyal öz yeterliği, öz yeterliğin ayrılmaz bir parçası olarak görür, bireysel organizasyon yeteneğine olan inancı ve belirli amaçların üretilmesi için gerekli işin uygulanması sosyalleşmeyi içerir (Bandura, 1997). Bireylerin sosyal iletişim kurma ve yeni arkadaşlıklar geliştirme konusunda yapabilecekleri ve girişimde bulunabilecekleri sosyal öz yeterlik inançlarını içermektedir (Wood ve

Olivier, 2004). Gaudiano ve Herbert (2003) sosyal özyeterlik, başkalarına olumlu bir izlenim iletme yeteneğine duyulan güven duygusu ve sosyal anksiyete bozukluğunda sosyal özyeterliğin önemini vurgulamaktadır. Fan ve Mak (1998) sosyal öz yeterliği şu şekilde tanımlamaktadır: Sosyal davranışın belirli bir sonuca ulaşmak için birey tarafından uygulanan istendik davranıştır. Bandura (1994), sosyal öz yeterliği şu şekilde tanımlamıştır: Bireyin sosyal ilişkiler kurmasını ve başkalarıyla birlikte çalışmasını, sosyal durumlar ve sosyal çatışmalarla uğraşmasını sağlayan yeteneklerinin farkında olmasıdır.

Sosyal öz yeterlilik, öz yeterliliğin az sayıda çalışıldığı bir dalıdır (Malinauskas vd., 2018; Zullig vd., 2011). Sosyal öz yeterlik, problem çözme, liderlik becerisi, etkili iletişim kurma, grup içinde etkili çalışma, yeni ortamlara ayak uydurma gibi becerileri içermektedir (Connolly, 1989). Sosyal öz yeterlilik düzeyi, kişilerin iyimser yönde ilişkilerinde ve geliştirici yönlerinin de gelişmesine katkı sağlamaktadır (Malinauskas vd., 2018). Sosyal öz yeterliği üst düzeyde olan bireyler herhangi bir olay karşısında çözüm geliştirmekte farklı yollar bulabilirler ve bu konuda kendilerine güvenleri yüksektir (Malinauskas vd., 2018). Sosyal öz yeterliliği düşük bireylerin önemli bir sorunu da öğrenme güçlüğü çeken ve antisosyal, şiddet içeren ve suç teşkil eden davranışlar sergileyen, herhangi bir diploma veya mesleki beceri olmadan okulu bırakan kişiler oluşturmakta ve bu kişiler istihdam yönünde de negatif yönde etkilenmektedirler (Akelaitis ve Lisinskiene, 2018).

2.4.3 Entelektüel Öz Yeterlik

Bandura (1977) tarafından ortaya atılan öz yeterlik beklentileri, öğrencilerin sınıftaki etkinlikleri seçmedeki ana belirleyicilerdendir. Ayrıca Moos ve Azevedo (2009) öz yeterliği “sosyal bilişsel kuramın” ilklerinden olduğunu söylemiştir. Sosyal bilişsel kuram, “öğrencilerin bulunmakta olan motivasyonlarına, gerekli eylemlerin süreci ve bilişsel yeteneklerin doğal sonucu olarak bireyin kendi içsel algısına dayalı karşılama becerileri olarak söylenmektedir. Sınıf ortamı bağlantılı olan belirli zihinsel veya bilişsel faktörler, katılımı veya etkileşimi sınırlandırabilmektedir. Öğrenciler arası iletişim ve olaylara katılma, öğrencilerin ödevlere merak duyma ve onları değerli bulma inancı ile bağlantılı olabilir (Pintrich ve De Groot, 1990).

Sınıf ortamları bazen öğrencilerin katılımını veya etkileşimini engelleyebilir, bu da öğrencinin entelektüel öz yeterliliğini veya içeriğin

bilinmesine yönelik güvenini artırır. Öğrencilerin içeriğe olan inancı ve sınıfa katılma isteği, bilginin alternatif bir açıdan ele alınmasını gerektirir. Eğitim ortamında, sadece yaşlılarıyla değil öğretmenleriyle de iletişim kurma becerisi okul öncesi dönemde başlayarak küçük yaştaki öğrencilerin eğitim yılları boyunca devam eder ve bunun sonucunda sınıf ortamının da etkisi olur. Mesela, bütün sınıftaki herkes tek bir konu üzerinde çalışır ve sınıfa katılmak için öğretmen bir öğrenciyi çağırır. Öğrenciler sıklıkla arkadaşları karşısında hatalı yanıt verme veya kendilerine karşı dikkat çekme konusunda endişe duyarlar ve bu da öğrenci hakkında genel bir kanı oluşmasına neden olur (Marty vd., 2007). Diğer öğrencilerin sınıftaki görüşleri sosyal karşılaştırmaya katkıda bulunur. Bilgisayar laboratuvarında yapılan derslerde sosyal karşılaştırma, özellikle sınıf deneyimleri akademik yetersizlikleri veya başarıları temsil ettiğinden, öz yeterliliğin gelişmesinde etkili bir rol oynar. Öğrencilerin yaşlıları ile bilgi açısından mukayese edilmesi, öğrencilerin öğrenim çevresinden farklı olarak becerileri ya da öğrenim kısıtlamaları konusunda bir miktar güvensizlik yaratabilmektedir. Sınıf içindeki iletişim ve etkileşimler, bir öğrencinin ödevlerini tamamlama konusundaki güvenini pekiştirir; bu da genellikle sınıfa uyum sağlayarak ödevlerini tamamlamasını sağlar (Usher ve Pajares, 2008).

2.4.4 Özyeterlik İnancının Kaynakları

“Öğrencilerin bilişsel ve psikomotor öğrenme süreçlerine aracılık eden duyuşsal özelliklerinden biri de öz-yeterlik inançlarıdır. Öz yeterlik, bireylerin kendileri için belirledikleri hedeflere ulaşmak için davranışlarını kontrol edebileceklerine olan inançlarıdır” (Bandura, 1997). Uсталık deneyimleri, öğrencilerin geçmiş performansları ve karşılaştıkları yeni olaylarla başa çıkabilmeleri konusunda bireylere en gerçekçi bilgiyi sağlayan faktördür (Bandura, 1994). Öğrencilerin başarıları öz-yeterliklerini artırırken, başarısız oldukları durumlar bu yeterliliği azaltmaktadır. Dolaylı deneyim, sınıf ortamındaki bireyleri izlemesi ile kazanılmaktadır.

Öğrenciler yaptıkları işlemler sonucunda başarı elde ediyor iseler bu onların öğrenme durumlarını geliştirmekte ve motivasyonlarını artırmaktadır (Elliot ve Dweck, 1988). Amaç yönelimli öğrenciler becerilerinin değiştirilebilir nitelikte olduğuna ve yeni bilgilerin ve becerilerin gayretle kazanılabileceğine inanırken; performans yönelimli bireyler becerinin değişmez nitelikte olduğuna ve önemli bir gayretin beceri eksikliğinden kaynaklı olduğuna

inanırlar (Dweck, 1986). Ayrıca, performans yönelimli kişiler yeteneklerini başkalarının görmesi yönünde de isteklidirler (Vandewalle, 2001).

Uzmanlar,” öğrenme-performans ayrımı” kapsamını derinleştirerek, amaç odak yapısı konusunda daha sonraki gelişmeleri önermektedir. Düzenleme odağının “yaklaşma” ve “kaçınma” yönelimleri, hem öğrenim ve performans odaklarının altında yer alan alt tipleri meydana getirmek üzere eklenmesi önerilmiştir (Elliot, 1999; Elliot ve Thrash, 2002).

Yaklaşım yönelimli kişiler bir olayda başarı elde etmenin zorluğu ile güdülenirken, kaçınma yönelimli kişiler başarısızlık karşısında duydukları endişe ile güdülenirler (Higgins, 1996). Hedef yönelimi ve düzenleyici odak bir araya geldiğinde, hedef yöneliminin dört farklı alt türünü oluşturur: “öğrenme yaklaşımı”, “performans yaklaşımı”, “performanstan kaçınma” ve “öğrenmeden kaçınma yönelimleri”. Bununla birlikte, hedef yöneliminden kaçınmayı öğrenmek iyi anlaşılan bir yapı olmayıp, bilim insanları genellikle burada olduğu gibi ilk üçe yoğunlaşmaktadır (DeShon ve Gillespie, 2005).

Bireyler önceki başarılarını sahip oldukları bilgi, yetenek veya becerilerine bağladıklarında etkin ustalık meydana gelir (Bandura, 1997). Bir davranışı gerçekleştirdikten sonra, bireyler kendi ilerlemeleri üzerinde düşünürler ve ulaştıkları performans düzeyine uyum sağlamak için yeterli inançlarını artırırılar; başarılı performans denemelerinden sonra yeterli inançları artmaktadır (Vancouver vd., 2001). Fakat söz konusu etkinin ortaya çıkması, kişilerin bu performansı kendi denetimleri haricindeki tesadüfi etkenlerden değil, tamamen bireysel davranışlarından kaynaklandığını düşünmelerine bağlıdır (Thomas ve Mathieu, 1994; Tolli ve Schmidt, 2008). Bu durum, öz yeterliliğin geçmiş olayların nesnel sonuçlarından ziyade bu olayların psikolojik olarak değerlendirilmesinden daha güçlü bir etkiye sahip olduğunu düşündürmektedir (Bandura, 1997).

“Duygusal uyarılma”, etkin uzmanlığa benzetilebilir ancak ikisi de kişilerin önceki başarı veya başarısızlıklarından kaynaklanır. Oysa etkin öğrenme önceki davranışlarının akılcı sonuçlarına yoğunlaşırken, duygusal uyarılma önceki çalışmalardan doğan duygusal neticelere yoğunlaşır (Kanfer ve Ackerman, 1989).

2.4.5 Eğitimde Öz Yeterlik

Eğitim bağlamında öz yeterlik kavramı son yıllara bakıldığında geniş çapta incelenmiştir. Çoğu bilim insanı öz yeterliliğin öğrenciler arası güdüleme ve öğrenim sürecine etkilerini araştırmıştır. Bu bulgular, öz yeterliğin, öğrencilerin bir görevi yerine getirirken gösterdikleri ilgiyi, azimlerini, belirledikleri hedefleri, karar vermelerini ve bilişsel ve üstbilişsel stratejileri kullanmalarını ve ayrıca kendiliklerini etkileyerek motivasyonu ve bilişi etkilediğini göstermektedir (Bouffard-Bouchard, 1990; Bouffard-Bouchard vd., 1991; Lent vd., 2002; Linnenbrink ve Pintrich, 2003; Pintrich ve Groot, 1990; Schunk, 2003; Zimmerman vd., 1992). Buna göre, öğrencinin öz yeterliği, yetenek (beceriler, bilgi, önceki başarılar vb.) ve bunun sonucunda ortaya çıkan performansla ilgili çeşitli yönler arasında aracılık eder (Bandura, 2006; Schunk ve Pajares, 2002).

Sosyal-bilişsel öğrenme teorisine göre, öz yeterliliği teşvik eden dört ana kaynak vardır: “becerileri uygulama”, “başkalarını gözlemlene”, “sosyal ikna” ve “kişinin fizyolojik ve psikolojik durumu” (Bandura, 1997). “Uygulama becerileri”, öğrencilerin görevi gerçekleştirebileceklerine ilişkin özgün deliller sunduğundan, öz yeterlilik duygusu inşa etmek adına kullanılan en değerli araçtır (Palmer, 2006). Katılımcılar davranışlarının neticelerini yorumlayıp bu yorumu kullanarak vazife ve aktiviteleri icra etme kabiliyetlerine ilişkin çeşitli fikirler edinirler. Genel olarak, bireyin daha sağlam ve dengeli bir öz-yeterlik duygusu oluşturmadan hata yapması durumunda, güçlü bir öz-yeterlik duygusunu başarı kolaylaştırır, başarısızlık ise bunu güçsüzleştirir. Fakat güçlü bir öz yeterliliğin oluşması yalnızca başarının elde edilmesini gerektirmez, aynı şekilde gayret ve sebatla zorlukların aşılması deneyimini de gerektirir (Bandura, 1997). İkinci kaynak, belirtildiği gibi, başkalarının deneyimlerini gözlemlemektir (Bandura, 1997). Öğrenciler, kendi yeteneklerini başkalarını gözlemleyerek, özellikle de karşılaştırma için doğal fırsatlar sağlayan akran gruplarında öğrenirler (Schunk, 1987).

Kısacası, istikrarlı bir öz yeterlilik hissi oluşturmak bakımından öğrencilerin becerilerinde uzmanlaştıkları deneyimlerin en önemli kaynak olduğu görülmüştür. Makalede incelenen ve öğrencileri pratik deneyime maruz bırakmanın ve beceri geliştirmenin önemine vurgu yapan çalışmaların neredeyse tamamında öz yeterlilik gelişiminin en fazla iyileşme gösteren alan olduğu araştırmacılar tarafından açıklanmıştır (Van Dinther vd., 2011).

2.4.6 Bilgi İşlemsel Düşünme ve Öz Yeterlilik

Son yıllarda eğitim ve öğretim araştırmalarında öz-yeterliliğin kullanımı popüler hale gelmektedir (Emurian, 2004). Eğitim araştırmaları (Bandura, 1997; Schunk, 1995) genellikle “öğrenme”, “eğitim” ve “öz yeterlilik” arasında kurulan ilişkiyi ele almaktadır (Akt. Pajares, 2002). Bu çalışmalar, bireylerin bilgi işlemsel düşünme, programlama alanlarındaki öz yeterliklerinin uzun süreli eğitim ve öğretim sayesinde arttığını göstermektedir (Abdunabi vd., 2019; Askar ve Davenport, 2009; Breslin, 2017; Dehghanpour, 2013; Feldhausen vd., 2018; Gist vd., 1989; Izquierdo ve Buelens, 2008; Mina vd., 2015; Özmen ve Altun, 2014; Shell vd., 2014; Shinnar vd., 2014). Öte yandan, öğrenme, eğitim ve öz yeterlik arasındaki ilişki, eğitim ve öğrenme öz yeterliğinin öğrenmeyi geliştirmek için araç olduğu, eğitimin ayrıca öz yeterlik ve öz yeterlik için önemli bir belirleyici olduğudur. Bu, öğrenme gerçekleştikten sonra beklenen daha iyi performansta önemli bir şart olmaktadır (Gist vd., 1989).

2.5 Yapılandırmacılık

Günümüzde bilgi toplumu insanı için sürekli öğrenme ve kendini geliştirme zorunluluğu vardır (Volles, 2014). Bu, eğitimin amacını, belirli bilgi, beceri, çalışma biçimleri, tutumlar veya değerler vermek yerine yeni ürünler, fikirler ve bakış açıları geliştirebilen insanlar yetiştirmek olarak değiştirir (Buckner, 2016). Böyle bir yaklaşım, eğitim programlarının ve okul içi aktivitelerin tasarımı ve sürdürülmesinde yapılandırmacı teoriye büyük önem vermektedir (Palmaru, 2016).

Öğretmenlerin bilgileri çocuklara aktaramayacağı, fakat onların bilgiyi kendi kafalarında aktif biçimde yapılandırmaları gerektiği, eğitimin önemli bir kısıtlıdır. Başka bir deyişle, öğrenciler bilgiyi keşfederek değiştirir, eski bilgilerle yeni bilgileri kıyaslar ve geçerliliğini yitirmiş olan bilgileri yeniden düzenlerler. Buna göre yapılandırmacı öğrenme görüşü, öğreneni bilgi edinme sürecinde aktif bir aktör olarak görmektedir. Yapılandırmacı öğrenme anlayışının kökeni “Dewey (1992)”, “Bruner (1961)”, “Vygotsky (1962)” ve “Piaget’in (1980)” araştırmalarında incelenmiştir. Bednar vd. (1992) ve Glasersfeld (1995), eğitim uzmanları için yapılandırmacı teorinin çeşitli etkilerini ortaya koymuş, öğrenme sonuçlarının bilgi geliştirme sürecine odaklanması gerektiği ve eğitim hedeflerinin spesifik görevlerdeki belirli özellikler tarafından belirlenmesi

gerektiğini belirtmiştir. Aynı biçimde, Glasersfeld (1995) öğrenmeyi bir uyarıcı-tepki olgusu olarak değil, öz-düzenleme ve yansıtma ve soyutlama aracılığıyla gelişimsel yapılar gerektiren bir süreç olarak ifade etmektedir. Dolayısıyla, yapılandırmacılığın farklı şekillerde uygulandığını ve bu görüşlerin birbirleriyle önemli ölçüde kesiştiğini, fakat aynı ölçüde de farklılaştığını belirtmekte fayda vardır.

Driscoll (2005) yapılandırmacı teorisinin bilgilerin yalnızca zihinlerde var olabileceği ve bunların gerçek dünya gerçekliğiyle örtüşmek zorunda olmadığını savunduğunu belirtmektedir. Gerçek dünyadan edindikleri tecrübelerden yola çıkarak kişisel modellerini sürekli olarak oluşturmaya gayret edeceklerdir. Öğrenciler her yeni tecrübeyi kavradıkça zihinsel kalıplarını sürekli olarak yeni bilgilere göre yenileyecek ve dolayısıyla kendi gerçek anlayışlarını inşa edeceklerdir. Jonassen'e (1994) göre, yapılandırmacılık sıklıkla öğrencilere "tekerleği yeniden icat ettiren" bir öğrenme teorisi olarak hatalı değerlendirilmektedir. Oysa yapılandırmacılık, öğrencilerin doğuştan getirdiği bilgiyi, dünyada işlerin nasıl işlediğine dair merakını kullanır ve tetikler.

Yapılandırmacılık aslında bireylerin nasıl öğrendikleri hakkında temel olarak bilimsel çalışma ve gözlem üzerine kurulu bir yaklaşımdır. Bu teori, bireylerin birtakım şeyleri yaşayarak ve bu yaşantıları derinlemesine inceleyerek kendi anlayışlarını ve dünya hakkındaki birikimlerini geliştirdiklerini savunur (Bereiter, 1994).

Fakat başka bir durumla karşılaştığımız zaman, önceki fikirlerimiz ve tecrübelerimiz ile bağdaştırmamız, belki düşüncelerimizi değiştirmemiz veya yeni bilgiyi ilgisiz bularak kabul etmememiz gerekir. Buna rağmen, bizler sahip olduğumuz bilgilerin birer etkin üreticisiyiz. Ancak bu amaçla soru sormalıyız, öğrendiklerimizi araştırmalı ve geliştirmeliyiz. Sınıf içi öğrenmeye ilişkin yapılandırmacı bir bakış açısı, birçok farklı eğitim uygulaması ile ilişkilendirilebilir. Bu en yaygın şekliyle, öğrencileri daha fazla bilgi edinmek için etkin teknikler ("deney yapma", "gerçek hayattaki olayları çözme") geliştirmeye ve daha sonra da ne yaptıklarını ve anlama süreçlerinin nasıl geliştiğini düşünmeye ve tartışmaya yönlendirmeğe çalışmaktadır. Eğitimciler, bu kavramları ele alan ve geliştiren aktivitelerle öğrencilere rehberlik ederek onların önceden var olan kavramlarını anlayabilmelerini amaçlar (Oliver, 2000).

Yapılandırmacılık felsefe, psikoloji, sosyoloji ve pedagojiden oluşan bir bileşen denilebilir. Ancak, öğretmenlerin yapılandırmacılığı kavraması önemli olmakla birlikte, öğrenmeye ilişkin bu bakış açısının eğitim ve öğretmenlerin uzmanlık gelişimi üzerindeki yansımalarını anlamak da son denli büyük önem taşımaktadır (Tam, 2000). Yapılandırmacılığın önemli teması, insan öğrenmesinin yapılandırıldığı ve öğrencilerin önceki öğrenmeye dayalı olarak yeni bilgileri yapılandırma işlemidir. Bu öğrenme görüşü, öğrenmenin bilginin bir kişiden diğerine pasif olarak aktarılması olduğu görüşüyle taban tabana zıttır. İki önemli kavram, basit yapılandırılmış bilgi fikri etrafında döner. Yeni anlayışlar oluşturmak için zaten bildiklerini kullanırlar. Yeni bilgiyi etkilemek için hiçbir değiştirilmeyecek madde veya kanun yoktur. Öğrenciler önceki tecrübelerinden gelen bilgilerle olgulara yaklaşır ve bu ön birikimler, yeni öğrenme tecrübelerinde edinecekleri yeni veya farklı bilgileri şekillendirir (Phillips, 1995).

İkincisinde ise öğrenim pasif değil aktiftir. Kişiler kendi kavramlarıyla yeni öğrenme durumunda karşılaştıkları şeylerin sonucunda yüzleşirler. Kişilerin yüzleştikleri şey halihazırdaki bilgileriyle uyumsuzsa, görüşleri yeni duruma uyum gösterecek şekilde yeniden şekillenebilir. Kişiler bu süre zarfında etkin olmaya devam eder, önceki bilgilerini kullanır, yeni öğrendikleri şeylerle ilgili öğeleri kaydeder, eski ve mevcut bilgiler arasındaki uyumu inceler ve bu incelemeye göre bilgilerini düzenlerler (Phillips, 1995).

Driscoll'e (2005) göre “yapılandırmacı öğrenme kuramı, öğrencilerin mantıksal ve kavramsal gelişimini destekleyen bir düşünce yapısıdır”. Yapılandırmacı öğrenme teorisinin temelinde bulunan düşünce, öğrenci eğitimindeki tecrübelerin veya çevresel bağlantıların üstlendiği kolektif görevdir. Yapılandırmacı öğrenme teorisi, kişilerin tecrübelerine bağlı olarak bilgi oluşturduklarını ve buna anlam kattıklarını öne sürer. Yapılandırmacı öğrenme teorisinde kişinin yeni öğrenmelerini oluştururken başvurduğu kilit noktalardan ikisi uyum ve özümsemedir. Özümseme, kişinin yeni tecrübelerini mevcut olanlara entegre edebilmesidir. Böylece kişi farklı görüş biçimleri kazanabilir, eskiden hatalı olarak anlaşılan konuları tekrar düşünüp neyin gerçekten değerli olabileceğini anlayabilir ve sonuçta bakış açısını değiştirir. Uyum, mevcut kapasite çerçevesinde hayatı ve ortaya çıkan yeni tecrübeleri tekrar çerçevelemektir. Kişiler bulunduğu zamanda uygulanan belli başlı kurallara uyum

sağlar. Bu bağlamda kurallar işlemediğinde, beklentileri sonuçlarla uyumlu hale getirmeli ve yeniden çerçevelemelidir.

Sokrates ilk büyük yapılandırmacı olarak kabul edilir (Sönmez, 2009). Sokrates'e göre; öğretmen ve öğrenenler birbirleriyle konuşarak ve birbirlerine sorular sorarak ruhlarında saklı olan bilgileri ortaya çıkarır, yorumlar ve yaratırlar. Sokrates “Pisagor Teoremini” eğitilmemiş bir köleye yönelttiği sorularla öğretmek, kölenin kafasındaki saklı kalmış olan bilgileri ortaya çıkmasını sağlamıştır (Cevizci, 2012). Glasersfeld, yapılandırmacılık kavramını “kökleri felsefe, psikoloji ve sibernetikte bulunan bir bilgi teorisi” şeklinde açıklamaktadır. Murphy (1997) ayrıca 18. Yüzyıl felsefecisi Giambattista Vico'yu bilinen ilk yapılandırmacılar olarak nitelendirir. Vico, yapılandırmacılığı “Bir şeyi bilen onu açıklayabilir” şeklinde belirtmektedir. Bilmek, öğrencinin o konuyu hangi yolla öğrendiğinin izahıdır (Özden, 2003). Bilme eylemi kişiden tamamen ayrı olarak meydana gelmeyip, açıklanınca ortaya çıkmaktadır. 20. yüzyıl düşünürlerinden Kuhn ve Rorty, kişinin etkin bir şekilde öğrenmeye dahil olmasını desteklemektedir. Bu düşünürler bilginin kesinliğinden daha çok uygulanabilir olmasına dikkat çekmişler ve kişinin bilgiyi spesifik olarak oluşturmasının önem taşıdığını ifade etmişlerdir (Erdem, 2002).

Örgün eğitim tasarımıdaki ilk uygulamaların büyük bir kısmı objektivist teoriden hareketle gerçekleştirilmiş olmasına rağmen, çağdaş akademik çevreler yapılandırmacı öğrenmenin ihtiyaçlarına uygun öğrenim çevrelerinin daha verimli sonuçlar doğurabileceğini kabul etmeye başlamıştır. Yapılandırmacı öğrenmenin bilinen yararları, “problem çözme” ve “eleştirel düşünme” becerileri de dahil olmak üzere kapsamlı becerilerin kazandırılması açısından çok önemli hale gelmektedir (Tam, 2000).

Yapılandırmacı teoremin öğrenimi tarif etmede kullanılan en iyi yöntem olarak benimsenmesi halinde, öğrencilerin öğrenmelerini desteklemek için öğrencileri çalıştıkları materyallere bizzat dahil eden öğrenme çevrelerinin oluşturulması gereklidir. Zira öğrenci dünyayı yalnızca bizzat tecrübe ederek anlamlandırabilmektedir. Bunun neticesinde yapılandırmacı öğrenmenin mutlaka doğru bir yapılandırmacı öğrenme çevresinde yer alması gerektiği görüşü doğmaktadır. Yapılandırmacı öğrenmenin kilit unsurlarından biri de bunun etkin bir süreç halinde gerçekleşmesi zorunluluğudur (Tam, 2000). Dolayısıyla, her yapılandırmacı öğrenme çevresi etkin öğrenme olanakları tanımalıdır.

2.5.1 Yapılandırmacı Öğrenme

“Yapılandırmacı öğrenme yönteminde” kişiler geçmiş bilgilerini temel olarak yeni bilgiler oluştururlar; bu nedenle bilgi oluşturma kişiye göre değişir ve eğitimciler yalnızca birer rehber konumundadır (Kanlı, 2010; Özmen, 2004; Ün Açıkgöz, 2014, s.60-61). “Bilişsel öğrenme bakımından yapılandırmacı öğrenme yaklaşımıyla örtüşen aktif öğrenme modelinde bireyler kendi öğrenme süreçlerinin sorumluluğunu almakta, bu süreçle ilgili kararlar almakta ve öz düzenleme olanağına sahiptir” (Ün Açıkgöz, 2014, s. 17).

21. yüzyıl becerilerine sahip kişilerin özellikleri, yapılandırmacı öğrenme yaklaşımı ve aktif öğrenme modeli ile kazanılan nitelikler ile karşılaştırıldığında, 21. yüzyıl becerilerine sahip bireylerin yetiştirilmesinde yapılandırmacı öğrenme yaklaşımı ve aktif öğrenme modelinin merkezi bir yere sahip olduğu görülmektedir. Birçok araştırma, yapılandırmacı öğrenme yaklaşımının öğrencilerde akademik başarı açısından "büyük" katkılar sağladığını göstermiştir. Ayaz ve Şekerci (2015), yapılandırmacı öğrenme yaklaşımının çocukların ders başarıları açısından yarattığı etkinin "büyük" düzeyde olduğundan bahsetmektedir.

Semerci ve Batdı (2015) yapılandırmacı öğrenme yaklaşımının çocukların akademik başarısına "büyük" bir katkısı ve faydası bulunduğunu saptamıştır. Yapılandırmacı öğrenme yaklaşımı esas hedefleri açısından benzer felsefeyi esas almakla birlikte, birbirinden farklı teorisyenler öğrenmede birbirinden farklı süreçlerin etkin olacağını ileri sürmüşlerdir. Piaget'nin “bilişsel gelişim kuramı”, öğrenmeyi ortamla etkileşimde bulunarak ve bilgileri şemalar vasıtasıyla ihtiyaç ve ilgisi doğrultusunda işleme tabi tutarak meydana geldiğini öne sürer. Bruner'in “keşif temelli öğrenme” teorisinde ise öğrenmenin içsel bir süreç olduğu, sezgi kullanımını ve ortamla etkileşimde bulunmayı kapsadığı, fakat ortamdan tamamen kopuk yanıtlar alındığı ifade edilmektedir. Ausubel, “anlamli öğrenme” teorisinde bilginin bir anlam oluşturma işlemi aracılığıyla edinildiğini ileri sürer. Dewey'in “deneyimsel öğrenme” teorisi, öğrenmenin ortamla olan etkileşimin sonucunda gerçekleşen tecrübeler ve keşiflerden elde edildiğini ifade eder. Vygotsky “sosyal gelişim teorisinde” bu dönemde oyunun çok değerli olduğunu belirterek öğrenmenin ortamla kurulan etkileşimle oluştuğunu ileri sürer (Ün Açıkgöz, 2014,

s. 67-76). Yapılandırmacı öğrenme yaklaşımı şu öğretim yöntem ve teknikleri kullanılarak kullanılabilir: probleme dayalı öğrenme, proje tabanlı öğrenme, bilgisayar destekli öğrenme, iş birlikli öğrenme, 5E ve 7E öğrenme döngüleri vd.

2.5.2 Yapılandırmacılık Temel Fikirler

Yapılandırmacı kuramını inceleyen kişilerin birçoğunun onay verdiği görüşler ortaya çıkmıştır. Bunlar, yapılandırmacı iddiaların ılımlı bir versiyonu olarak görülebilirler. Kısmen Taber'in (2006) analizine dayanan bu temel fikirlerden bazıları aşağıda verilmiştir.

- Kişiler bilgiyi kendileri oluşturur dış etkenler bu süreci etkilemez.
- Bazı fikirlerin çoğunluğu geçersizliğini yitirerek unutulur diğer fikirler gelişerek kalıcı hale gelmektedir.
- Öğrencilerin bulunduğu ortam ve yaşantı ile ilgili birçok fikirleri vardır. Bunların bazıları ortak ve benzer olmakla beraber kültürel ve sosyal alanda kabul görmektedir. Bu fikirler aralarında paylaşabilmektedirler ayrıca bu paylaşımların çoğu zaman birçok olguyu anlamak için de kullanılan bir araçtır.
- Kişilerin üretmiş olduğu bazı fikirler bilimsel kaynaklar ile çelişebilmektedir ve bunları değiştirmek için kişiler için zor bir hale dönüşmektedir. Bu fikirler kalıcı olmakla beraber değiştirilmesi imkânsız olmaktadır.
- Bilgi kavramsal yapılar halinde beyinde oluşmakta ve bunları ayrıntılı olarak bilmekteyiz.
- Bilgi kişilerin ürettiği kavramlar olsa bile kişi çevresi kültürel olarak bulunduğu ortamı sosyal çevresi ile etkileşim yaparak yeni kavramlar oluşturabilmektedir.

2.5.3 Yapılandırmacı Öğrenme ve Öğretmenin Özellikleri

Bu bölümdeki özelliklerin sunumu, yapılandırmacılığın literatürde ifade edildiği çeşitli yolları tanıdığı ve temsil etmeye çalıştığı için bu analojiye sadık kalmayı amaçlamaktadır. Bütün yaklaşımlar, öğrenme için kavramların işlevsel hale getirilme biçimine atıfta bulunur. Aşağıda yaklaşımları sıralanan kuramcılar ve bilim insanları “yapılandırmacı kuram” ve “uygulama yöntemleri” hakkında bilgi vermektedir. Bu yaklaşımlar, “tasarım”, “öğretim veya öğrenmeye yönelik yapılandırmacı” bir yaklaşımda yönlendirici bir bakış açısının temelini sunmaktadır.

Jonassen (1991) pek çok eğitimci ve bilişsel psikoloğun yapılandırmacılığı öğrenme çevrelerinin hazırlanmasında kullandığını söylemektedir. Kendisi bu uygulamaların içinden bir takım tasarım prensipleri çıkarmıştır.

- Eğitimin uygun olacağı içeriği kullanacak şekilde tasarlanmış gerçeklik odaklı çevreler oluşturun;
- Gerçeği yansıtan sorunların çözülmesine ilişkin yaklaşımlara ağırlık verilmesi;
- Eğitici, sorunları çözmeye uygulanan yöntemlerin koçu ve çözümleyicisidir;
- İçerik hakkında birden fazla sunum veya perspektif sunarak içeriğin birbiriyle olan kavramsal bağlantısını vurgulayın;
- Eğitim hedefleri ve gayeleri tartışılmalı ve zorla kabul ettirilmemelidir;
- Değerlendirme bir öz analiz aracı olarak kullanılmalıdır;
- Öğrencilerin farklı bakış açılarından bakarak olayları değerlendirmelerine destek olacak ortamlar ve yöntemler sunun;
- Öğrenmenin öğrenci aracılığıyla kendi kontrolünde gerçekleşmesi sağlanmalıdır.

Jonassen (1994), "yapılandırmacılığın öğretim tasarımı üzerindeki etkisi" olarak ileri sürdüğü görüşünü kısaca açıklamaktadır. Aşağıdaki ilkeler, bilgi oluşturma nasıl kolaylaştırılacağını göstermektedir:

- Gerçeğin çoklu temsillerini sunun.
- Gerçeğe uygun hayatın getirdiği karmaşıklığı temsil edin.
- Bir şeyleri tekrardan yapmaya değil, bilgi inşasına odaklanın.
- Gerçek olan görevleri sunun (bilgiyi soyutlamaktan ziyade bağlama oturtmak).
- Önceden planlanarak oluşturulmuş öğretiler yerine gerçek yaşamda olabilecek örnekler sunulmalıdır.
- Yansıtıcı uygulamayı teşvik edin.
- İçeriksel ve duruma özgü bilgi oluşturmaya sağlayın.
- Müzakere yoluyla bilginin işbirlikçi inşasını destekleyin.

Wilson ve Cole (1991), yapılandırmacı öğeleri "içeren" "bilişsel öğretim modellerinin" açıklamasını yapmaktadır. Yapılan bu açıklamalardan hareketle,

“yapılandırmacı tasarım, eğitim ve öğretim sürecinin odağında bulunan kavramları şöyle listeleyebiliriz:

- Zengin, sahici bir soru çözme alanına öğrenimi entegre edin.
- Öğrenme süreci boyunca otantik ve bilimsel içerikler sunun.
- Öğrencilerin denetimini mümkün kılın.
- Hataları, öğrencilerin kavrama düzeyleri konusunda dönüt sağlayan

bir yöntem haline getirin.

Paul Ernest (1995), birçok düşünce ekolünü ele aldığı çalışmasında hem köktenci hem de toplumsal yaklaşımlardan yola çıkarak yapılandırmacılığın aşağıda belirtilen sonuçlarını ortaya koymaktadır:

- Öğrencinin geçmiş dönem çalışmalarına yönelik hassasiyet ve özen.
- Öğrenci yanlışlarını düzeltmeye çalışan tanılayıcı eğitim.
- Öğrencilerin “üstbiliş” ve “yaratıcı” “öz-düzenlemeye” olan ilgisi.
- Matematiksel konseptin birden fazla gösteriminin birlikte kullanımı.
- Öğrenci hedeflerinin taşıdığı önem ve öğrenci ile öğretmen amaçları

konusundaki görüş ayrılıkları konusunda duyarlılık.

- Halk matematiği veya mahalle matematiği ile okul matematiği gibi toplumsal ortamların taşıdığı öneme ilişkin duyarlılık.

Honebein (1996) yapılandırmacı bir ortam tasarımı için gereken yedi amacı belirlemiştir:

- Bilgi yaratma süreci hakkında tecrübe kazandırın.
- Birden fazla perspektif için takdir ve tecrübe sunun.
- Öğrenmeyle ilgili gerçeklik içeren ve ilişkili ortamlar yaratın.
- Öğrenme aşamasında aidiyet ve fikir beyan etmeyi cesaretlendirin.
- Öğrenmeyi toplumsal tecrübeye entegre edin.
- Birden fazla temsilin katılımını destekleyin.
- Bilgi yaratma aşamasında öz farkındalığı destekleyin.

Sosyal yapılandırmacılar açısından oldukça mühim olan bir konu da öğrenciyi önceden bilinenlerden daha sonra öğrenileceklerle doğru rehberlik etme işlemi olarak tanımlanan kavramdır. Vygotsky’ye (1978) göre öğrencilerin problem çözme becerileri üç kategoriye ayrılır:

- Öğrencinin gerçekleştiremeyeceği beceriler
- Öğrencinin gerçekleştirebileceği beceriler

- Öğrencinin yardımla gerçekleştirebileceği beceriler

2.6 Çalışmada Kullanılacak Öğretim Yaklaşımları

Bu tezde kullanılacak öğretim yaklaşımları İnşacılık, Doğrudan öğretim ve GUTÜD tabanlıdır. Bu öğretim yaklaşımları aşağıda sunulmuştur.

2.6.1 İnşacılık

1980'lerin başında (yani seri üretim kişisel bilgisayarların ortaya çıkmasından önceki dönemde) matematik eğitimcisi Seymour Papert, bir gün tüm çocukların küçük ellerine sığacak ve kendi eğitimlerini somutlaştırmalarını ve bireyselleştirmelerini sağlayacak kişisel bir taşınabilir bilgisayara sahip olacağını öngörmüştür (Papert, 1980, 1985). Papert'in vizyonunun önemli bir kısmı onun yaşadığı dönemde gerçeğe dönüştürmüştü: 2010'lu yıllarda, Batılı zengin ülkelerde Common Sense Media (2016) ve birçok Afrika (Mahamud vd., 2015) ve Asya ülkesinde (Zaman vd., 2011) büyüyen çocukların çoğu hafif, taşınabilir ve iki yaşındaki çocuklar tarafından bile kolayca kullanabildiği tablet bilgisayarlara ve akıllı telefonlara erişebilmektedir. Bilgisayarları çocukların eğitimini kişiselleştirmek için kullanma fikri, çocukların aldığı içeriği bireyselleştirmek için kişisel profilleri kullanan veri yönetimi teknolojilerinin ortaya çıkmasıyla daha gerçekçi hale gelmiştir (Selwyn, 2016). Ancak bu cihazlar henüz Papert'in öngördüğü türden çocuk güdümlü keşif ve deneyler sunmamaktadır.

1980'lerde Papert yeni bir fikir öne sürmüştür: Yapılandırmacı psikoloji teorilerinden yola çıkarak öğrenmeyi bilginin aktarılmasından ziyade yeniden yapılandırılması olarak görüyoruz. Zengin bir eğitim deneyiminden yola çıkarak, öğrenmenin özellikle öğrencinin anlamlı bir ürün (örneğin, bir sanat eseri, işleyen bir makine, bir araştırma raporu veya bir bilgisayar programı) inşa ederken deneyimlediği bir faaliyetin içine yerleştirildiğinde etkili olduğu görüşünü benimsiyoruz. Papert ve Group (1986) “İnşacılık aynı zamanda öğrenme biliminin temelinde yatan derin yapıların merkezi olduğuna dair bilişselci ilkenin de ötesine geçer (yine bu ilkeyi de içerir): bilişsel derin yapılara bir dizi derin boyut ekler: en az bilişsel faktörler kadar ağırlık verdiğimiz duyuşsal, estetik ve sosyo-kültürel boyutlar” (Papert ve Group 1986, s. 10).

Papert'in teorisi, çocukların kendi içeriklerini ve eserlerini inşa ederek araçlarla öğrenmelerine ilişkin kapsamlı bir inceleme sunmaktadır. Teori, teknoloji odaklı çalışmalarda ve pedagojilerde yaygın olarak kullanılmaktadır (Kinshuk vd., 2013). Konu okuma ve anlatılar olduğunda, 'en bütünleştirici ve en

açıklayıcı güce sahiptir' (Owen, 2006, s.12). Mevcut argüman, Papert'in inşacı teorisi (Papert, 1980, 1993) ve daha sonraki türevi olan sosyal inşacı teori (Nightingale ve Cromby, 1999) tarafından ele alınan dört temel öğrenme ilkesi etrafında inşa edilmiştir: 1, bilişsel meydan okuma; 2, çocukların eylemliliği; 3, keşfederek öğrenme ve 4, çocuk liderliğinde öğrenme. Dört ilkenin de ortak bir özelliği vardır: çocuğun yeni araç ve anlayışları aktif olarak inşa etmesi. Bu fikirler, çocuklar için çeşitli robotik araçların ve programlama sistemlerinin (örneğin LOGO, Scratch) ve çocukların analitik ve problem çözme becerilerini destekleyen etkileşimli uyarım ortamlarının (örneğin Turtle, Talos) geliştirilmesinde merkezi bir rol oynamıştır.

Diğer birçok eğitim teorisinde olduğu gibi, inşacılığın da sağlam boylamsal verilere değil, felsefi fikirlere dayandığı unutulmamalıdır. Bununla birlikte, inşacılık en büyük ve en yenilikçi eğitim teknolojisi projelerinden bazılarında (örneğin, Çocuk Başına Bir Dizüstü Bilgisayar) ilham vermiş ve Papert'in mirası ilk çocuk programlama dili (Logo) ile bilişimsel okuryazarlık ve teknoloji destekli proje tabanlı öğrenme alanlarının temel taşı haline gelmiştir (Hoyles ve Noss, 2017).

Konfüçyüs (1986), Vygotsky (1978) ve diğerleri gibi Papert da yaptığımız işlemleri konuşarak ve düşünerek gerçekleştirsek gerçek öğrenmenin gerçekleştiğini savunmaktadır. Papert bu fikrini “Logo Felsefesi ve Uygulamasına” yazdığı makalede söz etmiştir (Papert, 1999).

Valente (1994), Papert'ın (1980) inşacılık perspektifinin, elle tutulur bir ürün ortaya çıkaran ve onu üretenlerin kişisel ilgisini çeken bir eylemin pratik uygulamasına dayanan bir bilgi inşası sunmasıyla öne çıktığına işaret etmektedir. Papert'a göre bu ortam içinde zihinsel yapı, soyutlamanın gelişimini destekleyen somut yapılarla desteklenmelidir: somut ve soyut arasında diyalektik bir hareket vardır. Bu bakış açısı altında, öğrencilerin süreci kontrol etmeleri öğrenmelerine yardımcı olur ve öğretmenin rolü, bilişsel gelişim düzeylerine saygı göstererek onları teşvik etmek için yeni zorluklar önermektir. Vurgu, öğrencilerin ortaya koydukları üründen ziyade öğrenme süreçleri üzerindedir.

Papert (1980), çocukların bilgisayarla temas ettiği çoğu çağdaş eğitim ortamında bilgisayar, çocukları sınamak, uygun zorluk derecesinde alıştırma yapdırmak, geri bildirim sağlamak ve bilgi vermek için kullanılır. Bilgisayar kullanıcısı olduğu bireyi yönlendirmektedir. LOGO ortamında ise bu ilişki tersine

döner: Çocuk, okul öncesi yaşlarda bile kontrolü elinde tutar ve çocuklar bilgisayarı programlar. Çocuklar bilgisayarlara nasıl düşüneceklerini öğretirken, aynı zamanda kendilerinin de nasıl düşündüklerini keşfediyorlar. Entelektüel yapıların öğrenen tarafından inşa edildiğini ve öğretmen tarafından öğretilmediğini söylemek, bunların yoktan var edildiği anlamına gelmez. Tam tersine: Diğer inşaçılar gibi çocuklar da çevrelerinde buldukları malzemeleri, en önemlisi de çevrelerindeki kültürün önerdiği model ve metaforları kendi kullanımlarına uygun hale getirirler. Papert (1999) Gary Stager'ın 2007 tarihli "An Investigation of Constructionism in the Maine Youth Çenter" başlıklı doktora tezi kapsamında yapılan araştırma projesinde inşaçı öğrenmenin ana hatlarını özetlemiştir:

İlk büyük fikir yaparak kazanmaktır. Gerçekten enteresan hissettiğimiz şeyleri yaparken öğrenme daha kolay gerçekleşir. Öğrendiğimiz bilgileri çok arzu duyduğumuz şeyleri gerçekleştirmek için kullanıyorsak daha kolay öğrenebiliriz.

İkinci önemli düşünce de bir araç olarak teknolojidir. Bazı işleri yapabilmek adına teknolojiyi kullanırsanız, daha enteresan işler çıkarabilirsiniz. Bunları gerçekleştirerek daha fazla şeyler öğrenebilirsiniz. Aynı husus bilhassa dijital teknoloji konusunda söz konusudur. Bilgisayar ortamında kullanılan LEGO programı da birçok şeyin öğrenilmesini sağlamaktadır.

Üçüncü düşünce ise eğlenirken zorlandığımız işlerdir. Yaptığımız işten keyif alırsak en iyi şekilde öğrenir ve en iyi şekilde çalışırız. Ama eğlenmek "kolay" demek değildir. Asıl eğlenceli şeyler zor eğlencelerdir. Sportif yıldızlarımız spor dallarında başarılı olabilmek amacıyla sıkı biçimde çalışırlar. En başarılı marangozlar marangozluktan işini yaparken o işten hoşlanırlar. Başarılı iş adamları başarılı iş görüşmeleri için çok çalışmayı severler.

Dördüncü önemli düşünce de kazanmayı öğrenmektir. Pek çoğu "öğrenmenin tek yolunun öğretilmek olduğu" düşüncesine sahiptir. Ancak böyle düşünmeleri onların hayatta ve okulda kaybetmelerine sebep olmaktadır. Hiç kimse öğrenmeniz gerektiği her şeyi size anlatamaz. Siz kendi öğrenmenizin mesuliyetini almalısınız.

Beşinci büyük düşünce işleriniz için uygun ve yeterli zamanı ayırabilmektir. Öğrenciler okulda genellikle yapılması gerekenlerin belirli zaman aralıkları ile söylendiği bir ortamda bulunurlar. Biri onlara ne yapmaları gerektiğini söylemezse sıkılırlar. Hayat böyle değildir. Zamanı kendi

kontrolünüzde tutmanız ve işleri o süreçte yönetmeniz gerekmektedir. Bu kısım öğrencilerin yapmakta en güçlük çektiği bölümdür.

Altıncı büyük düşünce ise fikirlerin en önemlisidir: Yanlış yapmaksızın doğruyu bulamazsınız. Hiçbir iş her defasında başarılı olmaz. Bir işi düzgün yapabilmenin yegâne yöntemi, işler yolunda gitmeyince ne olup bittiğine iyi bakabilmektir. Başarılı olabilmek istiyorsanız, bu süreçte hatalar yapma konusunda serbest olmalısınız.

Yedinci büyük düşünce, tıpkı çocuklarımıza yaptığımız gibi bunu kendimiz için de yapmamızdır. Her daim yeni şeyler öğrenmekteyiz. Başka projelerden pek çok tecrübemiz olmakla birlikte bunların her birisi birbirinden oldukça farklılık arz ediyor. Bu işin nasıl sonuçlanacağıyla ilgili hiçbir önyargımız yok. Biz işimizden keyif alıyoruz ama bunun zor olacağını da tahmin ediyoruz. İş düzgün yapabilmek adına gereken süreyi ayırmamız gerektiğini düşünüyoruz. Karşımıza çıkan her zorluk, bir şeyler öğrenme fırsatı sunar. Öğrencilere verebileceğimiz kuşkusuz ki en güzel mesaj, bizim de bir şeyler öğrenmek için çaba sarf ettiğimizi onlara göstermektir.

Sekizinci büyük düşünce, dijital bilginin okuma ve yazma kadar önemli olduğu dijital bir çağa doğru ilerliyoruz. Bu yüzden bilgisayarları öğrenmek, öğrencilerimizin geleceği için çok önemlidir, ancak en önemli amaç, öğrenme aracı olarak bilgisayar kullanmayı öğretmektir (Papert, 1999).

2.6.2 Doğrudan Öğretim Yöntemi

Öğrencilerin bilgi ve becerilerini geliştirmek için çeşitli öğretim yöntemleri geliştirilmiştir. Etkili öğretim, sistematik öğretim, stratejik öğretim ve doğrudan öğretim gibi yöntemler bulunmaktadır. Her biri kendi alanında deneysel araştırmalarla test edilir ve elde edilen başarılar karşılaştırılır. Bu araştırmaların sonuçları, eğitimin etkisini artırmak ve öğrenme güçlüğü olan öğrencileri öğrenmeye teşvik etmek için etkili uygulamaları içeren doğrudan öğretim modelinin kullanılabileceğini işaret etmektedir (Güneş, 2022).

Doğrudan öğretim yöntemi literatüre bakıldığında sistematik öğretim, etkin öğretim ve açık anlatımla öğretim şeklinde ifade edilmektedir (Güzel, 1998; Schug vd., 2001; Vuran ve Çelik, 2013).

Doğrudan öğretim modelleri, doğrudan, açıklayıcı, sistematik, artımlı, iyi yapılandırılmış ve öğretmen liderliğindeki öğretimi içerir. Bu modelin temel ilkesi şudur: Eğer öğrenciler öğrenmiyorsa, öğretmen konuyu öğrencilere

verimli bir şekilde öğretemiyor demektir. Öğretimin odak noktası önce bilgi aktarımı, sonra da öğrencilerle birlikte beceri ve stratejilerin öğrenilmesidir. Bundan dolayı öğretim yöntemi ve bilgi aktarımı noktasında bir bağlantı oluşmaktadır (Güneş, 2022). Modelleme yapılarak etkili öğrenmeyi ve sunulan içeriğin daha iyi öğrenilmesi sağlanmaktadır. Eğitim süreci, basit olandan karmaşığa, belirli bir periyotta, her bir noktaya dikkat edilerek ve etraflıca olacak şekilde yapılandırılır. Öğrenilenlerin zihinde yapılandırılması için tekrar vurgulanır. Öğrencinin süreçteki çabasına ve planlama dahilinde işlerin yapılmasına önem verilir (Güneş, 2022).

Doğrudan öğretim modelinde ilk odak noktası hedef kavram, beceri ya da stratejinin açıklanmasıdır. İkinci önemli nokta, uygulama sürecinin şeffaflığıdır. Bunun için aşağıdaki ilkeler çok önemlidir.

- Her şeyin anlaşılabilir olduğu bir ortam sunun: Zihinsel süreçlerin takibi zor olduğundan anlaşılması güç olan olayları, becerileri ve planları öğretim süreci boyunca öğrencilere her bir detayı anlatmalı ve öğrenciler sürece her zaman dahil olmalıdırlar.
- Öğrencilere lider olma: Uygulanacak konu ve problem üzerine öğrenciler ile tartışma başlatılmalı, belirli ipuçları gerekli yerlerde verilmeli, bu uygulamanın onlara ne gibi yararlar getireceğinin söylendiği süreçler oluşturmalıdır. İlk önce büyük gruplar ile bu işlemler yapılır. Daha sonrasında kişi sayısının az olduğu gruplar oluşturulur. En son bireysel olarak süreç ele alınır. Her bir adımda değerlendirme ve birbirleri arasındaki ilişki araştırılır.
- Bireysel uygulama: Öğrencilerden yapmış oldukları işlemleri farklı bir olay ya da probleme aktarmaları istenir ve yaptığı tüm süreç gözlemlenerek geri bildirim verilir.
- Uygulamayı geliştirme: Öğrenilen hüneri ve taktiği diğer durumlara aktarması ve farklı etkinlikler kullanarak farklı durum veya ortamlarda uygulaması istenir (Giasson, 2011; akt. Güneş, 2022).

Doğrudan öğretim modelinin özünü üç temel aşama oluşturur: hazırlık, öğrencilerle etkileşim ve öğrenilenlerin pekiştirilmesi. Bu aşamalar daha sonra "modelleme, rehberli uygulama, bağımsız uygulama" olarak adlandırılmış ve alanda kabul görmüştür (Gauthier vd., 2013).

Modelleme: Model, resim, şema, maket, ürün gibi somut bir unsur olabileceği gibi benzetme, zihinsel şema, düşünme biçimi gibi soyut bir unsur da olabilir. Modelleme ise modeli oluşturan aşamaları veya süreçleri açıklamak, somutlaştırmak, yeniden yapmak, göstermek veya sergilemektir. Diğer bir ifade ile yapılan işlemlerin daha net anlaşılması için kullanılır ve pratik bilgiler içerir (Güneş, 2022).

Rehberli uygulama: Bu aşamada öğretilen strateji hem öğretmen hem de öğrenci tarafından uygulanır. Modelleme aşamasında öğretmenler çeşitli sorular sorarak fikirlerini sözlü olarak ifade ederler. Öğrenciler ise kendilerine sorulan soruları dinler, tekrar eder, anlar ve cevaplar. Öğretmenler, öğrencilerin modelleme safhasında yaptıkları işlemleri birebir yapmalarını ister. Öğrenci, model olan öğretmeni gözlemledikten sonra onu taklit etmeye çalışır (Gauthier vd., 2009).

Kişisel uygulama: Bu kısımda öğrencilerin öğrenmiş oldukları süreçleri bağımsız bir şekilde uygulamaları için onlara yol gösterilir (Gauthier vd., 2013).

Doğrudan öğretim modeli üzerine yapılan çalışmalarda, beceriler ve stratejiler sistematik olarak dikkatli bir şekilde uygulandığında olumlu sonuçların çıktığı görülmektedir. Doğrudan öğretim uygulamalarında iyi hazırlanmış uygulamalar öğrenme başarısını artırmada oldukça etkili olduğu görülmektedir. Bu anlayıştan hareketle ve bir dizi deneysel çalışma sonucunda doğrudan öğretim modeline üç aşama daha eklenmiş ve altı aşamaya çıkarılmıştır (Conseil National Dévaluation Du Système Scolaire [CNESCO], 2016).

Gözden Geçirme: Daha öncesinde öğrenilmiş bilgilere ve kavramlara öğrencinin odaklanmasını sağlama sürecidir. Modelleme: Öğretilecek kavram, beceri ve stratejilerin sunulması, açıklanması, genelleştirilmesi, örneklendirilmesi, yararlarının tartışılması, özelliklerinin açıklanması ve nasıl uygulanacağı konusunda bilgilendirilmesi çalışmasıdır. Danışmanın olduğu uygulama: Öğrencilere yeni beceri ve stratejilerin uygulanması konusunda talimat verin ve onları denetleyin. Öğrencilere sorular sorun, süreci açıklamaları için onları teşvik edin ve tekniklerin ve becerilerin kullanımına ilişkin geri bildirim sağlayın. Düzeltme ve geri bildirim: Öğrenci hatalarını düzeltme, teknikleri kendi başınıza deneme, sesli düşünme, bunları kullanmanın farklı yollarını geliştirme vb.

etkinlikler. Bireysel uygulama: Öğrencilerin derste öğrendiklerini uygulamaları, ders ilerledikçe öğrencilere daha fazla sorumluluk verilmesi ve bireysel çalışmaya yönlendirilmeleri, Uygulamayı geliştirme: Öğrencilerin öğrendikleri beceri ve stratejileri okul dışı görevlerde ve bağımsız öğrenme durumlarında denemeleri ve geliştirmeleri (CNESCO, 2016).

Barak Rosenshine, doğrudan öğretimin sınıfta etkili bir şekilde uygulanması için bazı önerilerde bulunmaktadır (Rosenshine, 2010):

- 1- Öğrencilerin ön bilgilerini harekete geçirin veya hatırlatın,
- 2- Yeni konuları küçük adımlarla sunun, öğrencilere pratik yaptırın,
- 3- Bir kerede öğretilecek konuları sınırlandırın,
- 4- Net ve detaylı yönergeler ve açıklamalar verin,
- 5- Bol bol soru sorun ve öğrencilerin anladığını test edin,
- 6- Etkin alıştırmalar için olanaklar oluşturun,
- 7- Alıştırma aşamasının başlarında öğrencilere rehberlik edin,
- 8- Öğrenmenin her aşamasında yüksek sesle düşünün ve modeller gösterin,
- 9- Önceden yapılmış çözümlerden örnekler sunun,
- 10- Öğrencilerden öğrendiklerini sözlü olarak ifade etmelerini isteyin,
- 11- Bütün öğrencilerin cevaplarını denetleyin,
- 12- Sistemantik yorumlar ve düzenlemeler yapın,
- 13- Açıklamalar için geniş zaman ayırın,
- 14- Bol ve değişik örnekler sunun,
- 15- Gerekirse kavramları yeniden öğretin,
- 16- Öğrencileri serbest çalışmalara yeterince hazırlayın,
- 17- Bağımsız uygulama aşamasında öğrencileri destekleyin (Rosenshine, 2010).

Doğrudan öğretimin birincil amacı, ilkokul sınıflarında risk altındaki öğrencilerin öğrenmelerini hızlandırmak ve daha avantajlı akranlarıyla rekabet etmelerini sağlamaktır. Öğretim, önemli arka plan bilgilerini sistemantik olarak geliştirerek ve onu yeni bilgilerle ilişkilendirerek öğrenmenin yalnızca miktarını değil kalitesini de artırmayı amaçlar. Öğrenciler programda başarılı oldukça, öğrencilerin sosyal davranışlarını ve duygusal refahlarını geliştirir (Education Commission of the States, 1999). Doğrudan öğretim bileşenleri içerik ustalığına odaklanmayı içerir. Bu ustalık temel derslerde bir dizi beceri ve bilgi üzerine

kuruludur ve her yeni seviye bir öncekinin üzerine inşa edilir. Öğrencilerin ustalığa ulaştığından emin olmak, ekstra yardıma ihtiyaç duyabilecek öğrencileri tespit etmek ve yeniden gruplandırılması gereken öğrencileri belirlemek için kullanılan sık değerlendirmeler yapılmalıdır (Education Commission of the States, 1999).

Öğrencilerle artan öğretmen etkileşimini içeren küçük grup oturumları, tüm öğrencilerin her derste ustalaşmasına yardımcı olmak için pekiştirme ve düzeltme prosedürlerini içerir (Education Commission of the States, 1999). Ustalıkla okuma, uygun okuma becerilerinin aşamalı gelişimi için kritik öneme sahiptir. Brophy ve Good'a (1984) göre, başarı kazanımları, öğretmenler aktif öğretim ve öğrenmelerini doğrudan izleme yoluyla içeriği öğrencilere yönlendirdiğinde gerçekleşir. Başarı, öğretmenler "akademik öğretimi rollerinin önemli bir parçası olarak vurguladıklarında, öğrencilerin müfredatta ustalaşmalarını beklediklerinde ve mevcut zamanlarının çoğunu müfredatla ilgili etkinliklere ayırdığında" gerçekleşir (Brophy ve Good, 1984).

Usta öğrenci, öğrencilerin becerilerini geliştirmelerine ve üzerine inşa etmelerine yardımcı olur. Program, öğretmenin sağladığı örneklerin oranı ve türü, öğretmenin belirli kavramları ve becerileri öğretmek için kullandığı ifadeler, işlenen inceleme materyalinin sıklığı ve türü ve öğrencilerin materyali öğrenme girişimlerine öğretmenin tepkisi gibi değişkenleri kontrol eder (Stockard ve Engelmann, 2008). Program, birbiri üzerine inşa edilen izole beceri ve kavramları öğretmek için küçük adımlar kullanır ve bir dersin içeriğinin yalnızca yaklaşık %10'u yenidir (Stockard ve Engelmann, 2008). Bu tekrar, öğrencilerin kendilerini başarılı hissetmelerini ve zaman içinde materyale ulaşmalarını sağlar.

Doğrudan Öğretim yöntemleri, öğretimin davranışsal bir yönünü de içerir. Davranışçı yaklaşımın öğrencilerin ders ortamından gönderilmeden de davranışlarında pozitif yönde değişiklik yaptığını söylemiştir. Engelmann tarafından tasarlanan görev dizileri, daha önce karmaşık olarak kabul edilen davranışlara aşamalı olarak odaklanır (Ryder vd., 2006):

Doğrudan Öğretim şu adımları içerir;

a)küçük gruplarla çalışma

b)yüksek performans gösteren öğrencilerin düşük performans gösteren öğrencilerden önce yanıt vermesini önlemek için sözlü yanıtları (koro yanıtları) bekleme sürelerini ayarlama

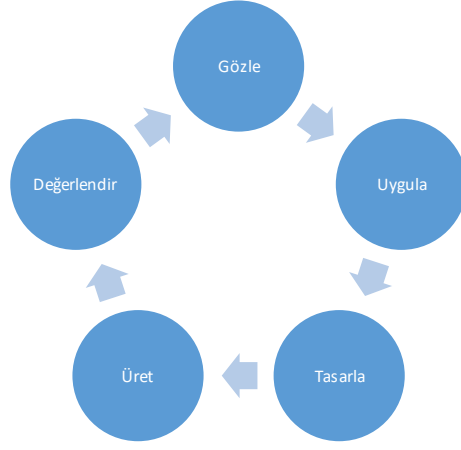
- c) hızlı ve kısa aralıklarla ilerleme,
- d) öğrenciler tarafından verilen yanıtları dikkatli dinleme,
- e) altı adımda hataları bulma ve geri bildirim verme (övgü, model olma, yol gösterme, test etme, sağlamlaştırma, gecikmeli test),
- f) motivasyon kaynağı olarak dışsal ödüller sunmak (Ryder vd., 2006).

Amaç nihayetinde "içsel motivasyonu" teşvik etmektir. Birçok motivasyon stratejisi, hızlı geri bildirim, yarışmalar, göreve katılım ve diğer dışsal olmayan teşviklerin yanı sıra puanlar ve ikramlar gibi ödülleri içerir (Ryder vd., 2006).

2.6.3 GUTÜD

Programlama eğitimi ile ilgili literatürde iki ana eğilim vardır. Bunlardan ilki, dersleri öğrencilerin keşfedebilecekleri ve sunabilecekleri öğrenci merkezli bir şekilde tasarlanmaktadır. Bu yaklaşıma göre öğrenciler süreçte aktif rol alırlar. Öğretmenler bilgiyi direk aktarmaktan çok öğrencilerin keşfetme sürecine katkıda bulunmak için rehber görevinde bulunmaktadır. İkinci yaklaşım merkeze öğretmeni alır. Burada asıl sorumluluğu öğretmen alır ve konuyu öğrencilere aktarmayı amaçlar (Çetin vd., 2021). Öğrencinin keşfetmesini temel alan ilk yaklaşımda, öğrenciler belirli bilgi işlemsel problemlerde yetkin olsalar da temel algoritmik kavramlarda ve dolayısıyla bunların farklı bağlamlara aktarımında sıkıntılar yaşamaktadır (Mayer, 2004). Öte yandan, bilgi aktarımına dayalı öğretmen merkezli doğrudan anlatım yöntemi, zamanımızın öğrencilerini hazırlamak için yeterli değildir (Papert, 1991).

Bu sebeplerden dolayı öğrencilerin belirli bir temelde üst düzey zihinsel faaliyetlerde bulunmalarını sağlamak ve farklı ortamlara aktarılabilecek bilgiler oluşturmak için keşfedici yöntemler ön plana çıkarılmak istenmektedir. Bu amaçlar için “**G**özle, **U**ygula, **T**asarla, **Ü**ret ve **D**eğerlendir” (GUTÜD) öğrenme döngüsü oluşturulmuştur (Çetin vd., 2021).



Şekil 2.3 Öğrenme döngüsü

Çetin vd. (2021) Öğrenme döngüsü Şekil 2.3 de görüldüğü üzere beş adımdan oluştuğunu söylemektedir. Bu beş adımı aşağıdaki gibi açıklamaktadır.

Gözle: Bu bölüm iki kısımdan oluşmaktadır. Birinci bölümde öğretmenler, öğrencilerin ön bilgilerini harekete geçirmek, dikkatlerini ve motivasyonlarını sağlamakla sorumludur. Bunu yapmak için, bir önceki derste yapılan etkinliği/araştırmayı basitçe özetleyebilir veya günlük hayattan veya bir bilim insanının hayatından ilginç örnekler kullanabilir. Bölümün ikinci bölümünde öğretmen aslında (göstererek) örnek olarak bir programlama konusunu açıklayabilir. Bu bölümde öğretmen daha aktiftir. Uygulamayı yaparken öğrencilere sorular sorabilir ve öğrencilerin sorularını cevaplayabilir.

Uygula: Bu bölümde öğretmen, öğrencilerden bir önceki bölümdeki alıştırmaların benzerlerini yapmalarını ister veya onlarla yapar. Örneğin öğretmen, öğrencilerden toplama işlemini program olarak yapmalarını ister onlardan da bir benzeri olan çıkarma işlemini oluşturmaları ister.

Tasarla: Bu bölümde öğrenciler daha aktif rol alırlar. Öğretmen rehber konumundadır. Öğretmenler, sıkıştıkları yerde öğrencileri desteklemek için yardımcı olabilirler. Ancak öğretmenler gereğinden fazla destek vermemelidir. Bu bölümde öğretmen öğrencilere açık uçlu bir problem verir. Öğrencilere verilmiş olan soruna bir çözüm yolu üretmeleri istenir. Tasarım aşamasında öğrenenler temel olarak bilinenler ile beklenenler arasında bağlantılar kurarak planlar yaparlar. Bunu yapmak için öğrenciler (bilgi işlemsel düşünme becerileri kullanarak ilerleyebilirler):

- ✓ Sırasıyla neyin bilindiğini ve neye ihtiyaç duyduklarını belirler.

- ✓ Gerekli öğeleri, eğer ayrılabilirse alt bileşenlerine ayırır.
- ✓ Daha önceden çözmüş olduğu benzer problemler var ise bunlardan yola çıkarak tanım yapar.
- ✓ Bilgisayar biliminde önceden belirlenmiş çözümlerle (sıralama ve arama algoritmaları gibi) bu problemlerin veya alt problemlerin çözümlerinin var olup olmadığına karar verme işlemini yapar.
- ✓ Önceki adımlarda ortaya çıkanları kullanarak bir çözüm formüle eder.

Bu aşamanın odak noktası, öğrencinin doğrudan çözüme başlamak yerine önce çözüm hakkında düşünmesi ve formüle etmesidir. Öğrenciler yukarıdaki beş adımı her seferinde yapmak istemeyebilir veya bunları yaparken sıkılabirler. Bu durumlarda, öğrenenleri bire bir adımları takip etmeye zorlamak yerine, doğrudan soruna değinmeden önce sorun hakkında düşünmeleri ve plan yapmaları istenir.

Üret: Öğrenciler bu bölümde de aktif rol alırlar. Öğretmen rehber konumundadır. Öğretmenler, sıkıştıkları yerde öğrencilere destek sağlarlar. "Yakınsal gelişim alanı" kavramında belirttiği gibi, bir kişinin yardımıyla başarabileceği ancak henüz bağımsız olarak başaramadığı bir durum meydana geldiğinde destek sağlanmalıdır (Vygotsky, 1978; akt. Üçgül vd., 2021). Oluşturma aşamasında, öğrencilerden bir önceki adımda geliştirdikleri planı kullanarak bir probleme algoritmik bir çözüm üretmeleri istenir. Öğrenenler bilgisayar ve robotları kullanarak gerekli donanım ve yazılım çözümlerini geliştirirler.

Değerlendir: Buradaki değerlendirme doğrudan öğrencinin performansını derecelendirmek değildir. Temel amaç, öğrencilerin yol boyunca deneyimledikleri ve öğrendikleri şeyler hakkında düşünmelerini sağlamaktır. Bu sayede öğrenciler, problem çözme süreçlerini, ders konularını ve kendilerini gözlemleyerek yeni öğrenme, öz değerlendirme ve planlama fırsatları elde edeceklerdir. Öğrencilerden aşağıdaki soruları yanıtlamaları istenebilir:

- ✓ Verilen problemi kendi ifadeleri ile açıklamaları istenir.
- ✓ Verilen problem ile ilgili hangi yöntemleri kullandığını ve neden bu yöntemleri seçtiği konusunda açıklama yapar.
- ✓ Verilen problem hakkında süreç boyun yaşamış olduğu sorunlar ve bu sorunlar ile nasıl başa çıktığı noktasında açıklamalarda bulunur.

- ✓ Problemlerin çözümünde kullanmış olduğunuz çözümlerin projenin ilerlemesinde yarar sağlayıp sağlamadığını açıklar.
- ✓ Grup arkadaşları ile ilgili çözüm sürecinde karşılaştığı problemleri ve problemlerin çözümü için ne yaptıklarını açıklar.
- ✓ Grup arkadaşların dan bu süreçte ne gibi yararlandığını açıklar.

Öğrencilerin burada tüm soruları cevaplamasına gerek yoktur. Verilen etkinlikle ilgili deneyimlerine bağlı olarak kendilerine uyan soruları yanıtlayabilirler. Öğrencilerden ayrıca yazılı olarak yanıt vermeleri istenebilir. Ancak öğrenenler bir süre sonra aynı soruları tekrar tekrar cevaplamaktan sıkılabilir. Bu durumda, bazı derslerin sonunda öğrencilerden dersteki genel deneyimlerini değerlendirmeleri istenebilir. Ayrıca değerlendirme sürecini daha etkili hale getirmek için çeşitli çalışmalar yapılabilir. Örneğin, öğrenciler bir daire şeklinde sıralanırlar. Bir öğretmen, öğrencilerden birine bir top (veya benzer bir nesne) atabilir ve ondan yukarıdaki sorulardan bir veya daha fazlasını yanıtlamasını veya sınıftaki genel deneyimini derecelendirmesini isteyebilir.

2.6.3.1 Eşli Programlama

Eşli programlama, yapılacak olan işlemleri iş birliği içinde bir bilgisayarda programlama şeklidir. Eşli programlama yazılım geliştirme metodolojisinin bir parçası olarak profesyonel yazılım mühendisleri tarafından uygulandığında (Beck, 2000), "pilot" olarak bilinen programcı klavye ve fareyi kontrol eder ve "yardımcı pilot" olarak adlandırılan programcı program tasarımı ve geliştirmesinde aktif bir katılımcı olarak hizmet eder. Programcılar belirli aralıklarla yapmış oldukları işleri değiştirirler. Eşli programlama araba yarışlarına benzetilebilir. Araba yarışlarında arabayı süren ve ona doğru yönde gitmesi için gereken yerleri gösteren bir kılavuz bulunur. Eşli programlamada da bilgisayarı kullanan bireyi sürücü olarak tanımlayabiliriz. Sürücünün görevi ise tasarlama ve programlama işlevlerini yapmak üzere gerekli programlardan yararlanmaktır. Sürücünün yanında bulunan kılavuzun görevi ise sürücünün yaptığı işlemler için değerlendirmede bulunmak ve çıkan sorunlar için çözüm önerisi üretmektir. Araba yarışından farklı olarak eşli programlamada sürücü ve kılavuz belli zamanlarda yer değiştirir (Çetin ve Berigel, 2017).

Çetin ve Berigel (2017), eşli programlama sürecini 3 bölüme ayırmıştır;

- Bir bilgisayar başına iki programcı verilir.

- Programcılardan bir tanesi kod yazım işlemlerini yaparken diğeri kodları kontrol eder ve süreci değerlendirir.

- Programcılar belirli bir süre sonra yerlerini değiştirirler.

Eşli programlama üzerine yapılan ilgili araştırmalar çoğunlukla yetişkin öğrencilerle gerçekleştirilmiştir. Bu araştırmalarda sıklıkla bireysel programlama ile eşli programlama karşılaştırılmış ve eşli programcılarının programlama derslerini başarıyla geçme, daha yüksek notlar alma, ödevleri tamamlama, problem çözme ve üst düzey düşünme becerilerini geliştirme, programlamayı öğrenme, programlama kavramlarını anlama, doğru kodlarla ve daha az hatayla kaliteli programlar üretme, üretkenliği artırma ve daha hızlı programlama gibi bir veya daha fazla performans ölçütünde daha etkili performans gösterdikleri rapor edilmiştir (Umapathy ve Ritzhaupt, 2017; Williams vd., 2002). Eşli programlama yöntemi için eşli gruplar oluşturulurken çeşitli yöntemler uygulanabilir. Eşli programlama yöntemini kullanan araştırmacılar, eşli grupları oluşturmak için farklı stratejiler kullanmışlardır. Williams ve Kessler (2000) çalışmalarında eşli programlama yöntemini kullanmışlar ve grubu aktif çalışan öğrenciler ya da pasif çalışan öğrenciler gibi öğrencilerin özelliklerine göre oluşturmuşlardır. McDowell vd. (2006) herhangi bir kural uygulamamış ve öğrenciler eşlerini özgürce seçmişlerdir. Öte yandan, Nagappan vd. (2003) farklı bir strateji uygulamış ve yazılım grupları rastgele oluşturmuştur.

Niculescu ve Plummer (2003) eşli programlama yönteminin bazı kurallara göre uygulanması gerektiğini şu şekilde belirtmişlerdir;

- Eşli grup üyeleri görevi başarıyla tamamlamak için aynı bilgisayar, klavye ve fare üzerinde birlikte çalışırlar. Tüm görevi birlikte tamamlarlar.

- Eşli programlama yönteminde her üyenin bir rolü vardır; bir üye sürücüyü gözlemleyen, yazılan kodları kontrol eden ve hataları düzelten, beyin fırtınası yapan ve önerilerde bulunan navigatör, diğer üye ise kodları yazan ve bilgisayarda tüm işleri yapan sürücüdür.

- Navigatör ve sürücü yazılımın tasarımına ve kodların türüne birlikte karar verirler.

- Sürücü ve navigatör her zaman birbirleriyle iletişim kurar ve görevin problemini çözmek için sorular sorar ve tartışır.

- Yazılım geliştirme sırasında sürücünün ve navigatörün rolü değişir, böylece grup üyeleri her iki rolde de deneyim kazanır.

Beck ve Andres (2004) liderliğindeki bir grup, eşli programlama sırasında tüm kodlamaların eşler tarafından birlikte yapılması gerektiğini söylemişlerdir. Ancak çiftin bu süreçte fiziksel olarak aynı ortamda bulunması gerekmemektedir. Başka ortamlarda bulunan eşler ekran paylaşımı ile bir araya gelerek eşli programlama yapabilirler.

Williams vd. (2006) Eşli programlama yapılırken eş seçme işleminde herhangi bir değişken dikkate alınmadan eşler belirlendiğinde bile %93 uyumluluk ve başarı elde edilebileceğini belirtmiştir. Bir başka çalışmada ise bireylerin rastgele seçilme durumlarında uyumluluk %90 olarak belirtilmiştir (Katira vd., 2004).

3. YÖNTEM

Bu bölümde, araştırmanın modeli, çalışma grubu, veri toplama araçları ve verilerin analiziyle ilgili bilgilere yer verilmiştir.

3.1 Araştırmanın Modeli

Bu araştırmada üç farklı öğrenme ortamının (GUTÜD, İnşacılık, Doğrudan Öğretim) altıncı sınıf öğrencilerinin kodlama başarısına, kodlamaya yönelik tutumuna, bilgi işlemsel düşünme becerisine yönelik öz yeterliğine ve bilgi işlemsel düşünme becerilerine etkisinin araştırılması hedeflenmiştir. Araştırmada nicel araştırma yöntemlerinden deneysel yöntem kullanılmıştır. Deneysel desen, değişkenler arasındaki nedensel ilişkiyi test etmek için kullanılan bir desendir. Ön test-son test kontrol gruplu deneysel desen, tekrarlanan ölçümleri (ön test-son test) ve farklı konu kategorilerini (deney-kontrol grubu) gösteren iki faktörlü deneysel desen olarak belirlenmiştir. Bu desende denekler deney veya kontrol gruplarından yalnızca birinde yer alır (Büyüköztürk, 2007). Fraenkel ve Wallen (2012), en az bir bağımsız değişkenin manipüle edildiği, diğer ilişkili değişkenlerin kontrol edildiği ve bir veya daha fazla bağımlı değişken üzerindeki etkilerin gözlemlendiği bir çalışma olarak tanımlamaktadır. Araştırmada deneysel desenin çeşitlerinden olan yarı deneysel desen yöntemi kullanılmıştır. Fraenkel ve Wallen (2012) araştırmacının denekleri gruplara rastgele atamadığı bir deneysel tasarım türü olarak tanımlamaktadır. Araştırma için Bolu ilinde merkez ilçeye bağlı bir ortaokul seçilip bu okuldan üç farklı altıncı sınıf şubesi kullanılmıştır. Bu şubeler Python yazılım dili ile üretilen random atama programı ile rastgele atanmıştır. Şubeler GUTÜD, İnşacılık, Doğrudan öğretim modeli şeklinde oluşturulmuştur. Şubeler belirlenip atama işlemlerinin yapılması ile tüm gruplara çalışma başlangıcında Gülbahar vd. (2018) Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeği (BİDBÖA) ve Otu ve Çetin'in (2020) Scratch Ortamı İçin Kodlama Başarı Testi (SOİKBT) ve Gül vd. (2019) geliştirmiş olduğu Ortaokul Öğrencilerinin Kodlamaya Yönelik Tutumları Ölçeği (KYTÖ) ile Çetin vd. (2020) Türkçeye uyarladığı Bilgi İşlemsel Düşünme Testi (BİDT) ön-test olarak uygulanmıştır. Üç ayrı öğretim 8 hafta boyunca belirlenen üç şubeye uygulanmıştır. Uygulanan üç farklı öğretimin konu ve etkinlik bazında içeriği aynıdır. Üç gruba da aynı öğretmen ders vermiştir. Öğretmen yüksek lisans derecesine sahiptir ve her üç öğretme/öğrenme ortamında

da tecrübesi bulunmaktadır. Haftalık uygulamaların her birinin bir hafta öncesinde, doğrudan öğretim yöntemi ile verilen içeriğin bir başka sınıfta pilot uygulaması gerçekleştirilmiştir. Tez yazarı, dersin öğretmeni ve tez danışmanı her hafta bir araya gelerek pilot uygulamayı değerlendirmiş ve içeriğe veya etkinliklere yönelik düzeltmeler üzerinde tartışmıştır. Süreç boyunca yalnızca bazı etkinliklerde küçük değişiklikler yapılması kararlaştırılmıştır. Yapılan uygulamalardan sonra gruplara BİDBÖA, SOİKBT, KYTÖ ve BİDT son-test olarak uygulanmıştır. Bu çalışma için hazırlanan yarı deneysel desenin özeti Tablo 3.1’de gösterilmiştir.

Tablo 3.1 Uygulanan yarı deneysel desenin özeti

Grup	Ön-Test	Öğrenme Ortamı	Son Test
Grup 1	BİDBÖA SOİKBT KYTÖ BİDT	GÜTUD	BİDBÖA SOİKBT KYTÖ BİDT
Grup 2	BİDBÖA SOİKBT KYTÖ BİDT	İnşacılık	BİDBÖA SOİKBT KYTÖ BİDT
Grup 3	BİDBÖA SOİKBT KYTÖ BİDT	D.Ö	BİDBÖA SOİKBT KYTÖ BİDT

- Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeği (BİDBÖA)

- Scratch Ortamı İçin Kodlama Başarı Testi (SOİKBT)
- Kodlamaya Yönelik Tutumları Ölçeği (KYTÖ)
- Bilgi İşlemsel Düşünme Testi (BİDT)
- Doğrudan Öğretim (DÖ)

3.2 Çalışma Grubu

Bu çalışmada uygun örnekleme yöntemi kullanılmıştır. Örnekleme teorisinin konusu, evrenden bir örneklemin seçilmesi ve evrenin yapısına en uygun örnekleme yöntemi kullanılarak bu örneklemden evrenin özelliklerinin

tahmin edilmesi işlemidir (Çıngı, 1994). Bu durum araştırmacılar için büyük bir avantajdır çünkü örnekleme yoluyla bilgi toplanır, daha az insan ve malzeme kaynağı kullanılır ve bilgi daha kısa sürede toplanabilmektedir (Büyüköztürk vd., 2016). Bolu ili merkez ilçesinde bulunan bir okuldan Python ile oluşturulmuş rastgele atama programı ile altıncı sınıfta bulunan altı şubenin üçü rastgele belirlenmiştir. Bu üç şubede bulunan öğrenciler ortalama olarak 10-12 yaş grubu aralığında bulunmaktadır. Yapılan bu çalışmaya toplam olarak 99 altıncı sınıf öğrencisi katılmıştır. Bu öğrencilerden 52'si kız, 47'si erkek öğrencidir. Erkek ve kız öğrencilerin dağılımları Tablo 3.2'de listelenmiştir.

Tablo 3.2 Araştırmaya dahil olan öğrencilerin demografik özellikleri

Şube Adı	Kız	Erkek	Toplam
GUTÜD	19	14	33
İnşacılık	14	20	34
Doğrudan Öğretim	19	13	32
Toplam	52	47	99

Öğrencilerin sınıf bazlı dağılımı Tablo 3.2'de gösterilmiştir. Birinci grupta (GUTÜD yöntemi) 19 kız öğrenci, 14 erkek öğrenci bulunmaktadır. İkinci grupta (İnşacılık yöntemi) 14 kız öğrenci ve 20 erkek öğrenci bulunmaktadır. Üçüncü grupta (doğrudan öğretim yöntemi) ise 19 kız öğrenci ve 13 erkek öğrenciden oluşmaktadır.

Çalışmanın yürütüldüğü okulda 5.sınıflarda yabancı dil ağırlıklı eğitim uygulaması yapılmaktadır. Talim ve Terbiye Kurulu Başkanlığının 20.02.2017 tarihli ve 10 sayılı Kurul Kararı eki İlköğretim Kurumları (İlkokul ve Ortaokul) Haftalık Ders Çizelgesi ile yabancı dil 15 ders saati, Türkçe 6 ders saati, Matematik 4 ders saati, -Fen Bilimleri 3 ders saati, Sosyal Bilgiler 3 ders saati, Din Kültürü ve Ahlak Bilgisi 2 ders saati seçmeli ders şeklinde uygulanacaktır. Söz konusu uygulama, ilgi (b) Kurul Kararı ile kabul edilen ve Ek-2'de yer alan "Yabancı Dil Ağırlıklı 5. Sınıf İngilizce Dersi Öğretim Programı" doğrultusunda yürütülecektir (Temel Eğitim Genel Müdürlüğü [TEGM], 2017). Araştırmaya

katılan öğrencilerin bilişim teknolojileri ve yazılım dersini ilk defa altıncı sınıfta almaktadırlar.

3.3 Veri Toplama Araçları

Bu araştırmada ön-test ve son test olarak BİDBÖA, SOİKBT, KYTÖ, BİDT uygulanmıştır. Bu ölçek ve testler ile ilgili detaylar alt başlıklar hâlinde verilmiştir.

3.3.1 Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeği

Ortaokul öğrencileri için bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısı ölçeği Gülbahar vd. (2018) tarafından geliştirilmiştir. Ölçek 5 faktöre ayrılmış ve toplamda 36 maddeden oluşmaktadır ve 3'lü Likert yapıda desenlenmiştir. Ölçekte alınacak toplam puan 36 ile 108 arasında değişmektedir. İfadeler küçükten büyüğe doğru sıralandığından, toplam puan hesaplanırken düşük puan, bireyin "bilgi işlemsel düşünme becerilerine yönelik öz yeterliliğinin" yüksek olduğunu göstermektedir. Unutulmamalıdır ki ölçekten alınan puanlar sadece beceri algısına yöneliktir ve bilgi işlemsel düşünme becerilerini ölçmek için ölçekten elde edilen verilerden daha kapsamlı verilerin elde edilmesi gerekmektedir. Bu testin ölçek geçerliğinin belirlenmesi için Cronbach's Alpha (α) analizinden yararlanılmıştır. Ölçek toplam güvenirlik katsayısı .943 olarak ortaya çıkmıştır.

3.3.2 Scratch Ortamı İçin Kodlama Başarı Testi

Araştırmada öğrencilerin Scratch ortamında kodlama başarılarını ölçmek için Otu (2020) tarafından geliştirilen test kullanılmıştır. Bu test 14 sorudan oluşmaktadır. Bu test üç sürümden oluşmakta olup bu çalışma kapsamında sadece Scratch kısmı kullanılmıştır. Bu test için oluşturulan belirtke tablosunda Millî Eğitim Bakanlığının altıncı sınıf bilişim teknolojileri ve yazılım dersi için oluşturduğu kazanımlar kullanılmıştır. 14 maddeden oluşan başarı testinin madde güçlükleri 0,26 ve 0,86 arasında değişmektedir. Testin ortalama güçlüğü 0,61 olarak bulunmuştur. Başarı testinin madde ayırt edicilik değerleri, 0,31 ile 0,73 arasında değişmektedir. Testin ortalama ayırt ediciliği 0,48 olarak ve KR20 güvenirlik değeri 0,74 olarak bulunmuştur.

3.3.3 Kodlamaya Yönelik Tutumları Ölçeği

Ortaokul öğrencilerinin kodlamaya yönelik tutum ölçeği Gül (2019) tarafından geliştirilmiştir ve bu çalışmada da kullanılmıştır. 5'li Likert tipte hazırlanmıştır. Bu ölçek toplam 13 maddeden oluşmaktadır. Bu maddelerin 5

tanisi olumlu madde, 8 tanesi olumsuz ifade olacak şekilde oluşturulmuştur. Ölçekten alınabilecek en yüksek puan 65, en düşük puan ise 13'tür. Güvenirlik katsayısı 0,93 olarak hesaplanmıştır.

3.3.4 Bilgi İşlemsel Düşünme Testi

Bilişimsel düşünme testi Román-González, Pérez-González ve Jiménez Fernández tarafından geliştirilmiştir (Román-González vd., 2017). Testin dili İspanyolcadır. Bu test bilgi işlemsel düşünme düzeyini ölçmek için tasarlanmıştır. Çetin vd. (2020) testi Türkçe hali ile yeniden düzenlemişlerdir. İspanyolcasında 28 soru varken Türkçe uyarlamasında bu sayı 24 olarak belirlenmiştir. Bilgi işlemsel düşünme testinin Türkçe uyarlamasındaki 24 sorunun kavram bazlı dağılımı şu şekilde olmuştur: temel sıralama (3 soru), defa tekrar et tipi döngü (4 soru), olana kadar tekrar et tipi döngü (3 soru), basit (if) karar ifadesi (3 soru), karmaşık (if-else) karar ifadesi (4 soru), while tipi döngü (3 soru), basit fonksiyonlar (4 soru). Bu testin iç tutarlılık değeri 0,78 ve farklı madde gücünde (0.27 ile 0.85 arasında) sorular içeren orta güçlükte bir test olduğu görülmektedir (Çetin vd., 2020).

3.4 Verilerin Toplanması

Araştırma 2020-2021 eğitim-öğretim yılı ikinci döneminde Bolu il merkezindeki bir ortaokulda bulunan üç altıncı sınıf şubesinde gerçekleştirilmiştir. Araştırmanın yapıldığı okulda beşinci sınıfta hazırlık sınıfı uygulaması bulunmaktadır. Bu nedenle öğrenciler beşinci sınıfta bilişim teknolojileri ve yazılım dersi almamışlardır. Araştırmada kullanılan ölçekler her öğrencinin tek bir bilgisayara oturacağı şekilde okulun bilişim sınıfı, kütüphanesi ve STEM atölyesi kullanılarak uygulanmıştır. Öğrencilerin ölçekleri doldurduğu anda iki bilişim öğretmeni ve BÖTE dördüncü sınıfta okumakta olan öğretmen adayları hazır bulunmuştur. Araştırma yapılmadan önce yetkili kurumlardan gerekli izinler alınmış ve bilgilendirmeler yapılmıştır.

3.5 Verilerin Analiz Edilmesi

Bu araştırmada verilerin analizi için SPSS 26 paket programı kullanılmıştır. Bu araştırmada GUTÜD eğitim modelinin İnşacılık ve Doğrudan Öğretim eğitim modelleri bakımından etkinliği sınanmıştır. Bu bağlamda veri analizlerine geçilmeden önce değişkenlerin normallik varsayımının varlığını sınamak için Kolmogorov-Smirnov testi yapılmıştır. Yapılan analizler sonucunda, araştırmada kullanılan tüm değişkenler bakımından normallik varsayımının

sağlandığı görülmüştür ($p>0,05$). Bu nedenle verilerin analizinde parametrik teknikler kullanılmıştır. GUTÜD, inşacılık ve doğrudan öğretim gruplarının varyansların eşit olup olmadığını belirlemek için ise Levene testi kullanılmıştır ve $p>0.05$ çıktığı için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir. Grupların ortalamaları arasında anlamlı fark olup olmadığını ortaya koyan ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Farklılık çıkan gruplarda farkın kaynağını belirlemek için ise Tukey çoklu karşılaştırma testinden yararlanılmıştır. Ayrıca değişkenlerin etki büyüklükleri incelenmiş ve orta düzeyde etkiye sahip oldukları gözlenmiştir. Eğitim modellerinin etkinliğini tespit etmek amacıyla öğrencilere ön test ve son test uygulaması gerçekleştirilmiş ve ön test ve son testler arasında farklılığın olup olmadığını ortaya koymak amacıyla İlişkili örneklem t-testi ve kovaryans analizleri yapılmıştır. Kovaryans analizinin yapılması için son olarak gruplardaki regresyon doğrularının eğimlerinin homojen olup olmadığı test edilmiştir. Araştırmada gerçekleştirilen bütün istatistiksel tekniklerde anlamlılık seviyesi .05 olacak şekilde belirlenmiştir.

3.6 Öğretim Ortamları

Bu çalışmada kullanılacak öğrenme ortamları için hazırlanmış olan etkinlikler Scratch ortamında yapılmıştır. 2 uzman akademisyen ve 1 uzman öğretmen tarafından kontrolleri sağlanarak oluşturulmuştur. Yapılan etkinlikler GUTÜD, Doğrudan öğretim ve İnşacılık tabanlı ders planları haline getirilmiştir. Bu planlar yine 2 uzman akademisyen ve 1 uzman öğretmen kontrolü ile ilerlemiştir. Öğretim ortamı olarak Bolu il merkezinde bulunan bir ortaokul seçilmiştir. Bu okulunun tercih sebebi öğrencilerin 5.sınıfta kodlama dersi almamış olmaları, uygun altyapıya sahip olması ve üç model hakkında da bilgi sahibi olan bir uzman öğretmen olmasından dolayı seçilmiştir. Öğretim modelleri uygulama yapılacak sınıflara öğretilmeden bir hafta öncesinde pilot olarak seçilmiş sınıflarda uygulanarak bir uzman akademisyen, iki uzman öğretmen tarafından kontrol edilip değişiklikler sağlanmıştır. Yapılan tüm uygulamalara tezin araştırmacısı da gözlemci olarak katılmıştır fakat olaylara hiçbir şekilde müdahalede bulunmamıştır. Birinci gruba GUTÜD tabanlı, ikinci gruba İnşacılık tabanlı, üçüncü gruba ise doğrudan öğretim tabanlı ders planı olarak işlenmiştir. Bütün gruplarda öğrenciler ikiye bölünmüş olarak eşli programlama yapılmıştır.

GUTÜD tabanında geliştirilen planda öğretmen gözle kısmında öğrencilere ilk olarak önceki derslerin ufak bir tekrarını yapar veya gerçek hayattan örnek vererek derse başlar. Örnek olarak iki sayıyı toplama işleminde rehber öğretmen değişkeninin tanımı ile konuya giriş yapar .Öğretmen öğrencilere konu ile ilgili çeşitli sorular sorarak ve öğrencilerin de sormuş olduğu sorulara yanıt vererek konu anlatımında bulunur. Öğrenciler toplama örneğinde ilk defa değişkenleri kullanacağı için Scratch programında değişkenlerin nasıl ekleneceği ve isimlendirileceği rehber öğretmen tarafından bir etkinlik üzerinden yaptırılır ve öğrencilerin de aynı örneği yapmaları istenir.

Uygulama kısmında rehber öğretmen öğrencilerden sizde iki sayı arasında çıkarma işlemi yapınız diyerek benzer bir örnek yapmalarını ister. Bu kısımda öğrencilere belirli bir süre tanınır. Öğretmen aralarda gezerek bazı noktalar da yardımcı olmakta fakat tüm çözümü vermemektedir.

Öğrenciler gözle ve uygula etkinlikleri ile yeterli bilgi birikimi oluşturduğunda tasarla etkinliklerine geçilir.Tasarla kısmında ise öğrencilerin daha önce öğrenmiş olduğu bilgiler bir bütün düşünülerek öğrencilerin bu kısmı yapmaları istenmektedir. Örnek olarak balık yakalama oyunu yapacakları söylenir. Balık yakalama ile ilgili öğrencilere bir problem verilir:

Ekranda birden fazla balık kuklası olacaktır. Bu balık kuklaları ekranda serbest bir biçimde sağa sola ve aşağı yukarı şekilde akvaryumda gezer gibi görüntü verilecektir. Bir adet köpek balığı kuklası eklenip bu kukla fare ile hareket etmesi sağlanacaktır. Bu kuklanın görevi balıkları yemek tir. Balıklara değdiği anda balıklar kaybolacaktır. Fakat belirli bir süre sonra balık tekrardan ekrana görünür halde çıkacaktır. Kukla ekle menüsünde denizde bulunan canlılardan herhangi biri eklenir ve bu canlılar oyunda yenilmemesi gereken kukla olarak kullanılır. Fare ile hareket ettirdiğimiz kukla bu kuklalara değdiği zaman puan menüsünden puan kaybedecek. Kaybedilecek puanı öğrencinin seçimine bırakılmıştır. Bu kuklalarda yenildiği anda ekranda kaybolacak ve belirli bir süre sonra tekrardan ekranda belirecektir. Oyunda puan sistemi uygulanacaktır. Bu puan sisteminde her yenilen balık için puan verilecek, her yenilen deniz canlısı için ise puan düşürülecektir.Oyuna zaman bloğu eklenecektir. Oyunun hangi saniyede biteceği öğrenciler tarafından belirlenecektir. Bu süre oyun içinde eksi puan getiren kuklaların ortaya çıkması içinde kullanılacaktır. Bu kuklalar ekrana çıkacak ve diğer eksi getiren kuklalardan farklı olarak renk değiştirme

animasyonu verilecek, bir daha kaybolmayacaktır. Örneğin 10.saniyede 1 kukla 20.saniyede başka bir kukla çıkarak oyunu zorlaştıracaktır. Oyunun sonunda herhangi bir ekrana eklenmiş kukla belirli bir puana ulaşıldığında veya belirli bir süre aşıldığında veya puana düştüğünde kaybettiniz veya kazandınız yazması istenmektedir.

Öğrencilere problem verilir ve öğrencilerden problem için bir çözüm tasarımlarını ister.Bu kısımda rehber öğretmen öğrencilerden problem hakkında düşünmesini ve çözüm getirmeleri istenir. Öğrenciler iki kişilik gruplar halinde bulunur ve birbirleri ile konuşarak çözüm önerileri için belli bir süre verilir.Rehber öğretmen öğrenciler arasında dolaşır takıldıkları kısımlarda yardım edebilir ama kesinlikle çözümün tamamını vermez. Onlara örnekler ile yardımcı olmaya çalışır. Örnek olarak öğrencinin balık sayısını nasıl bulacağını sorusu üzerine bir önceki çalışmada bulunan toplama-çıkarma işleminde neler yaptığımızı bir düşünün gibi cümleler ile onlara destek olunabilir.

Üret kısmında ise öğrencilere tasarla bölümünde oluşturdukları şablona göre çözümü üretmesi için belirli bir süre verilir ve Scratch programını açarak planı yapmaları istenir. Öğrenciler bu işlemi yaparken öğretmen sınıf içerisinde dolaşır ve öğrencilerin takıldıkları noktalarda rehber görevinde bulunur fakat kesinlikle çözümü vermez öğrencilerin çözümü bulması sağlanır.

Değerlendirme bölümünde öğretmen öğrencilerin öğrenme sürecinde yaşadıkları ve öğrenme sürecini düşünmelerini ister. Bu kısımda öğretmen öğrencilerin dikkatinin dağılmaması için bilgisayarları kapatıp sadece bu kısım ile ilgilenmeleri sağlar. Tüm sınıfa öğretmen balık yakalama uygulamasında hangi problemler ile karşılaştınız sorusunu sorar ve öğrencilerden cevapları alır.Verilen görevler göz önünde bulundurduğunuzda en çok hangi kısımları yaparken zorlandınız sorusu sorulur ve sınıftan cevaplar alınır.Bu zorlukların üstesinden nasıl geldiklerini arkadaşları ile paylaşması istenir.Daha ayrıntılı bir tartışma olması için rehber öğretmen grup arkadaşınızdan ne öğrendiniz birbirinize ne gibi katkınız oldu soruları sorarak değerlendirme aşaması tamamlanır.

İnşacılık tabanlı geliştirilen ders planında öğrencilere daha öncesinde hazırlanmış olan Scratch kartları dağıtılır. Öğrenciler ile birlikte Scratch kartları incelenir. Öğretmen tarafından öğrencilere istenilen işlemler söylenir. Kelebek motifi çizelim uygulamasını örnek verecek olursak öğretmen öğrencilere Scratch kartlarının içerisindeki içindekiler bölümünden yapılacak işlemlerin neler olduğunu

söyler ve öğrencilerin geçmiş bilgilerinde kullanarak uygulamaya başlaması istenir. Öğrenciler uygulamayı yaparken öğretmen öğrenciler arasında dolaşarak sıkıştıkları kısımlarda çözümün tamamını vermeyerek sadece yönlendirmelerde bulunur. Örnek olarak öğrenciler “kalemin ucu değil de sadece orta kısmı çizgi çiziyor” şeklinde soru geldiğinde öğretmen “Scratch kartlarındaki dikkat kısımlarını daha iyi okumalısın” veya “kaleminizin merkezi sizce neresi kontrol ettiniz mi?” şeklinde sorular ile öğrenciler yönlendirilir. Konuyu erken bitiren öğrencilere öğretmen siz olsaydınız bu uygulamaya başka neler eklemek isterdiniz sorusu sorarak öğrencilerin fikirlerini alır ve eklemelerini ister. Dersin sonunda öğretmen benzer örnek içerikleri içeren bir ödev verir.

Doğrudan öğretim tabanlı geliştirilen ders planında öğretmen model olarak ders anlatımına başlar bu tez için hazırlanmış uygulamalardan olan kalem ile kare, üçgen ve renkli kesikli çizgi çizme örneğini ele alırsak. Öğretmen öğrencilere “geometrik şekil nedir? ”Hangi geometrik şekilleri biliyorsunuz” şeklinde sorular yöneltir. Bu sorular öğrenciler ile tartışıldıktan sonra öğretmen bu geometrik şekillerin Scratch programında bulunan kalem eklentisi ile de çizelebildiğini söyler ve kalem eklentisinin nasıl ekleneceğini adım adım anlatır. Bu kısımda bulunan tümünü sil komutu ile sahnede daha önceden çizilmiş şekillerin tümü silineceği söylenir. Kalemi bastır komutunda ise ile program ilk açıldığında kalem kuklası elimizde havada duruyor gibi düşünebiliriz. Kalem kuklasının çizebilmesi için kalemi bastır komutu kullanılması gerektiği anlatılır. Öğretmen kare çizilebilmesi için gerekli bütün kodları kendi göstererek oluşturur ve öğrencilerden de aynısını yapmasını ister. Daha sonrasında öğretmen öğrencilerden bilginin daha iyi pekişmesi için üçgen çizelim der ve öğrencilere bu kısımda hangi kodları kullanmalıyız, kalemi orta noktada tutmak için hangi işlemleri yapmalıyız gibi sorular sorar. Bu kısımda öğretmen tüm sınıfın derse katılmasını sağlar. Öğrencilerin bağımsız bir uygulama yapabilmesi ve bol örnek kullanılarak dersin daha iyi anlaşılması için öğretmen “sizlerden kesikli çizgilerden oluşan gittikçe kalınlaşan bir uygulama yapmanızı istiyorum” der. Öğrencileri küçük gruplara ayırır ve örneği yapmalarını ister. Öğretmen bu sırada sınıftan gelen soruları da cevaplandırır.

4. BULGULAR

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

4.1 Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeğine Yönelik Bulgular

GUTÜD, inşacılık ve doğrudan öğretim grubunda bulunan öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlik ön test ve son test düzeylerini belirlemek üzere betimsel ve çıkarımsal istatistiklerden yararlanılmıştır. Tablo 4.1’de bilgi işlemsel düşünme becerisine yönelik öz yeterlilik ile ilgili betimsel istatistikler gösterilmiştir. GUTÜD grubunda bulunan öğrencilerin ön test bilgi işlemsel düşünme becerisine yönelik öz yeterlik ortalama puanları $\bar{X}= 141.76$ olarak bulunmuşken, son test bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanları ise $\bar{X}= 152,09$ olarak bulunmuştur. İkinci olarak inşacılık grubunda bulunan öğrencilerin ön test ortalama bilgi işlemsel düşünme becerisine yönelik öz yeterlik ortalama puanları, $\bar{X}= 130.68$ olarak; son test bilgi işlemsel düşünme becerisine yönelik öz yeterlik ortalama puanları ise $\bar{X}= 133.94$ şeklinde bulunmuştur. Son olarak doğrudan öğretim grubunda bulunan öğrencilerin ön test ortalama bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanları $\bar{X}= 124,06$ şeklinde; son test ortalama bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanları ise $\bar{X}= 130.09$ olarak bulunmuştur.

Tüm bu sonuçlar göz önünde bulundurulduğunda, bütün gruplarda son test bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanları ön test bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanlarına göre daha yüksek çıkmıştır.

Uygulama öncesinde ve sonrasında elde edilen bilgi işlemsel düşünme becerisine yönelik öz yeterlilik puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalamanın GUTÜD grubunda olduğu; ardından inşacılık grubu ve son olarak da doğrudan öğretim grubunun geldiği görülmüştür.

Tablo 4.1 Bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısına ait betimsel istatistikler

	Test	N	$\bar{X}$	Ortanc a	Mod	SS	Çarpı klık	Basıkl ık	Mi n	Mak
GUTÜD	Ön Test	3	141,76	142	141	19,39	-0,094	-0,514	10	180
	Son Test	3	152,09	158	170	18,69	-0,766	-0,184	10	176
İnşacılık	Ön Test	3	130,68	128	136	25,75	-0,135	-0,809	73	171
	Son Test	4	133,94	134	134	27,22	-0,243	-0,964	82	180
Doğrudan Öğretim	Ön Test	3	124,06	125	108	20,04	0,444	0,473	82	173
	Son Test	2	130,09	130	132	20,19	0,641	0,099	10	174
		2							0	

GUTÜD, inşacılık ve doğrudan öğretim gruplarından bulunan öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlik ön test puanları arasında fark olup olmadığını sınamak üzere ilişkisiz örneklemeler için tek yönlü varyans analizi yapılmıştır. Asıl analiz yapılmadan önce ilişkisiz örneklemeler için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.1’de Bilgi işlemsel düşünme becerisine yönelik öz yeterlik 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 bunların ışığında grupların normal dağıldığı varsayılmıştır.

Tablo 4.2 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.2 Kolmogorov-Smirnov testi sonuçları

	İstatistik	sd	p
GUTÜD	,107	33	,200
İnşacılık	,131	34	,145
Doğrudan Anlatım	,101	32	,200

GUTÜD, inşacılık ve doğrudan öğretim 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. Öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlik testini kişisel olarak tamamlamışlardır ve birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.3 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
2,499	2	96	,088

Grupların ortalamaları arasında anlamlı fark olup olmadığını ortaya koyan ilişkisiz örneklemeler için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları Tablo 4.4'te sunulmuştur. Analiz sonucunda GUTÜD grubunda bulunan öğrencilerin ortalaması ($\bar{X}$ =141,76), inşacılık grubunda bulunan öğrencilerin ortalaması ($\bar{X}$ =130,68) ve doğrudan öğretim grubunda bulunan öğrencilerin ortalamasının ($\bar{X}$ =124,06) en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir [$F(2-96)=5.397$, $p<0.05$]. Test sonucunda hesaplanan etki büyüklüğü ($\eta^2=(5212,280/51567,657)=0,10$) olarak orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Yapılan Tukey çoklu karşılaştırma testi sonucunda anlamlı farkın GUTÜD ve Doğrudan öğretim 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	5212,280	2	2606,140	5,397	,006	GUTÜD>Doğrudan Anlatım
Gruplar içi	46355,377	96	482,869			
Toplam	51567,657	98				

Uygulanan üç farklı öğretim yönteminin öğrencilerin, bilgi işlemsel düşünme becerisine yönelik öz yeterlilik 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 bilgi işlemsel düşünme becerisine yönelik öz yeterlilik son test puanlarının her bir grup içerisinde normal dağılımlarına 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
GUTÜD	,149	33	,062
İnşacılık	,134	34	,126
Doğrudan Öğretim	,130	32	,181

Gruplarda bilgi işlemsel düşünme becerisine yönelik öz yeterlik 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
1,787	2	96	,173

Sonuçlar incelendiğinde GUTÜD grubu bilgi işlemsel düşünme becerisine yönelik öz yeterlilik son test ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.73$) gözlenmiştir [$F_{(1-31)}=84.96$, $p<0.05$]; İnşacılık grubu bilgi işlemsel düşünme becerisine yönelik öz yeterlilik son test ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.83$) gözlenmiştir [$F_{(1-32)}=156.85$, $p<0.05$]; ve son olarak Doğrudan öğretim grubu bilgi işlemsel düşünme becerisine yönelik öz yeterlilik son test ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.84$) gözlenmiştir [$F_{(1-30)}=151.82$, $p<0.05$].

Kovaryans analizinin yapılması için son olarak gruptaki regresyon doğrularının eğimlerinin homojen olup olmadığını test edilmiştir. Yapılan analiz sonucunda $p>0.05$ ($p=0.473$) bulunmuştur. Bu sonuca göre regresyon doğrularının eğimleri homojendir denilebilir.

Kovaryans analizinin tüm varsayımlarını test ettikten sonra asıl test yapılmıştır. Kovaryans analizi sonuçlarına göre, her grubun bilgi işlemsel düşünme becerisine yönelik öz yeterlilik ön test puanları ile düzeltilmiş ortalama bilgi işlemsel düşünme becerisine yönelik öz yeterlilik son test puanlarında anlamlı bir fark yoktur. [$F(2-93)=0,755$, $p>0,05$].

Kovaryans analizinin sonuçları Tablo 4.7’de sunulmuştur.

Tablo 4.7 Kovaryans analizi sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p
Öz Yeterlilik ön test	35093,749	1	35093,749	354,109	,000
Grup	149,701	2	74,850	,755	,473
Hata	9216,701	93	99,104		
Toplam (düzeltilmiş)	57324,687	98			

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan bilgi işlemsel düşünme becerisine yönelik öz yeterlilik 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. Öncelikle ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılmalıdır. Tablo 4.8’de bilgi işlemsel düşünme becerisine yönelik öz yeterlilik 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

	N	Çarpıklık	Basıklık
GUTÜD	33	1,073	1,480
İnşacılık	34	,121	-,507
Doğrudan Anlatım	32	,105	,049

GUTÜD, inşacılık ve doğrudan öğretim yöntemlerinin öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik inançlarına etkisinin incelendiği bu bölümde GUTÜD ön test puan ortalaması ($\bar{X} = 141,76$) ile GUTÜD son test puan ortalaması ($\bar{X} = 152,09$) arasında anlamlı bir farklılığın olduğu ($t_{(32)} = -5,795$, $p < 0.00$) ve bu farkın orta büyüklükte bir etki gücüne sahip olduğu ifade edilebilir. İnşacılık yönteminin öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik inançları üzerindeki etkisi incelendiğinde ise ön test puan ortalaması ($\bar{X} = 130,68$) ile son test puan ortalaması ($\bar{X} = 133,94$) arasında anlamlı bir farklılığın olmadığı söylenebilir ($t_{(33)} = -1,693$, $p = 0,10$). Doğrudan öğretim yönteminin öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik inançlarına olan etkisi açısından incelendiğinde ise öğrencilerin ön test puan ortalaması ($\bar{X} = 124,06$) ile son test puan ortalaması ($\bar{X} = 130,09$) arasında anlamlı bir farklılığın olduğu ($t_{(31)} = -4,084$, $p < 0.00$) ve bu farkın orta büyüklükte bir etki gücüyle temsil edildiği ifade edilebilir.

Tablo 4.9 İlişkili örneklem t-testi

	N	Ort.	SS	Ortalamalar Arası Fark	SS	t	sd	p
GUTÜD Ön	33	141,76	19,39	-10,33	10,24	-5,795	32	,000
GUTÜD Son	33	152,09	18,69					
İnşacılık Ön	34	130,68	25,75	-3,26	11,24	-1,693	33	,100
İnşacılık Son	34	133,94	27,22					
Doğrudan Öğretim Ön	32	124,06	20,04	-6,03	8,35	-4,084	31	,000
Doğrudan Öğretim Son	32	130,09	20,19					

4.2 Bilgi İşlemsel Düşünme Testine Yönelik Bulgular

GUTÜD, inşacılık ve doğrudan öğretim grubunda bulunan öğrencilerin BİDT ön test ve son test düzeylerini belirlemek üzere betimsel ve çıkarımsal istatistiklerden yararlanılmıştır. Tablo 4.11’de BİDT yönelik betimsel istatistikler gösterilmiştir. GUTÜD grubunda bulunan öğrencilerin ön test ortalama BİDT puanları $\bar{X} = 11.61$ olarak bulunmuşken, GUTÜD grubunda bulunan öğrencilerin son test ortalama BİDT puanları ise $\bar{X} = 16,88$ olarak bulunmuştur. İkinci olarak İnşacılık grubunda bulunan öğrencilerin ön test ortalama BİDT puanları, $\bar{X} = 10.18$ olarak; İnşacılık grubunda bulunan öğrencilerin son test ortalama BİDT puanları, $\bar{X} = 14.18$ olarak bulunmuştur. Son olarak doğrudan öğretim bulunan öğrencilerin ön test ortalama BİDT puanları ise $\bar{X} = 9.47$ olarak; doğrudan öğretim grubunda bulunan öğrencilerin son test ortalama BİDT puanları $\bar{X} = 13.47$ bulunmuştur.

Tüm bu sonuçlar göz önünde bulundurulduğunda bütün gruplarda son test BİDT puanları ön test BİDT puanlarına göre daha yüksektir. Uygulama öncesinde ve sonrasında elde edilen BİDT puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama GUTÜD grubuna ardından inşacılık grubuna ve son olarak en düşük ortalama doğrudan öğretim grubuna aittir.

Tablo 4.10 Bilgi işlemsel düşünmeye ait betimsel istatistikler

	Test	N	$\bar{X}$	Ortanca	Mod	SS	Çarpıklık	Basıklık	Min	Mak
GUTÜD	Ön Test	33	11,61	11	10	3,99	-0,579	-0,254	2	18
	Son Test	33	16,88	17	17	3,27	-0,360	-0,556	11	22
İnşacılık	Ön Test	34	10,18	11	11	3,54	-0,220	0,966	2	18
	Son Test	34	14,18	15	16	4,00	-0,456	-0,022	5	21
Doğrudan Öğretim	Ön Test	32	9,47	10	10	2,78	0,371	-0,281	4	15
	Son Test	32	13,47	13	13	3,74	0,360	-0,316	7	21

GUTÜD, inşacılık ve doğrudan öğretim gruplarından bulunan öğrencilerin BİDT ön test puanları arasında fark olup olmadığını bakmak üzere ilişkisiz örneklem için tek yönlü varyans analizi yapılmıştır. Asıl analiz yapılmadan önce ilişkisiz örneklem için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.10’de BİDT ön test puanlarının her bir grup içerisinde normal dağıldıklarını işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.10 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 4.11 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.11 Kolmogorov-Smirnov testi sonuçları

	İstatistik	Sd	p
GUTÜD	,150	33	,058
İnşacılık	,127	34	,179

Doğrudan Öğretim	,174	32	,055

GUTÜD, inşacılık ve doğrudan öğretim 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 4.12’te sunulmuştur. Öğrencilerin Öz yeterlik testini kişisel olarak tamamlamışlardır ve birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.12 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
2,258	2	96	,110

Grupların ortalamaları arasında anlamlı fark olup olmadığını ortaya koyan ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları Tablo 4.13’te sunulmuştur. Analiz sonucunda GUTÜD grubunda bulunan öğrencilerin ortalaması ($\bar{X}$ =11,61), İnşacılık grubunda bulunan öğrencilerin ortalaması ($\bar{X}$ =10,18) ve doğrudan öğretim grubunda bulunan öğrencilerin ortalamasının ($\bar{X}$ =9,47) en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir [$F(2-96)=3.195$, $p<0.05$]. Test sonucunda hesaplanan etki büyüklüğü ($\eta^2=(77,393/1240,182)=0,06$) olarak orta düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Yapılan Tukey çoklu karşılaştırma testi sonucunda anlamlı farkın GUTÜD ve doğrudan öğretim grubunda bulunan öğrencilerin puanları arasında olduğu görülmüştür.

Tablo 4.13 ANOVA sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	Anlamlı Fark
Gruplar arası	77,393	2	38,697	3,195	,045	GUTÜD>Doğrudan Anlatım
Gruplar içi	1162,789	96	12,112			
Toplam	1240,182	98				

Uygulanan üç farklı öğretim yönteminin öğrencilerin, öz yeterlilik 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 4.10’de BİDT son test puanlarının her bir grup içerisinde normal dağıldıklarına işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.10 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 4.14 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.14 Kolmogorov-Smirnov testi sonuçları

	İstatistik	Sd	p
GUTÜD	,121	33	,200
İnşacılık	,120	34	,200*
Doğrudan Anlatım	,175	32	,054

Gruplarda Öz yeterlik son test değişkeni açısından grup varyanslarının homojenliği Levene testi sonuçları ile belirlenmiştir. Levene testi sonuçları Tablo 4.15’te sunulmuştur. Levene testine göre $p > 0.05$ olduğu için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir.

Tablo 4.15 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
0,437	2	96	,647

Sonuçlar incelendiğinde GUTÜD grubu Bilgi İşlemsel Düşünme son test ve Bilgi İşlemsel Düşünme ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.18$) gözlenmiştir [$F_{(1-31)}=6.841$, $p<0.05$]; İnşacılık grubu Bilgi İşlemsel Düşünme son test ve Bilgi İşlemsel Düşünme ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.02$) bulunamamıştır [$F_{(1-32)}=0.613$, $p>0.05$]; ve son olarak Doğrudan öğretim grubu Bilgi İşlemsel Düşünme son test ve Bilgi İşlemsel Düşünme ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.09$) bulunamamıştır [$F_{(1-30)}=3.51$, $p>0.05$].

Kovaryans analizinin yapılması için son olarak gruplardaki regresyon doğrularının eğimlerinin homojen olup olmadığını test edilmiştir. Yapılan analiz sonucunda $p>0.05$ ($p=0.607$) bulunmuştur. Bu sonuca göre regresyon doğrularının eğimleri homojendir denilebilir.

Tablo 4.16 Kovaryans analizi sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p
Bilgi İşlemsel Düşünme ön test	97,971	1	97,971	7,657	,007
Grup	12,845	2	6,423	,502	,607
Hata	1189,968	93	12,795		
Toplam (Düzeltilmiş)	1514,727	98			

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 4.17’de BİDT son test- ön test fark puanlarının her bir grup içerisinde normal dağıldıklarını gösteren basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.17 incelendiğinde basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir.

Tablo 4.17 Fark puanlarının basıklık ve çarpıklık değerleri

	N	Çarpıklık	Basıklık
GUTÜD	33	,603	-,486
İnşacılık	34	,646	,953
Doğrudan Anlatım	32	,374	-,327

GUTÜD, inşacılık ve doğrudan öğretim yöntemlerinin öğrencilerin BİDT etkisinin incelendiği bu bölümde GUTÜD ön test puan ortalaması ($\bar{X} = 11,61$) ile Gütüd son test puan ortalaması ($\bar{X} = 16,88$) arasında anlamlı bir farklılığın olduğu ($t_{(32)} = -7,689$, $p=0.00$) ve bu farkın orta büyüklükte bir etki gücüne sahip olduğu ifade edilebilir. İnşacılık yönteminin öğrencilerin BİDT üzerindeki etkisi incelendiğinde ise ön test puan ortalaması ($\bar{X} = 10,18$) ile son test puan ortalaması

($\bar{X} = 14,18$) arasında anlamlı bir farklılığın olduğu söylenebilir ($t_{(33)} = -4,702$), $p=0.00$) ve bu farkın orta büyüklükte etki gücüne sahip olduğu ifade edilebilir. Doğrudan öğretim yönteminin öğrencilerin BİDT olan etkisi açısından incelendiğinde ise öğrencilerin ön test puan ortalaması ($\bar{X} = 9,47$) ile son test puan ortalaması ($\bar{X} = 13,47$) arasında anlamlı bir farklılığın olduğu ($t_{(32)} = -5,774$), $p=0.00$) ve bu farkın orta etki gücüyle temsil edildiği ifade edilebilir.

Tablo 4.18 İlişkili örneklem t-testi

	N	Ort.	SS	Ortalamlar Arası Fark	SS	t	sd	p
GUTÜD Ön	33	11,61	3,99	-5,27	3,94	-	32	,000
GUTÜD Son	33	16,88	3,27			7,689		
İnşacılık Ön	34	10,18	3,54	-4,00	4,96	-	33	,000
İnşacılık Son	34	14,18	4,00			4,702		
Doğrudan öğretim Ön	32	9,47	2,78	-4,00	3,92	-	31	,000
Doğrudan öğretim Son	32	13,47	3,74			5,774		

4.3 Ortaokul Öğrencilerinin Kodlamaya Yönelik Tutumları Ölçeğine Yönelik Bulgular

GUTÜD, inşacılık ve doğrudan öğretim grubunda bulunan öğrencilerin kodlamaya yönelik tutum ön test ve son test düzeylerini belirlemek üzere betimsel ve çıkarımsal istatistiklerden yararlanılmıştır. Tablo 4.21’de kodlamaya yönelik tutum testine yönelik betimsel istatistikler gösterilmiştir. GUTÜD grubunda bulunan öğrencilerin ön test ortalama kodlamaya yönelik tutum puanları, $\bar{X} = 51,33$ olarak bulunmuşken, GUTÜD grubunda bulunan öğrencilerin son test ortalama kodlamaya yönelik tutum puanları ise $\bar{X} = 56,06$ olarak görülmüştür. İkinci olarak İnşacılık grubunda bulunan öğrencilerin ön test ortalama kodlamaya yönelik tutum puanları, $\bar{X} = 48,56$ olarak; İnşacılık grubunda bulunan öğrencilerin son test ortalama kodlamaya yönelik tutum puanları, $\bar{X} = 52,03$ şeklinde; Doğrudan öğretim grubunda bulunan öğrencilerin ön test ortalama kodlamaya yönelik tutum puanları, $\bar{X} = 37,41$ bulunmuştur. Doğrudan öğretim grubunda bulunan öğrencilerin son test ortalama kodlamaya yönelik tutum puanları, $\bar{X} = 52$ olarak görülmüştür.

Tüm bu sonuçlar göz önünde bulundurulduğunda bütün gruplarda son test kodlamaya yönelik tutum testi puanları ön test kodlamaya yönelik tutum testi puanlarına göre daha yüksektir.

Uygulama öncesinde ve sonrasında elde edilen kodlamaya yönelik tutum testi puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama GUTÜD grubuna ardından İnşacılık grubuna ve son olarak en düşük ortalama doğrudan öğretim grubuna aittir.

Tablo 4.19 Ortaokul öğrencilerinin kodlamaya yönelik tutuma ait betimsel istatistikler

	Test	N	$\bar{X}$	Ortanca	Mod	SS	Çarpıklık	Basıklık	Min	Mak
GUTÜD	Ön Test	33	51,33	52	52	5,96	-0,400	-0,260	37	61
	Son Test	33	56,06	58	58	5,67	-0,825	0,237	41	64
İnşacılık	Ön Test	34	48,56	47	47	4,65	0,415	-0,763	41	58
	Son Test	34	52,03	52	55	8,40	-0,016	-0,988	37	65
Doğrudan Anlatım	Ön Test	32	37,41	37	36	4,94	0,273	0,022	28	48
	Son Test	32	52,00	52	52	5,04	-0,307	-0,459	41	61

GUTÜD, İnşacılık ve Doğrudan öğretim gruplarından bulunan öğrencilerin kodlamaya yönelik tutum ön test puanları arasında fark olup olmadığını sınamak üzere ilişkisiz örneklemeler için tek yönlü varyans analizi yapılmıştır. Asıl analizi gerçekleştirmeden önce ilişkisiz örneklemeler için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.19’de kodlamaya yönelik tutum ön test puanlarının her bir grup içerisinde normal dağıldıklarını işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.19 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 4.20 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.20 Kolmogorov-Smirnov testi sonuçları

	İstatistik	Sd	p
GUTÜD	,106	33	,200*
İnşacılık	,081	34	,200*

Doğrudan Anlatım	,107	32	,200*

GUTÜD, inşacılık ve doğrudan öğretim 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 4.21’te sunulmuştur. Öğrencilerin kodlamaya yönelik tutum testini kişisel olarak tamamlamışlardır ve birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.21 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
,970	2	96	,383

Grupların ortalamaları arasında anlamlı fark olup olmadığını ortaya koyan ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları Tablo 4.22’te sunulmuştur. Analiz sonucunda GUTÜD grubunda bulunan öğrencilerin ortalaması ($\bar{X} = 51,33$), inşacılık grubunda bulunan öğrencilerin ortalaması ($\bar{X} = 48,56$) ve doğrudan öğretim grubunda bulunan öğrencilerin ortalamasının ($\bar{X} = 37,41$) en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir [$F(2-96)=64,807$, $p < 0.05$]. Test sonucunda hesaplanan etki büyüklüğü ($\eta^2 = (3523,111/6132,545) = 0,57$) olarak etki büyüklüğü büyük düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Yapılan Tukey çoklu karşılaştırma testi sonucunda anlamlı farkın GUTÜD ve doğrudan öğretim grubunda ve inşacılık ve doğrudan öğretim grubunda bulunan öğrencilerin puanları arasında olduğu görülmüştür.

Tablo 4.22 ANOVA sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	P	Anlamlı Fark
Gruplar arası	3523,111	2	1761,556	64,807	,000	GUTÜD>Doğrudan Öğretim İnşacılık>Doğrudan Öğretim
Gruplar içi	2609,434	96	27,182			
Toplam	6132,545	98				

Uygulanan üç farklı öğretim yönteminin öğrencilerin, kodlamaya yönelik tutum 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 4.20’de kodlamaya yönelik tutum son test puanlarının her bir grup içerisinde normal dağıldıklarına işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.20 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 4.23 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.23 Kolmogorov-Smirnov testi sonuçları

	İstatistik	Sd	p
GUTÜD	,149	33	,061
İnşacılık	,081	34	,200*
Doğrudan Anlatım	,099	32	,200*

Gruplarda kodlamaya yönelik tutum son test değişkeni açısından grup varyanslarının homojenliği Levene testi sonuçları ile belirlenmiştir. Levene testi sonuçları Tablo 4.24’te sunulmuştur.

Levene testine göre $p < 0.05$ olduğu için grupların varyansları aralarında anlamlı farkın olduğu kabul edilmiştir.

Tablo 4.24 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
7,936	2	96	,001

Sonuçlar incelendiğinde GUTÜD grubu kodlamaya yönelik tutum son test ve kodlamaya yönelik tutum ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.59$) gözlenmiştir [$F_{(1-31)}=24,834$ $p < 0.05$]; inşacılık grubu kodlamaya yönelik tutum son test ve kodlamaya yönelik tutum ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.29$) gözlenmiştir [$F_{(1-32)}=13.077$, $p < 0.05$]; ve son olarak doğrudan öğretim grubu kodlamaya yönelik tutum son test ve kodlamaya

yönelik tutum ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.00$) bulunamamıştır [$F_{(1-30)}=0.062$, $p>0.05$].

Kovaryans analizinin yapılması için son olarak gruptaki regresyon doğrularının eğimlerinin homojen olup olmadığını test edilmiştir. Yapılan analiz sonucunda $p<0.05$ ($p=0.001$) bulunmuştur. Bu sonuca göre regresyon doğrularının eğimleri homojen değildir denilebilir.

Tablo 4.25 Kovaryans analizi sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p
Tutum ön test	762,789	1	762,789	24,834	,000
Grup	432,654	2	216,327	7,043	,001
Hata	9216,701	93	99,104		
Toplam (Düzeltilmiş)	4500,909	98			

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan kodlamaya yönelik tutum testi 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 4.26’de kodlamaya yönelik tutum son test- ön test fark puanlarının her bir grup içerisinde normal dağıldıklarını gösteren basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.26 incelendiğinde basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir.

Tablo 4.26 Fark puanlarının basıklık ve çarpıklık değerleri

	N	Çarpıklık	Basıklık
GUTÜD	33	1,524	1,392
İnşacılık	34	-,011	-,692
Doğrudan Öğretim	32	-1,030	1,650

GUTÜD, inşacılık ve doğrudan öğretim yöntemlerinin öğrencilerin kodlamaya yönelik tutum testine etkisinin incelendiği bu bölümde GUTÜD ön test puan ortalaması ($\bar{X} = 51,33$) ile gutüd son test puan ortalaması ($\bar{X} = 56,06$) arasında anlamlı bir farklılığın olduğu ($t_{(32)} = -6,853$, $p=0.00$) ve bunun büyük düzeyde bir etki gücüne sahip olduğu ifade edilebilir. İnşacılık yönteminin

öğrencilerin kodlamaya yönelik tutum inançları üzerindeki etkisi incelendiğinde ise ön test puan ortalaması ($\bar{X}$ =48,56) ile son test puan ortalaması ($\bar{X}$ =52,03) arasında anlamlı bir farklılığın olduğu söylenebilir ($t_{(33)} = -2,860$), $p < 0,00$) ve bu farkın büyük düzeyde ile temsil ettiği söylenebilir. Doğrudan öğretim yönteminin öğrencilerin kodlamaya yönelik tutum testine olan etkisi açısından incelendiğinde öğrencilerin ön test puan ortalaması ($\bar{X}$ =37,41) ile son test puan ortalaması ($\bar{X}$ =52,00) arasında anlamlı bir farklılığın olduğu ve bu farkın büyük düzeyde bir etki gücüyle temsil edildiği ifade edilebilir.

Tablo 4.27 İlişkili örneklem t-testi

	N	Ort.	SS	Ortalamalar Arası Fark	SS	t	sd	p
GUTÜD Ön	33	51,33	5,96	-4,73	3,96	-6,853	32	,000
GUTÜD Son	33	56,06	5,67					
İnşacılık Ön	34	48,56	4,65	-3,47	7,08	-2,860	33	,007
İnşacılık Son	34	52,03	8,40					
Doğrudan öğretim Ön	32	37,41	4,94	-14,59	7,21	-11,442	31	,000
Doğrudan öğretim Son	32	52,00	5,04					

4.4 Scratch Ortamı İçin Kodlama Başarısı Testine Yönelik Bulgular

GUTÜD, inşacılık ve doğrudan öğretim grubunda bulunan öğrencilerin kodlama başarısı ön test ve son test düzeylerini belirlemek üzere betimsel ve çıkarımsal istatistiklerden yararlanılmıştır. Tablo 4.28’de kodlama başarısına yönelik betimsel istatistikler gösterilmiştir. GUTÜD grubunda bulunan öğrencilerin ön test ortalama kodlama başarı puanları, $\bar{X}$ = 5,18 olarak bulunmuşken, GUTÜD grubunda bulunan öğrencilerin son test ortalama kodlama başarı puanları ise $\bar{X}$ = 10,30 olarak bulunmuştur. İkinci olarak İnşacılık grubunda bulunan öğrencilerin ön test ortalama kodlama başarı puanları, $\bar{X}$ = 4,59 olarak; inşacılık grubunda bulunan öğrencilerin son test ortalama kodlama başarı puanları ise $\bar{X}$ = 6,85 olarak bulunmuştur. Son olarak doğrudan öğretim bulunan öğrencilerin ön test ortalama kodlama başarı puanları, $\bar{X}$ = 4,81 bulunmuştur. Doğrudan öğretim grubunda bulunan öğrencilerin son test ortalama kodlama başarı puanları, $\bar{X}$ = 8,16 görülmüştür.

Tüm bu sonuçlar göz önünde bulundurulduğunda bütün gruplarda son test kodlama başarı puanları ön test kodlama başarı puanlarına göre daha yüksektir.

Uygulama öncesinde ve sonrasında elde edilen kodlama başarı puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama GUTÜD grubuna ardından doğrudan öğretim grubuna ve son olarak en düşük ortalama inşacılık grubuna aittir.

Tablo 4.28 Scratch ortamı için kodlama başarısına ait betimsel istatistikler

	Test	N	$\bar{X}$	Ortanca	Mod	SS	Çarpıklık	Basıklık	Min	Mak
GUTÜD	Ön Test	33	5,18	5	6	1,33	0,151	-0,265	3	8
	Son Test	33	10,30	10	11	1,63	-0,155	-0,772	7	13
İnşacılık	Ön Test	34	4,59	5	5	1,76	-0,209	-0,639	1	8
	Son Test	34	6,85	7	6	2,02	0,237	-0,928	4	11
Doğrudan Anlatım	Ön Test	32	4,81	5	7	2,10	-0,092	-1,252	1	8
	Son Test	32	8,16	8	8	2,14	-0,426	-0,355	4	12

GUTÜD, inşacılık ve doğrudan öğretim gruplarından bulunan öğrencilerin kodlama başarı ön test puanları arasında fark olup olmadığını sınamak üzere ilişkisiz örneklemeler için tek yönlü varyans analizi yapılmıştır. Asıl analiz yapılmadan önce ilişkisiz örneklemeler için tek yönlü varyans analizinin varsayımları kontrol edilmiştir.

Tablo 4.28’de Kodlama başarı ön test puanlarının her bir grup içerisinde normal dağıldıklarını işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.28 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 4.29 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.29 Kolmogorov-Smirnov testi sonuçları

	İstatistik	Sd	p
GUTÜD	,150	33	,056

İnşacılık	,134	34	,125
Doğrudan Anlatım	,152	32	,060

GUTÜD, inşacılık ve doğrudan öğretim gruplarının varyansların eşit olup olmadığı Levene testi ile denenmiştir. Levene testine göre $p>0.05$ olduğu için grupların varyansları aralarında anlamlı fark olmadığı kabul edilmiştir. Levene testinin sonuçları Tablo 4.30’te sunulmuştur. Öğrencilerin kodlama başarı testini kişisel olarak tamamlamışlardır ve birbirlerinden etkilenmemeleri sağlanmıştır.

Tablo 4.30 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
5,486	2	96	,066

Grupların ortalamaları arasında anlamlı fark olup olmadığını ortaya koyan ilişkisiz örneklem için tek yönlü varyans analizi gerçekleştirilmiştir. Analiz sonuçları Tablo 4.31’te sunulmuştur. Analiz sonucunda GUTÜD grubunda bulunan öğrencilerin ortalaması ($\bar{X}$ =5,18), inşacılık grubunda bulunan öğrencilerin ortalaması ($\bar{X}$ =4,59) ve doğrudan öğretim grubunda bulunan öğrencilerin ortalamasının ($\bar{X}$ =4,81) en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiştir [$F(2-96)=4,973$, $p<0.05$]. Test sonucunda hesaplanan etki büyüklüğü ($\eta^2=(146.001/302.020)=0,48$) olarak büyük düzeyde olduğunu göstermektedir (Green ve Salkind, 2005). Yapılan Tukey çoklu karşılaştırma testi sonucunda anlamlı farkın GUTÜD, doğrudan öğretim ve inşacılık grupları arasında olduğu görülmüştür.

Tablo 4.31 ANOVA sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	Anlamlı Fark
Gruplar arası	146,001	2	3,000	4,973	,002	GUTÜD>Doğrudan Anlatım Doğrudan Anlatım>İnşacılık
Gruplar içi	296,019	96	3,084			
Toplam	302,020	98				

Uygulanan üç farklı öğretim yönteminin öğrencilerin, kodlama başarı 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 4.28’de Kodlama başarı son test puanlarının her bir grup içerisinde normal dağıldıklarına işaret eden ortalama, ortanca, mod, basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.28 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 4.32 incelendiğinde Kolmogorov-Smirnov testinin sonuçları verilerin normal dağıldığını göstermektedir.

Tablo 4.32 Kolmogorov-Smirnov testi sonuçları

	İstatistik	Sd	p
GUTÜD	,150	33	,056
İnşacılık	,134	34	,125
Doğrudan Anlatım	,138	32	,123

Gruplarda Öz yeterlik son test değişkeni açısından grup varyanslarının homojenliği Levene testi sonuçları ile belirlenmiştir. Levene testi sonuçları Tablo 4.30’da sunulmuştur.

Levene testine göre $p < 0.05$ olduğu için grupların varyansları aralarında anlamlı fark olduğu görülmüştür.

Tablo 4.33 Levene testi sonucu

Levene İstatistik	sd1	sd2	p
4,689	2	96	,011

Sonuçlar incelendiğinde GUTÜD grubu Kodlama Başarısı son test ve Kodlama Başarısı ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.41$) gözlenmiştir [$F_{(1-31)}=24.462$, $p < 0.05$]; İnşacılık grubu Kodlama Başarısı son test ve Kodlama Başarısı ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.64$) gözlenmiştir [$F_{(1-32)}=57.613$, $p < 0.05$]; ve son olarak Doğrudan öğretim grubu Kodlama Başarısı son test ve Kodlama Başarısı ön test puanları arasında anlamlı doğrusal bir ilişki ($R^2=0.18$) gözlenmiştir [$F_{(1-30)}=6.516$, $p < 0.05$].

Kovaryans analizinin yapılması için son olarak gruplardaki regresyon doğrularının eğimlerinin homojen olup olmadığını test edilmiştir. Yapılan analiz sonucunda $p > 0.05$ ($p = 0.039$) bulunmuştur. Bu sonuca göre regresyon doğrularının eğimleri homojen değildir denilebilir.

Tablo 4.34 Kovaryans analizi sonuçları

Varyansın Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p
Kodlama Başarısı ön test	134,734	1	134,734	59,028	,000
Grup	15,315	2	7,657	3,355	,039
Hata	212,278	93	2,283		
Toplam (Düzeltilmiş)	564,182	98			

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan kodlama başarı testi 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.

Öncelikle ortalamaları kıyaslanacak verilerin farklarının oluşturduğu veri dizisi normal dağılmalıdır. Tablo 4.35'te kodlama başarı son test- ön test fark puanlarının her bir grup içerisinde normal dağıldıklarını gösteren basıklık ve çarpıklık değerleri verilmiştir. Tablo 4.35 incelendiğinde basıklık ve çarpıklık kat sayılarının -1.96 ve 1.96 aralığında olduğu görülmektedir.

Tablo 4.35 Fark puanlarının basıklık ve çarpıklık değerleri

	N	Çarpıklık	Basıklık
GUTÜD	33	,068	-,158
İnşacılık	34	,104	1,741
Doğrudan Öğretim	32	,956	1,402

GUTÜD, inşacılık ve doğrudan öğretim yöntemlerinin öğrencilerin kodlama başarı etkisinin incelendiği bu bölümde GUTÜD ön test puan ortalaması ($\bar{X} = 5,18$) ile Gütüd son test puan ortalaması ($\bar{X} = 10,30$) arasında anlamlı bir farklılığın olduğu ($t_{(32)} = -23,650$, $p = 0.00$) ve bu farkın orta bir etki gücüne sahip olduğu ifade edilebilir. İnşacılık yönteminin öğrencilerin kodlama başarı üzerindeki etkisi incelendiğinde ise ön test puan ortalaması ($\bar{X} = 4,59$) ile son test puan ortalaması ($\bar{X} = 6,85$) arasında anlamlı bir farklılığın olduğu ve ($t_{(33)} = -$

10,880), $p=0,00$) ve bu farkın orta büyük bir etki gücüne sahip olduğu. Doğrudan öğretim yönteminin öğrencilerin kodlama başarısına olan etkisi açısından incelendiğinde öğrencilerin ön test puan ortalaması ($\bar{X} =4,81$) ile son test puan ortalaması ($\bar{X} =8,16$) arasında anlamlı bir farklılığın olduğu ve bu farkın orta büyük bir etki gücüyle temsil edildiği ifade edilebilir.

Tablo 4.36 İlişkili örneklem t-testi

	N	Ort.	SS	Ortalamalar Arası Fark	SS	t	sd	P
GUTÜD Ön	33	5,18	1,33	-5,12	1,24	-23,650	32	,000
GUTÜD Son	33	10,30	1,63					
İnşacılık Ön	34	4,59	1,76	-2,26	1,21	-10,880	33	,000
İnşacılık Son	34	6,85	2,02					
Doğrudan öğretim Ön	32	4,81	2,10	-3,34	2,28	-8,294	31	,000
Doğrudan öğretim Son	32	8,16	2,14					

5. TARTIŞMA

Bu araştırmada, GUTÜD, inşacılık ve doğrudan öğretim modelleri arasında; ortaokul öğrencileri için bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısı, Scratch ortamı için kodlama başarısı, ortaokul öğrencilerinin kodlamaya yönelik tutumu, bilgi işlemsel düşünme durumları karşılaştırmaları yapılmıştır. Bu bölümde elde edilen bulgular tartışılmıştır ve çeşitli sonuçlara ulaşılmıştır.

5.1.1 Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı Ölçeğine Yönelik Tartışma

Yapılan analizler sonucunda bütün gruplarda son test bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanları ön test bilgi işlemsel düşünme becerisine yönelik öz yeterlik puanlarına göre daha yüksek çıkmıştır. Uygulama öncesinde ve sonrasında elde edilen bilgi işlemsel düşünme becerisine yönelik öz yeterlilik puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalamanın GUTÜD grubunda olduğu; ardından inşacılık grubu ve son olarak da doğrudan öğretim grubunun geldiği görülmüştür.

Grupların bilgi işlemsel düşünme becerisine yönelik öz yeterlilik ön test puanları ile düzeltilmiş ortalama bilgi işlemsel düşünme becerisine yönelik öz yeterlilik son test puanlarında anlamlı bir fark çıkmamıştır. Fakat gruplar kendi içlerinde değerlendirildiğinde GUTÜD ve doğrudan öğretim gruplarında ön test ve son test arasında anlamlı bir fark bulunurken inşacılık grubunun ön test son test puanları arasında anlamlı bir fark bulunamamıştır.

Her bir grubun ön test ve son test puanları incelendiğinde, bütün grupların ortalamalarında bir artış gözlemlenmesine rağmen bu artış GUTÜD ve doğrudan anlatım gruplarında anlamlı iken inşacılık grubunda gözlenen farkın anlamlı olmadığı bulunmuştur. Doğrudan öğretim ile yapılan öğretimde öğretmen büyük bir oranda aktif rol üstlenmektedir. GUTÜD ile yapılan öğretimde ise öğretmen konunun temellerini verirken aktif bir rol üstlenir fakat ileri seviye görevlerde öğrenciler aktif bir rol üstlenmektedir ve öğretmen rehber konumundadır. İnşacılık grubunda ise dersin başından sonuna kadar öğrenciler aktif bir rol üstlenirken, öğretmen rehber pozisyonundadır. Öğrencilerin bulunduğu okulda verilen dersler ağırlıklı olarak öğretmenin aktif olduğu doğrudan öğretim yolu ile öğretime daha yakındır. Öğrenciler kendilerinin

dersin başından itibaren aktif bir şekilde öğretimlerinin sorumluluğunu aldığı öğretim ortamlarına görece olarak alışkın değildir. Öğrenciler görece daha alışkın olduğu ortamlarda öğrendiklerinde kendilerini daha verimli ve yeterli hissedebilir (Schunk, 1991). Ya da inşacılık grubundaki öğrenciler 8 haftalık bir süre içerisinde verilen inşacılık gibi görece alışık olmadığı farklı bir ortamda verilen öğretimde beklenen özyeterliliği tam olarak geliştirememiş olabilir. Bu ihtimal dahilinde düşünüldüğünde, öğrencilere daha uzun süreli inşacılık yaklaşımı ile öğretim verilirse öğrenciler beklenen öz yeterlik algısını geliştirebilir.

Bu çalışmada grupları birbirlerine kıyasla değerlendirmek için kovaryans analizi kullanılmıştır. Kovaryans analizinin sonuçlarına göre gruplar arasında bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algıları açısından anlamlı bir fark bulunamamıştır.

Bilgi işlemsel düşünme becerisine yönelik özyeterlilik algısı temelde bir problem çözme sürecine yönelik bir algıdır (Kalelioğlu vd., 2016). Yapılan ölçme işleminde öğrencilerin bilişsel alt boyutlar ile duyuşsal anlamda özgüvenleri de ölçülmektedir (Gülbahar vd., 2019). ISTE ve CSTA'nın (2011) ifadesiyle bilgi işlemsel düşünme becerisine yönelik özyeterlilik algısının önemli bir boyutu da insanların tutum ve eğilimleridir. Bunlar, karmaşıklıkla başa çıkmada güven, zor problemlerle uğraşırken sabır, belirsizlikle başa çıkmada esneklik, açık uçlu problemlerle başa çıkma ve problem çözmede başkalarıyla iletişim kurma ve iş birliği yapma becerisini içermektedir. Bu tarz becerilerin kıyaslamalı olarak gelişimi için 8 haftalık süre yeterli olmayabilir. Daha uzun süreli verilecek eğitimlerde beklenen kıyaslamalı anlamlı farklara ulaşılabilir.

Bir başka faktör ise bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısının, programlama ile doğrudan ilişkili olduğu görülse de bunun sadece bir programlama becerisi olmadığına dikkat çekilmiştir (Lu ve Fletcher, 2009; Qualls ve Sherrell, 2010). Uslu (2018) yaptığı araştırmada ortaokul düzeyinde gerçekleştirilen görsel programlama etkinliklerinin öğrencilerin bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algısının üzerinde etkisinin olup olmadığı incelenmiş ve istatistiksel olarak anlamlı bir değişim meydana gelmediğini bulmuştur.

5.1.2 Bilgi İşlemsel Düşünme Testine Yönelik Tartışma

Yapılan analizler sonucunda bütün gruplarda son test bilgi işlemsel düşünme testi puanları ön test bilgi işlemsel düşünme testi puanlarına göre daha yüksektir. Uygulama öncesinde ve sonrasında elde edilen bilgi işlemsel düşünme testi puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama GUTÜD grubuna, ardından inşacılık grubuna ve son olarak en düşük ortalama doğrudan öğretim grubuna aittir.

GUTÜD grubu bilgi işlemsel düşünme son test ve bilgi işlemsel düşünme ön test puanları arasında anlamlı doğrusal bir ilişki gözlenmiştir. Buna karşın inşacılık grubu bilgi işlemsel düşünme son test ve bilgi işlemsel düşünme ön test puanları arasında anlamlı doğrusal bir ilişki bulunamamıştır. Son olarak doğrudan öğretim grubu bilgi işlemsel düşünme son test ve bilgi işlemsel düşünme ön test puanları arasında da anlamlı doğrusal bir ilişki bulunamamıştır.

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan bilgi işlemsel düşünme testi sonuçlarının ortalamaları arasında anlamlı bir fark olup olmadığı test edildiğinde değişkenler arasında anlamlı bir farklılığın olduğu görülmüştür.

GUTÜD, inşacılık ve doğrudan öğretim yöntemlerinin öğrencilerin bilgi işlemsel düşünme testi etkisinin incelendiğinde GUTÜD ön test puan ortalaması ile Gütüd son test puan ortalaması arasında anlamlı bir farklılığın olduğu belirlenmiştir. İnşacılık yönteminin öğrencilerin bilgi işlemsel düşünme testi üzerindeki etkisi incelendiğinde ise ön test puan ortalaması ile son test puan ortalaması arasında anlamlı bir farklılığın olduğu görülmüştür. Doğrudan öğretim yönteminin öğrencilerin bilgi işlemsel düşünme testine olan etkisi açısından incelendiğinde ise öğrencilerin ön test puan ortalaması ile son test puan ortalaması arasında anlamlı bir farklılığın olduğu ortaya çıkmıştır.

Her bir grubun ön test ve son test puanları incelendiğinde, bütün grupların ortalamalarında bir artış gözlemlenmesine karşın bu artış GUTÜD ve doğrudan öğretim gruplarında anlamlı iken inşacılık grubunda gözlenen farkın anlamlı olmadığı bulunmuştur. CSTA (2016) Bilgi işlemsel düşünme becerileri, karmaşık problemleri daha küçük parçalara ayırmayı, bu parçalar arasındaki ilişkileri ve kalıpları belirlemeyi ve mantıksal ve analitik düşünce kullanarak çözümleri geliştirmeyi içermektedir. Çetin vd. (2021) GUTÜD öğrenme döngüsüne bakıldığında öğrencilerin proje geliştirme sürecinde aktif olarak bulunmasını

sağlamakta ve öğrencilerin öğrenme sürecinde aktif katılımı teşvik edilir ve öğrenciler kendi hatalarını fark ederek ve düzelterek öğrenir. GUTÜD, öğrencilerin üst düzey becerilerini geliştirmelerine yardımcı olabileceği söylenmektedir. GUTÜD ve doğrudan öğretim modellerinde öğrenciler bir öğretici ile konuyu daha küçük parçalara bölerek incelemekte ve alışmış olduğu geleneksel eğitime biraz daha yakın olduğundan konuları daha rahat kavradığı düşünülmektedir. İnşacılık modelinde ise öğrenciler daha az alışkın olduğu ortamda olmakla beraber konuyu anlamada bu açıdan zorluk çekmişlerdir. Fosnot ve Perry (1996) inşacılık modeli öğrenme sürecini yaparak yaşayarak deneyim elde edilmesi süreci şeklinde ifade etmektedir. Bu yöntemin genellikle pratik becerileri ve problem çözme becerilerini geliştirmede yardımcı olmaktadır. Kalelioğlu vd. (2016) ise bilgi işlemsel düşünme becerileri konusunda gerçekleştirdikleri çalışmada, doğru problem çözme, soyutlama, bilgisayar, süreç, bilim, veri, algoritma bağlamlarında ele alındığını belirtmişlerdir. Bilgi işlemsel düşünmenin birden çok boyutu olduğundan dolayı inşacılık modeli bilgi işlemsel boyutların gelişimine tek başına yeterli gelmediği farklı yöntemler ve daha uzun zaman kullanarak daha iyi sonuçlar elde edileceği düşünülmektedir.

Bu çalışmada grupları birbirlerine kıyasla değerlendirmek için kovaryans analizi kullanılmıştır. Kovaryans analizinin sonuçlarına göre gruplar arasında bilgi işlemsel düşünme testine yönelik anlamlı bir fark bulunamamıştır.

León ve Robles (2015) Scratch internet sayfasında paylaşılan ve rastgele seçilmiş olan projeleri inceleyerek Scratch çalışmalarının bilgi işlemsel düşünme ile ilgili kriterleri ölçmede yeterli olmayacağını ifade etmişlerdir. Çalışmada scratch uygulaması haricinde birkaç programlama ortamı kullanılarak planların uygulanması daha iyi sonuçların ortaya çıkması sağlanabilir.

5.1.3 Ortaokul Öğrencilerinin Kodlamaya Yönelik Tutumları Ölçeğine Yönelik Tartışma

Yapılan analizler sonucunda bütün gruplarda son test kodlamaya yönelik tutum testi puanları ön test kodlamaya yönelik tutum testi puanlarına göre daha yüksektir. Uygulama öncesinde ve sonrasında elde edilen kodlamaya yönelik tutum testi puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama GUTÜD grubuna ardından İnşacılık grubuna ve son olarak en düşük ortalama doğrudan öğretim grubuna aittir.

Uygulanan üç farklı öğretim yönteminin öğrencilerin, kodlamaya yönelik tutum ön test puanları kontrol edilerek, son test puanlarına etkisi incelendiğinde GUTÜD grubu kodlamaya yönelik tutum son test ve kodlamaya yönelik tutum ön test puanları arasında anlamlı doğrusal bir ilişki gözlenmiştir. Bununla birlikte inşacılık grubu kodlamaya yönelik tutum son test ve kodlamaya yönelik tutum ön test puanları arasında da anlamlı doğrusal bir ilişki gözlenmiştir. Ve son olarak doğrudan öğretim grubu kodlamaya yönelik tutum son test ve kodlamaya yönelik tutum ön test puanları arasında anlamlı doğrusal bir ilişki bulunamamıştır.

Her bir grubun ön test ve son test puanları incelendiğinde, bütün grupların ortalamalarında bir artış gözlenmesine rağmen bu artış GUTÜD modeli ile doğrudan öğretim modeli arasında anlamlı bir fark olarak çıkmıştır. Aynı zamanda inşacılık modeli ile doğrudan öğretim arasında anlamlı bir fark çıkmıştır. Fakat GUTÜD modeli ile inşacılık modeli arasında anlamlı bir fark bulunamamıştır. Bu çalışmada grupları birbirlerine kıyasla değerlendirmek için kovaryans analizi kullanılmıştır. Kovaryans analizinin sonuçlarına göre öğrencilerin kodlamaya yönelik tutumları açısından anlamlı bir fark bulunmuştur.

Wang vd. (2017) yapmış olduğu çalışmada da benzer sonuçlar bulunmaktadır. GUTÜD ve inşacılık modelindeki gibi öğrencilerin daha aktif olduğu modeller, doğrudan öğretim modeline göre daha etkili olduğu görülmektedir. Hanks (2006) yaptığı çalışmada öğrenci tutumlarını öğretmenlerin etkilediği sonucuna ulaşmıştır. Doğrudan öğretim modelinde diğer modellere göre öğretmen daha aktif olması bu çalışmayı desteklemektedir. Kert ve Uğraş (2009) basit ve kullanımı kolay bir yazılım olan Scratch ortamının kodlama öğretimindeki önemini anlatmışlardır. Bu araştırmaya ek olarak Scratch yazılımı, eğitim ve öğretim süreçleri tartışılmış ve öneriler sunulmuştur. Kodlamanın çocuklara küçük yaşta kazandırılmasının çocukların düşünme becerisini olumlu yönde geliştirdiği, yaşlılarıyla işbirlikli öğrenmelerine ve bilgiyi kendilerinin bulmasına katkıda bulunacağı ifade edilmiştir. Çalışmada, verilecek kodlama eğitiminin öğrencilerin düzeyine göre olması ve eğitim verilecek ortamın eğlenceli bir forma bürünmesi üzerinde vurgu yapmıştır. GUTÜD modeli ve inşacılık modeli doğrudan öğretime göre öğrencilerin daha aktif ve eğlenceli buldukları ortam sağlamak ve bu da öğrenciler üzerindeki tutuma etki etmektedir. Yapılan çalışmada tüm gruplarda ön ve son testler sonucunda tutumda olumlu yönde bir etki olduğu görülmektedir. Fidan (2016), tarafından yapılan

“Scratch ile Programlama Öğretiminde Oyunlaştırmanın Öğrenci Katılımına Etkisi” çalışmada scratch ortamında yapılan uygulamaların öğrencilerin derse katılım ve derse olana motivasyonlarında artış göstermiştir.

5.1.4 Kodlama Başarısı Testine Yönelik Tartışma

Yapılan analizler sonucunda bütün gruplarda son test kodlama başarı puanları ön test kodlama başarı puanlarına göre daha yüksek çıkmıştır. Uygulama öncesinde ve sonrasında elde edilen kodlama başarı puanları incelendiğinde hem ön test hem de son test puanlarında en yüksek ortalama GUTÜD grubuna ardından doğrudan öğretim grubuna ve son olarak en düşük ortalama inşacılık grubuna aittir.

GUTÜD, inşacılık ve doğrudan öğretim gruplarından bulunan öğrencilerin kodlama başarı ön test puanları arasında fark olup olmadığı incelendiğinde en az ikisi arasında istatistiksel olarak anlamlı fark gözlenmiş ve anlamlı farkın GUTÜD, doğrudan öğretim ve inşacılık grupları arasında olduğu görülmüştür.

Uygulanan üç farklı öğretim yönteminin her bir grup içerisinde uygulama öncesinde ve sonrasında yapılan kodlama başarı testi sonuçlarının ortalamaları arasında anlamlı bir fark olup olmadığı incelendiğinde GUTÜD, inşacılık ve doğrudan öğretim yöntemlerinin öğrencilerin kodlama başarı etkisinin incelendiği bu bölümde GUTÜD ön test puan ortalaması ile GUTÜD son test puan ortalaması arasında anlamlı bir farklılığın olduğu edilebilir. İnşacılık yönteminin öğrencilerin kodlama başarı üzerindeki etkisi incelendiğinde ise ön test puan ortalaması ile son test puan ortalaması arasında anlamlı bir farklılığın olduğu görülmüştür. Doğrudan öğretim yönteminin öğrencilerin kodlama başarısına olan etkisi açısından incelendiğinde öğrencilerin ön test puan ortalaması ile son test puan ortalaması arasında anlamlı bir farklılığın olduğu ifade edilebilir.

Yapılan analizler sonucunda GUTÜD, inşacılık ve doğrudan anlatım grupları arasında kodlama başarısına yönelik bakıldığında GUTÜD modelinin doğrudan öğretim modeline göre daha etkili olduğu görülmektedir. Doğrudan öğretim modelinin de inşacılık modeline göre daha etkili olduğu görülmektedir. Bu çalışmada grupları birbirlerine kıyasla değerlendirmek için kovaryans analizi kullanılmıştır. Kovaryans analizinin sonuçlarına göre öğrencilerin kodlamaya yönelik tutumları açısından anlamlı bir fark bulunmuştur.

Yapılan çalışmada GUTÜD grubu Doğrudan anlatım ve İnşacılık grubuna göre başarılı olduğu görülmektedir. Doğrudan öğretim de öğretmen

öğrencilere konuyu anlatır, öğrenciler ise bu konu hakkında ödev yapmaktadırlar (Education Commission of the States, 1999). Doğrudan öğretimde öğretmen sunulan örneklerin oranı ve türü, belirli kavram ve becerileri öğretmek için kullanılan ifadeler, kullanılan materyallerin gözden geçirilmesi, öğrencilerin kullanılan materyal hakkında verdiği yanıtları kontrol etmektedir (Stockard ve Engelmann, 2008). Öğretmen doğrudan öğretimde öğrenciye oranla daha çok rol almaktadır ve öğrencilerin keşfetmesine engel oluşturmaktadır. Denner vd. (2012) öğrencilerin programlama konusunda daha iyi işler çıkarabilmesi için öğrencilerin özgür olması gerektiğini ve yaratıcı düşüncelerinin ortaya çıkması için imkân sağlanmasını söylemektedir. Papert (1999) öğretmenin rolünün, öğrencinin öğrenme deneyimini yönetmesine yardımcı olan bir rehber olduğunu söyler. Öğretmenin çocuğun öğrenmesini kontrol etmesi yerine, öğrencinin kontrolü elinde tutması gerektiğini söyler. İnşacılık modelinde ise öğrencini özgür ve keşfetmeye açık bırakıldığı görülmektedir. Denner vd. (2012) aşırı özgürlük ve esneklik öğrencilerin yaratıcılığını etkileyebileceğini beklenen sonuçların elde edilemeyeceğini vurgulamaktadır. Çetin vd. (2021) GUTÜD döngüsünde iki temel yaklaşım vardır. Bunların ilki öğretmenin dersi anlattığı yöntemdir. Rehber öğretmenin asıl sorumluluğu aldığı, konuyu doğrudan anlatan veya model olarak problemleri nasıl çözüldüğünü gösteren rolde bulunmaktadır. İkinci aşamada ise öğretmen bilgiyi aktarandan ziyade, öğrencinin keşfetme veya oluşturma sürecinde yardımcı olan, öğrencinin aktif olduğu bir ortam hazırlamaktadır. Öğrencilerin temel bilgileri belli bir disipline alınması sağlanarak keşfetme aşamasında öğrencinin kontrolü kaybetmemesini sağlamaktadır. Alan yazında belirtildiği üzere öğrencilerin bir rehber tarafından yönlendirilmesi ve aynı zamanda öğrencilerin kendini aktif bir ortamda bulması GUTÜD grubunun diğer iki grubu göre daha etkili çıkmasına neden olabileceği sonucuna ulaşılmaktadır.

Urquiza-Fuentes ve Velázquez-Iturbide (2007) yaptıkları çalışmada öğrencileri yapılandırma grubu ve izleme grubu olarak iki gruba ayırdılar. İzleyen grup animasyonu sadece izleyerek öğrenirken, yapılandırma grubu animasyonda değişiklik yaparak öğrenir. Araştırma sonucunda grupların etkinlikleri değerlendirildiğinde yapıcı gruplarda yer alan öğrencilerin sonraki gruplarda yer alan öğrencilere göre daha başarılı olduğu görülmektedir. Yurdugül ve Gültekin (2009) öğrencilerin daha aktif olduğu çoklu ortama dayalı

öğretim ile gerçekleştirilen öğretim süreci, alışlagelmiş yöntem ile gerçekleştirilen öğretim sürecine göre, öğrencilerin başarı puanlarının artması yönünde olumlu etki yaptığı göstermiştir. Bu çalışmadan çıkan sonuçla öğrencilerin daha aktif olduğu inşacılık grubunun doğrudan öğretim gurubundan daha etkili olması ile paralellik göstermektedir.



6. SONUÇ VE ÖNERİLER

Araştırmadaki bulgulara göre eğitimcilere ve araştırmacılara yapacakları çalışmalar için birtakım öneriler sunulmuştur.

6.1.1 Araştırmacılar İçin Öneriler

- Bu çalışmada incelenen sorular farklı bağlamlarda incelendiğinde farklı sonuçlar elde edilebilir. Yapılacak çalışmanın sonuçlarının farklı olup olmayacağı test edilebilir.
- Çalışma süresi artırılarak öğrencilere daha fazla etkinlik verilerek çalışmaya katkı sağlanabilir.
- Farklı sınıf seviyelerine uygun planlar hazırlanarak modeller arasındaki farklar değerlendirilebilir.
- Çalışmada yapılan değerlendirmelerin farklı bir programlama dili (Scratch hariç) kullanılarak yapılabilir.
- Yapılan çalışmada GUTÜD, inşacılık grupları arasında bilgi işlemsel düşünme becerisine yönelik yeterlilik algısına bakıldığında herhangi bir fark bulunmamaktadır. Öğrencilerin bilgi işlemsel düşünme becerisine yönelik algılar farklı değişkenler açısından incelenebilir.
- Ortaöğretim ve yüksek öğretim kurumlarında bulunan öğrencilerin üç model kullanılarak araştırma yapılabilir.
- Farklı ölçekler kullanarak üç model hakkında daha fazla araştırma yapılabilir.
- Alan yazına bakıldığında inşacılık modeli ile ilgili birçok araştırma yapıldığı görülmektedir. Ülkemizde ise bu çalışmalar yeterli sayıda bulunmamaktadır. Yapılacak araştırmalar ile inşacılık modeli ile ilgili veriler artırılabilir.

6.1.2 Eğitimciler İçin Öneriler

- Yapılan çalışma sonucunda GUTÜD, inşacılık ve doğrudan anlatım grupları arasında kodlama başarısına yönelik bakıldığında GUTÜD modelinin doğrudan öğretim modeline göre daha etkili olduğu görülmektedir. Doğrudan öğretim modelinin de inşacılık modeline göre daha etkili olduğu görülmektedir. 5 ve 6. Sınıfların bilişim teknolojileri konuları GUTÜD modeline göre düzenlenip öğrencilere kodlama öğretilir.

- Eđitimciler yapılmıř olan bu planları rnek olarak robotik-kodlama ile ilgili tm programlar iin plan geliřtirebilir ve uygulayabilir.
- Eđitimci okulunun ve sınıfının durumuna gre  farklı đretim modelinin herhangi birini seerek uygulayabilir.
- Geliřtirilen planlar biliřim teknolojileri ve yazılım dersi iin retilmiřtir. STEM yaklařımı erevesinde birkaç tane disiplin eklenerek geliřtirilebilir.



7. KAYNAKLAR

Bu tez çalışmasında APA 7th Edition atıf sistemi kullanılmıştır.

- Abdunabi, R., Hbaci, I. & Ku, H.Y. (2019) Towards enhancing programming self-efficacy perceptions among undergraduate information systems students, *Journal of Information Technology Education: Research*, 18, 185–206. <https://doi.org/10.28945/4308>.
- Açıkgöz, K. Ü. (2005). *Etkili öğrenme ve öğretme*. Eğitim Dünyası Yayınları.
- Aho, A. V. (2012). *Computation and computational thinking*. The computer journal, 55(7), 832-835.
- Akelaitis, A., & Lisinskienė, A. (2018). *Social emotional skills and prosocial behaviour among 15-16-year-old adolescent*. European journal of contemporary education, 21-28.
- Akpınar, Y., & Altun, A. (2014). *Bilgi toplumu okullarında programlama eğitimi gereksinimi*. Elementary Education Online.
- Ala-Mutka, K. (2004). *Problems in learning and teaching programming-a literature study for developing visualizations in the Codewitz-Minerva project*. Codewitz needs analysis.
- Allport, G. W. (1933). *1. Attitudes*. Terminology.
- Allport, G. W. Attitudes. In C. A. Murchison (Ed.), *A handbook of social psychology*. Clark Univ. Press,
- Anderson, L. W. (1988). Attitudes and their measurement. Keeves, J. P. (Ed.). *Educational research, methodology and measurement: An international handbook*. Pergamon Press.
- Apostolellis, P., Stewart, M., Frisina, C., & Kafura, D. (2014). *RaBit escApe: A board game for computational thinking*. Paper presented at the Interaction Design and Children Conference, Denmark.
- Arabacıoğlu, C., Bülbül, H., & Filiz, A. (2007). *A new approach to teaching computer programming*. Paper presented at the Academic Computing Conference, the Kutahya, Turkey.
- Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). *From Scratch to "real" programming*. ACM Transactions on Computing Education.
- Artym, C., Carbonaro, M., & Boechler, P. (2017). Evaluating pre-service teachers' computational thinking skills in scratch. *Ubiquitous Learning*, 10(2), 43.
- Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java Programming among engineering students. *Online Submission*, 8(1).
- Aşkar, P., & Umay, A. (2001). İlköğretim matematik öğretmenliği öğrencilerinin bilgisayarla ilgili öz-yeterlik algısı. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 21. 1-8.
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding priorities, school curricula and initiatives across Europe*. European Schoolnet.
- Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding- Priorities, school curricula and initiatives across Europe*.
- Bandura, A. (1977). *Self-efficacy: Toward a unifying theory of behavioral change*. Psychological Review, 84(2)
- Bandura, A. (1986). *Social foundations of thought and action*. Englewood Cliffs, NJ, 1986(23-28).

- Bandura, A. (1994). *Self-efficacy*. In V. S. Ramachaudran (Ed.), *Encyclopedia of human behavior* (Vol. 4., pp. 71-81). Academic Press
- Bandura, A. (1997). *Self efficacy: The exercise of control*. Newyork: Freeman. 25(1), 115-144.
- Bandura, A. (2006). Adolescent development from an agentic perspective. *Self-efficacy beliefs of adolescents*, 5(1-43).
- Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- Baser, M. (2013). Attitude, gender and achievement in computer programming. *Online Submission*, 14(2), 248-255.
- Baser, M., & Geban, Ö. (2007). Effectiveness of conceptual change instruction on understanding of heat and temperature concepts. *Research in science & technological education*, 25(1), 115-133.
- Başer, M. (2013). Bilgisayar programlamaya karşı tutum ölçeği geliştirme çalışması. *The Journal of Academic Social Science Studies*, 6(6).
- Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change*. Addison-Wesley Yayıncılık.
- Beecher, K. (2017, August). *Computational thinking*. BCS.
- Bereiter, C. (1994). Constructivism, socioculturalism, and Popper's world 3. *Educational researcher*, 23(7), 21-23.
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- Bissonnette, S., Richard, M., Gauthier, C. & Bouchard, C. (2010). Quelles sont les stratégies d'enseignement efficaces favorisant les apprentissages fondamentaux auprès des élèves en difficulté de niveau élémentaire? Revue de recherche appliquée sur l'apprentissage? Résultats d'une méga-analyse. *Revue de recherche appliquée sur l'apprentissage*, 3(1), 1-35.
- Bouffard-Bouchard, T. (1990). Influence of self-efficacy on performance in a cognitive task. *The journal of social Psychology*, 130(3), 353-363.
- Bouffard-Bouchard, T., Parent, S., & Larivee, S. (1991). Influence of self-efficacy on self-regulation and performance among junior and senior high-school age students. *International journal of behavioral development*, 14(2), 153-164.
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).
- Brennan, K., & Resnick, M. (2012, April). Using artifact-based interviews to study the development of computational thinking in interactive media design. In *annual American Educational Research Association meeting, Vancouver, BC, Canada* (pp. 1-25).
- Breslin, D. (2017). Learning to evolve: increasing entrepreneurial self-efficacy and putting the market first. In *Entrepreneurship Education*. Emerald Publishing Limited.
- Brophy, J. E. (1984). *Teacher behavior and student achievement* (No. 73). Institute for Research on Teaching, Michigan State University.
- Brophy, J. E., & Evertson, C. M. (1976). *Learning from teaching: A developmental perspective*. Allyn and Bacon.
- Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review*, 31(1), 21-32.

- Buckner, E. S. (2017). The changing discourse on higher education and the nation-state, 1960–2010. *Higher Education*, 74(3), 473-489.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- Büyüköztürk, Ş. (2007). *Sosyal bilimler için veri analizi el kitabı* [Data analysis handbook for social sciences].
- Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2016). *Bilimsel araştırma yöntemleri*. Pegem Akademi.
- Byrne, P., & Lyons, G. (2001, June). The effect of student attributes on success in programming. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education* (pp. 49-52).
- Cevizci, A. (2012). *Eğitim Felsefesi*. Say Yayınları.
- Chein, I., Harding, J., & Proshansky, H. (1954). Prejudice and ethnic relations. *Handbook of Social Psychology G. Lindzey (editor)*, Camiibridge, Mass.: Addison-Wesley, 2, 1022.
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & education*, 109, 162-175.
- CNESCO. (2016). Lire, comprendre, apprendre: comment soutenir le développement des compétences en lecture? Rapport pour la préparation de la conférence de consensus sur la lecture, Lyon: CNESCO-IFE.
- Code.org (2022). *The hour of code is here*. <https://code.org/> adresinden 10 Ağustos 2022 tarihinde alınmıştır.
- Codeweek about (2022). Codeweek hakkında. <https://codeweek.eu/about> adresinden 08 Kasım 2021 tarihinde alınmıştır.
- Codeweekturkiye (2022). Codeweekturkiye hakkında. https://codeweekturkiye.eba.gov.tr/?page_id=24,14/08/2022 adresinden 28 Kasım 2022 tarihinde alınmıştır.
- Common Sense Media, & Rideout, V. (2016). *Zero to eight: Children's media use in America*. Common Sense Media.
- Connolly, J. (1989). Social self-efficacy in adolescence: relations with self-concept, social adjustment, and mental health. *Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement*, 21(3), 258.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking-A guide for teachers.
- CSTA (2016). K-12 Computer Science Standards. <https://csteachers.org/k12standards/interactive/> adresinden 3 Aralık 2022 tarihinde alınmıştır.
- CSTA and ISTE (2011). Computational Thinking in K-12 Education Leadership Toolkit. https://csta.acm.org/Curriculum/sub/Curr_Files/471.11CTLeadership_Toolkit-SP-v_F.pdf adresinden 12 Ağustos 2022 tarihinde alınmıştır.
- Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing computational thinking in the classroom: a framework.
- Curzon, P., Peckham, J., Taylor, H., Settle, A., & Roberts, E. (2009, July). Computational thinking (CT) On weaving it in. In *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education* (pp. 201-202).

- Çatak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: Bir döküman inceleme çalışması. *Journal of Instructional Technologies and Teacher Education*, 4(3).
- Çetin, E. (2016). *Okul öncesi çocukların problem çözme sürecinde teknoloji destekli şematik düzenleyicilerin kullanımına yönelik bir durum çalışması* [Doktora tezi]. Gazi Üniversitesi.
- Çetin, İ., & Berigel, M. (2017). Bilgisayar Bilimi Eğitiminde Kavram ve Kuramlar. *Bilgi İşlemsel Düşünmeden Programlamaya*, 118-124. Pegem.
- Çetin, İ., & Uçar, Z. T. (2017). *Bilgi işlemsel düşünme tanımı ve kapsamı*. Pegem.
- Çıngı, H. (1994). *Örnekleme kuramı*. Hacettepe Üniversitesi Basımevi.
- Dagdilelis, V., Satratzemi, M., & Evangelidis, G. (2004). *Introducing secondary education students to algorithms and programming*. *Education and Information Technologies*, 9(2), 159-173.
- Dasgupta, A., & Purzer, S. (2016). No patterns in pattern recognition: A systematic literature review. In *2016 IEEE Frontiers in Education Conference (FIE)* (pp. 1-3). IEEE.
- Dehghanpour Farashah, A. (2013). The process of impact of entrepreneurship education and training on entrepreneurship perception and intention: Study of educational system of Iran. *Education+ Training*, 55(8/9), 868-885.
- Demirbaş, A. (2019). Kodlama eğitiminin değerlendirilmesi araştırması. https://yegitek.meb.gov.tr/meb_iys_dosyalar/2020_03/26131559_kodlamaegitiminindeYerlendirilmesiarastirmasi.pdf adresinde 4 Eylül 2022 tarihinde alınmıştır.
- Demirer, V., & Nurcan, S. A. K. (2016). *Programming education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar*. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
- Deneyap hakkımızda (2022). Dene yap hakkımızda. Erişim adresi: <https://deneyap.org/tr/kurumsal/deneyap/> adresinden 20 Mayıs 2022 tarihinde ulaşılmıştır.
- Denner, J., Werner, L., & Ortiz, E. (2012). *Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts*. *Computers & Education*, 58(1), 240-249.
- Denning, P. J. (2009). *The profession of IT Beyond computational thinking*. *Communications of the ACM*, 52(6), 28-30.
- Denning, P. J. (2017). *Computational thinking in science*. *The Best Writing on Mathematics*, 67.
- Denning, P.J., & Tedre, M. (2019). *Computational thinking from*. <https://mitpress.mit.edu/books/computational-thinking> adresinden 05 Mayıs 2022 tarihinde alınmıştır.
- DeShon, R. P., & Gillespie, J. Z. (2005). *A motivated action theory account of goal orientation*. *Journal of Applied Psychology*, 90(6), 1096.
- Dewey, J. (1929). *The quest for certainty: A study of the relation of knowledge and action*. *The Journal of Philosophy*, 27(1), 14-25.
- Dewey, J. (1986). *Experience and education*. In *The educational forum* (Vol. 50, No. 3, pp. 241-252). Taylor & Francis Group.
- Di Martino, P., & Zan, R. (2001). *Attitude toward mathematics: some theoretical issues*. In *PME conference* (Vol. 2, pp. 3-351).
- Driscoll, M. P., & Burner, K. J. (2005). *Psychology of learning for instruction*. Florida State University.

- Duman, B. (2007). *Lise öğrencilerinin İngilizceye yönelik öz yeterlik algı puanlarının cinsiyete, alanlara ve farklı düzeylere göre İngilizce başarısını yordama gücü* [Yüksek lisans tezi]. Yıldız Teknik Üniversitesi.
- Dweck, C. S. (1986). Motivational processes affecting longitudinal study. *Research Quarterly for Exercise Learning*, 41, 1040-1048.
- Eisenberg, E., & Selivansky, E. O. (2019). *Adapting Israel's Education system for the challenges of the 21st century*. Jerusalem: The Israeli Democracy Institute. Available online: <https://www.idi.org.il/media/13079/adapting-israel-s-education-system-for-the-challenges-of-the-21st-centur.pdf> adresinden 26 Mayıs 2022 tarihinde ulaşılmıştır.
- Eker, M. (2011). *Algoritmayı anlamak*. Nirvana Yayınları.
- Ekici, G. (2006). Meslek lisesi öğretmenlerinin öğretmen öz-yeterlik inançları üzerine bir araştırma. *Eurasian Journal of Educational Research (EJER)*, (24), 87-96.
- Elliot, A. J. (1999). Achievement goals and approach-avoidance motivation. *Educational Psychologist*, 34, 169-189.
- Elliot, A. J., & Thrash, T. M. (2002). Approach-avoidance motivation in personality: Approach and avoidance temperaments and goals. *Journal of Personality and Social Psychology*, 82(5), 804-818. <https://doi.org/10.1037/0022-3514.82.5.804>
- Elliott, E. S., & Dweck, C. S. (1988). Goals: An approach to motivation and achievement. *Journal of personality and social psychology*, 54(1), 5.
- Emurian, H. H. (2004). A programmed instruction tutoring system for java™: consideration of learning performance and software self-efficacy. *Computers in Human Behavior*, 20(3), 423-459.
- Erdem, E., & Demirel, Ö. (2002). *Program geliştirmede yapılandırmacılık yaklaşımı*. Hacettepe Üniversitesi Eğitim Fakültesi Dergisi, 23, 81-87.
- Erdogan, Y., Aydın, E., & Kabaca, T. (2008). Exploring the psychological predictors of programming achievement. *Journal of Instructional Psychology*, 35(3), 264.
- Ernest, P. (1995). The one and the many. In L. Steffe & J. Gale (Eds.). *Constructivism in education* (pp.459-486). New Jersey: Lawrence Erlbaum Associates, Inc
- Erümit, A. K. (2020). Effects of different teaching approaches on programming skills. *Education and Information Technologies*, 25(2), 1013-1037.
- Eryılmaz, S. (2003). *Algoritma tasarlama ve programlamaya giriş*. Detay Yayıncılık.
- European Schoolnet. (2015). *Computing our future: Computer programming and coding*. http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03 adresinden 12 ocak 2022 tarihinde alınmıştır.
- Falkner, K. (2015). Computational thinking as the 'new literacy': professional development opportunities. *Curriculum & Leadership Journal*, 13(9).
- Fan, C., & Mak, A. S. (1998). Measuring social self-efficacy in a culturally diverse student population. *Social Behavior and Personality: an international journal*, 26(2), 131-144.
- Farkas, D., & Murthy, N. (2005, June). Attitudes Toward Computers, the Introductory Course and Recruiting New Majors: Preliminary Results. In *PPIG* (p. 22).
- Fazio, R. H. (1995). Attitudes as object-evaluation associations: Determinants, consequences, and correlates of attitude accessibility. *Attitude strength: Antecedents and consequences*, 4, 247-282.
- Feldhausen, R., Weese, J. L., & Bean, N. H. (2018). Increasing student self-efficacy in computational thinking via STEM outreach programs. In *Proceedings of the 49th ACM*

technical symposium on computer science education (pp. 302-307).
<https://doi.org/10.1145/3159450.3159593>

Fernández, J. M., Zúñiga, M. E., Rosas, M. V., & Guerrero, R. A. (2018). Experiences in learning problem-solving through computational thinking. *Journal of Computer Science and Technology*, 18(2), 136-142. <https://doi.org/10.24215/16666038.18.e15>

Fidan, A., (2016). *Scratch ile Programlama Öğretiminde Oyunlaştırmanın Öğrenci Katılımına Etkisi* [Yüksek lisans tezi]. Uludağ Üniversitesi.

Fink, D. G. (1966). *Computers and the human mind*. Doubleday.

Fosnot, C. T., & Perry, R. S. (1996). Constructivism: A psychological theory of learning. *Constructivism: Theory, perspectives, and practice*, 2(1), 8-33.

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education*. McGraw-hill.

Fraser, N. (2013). *Blockly*. <https://developers.google.com/blockly/> adresinden 16 Kasım 2022 tarihinde alınmıştır.

Frauenberger, C., & Purgathofer, P. (2019). *Ways of thinking in informatics*. Communications of the ACM.

Frey, C. B., & Osborne, M. A. (2013). *The future of employment: How susceptible are jobs to computerization?*. Oxford Martin School.

Futschek, G. (2006). Algorithmic thinking: the key for understanding computer science. In *Informatics Education–The Bridge between Using and Understanding Computers: International Conference in Informatics in Secondary Schools–Evolution and Perspectives, ISSEP 2006, Vilnius, Lithuania, November 7-11, 2006. Proceedings* (pp. 159-168). Springer Berlin Heidelberg.

Gauthier, C., Bissonnette, S., & Richard, M. (2009). *Passer du paradigme de l'enseignement au paradigme de l'apprentissage. Les effets néfastes d'un slogan*. Actes de la Recherche, (7), 239-271.

Gauthier, C., Bissonnette, S., Richard, M., & Castonguay, M. (2013). *Enseignement explicite et réussite des élèves: la gestion des apprentissages*. Pearson.

Gaudio, B. A., & Herbert, J. D. (2003). Preliminary psychometric evaluation of a new self-efficacy scale and its relationship to treatment outcome in social anxiety disorder. *Cognitive Therapy and Research*, 27, 537-555.

Genç, Z. & Karakuş, S. (2011). *Tasarımla öğrenme: Eğitsel bilgisayar oyunları tasarımı Scratch kullanımı*. 5. International Computer & Instructional Technologies Symposium, Fırat Üniversitesi, Elâzığ.

Gist, M. E., Schwoerer, C., & Rosen, B. (1989). Effects of alternative training methods on self-efficacy and performance in computer software training. *Journal of applied psychology*, 74(6), 884.

Glaserfeld, E. V. (1995). A constructivist approach to teaching. P. Steffe and J. Gale (Eds), *Constructivism in Education*, (3-15).

Gomes, A., & Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 7).

Gore Jr, P. A. (2006). *Academic self-efficacy as a predictor of college outcomes: Two incremental validity studies*. Journal of career assessment.

Gouws, L. A., Bradshaw, K., & Wentworth, P. (2013, July). Computational thinking in educational activities: an evaluation of the educational game light-bot. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 10-15).

Grossman, P. (2009). *Research on pedagogical approaches in teacher education*. In *Studying teacher education*. Routledge.

- Grover, S., & Pea, R. (2013). *Computational thinking in K-12: A review of the state of the field*. Educational researcher.
- Güneş, F. (2022). *The Direct Instruction Model*. The Journal of Limitless Education and Research.
- Guzdial, M. (2008). *Education paving the way for computational thinking*. Communications of the ACM.
- Gülbahar, Y., & Kalelioğlu, F. (2018). *Bilişim Teknolojileri ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci*. Millî Eğitim Dergisi.
- Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 10(1), 1-29. <https://doi.org/10.16949/turkbilmat.385097>
- Güler, Ç. (2021). Algorithmic thinking skills without computers for prospective computer science teachers. *Journal of Theoretical Educational Science*, 14(4), 570-585. <https://doi.org/10.30831/akukey.892869>
- Güney, S. (2008). *Davranış Bilimleri*. Nobel Yayın Dağıtım.
- Günüş, S., Odabaşı, H., & Kuzu, A. (2013). 21. yüzyıl öğrenci özelliklerinin öğretmen adayları tarafından tanımlanması: bir twitter uygulaması/the defining characteristics of students of the 21st century by student teachers: a twitter activity. *Eğitimde Kuram ve Uygulama*, 9(4), 436-455.
- Güzel, R. (1998). *Alt özel sınıflardaki öğrencilerin sesli okudukları öyküyü anlama becerilerini kazanmalarında doğrudan öğretim yöntemiyle sunulan bireyselleştirilmiş okuduğunu anlama materyalinin etkililiği* [Yayınlanmamış doktora tezi]. Gazi Üniversitesi.
- Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive learning environments*, 1(1), 1-32. <https://doi.org/10.1080/1049482900010102>.
- Harvey, B., & Mönig, J. (2010). Bringing “no ceiling” to scratch: Can one language serve kids and computer scientists. *Proc. Constructionism*, 1-10.
- Heersink, D., & Moskal, B. M. (2010, March). Measuring high school students' attitudes toward computing. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 446-450). <https://doi.org/10.1145/1734263.1734413>
- Higgins, E. T. (1996). Ideals, oughts, and regulatory focus: Affect and motivation from distinct pains and pleasures.
- Honebein, P. C. (1996). Seven goals for the design of constructivist learning environments. *Constructivist learning environments: Case studies in instructional design*, 11(12), 11.
- Hoyle, D. C., & Noss, R. (2017). Visions for mathematical learning: the inspirational legacy of Seymour Papert (1928–2016). *European Mathematical Society Magazine*, (103), 34-36.
- Insko, C. A., & Schopler, J. (1967). Triadic Consistency: A statement of affective-cognitive-conative consistency. *Psychological review*, 74(5), 361. <https://doi.org/10.1037/h0020278>
- Isbell, C. L., Stein, L. A., Cutler, R., Forbes, J., Fraser, L., Impagliazzo, J., ... & Xu, Y. (2010). (Re) defining computing curricula by (re) defining computing. *Acm Sigcse Bulletin*, 41(4), 195-207. <https://doi.org/10.1145/1709424.1709462>.
- International Society for Technology in Education [ISTE] (2016). ISTE Standards For Students. <https://www.iste.org/standards/iste-standards-for-students> adresinden 12 Temmuz 2022 tarihinde alınmıştır.
- İnceoğlu, M. (2010). *Tutum algı iletişim*. Beykent Üniversitesi Yayınları.

- İsen, G. ve Batmaz, V. (2006). *Ben ve toplum*. Salyangoz Yayınları.
- Jonassen, D. H. (1991). Objectivism versus constructivism: Do we need a new philosophical paradigm. *Educational technology research and development*, 39, 5-14.
- Jonassen, D. (1994). *Thinking technology*. Educational Technology.
- Jonassen, D. H. (1994). *Toward a Constructivist Design Model*. Educational Technology.
- Kağıtçıbaşı, Ç. (2010). *Günümüzde İnsan ve İnsanlar*. Evrim Yayınevi.
- Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). *A framework for computational thinking based on a systematic research review*. *Baltic Journal of Modern Computing*.
- Kalelioglu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in education*, 13(1), 33-50.
- Kanfer, R., & Ackerman, P. L. (1989). Motivation and cognitive abilities: An integrative/aptitude-treatment interaction approach to skill acquisition. *Journal of applied psychology*, 74(4), 657. <https://doi.org/10.1037/0021-9010.74.4.657>
- Kanlı, U. (2010). *Yapılandırmacı Kuramın Işığında Öğrenme Halkası'nın Kökleri ve Evrimi-Örnek Bir Etkinlik*. Eğitim ve Bilim.
- Kaptan, F. & Korkmaz, H. (2002). *Probleme Dayalı Öğrenme Yaklaşımının Hizmet Öncesi Fen Öğretmenlerinin Problem Çözme Becerileri ve Öz Yeterlik İnanç Düzeylerine Etkisi*. V. Ulusal Fen Bilimleri ve Matematik Eğitimi Kongresi Bildiri Kitabı, 68-94.
- Karabak, D., & Güneş, A. (2013). *Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi*. Eğitim ve Öğretim Araştırmaları Dergisi, 21(2-3), 163-169.
- Karataş, H. (2021). 21. Yy. Becerilerinden Robotik ve Kodlama Eğitiminin Türkiye ve Dünyadaki Yeri. 21. Yüzyılda Eğitim ve Toplum Eğitim Bilimleri ve Sosyal Araştırmalar Dergisi, 10(30), 693-729.
- Katira, N., Williams, L., Wiebe, E., Miller, C., Balik, S., & Gehringer, E. (2004, March). On understanding compatibility of student pair programmers. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education* (pp. 7-11). <https://doi.org/10.1145/971300.971307>
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM computing surveys (CSUR)*, 37(2), 83-137. <https://doi.org/10.1145/1089733.1089734>
- Kerlinger, F. N., Lee, H. B., & Bhanthumnavin, D. (2000). Foundations of behavioral research: The most sustainable popular textbook by Kerlinger & Lee (2000). *Journal of Social Development*, 13, 131-144.
- Kert, S. B., & Uğraş, T. (2009, May). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. In *The First International Congress of Educational Research, Çanakkale, Turkey*.
- Kinshuk, Huang, H. W., Sampson, D., & Chen, N. S. (2013). Trends in educational technology through the lens of the highly cited articles published in the journal of educational technology and society. *Journal of Educational Technology & Society*, 16(2), 3-20.
- Knuth, D. E. (2005, June). Algorithms in modern mathematics and computer science. In *Algorithms in Modern Mathematics and Computer Science: Proceedings, Urgench, Uzbek SSR September 16-22, 1979* (pp. 82-99). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Konuk, N. E. (2011). *Müzik öğretmeni adaylarının öğretmenliğe ilişkin Tutum ve Mesleki Yeterlik algı düzeylerinin farklı değişkenlere göre İncelenmesi* [Doktora tezi]. Marmara Üniversitesi.

- Korkmaz, Ö., Çakır, R., Özden, Y.M., Oluk, A. & Sarıoğlu, S., (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, 34(2), 68-87.
- Koupritzioti, D., & Xinogalos, S. (2020). PyDiophantus maze game: *Play it to learn mathematics or implement it to learn game programming in Python*. *Education and Information Technologies*, 25(4), 2747-2764.
- Kurbanoglu, N. İ., & Takunyacı, M. (2012). *Lise öğrencilerinin matematik dersine yönelik kaygı, tutum ve öz-yeterlik inançlarının cinsiyet, okul türü ve sınıf düzeyi açısından incelenmesi*. *Uluslararası İnsan Bilimleri Dergisi*, 9(1), 110-130.
- Kurbanoglu, S. S. (2004). *Öz-yeterlik inancı ve bilgi profesyonelleri için önemi*. *Bilgi Dünyası*, 5(2), 137-152.
- Lai, A. F., & Yang, S. M. (2011). *The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities*. In 2011 International Conference on Electrical and Control Engineering (pp. 6940-6944). IEEE.
- Land, S. M. (2000). *Cognitive requirements for learning with open-ended learning environments*. *Educational Technology Research and Development*, 48(3), 61-78.
- Leigh, A. and Mead, S. 2005, *Lifting Teacher Performance*, Policy Report, Progressive Policy Institute, Washington, DC .
- Lent, R. W., Brown, S. D., & Hackett, G. (2002). *Social cognitive career theory*. *Career choice and development*, 4(1), 255-311.
- Linnenbrink, E. A., & Pintrich, P. R. (2003). *The role of self-efficacy beliefs instudent engagement and learning inthe classroom*. *Reading & Writing Quarterly*, 19(2), 119-137.
- Lu, J. J., & Fletcher, G. H. (2009). *Thinking about computational thinking*. In *Proceedings of the 40th ACM technical symposium on Computer science education* (pp. 260-264).
- Luo, F., Antonenko, P. D., & Davis, E. C. (2020). *Exploring the evolution of two girls' conceptions and practices in computational thinking in science*. *Computers & Education*, 146, 103-127.
- Maio, G., & Haddock, G. (2009). *The psychology of attitudes and attitude change*. *Publications Limited*, 45- 78.
- Malinauskas, R., Sniras, S., & Malinauskiene, V. (2018). *Social self-efficacy training programme for basketball-playing students: a case study*. *Revista de psicología del deporte*, 27(1), 165-185.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004, January). *Scratch: a sneak preview [education]*. In *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing, 2004*. (pp. 104-109). IEEE.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). *The Scratch programming language and environment*. *ACM Transactions on Computing Education*, 10(4), 1-16.
- Manela, M. (2017). *Research: 65% of the jobs that Z Generation will engage in do not yet exist*. *Calcalist* 128-154.
- Marín, B., Frez, J., Cruz-Lemus, J., & Genero, M. (2018). *An empirical investigation on the benefits of gamification in programming courses*. *ACM Transactions on Computing Education (TOCE)*, 19(1), 4. <https://doi.org/10.1145/3231709>
- Marty, J.-C., Heraud, J.-M., Carron, T., & France, L. (2007). *Matching the Performed Activity on an Educational Platform with a Recommended Pedagogical Scenario: A Multi-Source Approach*. *Journal of Interactive Learning Research*, 18(2), 267.

- McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simone, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. *Computer Science Education*, 18(2), 67- 92. <https://doi.org/10.1080/08993400802114581>
- McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8), 90–95.
- McGuire, W. J. (1966). *The current status of cognitive consistency theories*. Cognitive consistency: Motivational antecedents and behavioral consequents, 1-26.
- McLeod, D. B. (1992). *Research on affect in mathematics education: A reconceptualization*. Handbook of research on mathematics teaching and learning, 1, 575-596.
- Millî Eğitim Bakanlığı [MEB] (2012). *Programlar*. <https://mufredat.meb.gov.tr/Programlar.aspx> adresinden 30 Eylül 2022 tarihinde alınmıştır.
- Millî Eğitim Bakanlığı Talim ve Terbiye Kurulu Başkanlığı [MEB-TTKB] (2015). *Bilişim Teknolojileri ve Yazılım Dersi Programı*, Ankara: Talim Terbiye Kurulu Başkanlığı.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2011, June). Habits of programming in scratch. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 168-172). <https://doi.org/10.1145/1999747.1999796>
- Meger, J. P., & Allen, N. J. (1991). A three-component conceptualization of organizational commitment. *Human resource management review*, 1(1), 61-89. [https://doi.org/10.1016/1053-4822\(91\)90011-Z](https://doi.org/10.1016/1053-4822(91)90011-Z)
- Menggo, S., Suastra, I., Budiarsa, M., & Padmadewi, N. N. (2019). Needs Analysis of Academic-English Speaking Material in Promoting 21st Century Skills. *International Journal of Instruction*, 12(2), 739-754.
- Michael, A., & Omoloye, A. (2014). Improving structural designs with computer programming in building construction. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 16(3), 10-16.
- Millburg, S. N. (2009). *The effects of environmental risk factors on at-risk urban high school students' academic self-efficacy* [Yayımlanmamış Doktora Tezi]. University of Cincinnati.
- Mina, M., Cowan, J., & Heywood, J. (2015, October). Case for reflection in engineering education-and an alternative. In *2015 IEEE frontiers in education conference (FIE)* (pp. 1-6). IEEE.
- Moos, D. C., & Azevedo, R. (2009). Learning with computer-based learning environments: A literature review of computer self-efficacy. *Review of educational research*, 79(2), 576-600 <https://doi.org/10.3102/0034654308326083>.
- Morgan, W. R. (1995). "Critical Thinking"-What Does that Mean? Searching for a Definition of a Crucial Intellectual Process. *Journal of College Science*
- Mouly, G. J. (1960). *Psychology of effective teaching*. Holt, Rinehart, and Winston.
- Murphy, E. (1997). Constructivism: From Philosophy to Practice. Retrieved from <http://www.cdli.ca/~elmurphy/emurphy/cle.html> mufredat önerisi.” *Journal of Research in Education and Teaching*, 2(3), 175-181 (2013).
- Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. *ACM Sigcse Bulletin*, 35(1), 359-362. <https://doi.org/10.1145/792548.612006>
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.

- Niculescu, R., & Plummer, R. (2003). A pair programming experiment in a large computing course. *Romanian Journal of Information Science and Technology*, 6(1-2), 199-216.
- Nightingale, D., & Cromby, J. (1999). *Social constructionist psychology: A critical analysis of theory and practice*. McGraw-Hill Education (UK).
- Nisbet, S. (2006, May). Mathematics without attitude. In *Keynote address to the Annual Conference of the Queensland Association of Mathematics Teachers, Brisbane*.
- Oliver, K. M. (2000). Methods for developing constructivist learning on the web. *Educational technology*, 40(6), 5-18.
- Organisation for Economic Cooperation and Development [OECD]. (2020). *Education policy outlook Turkey*. <https://www.oecd.org/education/policy-outlook/country-profile-Turkey2020.pdf> adresin 14 Şubat 2022 tarihinde ulaşılmıştır.
- Organisation for Economic Cooperation and Development [OECD]. (2005). The definition and selection of key competencies. Executive summary. Author.
- Owen, K. B. (2006). *Narrative comprehension and enjoyment of feature films: An experimental study*. University of Southern California.
- Özden, (2015). *Computational thinking*. <http://myozden.blogspot.com.tr/2015/06/computational-thinking-bilgisayarca.html> adresinden 21 Kasım 2022 tarihinde alınmıştır.
- Özçınar, H., & Öztürk, E. (2018). Hesaplamalı Düşünmenin Öğretimine İlişkin Özyeterlik Algısı Ölçeği: Geçerlik ve Güvenirlik çalışması. *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, (30), 173-195.
- Özden, Y. (2003). *Öğrenme ve Öğretme*. Pegem A Yayıncılık.
- Özmen, B., & Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles. *Turkish Online Journal of Qualitative Inquiry*, 5(3), 9- 29. <https://doi.org/10.17569/tojq.20328>
- Özmen, H. (2004). Fen öğretiminde öğrenme teorileri ve teknoloji destekli yapılandırmacı (constructivist) öğrenme. *The Turkish Online Journal of Educational Technology – TOJET*, 3 (1), 100-111.
- Pajares, F., & Miller, M. D. (1994). Role of Self-Efficacy and Self-Concept Beliefs in Mathematical Problem Solving: A Path Analysis. *Journal of Educational Psychology*, 86(2), 193-203. <https://doi.org/10.1037/0022-0663.86.2.193>
- Palmaru, R. (2016). Constructivism as a key towards further understanding of communication, culture and society. *Constructivist Foundations* 12 (1), 30-38.
- Palmer, D. H. (2006). *Sources of self-efficacy in a science methods course for primary teacher education students*. Research in Science Education, 36, 337–353.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, 23-29.
- Papert, S. (1985). *Computer Criticism vs. Technocentric Thinking. Logo 85 Theoretical Papers*. MIT, 22-27.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. Paperback, 3-8.
- Papert, S. (1999). *Eight big ideas behind the constructionist learning lab*. Constructive technology as the key to entering the community of learners, 4-5.
- Papert, S. (1999). Logo Philosophy and Implementation. Lcsi, 43-62.
- Papert, S., & Harel, I. (1991). *Situating constructionism. Constructionism*. MIT, 1-11.

- Papert, S., & Harel, I. (Eds.). (1991). *Constructionism: research reports and essays, 1985-1990*. Ablex publishing Corporation, 23-29.
- Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children. *Computers in Human Behavior*, 105, 105849. <https://doi.org/10.1016/j.chb.2018.12.027>
- Phillips, D.C. (1995). The good, the bad, and the ugly: The many faces of constructivism. *Educational Researcher*, 24 (7), 5-12. <https://doi.org/10.3102/0013189X024007005>
- Piaget, J. (1980). The psychogenesis of knowledge and its epistemological significance. In M. Piattelli-Palmarini (Ed.), *Language and learning* (pp. 23-34). Cambridge, MA: Harvard University Press.
- Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of educational psychology*, 82(1), 33. <https://doi.org/10.1037/0022-0663.82.1.33>
- Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, 25(5), 66-71.
- Resnick, M., Kafai, Y., Maeda, J., Rusk, N., & Maloney, J. (2003). A networked, media-rich programming environment to enhance technological fluency at after-school centers in economically-disadvantaged communities. *Proposal to National Science Foundation*.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., & Silverman, B. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
- Rich, P.T., Egan, G., & Ellsworth, J. (2019). *A Framework for Decomposition in Computational Thinking*. In ITICSE '19: Proceedings of the 2019 Conference on Innovation and Technology in Computer Science Education (pp. 416-421). Association for Computing Machinery. <https://doi.org/10.1145/3304221.3319793>
- Román-González, M., Perez-González, J. C. & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, (72), 678–691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Rosenberg, M. J. (1960). Cognitive, affective, and behavioral components of attitudes. *Attitude organization and change*, 1-14.
- Rosenshine, B. (2010). Principes d'enseignement. *Série pratiques éducatives*, 21, 1-31.
- Ruffell, M., Mason, J., & Allen, B. (1998). Studying attitude to mathematics. *Educational Studies in Mathematics*, 35, 1-18.
- Ryder, R. J., Burton, J. L., & Silberg, A. (2006). Longitudinal study of direct instruction effects from first through third grades. *The Journal of Educational Research*, 99(3), 179-192. <https://doi.org/10.3200/JOER.99.3.179-192>
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Saygılı Yıldırım, T. (2020). *Robotik Kodlama Öğretiminde Probleme Dayalı Öğrenme Yaklaşımının Başarı, Pozitif Duygu ve Bilgi İşlemsel Düşünmeye Etkisi* [Yüksek lisans tezi]. Onsekiz Mart Üniversitesi.

- Sayın, Z., & Seferoğlu S. S. (2016), *Yeni Bir 21. Yüzyıl Becerisi Olarak Kodlama Eğitimi ve Kodlamanın Eğitim Politikalarına Etkisi*, Akademik Bilişim Konferansı 2016, 3-5 Şubat, Aydın, 1-13.
- Scherer, R., Siddiq, F., & Viveros, B. S. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in Human Behavior*, 109, 106349. <https://doi.org/10.1016/j.chb.2020.106349>
- Schug M.C., Tarver, S.G. ve Western R.D. (2001). Direct Instruction and Teaching of Early Reading, *Policy Research Institute*, 14(2), 5-21.
- Schunk, D. H. (1987). Peer models and children's behavioral change. *Review of Educational Research*, 57(2), 149-174. <https://doi.org/10.3102/00346543057002149>
- Schunk, D. H. (1991). Self-efficacy and academic motivation. *Educational psychologist*, 26(3-4), 207-231. <https://doi.org/10.1080/00461520.1991.9653133>
- Schunk, D. H. (1995). Self-efficacy, motivation, and performance. *Journal of applied sport psychology*, 7(2), 112-137. <https://doi.org/10.1080/10413209508406961>
- Schunk, D. H. (2003). Self-efficacy for reading and writing: Influence of modeling, goal setting, and self-evaluation. *Reading & Writing Quarterly*, 19(2), 159-172. <https://doi.org/10.1080/10573560308219>
- Schunk, D. H., & Pajares, F. (2002). The development of academic self-efficacy. In *Development of achievement motivation* (pp. 15-31). Academic Press. <https://doi.org/10.1016/B978-012750053-9/50003-6>
- Scratch About. (2022). Scratch Hakkında. <https://scratch.mit.edu/about/> adresinden 20 Aralık 2023 tarihinde alınmıştır.
- Scratch educators (2022). Eğitimciler İçin Kaynaklar. <https://scratch.mit.edu/educators> adresinden 18 Mayıs 2022 tarihinde alınmıştır.
- Scratch foundation (2022). *Our Story*. <https://www.scratchfoundation.org/our-story> adresinden 12 Mart 2022 tarihinde alınmıştır.
- Selby, C. C. (2015, November). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In *Proceedings of the workshop in primary and secondary computing education* (pp. 80-87). <https://doi.org/10.1145/2818314.2818315>
- Selwyn, N. (2016). *Education and technology: Key issues and debates*. Bloomsbury Publishing.
- Semerçi, Ç., & Batdı, V. (2015). A meta-analysis of constructivist learning approach on learners' academic achievements, retention and attitudes. *Journal of Education and Training Studies*, 3 (2), 171-180. <https://doi.org/10.11114/jets.v3i2.644>
- Senemoğlu, N. (2003). *Gelişim Öğrenme ve Öğretim: Kuramdan uygulamaya*. Gazi kitapevi
- Seymour Papert and MIT Epistemology & Learning Research Group. (1986). Constructionism: A New Opportunity for Elementary Science Education. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group. <https://books.google.com.sg/books>
- Shell, D. F., Hazley, M. P., Soh, L. K., Miller, L. D., Chiriacescu, V., & Ingraham, E. (2014, October). Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1-7). IEEE.
- Sherif, M. & Cantrel, H. (1965). The psychology of attitudes. *Psychological Review*, 52, 301-313.
- Shinnar, R. S., Hsu, D. K., & Powell, B. C. (2014). Self-efficacy, Entrepreneurial Intentions, and Gender: Assessing the Impact of Entrepreneurship Education Longitudinally. *The International Journal of Management Education*, 1-10. <https://doi.org/10.1016/j.ijme.2014.09.005>

- Smith, P., Smith, J. U., Powlson, D. S., McGill, W. B., Arah, J. R. M., Chertov, O. G., ... & Whitmore, A. P. (1997). A comparison of the performance of nine soil organic matter models using datasets from seven long-term experiments. *Geoderma*, 81(1-2), 153-225. [https://doi.org/10.1016/S0016-7061\(97\)00087-6](https://doi.org/10.1016/S0016-7061(97)00087-6)
- Sönmez, V. (2011). *Eğitim Felsefesi*. Anı Yayıncılık.
- Stamatios, P. (2022). Can preschoolers learn computational thinking and coding skills with ScratchJr? A systematic literature review. *International Journal of Educational Reform*, 3(4), 16-19. <https://doi.org/10.1177/10567879221076077>
- Taber, K. S. (2006). Beyond constructivism: The progressive research programme into learning science. *Studies in Science Education* 42, 125–184. <https://doi.org/10.1080/03057260608560222>
- Tai, D.W.S., Yu, C.H. Laive, L.C. & Lin, S.J. (2003). A study on the effects of spatialability in promoting the logical thinking abilities of students with regard to programming language. *World Transactions on Engineering and Technology Education*, 2(2), 251-254.
- Talim Terbiye Kurulu Başkanlığı [TTKB] (2012), *Bilişim Teknolojileri ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı*, <http://ttkb.meb.gov.tr/> sayfasından 12 Eylül 2022 tarihinde alınmıştır.
- Tam, M. (2000). Constructivism, instructional design, and technology: Implications for transforming distance learning. *Journal of Educational Technology & Society*, 3(2), 50-60.
- Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences*, 8, 561-570. <https://doi.org/10.1016/j.sbspro.2010.12.078>
- Teknofest (2022). *Sıkça sorulan sorular*. <https://www.teknofest.org/tr/corporate/faq/> adresinden 25 Mart 2023 tarihinde alınmıştır.
- Temel Eğitim Genel Müdürlüğü [TEGM] (2017). *Ortaokul 5. Sınıflarda Yabancı Dil Ağırlıklı Eğitim Uygulaması*. <https://tegm.meb.gov.tr/www/ortaokul-5-siniflarda-yabanci-dil-agirlikli-egitim%20uygulamasi/icerik/471> adresinden 10 Ocak 2023 tarihinde alınmıştır.
- Tepgeç, M. (2017). *Algoritma öğretiminde çözümlü örnek kullanımının öğrenci başarısına ve bilişsel yüke etkileri* (Yüksek lisans tezi). Hacettepe Üniversitesi.
- Tezbaşaran, A. A. (1996). *Likert tipi ölçek geliştirme kılavuzu*. Türk Psikologlar Derneği.
- Thomas, K. M., & Mathieu, J. E. (1994). Role of causal attributions in dynamic self-regulation and goal processes. *Journal of Applied Psychology*, 79(6), 812. <https://doi.org/10.1037/0021-9010.79.6.812>
- Tolli, A. P., & Schmidt, A. M. (2008). The role of feedback, causal attributions, and self-efficacy in goal revision. *Journal of Applied Psychology*, 93(3), 692. <https://doi.org/10.1037/0021-9010.93.3.692>
- Triandis, H. C. (1971). *Attitude and Attitude Change*. New York: John Willey & Sons. Inc Link.
- Tyng, K. S., Zaman, H. B., & Ahmad, A. (2011). Visual application in multi-touch tabletop for mathematics learning: A preliminary study. In *Visual Informatics: Sustaining Research and Innovations: Second International Visual Informatics Conference, IVIC 2011, Selangor, Malaysia, November 9-11, 2011, Proceedings, Part I 2* (pp. 319-328). Springer Berlin Heidelberg.
- Umapathy, K., & Ritzhaupt, A. D. (2017). A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education (TOCE)*, 17(4), 1-13. <https://doi.org/10.1145/2996201>

- Unal, A., & Topu, F. B. (2021). Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students. *Education and Information Technologies*, 26(5), 5291-5309. <https://doi.org/10.1007/s10639-021-10536-w>
- Ural, G., & Bümen, N. (2016). *Türkiye’de fen ve teknoloji öğretiminde yapılandırmacılığın öğretimsel uygulamaları üzerine bir meta-analiz*. Eğitim ve Bilim, 41 (185), 51-82.
- Urquiza-Fuentes, J., & Velázquez-Iturbide, J. A. (2007). An evaluation of the effortless approach to build algorithm animations with WinHIPE. *Electronic Notes in Theoretical Computer Science*, 178, 3-13. <https://doi.org/10.1016/j.entcs.2007.01.038>
- Usher, E. L., & Pajares, F. (2008). Sources of Self-Efficacy in School: Critical Review of the Literature and Future Directions. *Review of Educational Research*, 78(4), 751-796. <https://doi.org/10.3102/0034654308321456>
- Üçgül, M., Çetin, İ. Yükseltürk, E. & Top, E. (2021). *Dene yap teknoloji atölyeleri robotik ve kodlama*. TUBİTAK Yayınları,1-25.
- Üçgül, M., Çetin, İ., Yükseltürk, E., & Top, E. (2021). *Deneyap Türkiye Robotik ve Kodlama*. TUBİTAK Yayınları,1-13.
- Açıkgöz, K. Ü. (2014). *Aktif öğrenme*. Biliş Yayınları,123-136.
- Valente, J. A. (1993). Por que o computador na educação. *Computadores e Conhecimento: repensando a educação*. Campinas: Gráfica da UNICAMP, 24-44.
- Van Dinther, M., Dochy, F., & Segers, M. (2011). Factors affecting students’ self-efficacy in higher education. *Educational research review*, 6(2), 95-108. <https://doi.org/10.1016/j.edurev.2010.10.003>
- Vancouver, J. B., Thompson, C. M. & Williams, A. A. (2001), ‘The changing signs in the relationships among self-efficacy, personal goals, and performance’, *Journal of Applied Psychology*, 86 (4), 605–20. <https://doi.org/10.1037/0021-9010.86.4.605>
- VandeWalle, D. (2001). Goal orientation: Why wanting to look successful doesn’t always lead to success. *Organizational Dynamics*, 30(2), 162-171.
- Vasquez, J. A. (2013). STEM Lesson Essentials, Grades 3-8: integrating Science, Technology, Engineering, and Mathematics. *Teacher Education and Practice*, 26(2), 358-364.
- Volles, N. (2014). Lifelong learning in the EU: changing conceptualisations, actors, and policies. *Studies in Higher Education* 41 (2), 343-363. <https://doi.org/10.1080/03075079.2014.927852>
- Vygotsky, L. S. (1962). *Thought and language*. Cambridge. MIT Press
- Vygotsky, L. S., & Cole, M. (1978). *Mind in society: Development of higher psychological processes*. Harvard university press,123-128.
- Wang, X. M., Hwang, G. J., Liang, Z. Y., & Wang, H. Y. (2017). *Enhancing students’ computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt*. Journal of Educational Technology & Society, 20(4), 58-68.
- Weintrop, D., & Wilensky, U. (2012, August). RoboBuilder: A program-to-play constructionist video game. In Proceedings of the constructionism 2012 conference. Athens, Greece.
- Wing, J. (2017). Computational thinking’s influence on research and education for all. Italian Journal of Educational Technology, 25(2), 7-14.
- Wilson, B., & Cole, P. (1991). *A review of cognitive teaching models*. Educational Technology Research and Development, 39, 47-64.

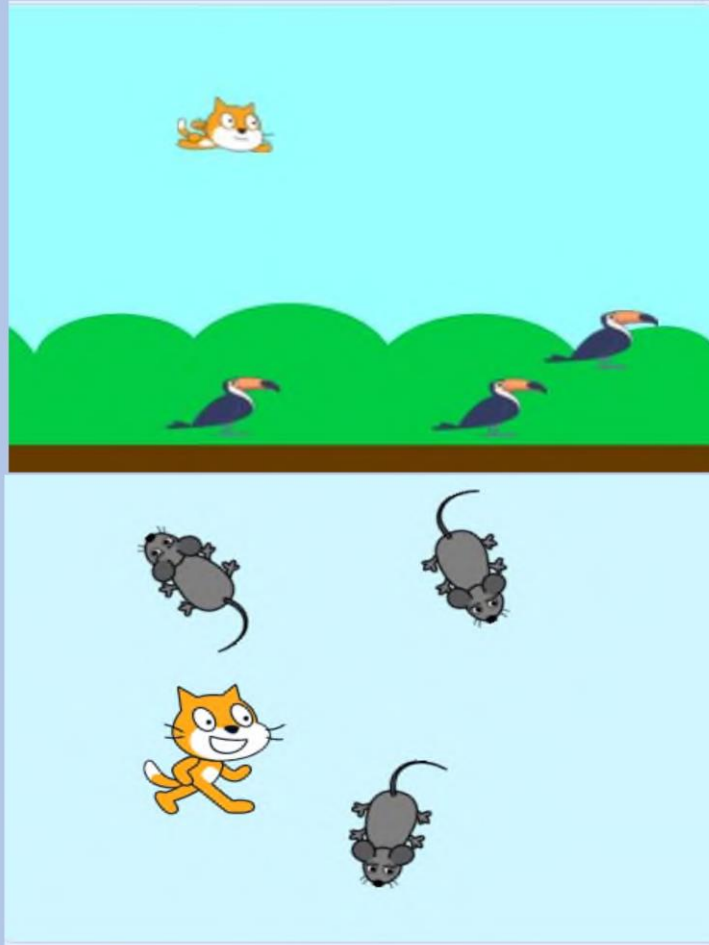
- Williams, L., Layman, L., Osborne, J., & Katira, N. (2006, July). Examining the compatibility of student pair programmers. In *AGILE 2006 (AGILE'06)* (pp. 10-pp). IEEE.
- Williams, L. A., & Kessler, R. R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, 43(5), 108-114. <https://doi.org/10.1145/332833.332848>
- Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, 12(3), 197-212. <https://doi.org/10.1076/csed.12.3.197.8618>
- Wing, J. M. (2006). Computational thinking. *Commun. ACM* 49, 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wood, A. & Olivier, M. (2004). A self-efficacy approach to holistic student development. *South African Journal of Education*, 24, 289-294.
- Woolfolk, A. E. & Hoy, W. K. (1990). Prospective teachers' sense of efficacy and beliefs about control. *Journal of Educational Psychology*, 82, 81-91. <https://doi.org/10.1037/0022-0663.82.1.81>
- Yaşar, S. (2021). Examination of primary school fourth grade students' perceptions of 21st century learning experiences [Doktora tezi]. Anadolu Üniversitesi.
- Yıldırım, F., & İlhan, İ. Ö. (2010). Genel öz yeterlilik ölçeği Türkçe formunun geçerlilik ve güvenilirlik çalışması. *Türk Psikiyatri Dergisi*, 21(4), 301-308.
- Yıldız, A. G. M., Çiftçi, E., & Karal, H. (2017). Bilişimsel düşünme ve programlama. Eğitim teknolojileri okumaları, 75-86.
- Yukselturk, E., & Altioğ, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789-801. <https://doi.org/10.1111/bjet.12453>
- Yu, J. & Roque, R. (2018). Küçük çocuklar için hesaplama kitleri araştırması. Etkileşim tasarımı ve çocuklar üzerine 17. ACM konferansı Bildiri Kitabında (s. 289-299).
- Yurdugül, H., & Gültekin, K. (2009). Çoklu ortamın Bilgisayar Programlama Başarısı Üzerine Etkisi The Effects Of Multimedia On Computer Programming Achievement. In Proceedings Of 9 Th International Educational Technology Conference.
- Zamankasy, M., Oliver, A., & Yadav, A. (2019) How an Unplugged Approach to Computational Thinking Can Move Schools to Computer Science. <https://www.edsurge.com/news/2019-12-19-how-an-unplugged-approach-to-computational-thinking-can-move-schools-to-computer-science> adresinden 3 Ocak 2022 tarihinde ulaşılmıştır.
- Zapata-Ros, M. (2015). Pensamiento computacional: Una nueva alfabetización digital. *Revista de Educación a Distancia (RED)*, (46).
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>
- Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, 141, 103633. <https://doi.org/10.1016/j.compedu.2019.103633>
- Zimmerman, B. J., Bandura, A., & Martinez-Pons, M. (1992). Self-motivation for academic achievement: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal*, 29, 663-676. <https://doi.org/10.2307/1163261>
- Zullig, K. J., Teoli, D. A., & Valois, R. F. (2011). Evaluating a brief measure of social self-efficacy among US adolescents. *Psychological reports*, 109(3), 907-920.

8. EKLER

EK 1 İnşacılık Modeli İçin Hazırlanan Ders Planları

SCRATCH OYUN KARTLARI YAPIMI

Doğadaki avcılık kolonisini gösteren oyun yapalım.





TEKİR UÇUYOR

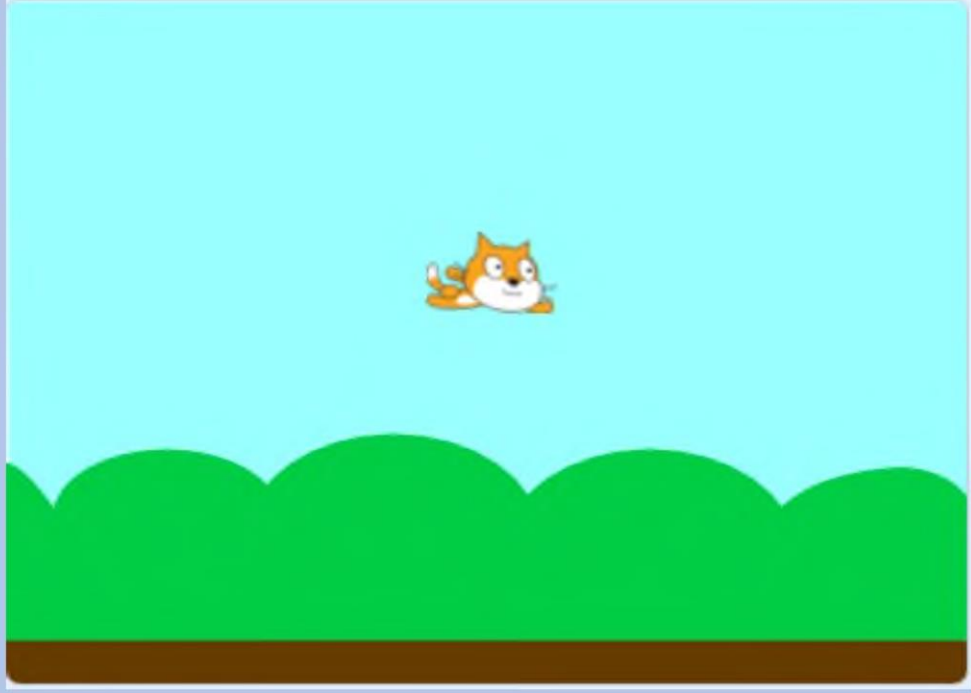


KUŞLAR SAĞA-SOLA RASTGELE



KUŞLARI GİZLENİP GÖSTERİLMESİ

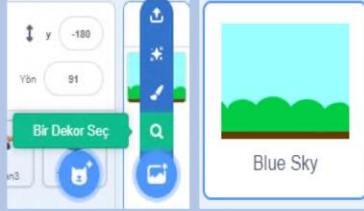
TEKİRİ EKRANDA HAREKET ETTİRELİM



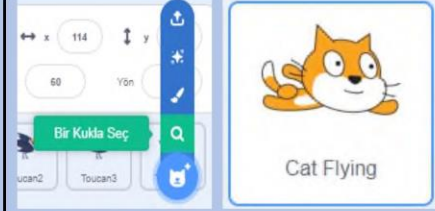
Tekir kedimizin fareyi takip ederek takip etmesi

TEKİR UÇUYOR

HAZIRLIK



Bu kısımda arka plandaki dekoru seçiyoruz.



Bu kısımda kukla seçimi yapıyoruz

BU KODU EKLE



Kedi karakterinin fare imlecini takip ederek ekranda serbest bir biçimde gezinmesini sağlar

Tıklandığında: Yeşil bayrağa tıkladığımızda, bu konunun altındaki tüm komutları çalıştırır.

Sürekli tekrarla: İçerisinde bulunan kod bloklarını sürekli olarak çalıştırır.

Fare imlecine git: Kuklanın fare imlecinin bulunduğu yere gitmesini sağlar

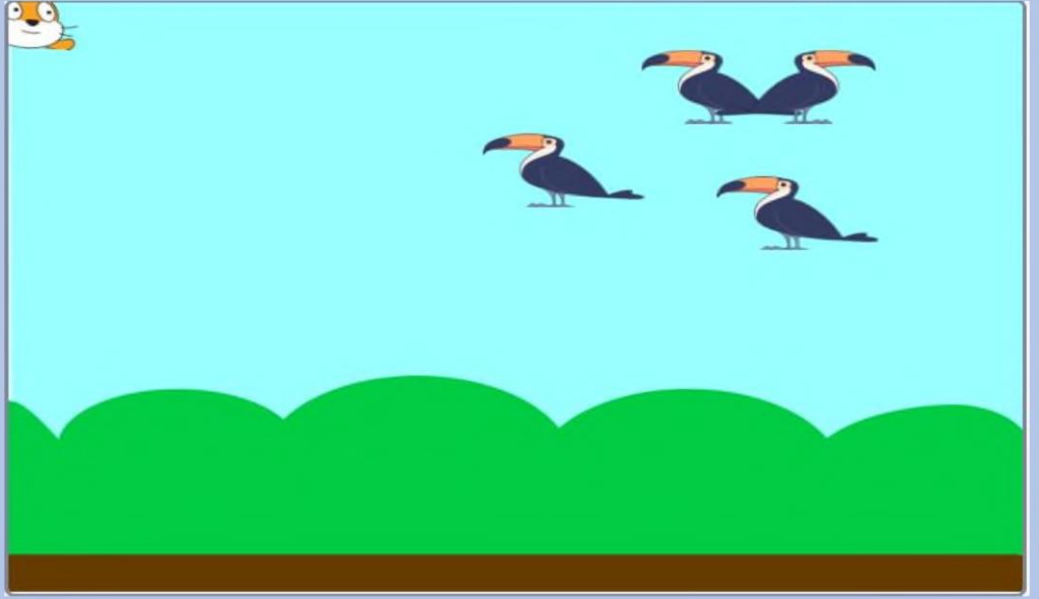
SEN DE DENE



Yeşil bayrağa tıkladığımız zaman kodların çalışmasını sağlar.

Çalışan kodların durmasını sağlar

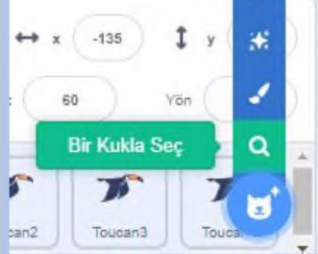
KUŞLARIN RASTGELE BİR ŞEKLİDE EKRANDA UÇMASINI SAĞLA



KUŞLAR SAĞA-SOLA RASTGELE UÇUYOR

HAZIRLIK

Bu kısımda kukla seçimi yapıyoruz



BU KODU EKLE



Papağan kuklasının ekranda sağdan sola soldan sağa şeklinde hareket etmesini ve kostümünün değişmesini sağlar.

Adım git: Sahneyi kuklanın bulunduğu yönde hareket ettirir.

Eğer ise : Bu kod ile eğer'den sonra gelen koşul doğru ise sonraki bloklar çalışmaya başlar. Yan taraftaki kod bloğundaki tekir kuşlara değdiği zaman şart doğru hale gelir ve eğer bloğundan sonraki kodların çalışması başlatılır.

Gizle: Bu kod bloğu ekrandaki kuklanın ekrandan kaybolmasını sağlar. Bu kod bloğunda değme işlemi yapıldıktan sonra gizlenme işlemi gerçekleştirilir.

Sürekli tekrarla: Bu kod bloğu altındaki kodların sürekli çalışmasını sağlar.

SENDE DENE



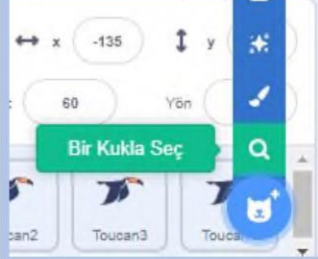
Yeşil bayrağa tıkladığımız zaman kodların çalışmasını sağlar.

Çalışan kodların durmasını sağlar

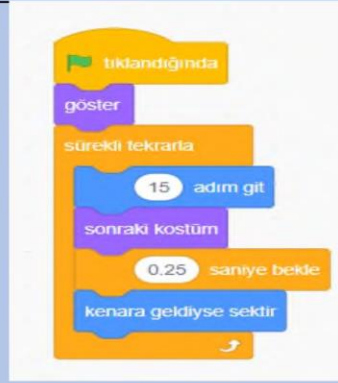
KUŞLARI GİZLENİP GÖSTERİLMESİ

HAZIRLIK

Bu kısımda kukla seçimi yapıyoruz



BU KODU EKLE



Göster: Bu komut ile kuşun kaybolduktan sonra tekrar ekranda gösterilmesi sağlanır. Bir önceki kod bloğunda ekranda gizlemiş olduğumuz kukla tekrardan ekrandan gözükmesi sağlanır.

Adım Git: Kuklanın bulunduğu yöne doğru yazılan sayı kadar ilerlemesi sağlanır.

Sonraki Kostüm: Kuklaları hareketli göstermek için birçok kostüm kullanırız. Bu kostümlerin arasında geçiş yapabilmek için bu kod bloğu kullanılır.

Saniye Bekle: uygulamayı istenilen süre kadar bekletir, süre tamamlanınca sıradaki kod blokların çalışmasını sağlar. Bu uygulamamızda kanat çırpma hareketlerini yavaşlatmak için kullanırız.

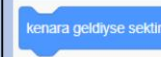
Kenara Geldiyse Sektir: Kukla ekranın kenarına değdiği ana karakteri ters yöne döndürür.

SENDE DENE



Yeşil bayrak tıklanarak uygulamayı çalıştırınız.

İPUCU



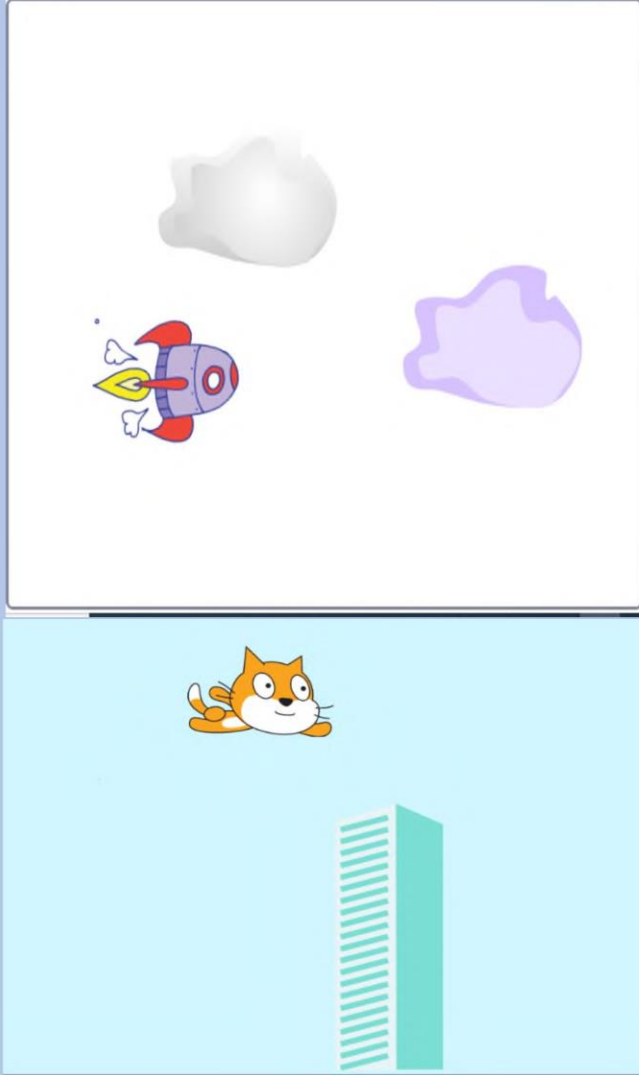
Komutun kullanımında kukla kenara değdiği anda ters bir hal almaktadır veya istediğimiz yöne gitmesi istendiği an yön kısmından ayarlamaları yapılır.

Alternatif olarak dönüş stilini sağ-sol yap komutu da kullanılabilir.



SCRATCH OYUN KARTLARI YAPIMI

Engellerden kaçan kukla uygulaması yapalım.





TEKİR UÇUYOR



BAŞLAMA KONUMU VE

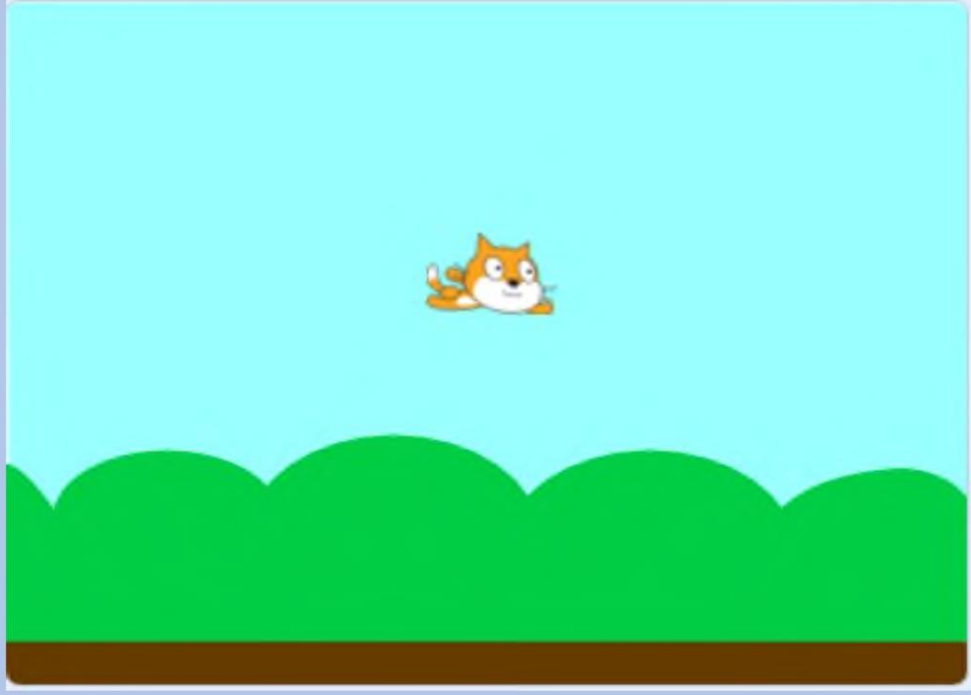


BİNALARA DEĞDİĞİ ANDA YANMA



BİNALARI HAREKETLİ HALE
GETİRME

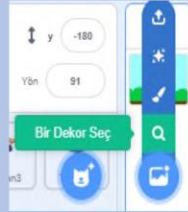
TEKİR İLE GÖKDELENLERE DEĞMEDEN UÇALIM



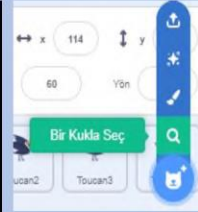
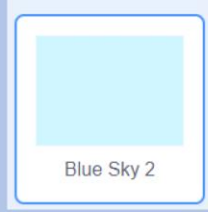
Tekir kedimizin fareyi takip ederek takip etmesi

TEKİR UÇUYOR

HAZIRLIK



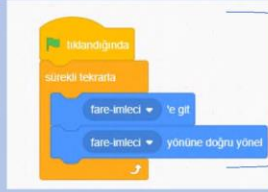
Bu kısımda arka plandaki dekor seçilir.



Bu kısımda kukla seçimi yapılır.



BU KODU EKLE



Bu kod bloğu bütünüyle kedi karakterinin fare imlecini takip ederek ekranda serbest bir biçimde gezinmesini sağlar

Sürekli tekrarla: İçerisinde bulunan kod bloklarını sürekli olarak çalıştırır.

Fare imlecine git: Kuklanın fare imlecinin bulunduğu yere gitmesini sağlar

Fare imleci yönüne doğru yönel: Kuklanın fare imlecinin olduğu yöne veya sahnede bulunan diğer kuklalara doğru yönelmesini sağlar.

SENDE DENE



Yeşil bayrağa tıkladığımız zaman kodların çalışmasını sağlar.

Çalışan kodların durmasını sağlar

BAŞLAMA KONUMU VE KONUŞMA

HAZIRLIK



Uçan kedi kuklasını tıklayıp aşağıdaki kod giriş işlemleri yapılır.

BU KODU EKLE



Kedi kuklasının oyunun başlangıçta hangi konumdan başlayacağı belirlenir ve hangi sözü söyleceği yazılır.
X,Y Konumuna Git:Kuklanın sahne üzerinde nereye gideceği girilen x ve y değerleri ile belirlenir.

Ön Katmanına Git:Kuklanın diğer kuklaların arkasında kalmaması için kod bloğu kullanılır.Kuklayı sahnede tüm kuklaların önüne gelir.

Saniye Boyunca De: Kuklanın yazılan metni girilen süre boyunca konuşma balonu içerisinde söylemesi sağlanır.

SENDE DENE



Denemek için yeşil bayrağa tıklayın.

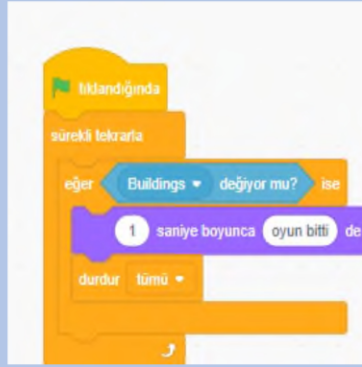
BİNALARA DEĞDİĞİ ANDA YANMA

HAZIRLIK

Uçan kedi kuklasını tıklayıp aşağıdaki kod giriş işlemlerini yapınız.



BU KODU EKLE



Değiyormu: Bu komut ile eklemiş olduğumuz herhangi bir kuklaya değdiği anda döngünün alt kısmındaki kodu çalıştırır değme işlemi gerçekleşmiyor ise herhangi bir işlem yapılmaz.

Durdur Tümü: Bu komut ile tüm işlemleri veya menüden seçilen özellikleri durdurabiliriz.

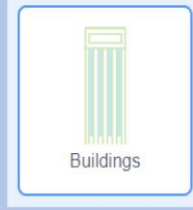
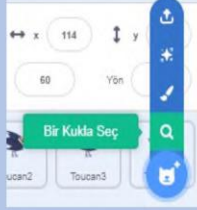
İPUCU



Oyun çalışırken değme işleminin devamlı olarak kontrol edilmesi gerekmektedir. Bu işlem için döngümüzü sürekli tekrarla komutu arasına alarak değme işlemini kontrol ettiririz.

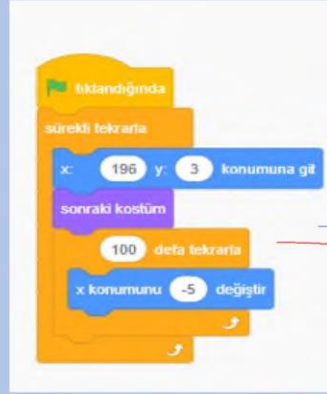
BİNALARI HAREKETLİ HALE GETİRME

HAZIRLIK



Buildings kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.

BU KODU EKLE



X ve Y konumuna Git: Bu komut ile kuklamızın başlangıç noktasını belli etmiş oluruz. Burada dikkat edilmesi gereken kedinin olduğu konum ile binanın başlangıç noktası aynı olmamalı oyun sürekli olarak bitti olarak hata verir.

Sonraki Kostüm: Bu komut ile kuklanın içinde oluşturulan bir sonraki kostüme geçmesi sağlanır.

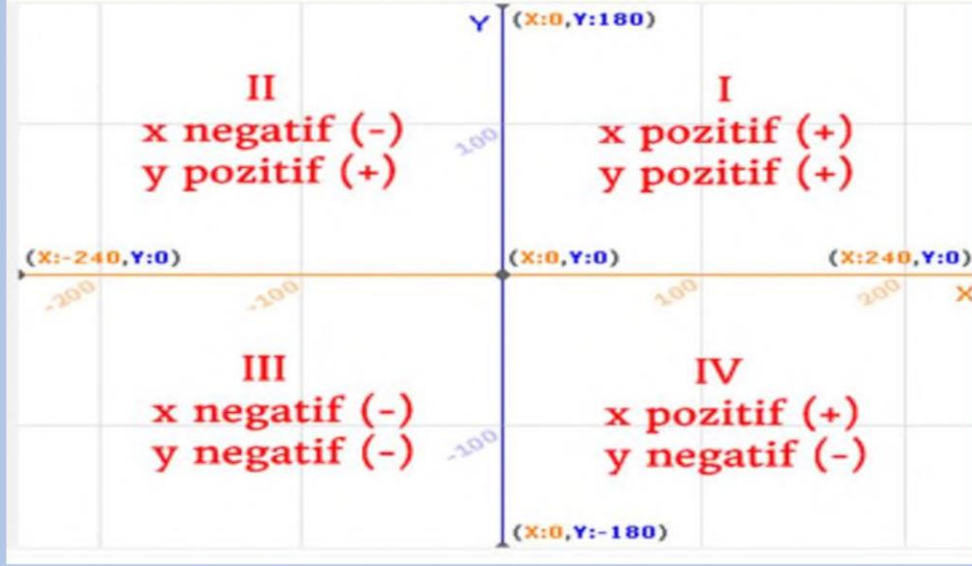
100 defa tekrarla: Buradaki komutumuz ile binalarımız ekranımızın sonuna kadar gelmesi için gerekli sayıyı giriyoruz. Ve ekranda binaların kayıyormuş gibi görünmesini sağlıyoruz.

SENDE DENE



Oyun tamamlandı yeşil bayrağa tıklayarak oyunu oynamaya başlayabilirsiniz.

UYARI



Scratch programında sahne, yatay ekseninde (x) $240+240=480$ pixel, düşey ekseninde (y) ise $180+180=360$ pixel bir dikdörtgen çalışma alanı sunar. Sahnenin tam ortasında $x=0$ ve $y=0$ koordinatlarını içerir.

Örnek verecek olursak herhangi bir kuklayı 1.bölümde olmasını istersek x ve y sayılarına pozitif sayılar girilmelidir.

4.bölümde olmasını istersek x pozitif y sayısına ise negatif değer girilmelidir.

SCRATCH OYUN KARTLARI YAPIMI

Animasyon çizgi film yapalım.





ALİ YÜRÜYOR



ALİNİN ANİNA ALDIĞI NOTU
SÖYLEMESİ



ANİNA SEVİNİYOR



ANİNA DANS EDİYOR

Anina nın Not Sevinci



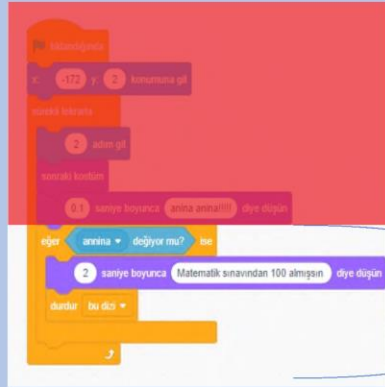
Ali'nin Anina Aldığı Notu Söylemesi

HAZIRLIK



Ali kuklasını tıklayıp aşağıdaki kod giriş işlemlerini yapınız.

BU KODU EKLE



Ali'nin yürümekte olduğu kodun alt kısmına kodlarımızı ekliyoruz.

Bu kod bloğu ile Ali'nin Anina ya değdiği anda sınavdan almış olduğu notu söyleyip Alinin bulunduğu konumda hareketsiz durmasını sağlar

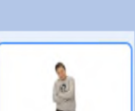
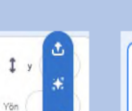
SENDE DENE



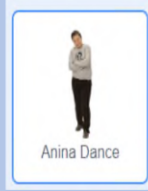
Denemek için yeşil bayrağa tıklayın.

Anina Seviniyor

HAZIRLIK

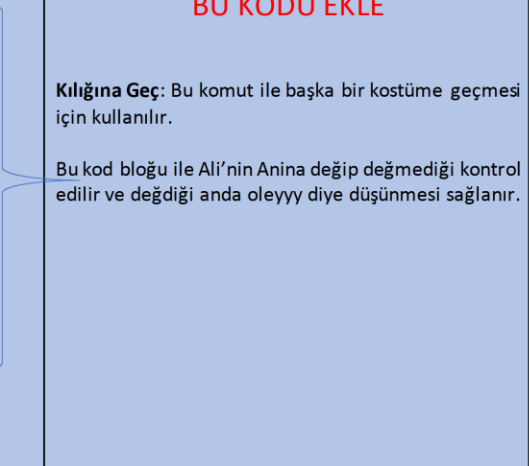


Anina Dance kuklası seçimi yapılır. Aşağıdaki kodla işlemleri yapınız.



Anına Dance kuklası seçimi yapılır. Aşağıdaki kodla işlemlerini yapınız.

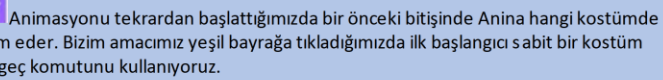
The diagram shows a Scratch script for a costume change. It starts with a 'When green flag clicked' event block. This is followed by an 'Anina stance' block with the costume set to 'Ali' and a 'say' block with the text 'Ali' and a duration of 2 seconds. Then, there is an 'if-then' conditional block. The 'if' part is 'Ali = Ali?'. The 'then' part contains two blocks: a 'say' block with the text 'Ali' and a duration of 2 seconds, and a 'say' block with the text 'Ali' and a duration of 2 seconds.



İPUCU

😊 **anina stance ▾ kılığına geç**

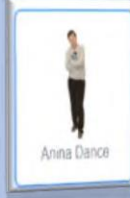
Animasyonu tekrardan başlattığımızda bir önceki bitişinde Anina hangi kostümde kalmış ise o kostümden devam eder. Bizim amacımız yeşil bayrağa tıkladığımızda ilk başlangıcı sabit bir kostüm olmasıdır. Bu sebeple kılığına geç komutunu kullanıyoruz.



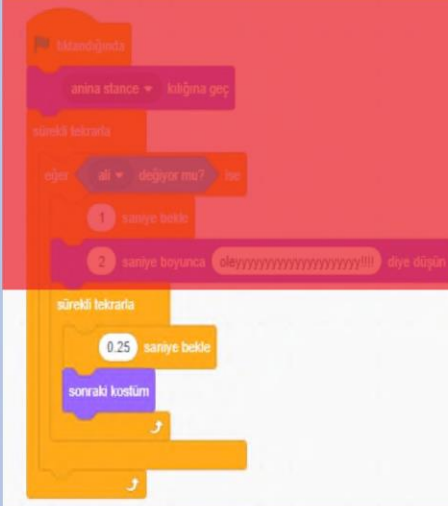
Anina Dans Ediyor

HAZIRLIK

Anina kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE



Kostümler seçeneğinde Anina birçok hareketi bulunmaktadır. Bunlar dans etme hareketlerinden bazılarıdır. Bu kod bloğu ile belirli saniye aralıkları ile kostümler arası geçiş yaparak kuklanın dans etme görüntüsünü sağlamış oluyoruz.

SENDE DENE



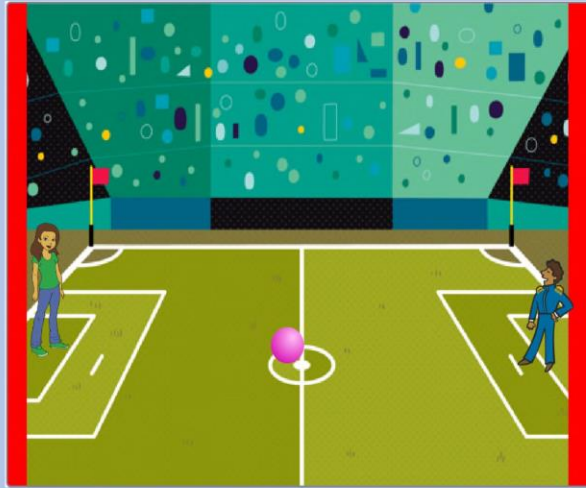
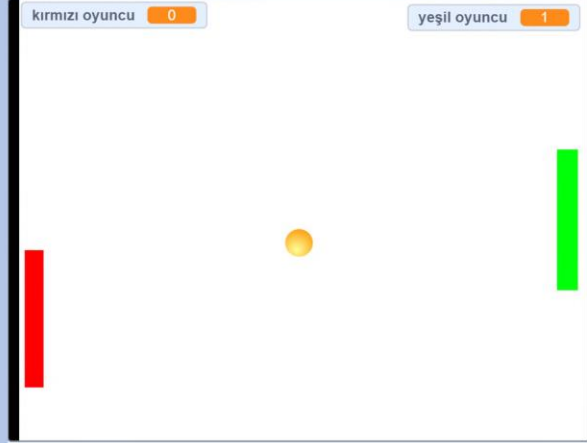
Animasyon tamamlandı yeşil bayrağa tıklayarak animasyonu çalıştırabilirsiniz.

ÖDEV

Yukarıdaki uygulamada iki kukla arasında konuşma sağlanmıştı. Sizler de kukla sayısını artırarak ve konuşma metnini ve animasyonlarını değiştirerek kuklaların konuşmalarını sağlayınız.

SCRATCH OYUN KARTLARI YAPIMI

Buz Hokeyi Oyunu Yapalım





BUZ HOKEYİ KUKLASI OLUŞTURMA

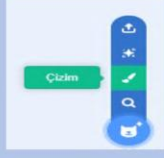


TOP HAREKET EDİYOR



KİM KAZANDI?

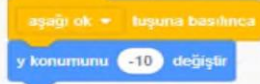
BUZ HOKEYİ KUKLASI OLUŞTURMA



HAZIRLIK

Bu kısımda hokey olarak kullanılacak iki adet çubuk çizilir. Bu çubuklardan bir tanesi kırmızı hokey çubuğu kuklası diğeri ise yeşil hokey çubuğu şeklinde adlandırılmalıdır.

BU KODU EKLE



Yeşil hokey çubuğuna yukarı ok ve aşağı ok tuşuna basıldığı anda y konumunu 10 ve -10 değiştir kodları eklenir.

Kırmızı hokey çubuğuna q ve z tuşuna basıldığı anda y konumunu 10 ve -10 değiştir kodları eklenir.

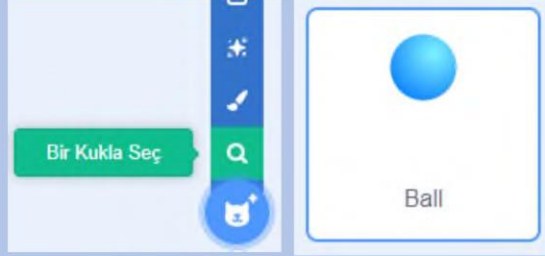
SENDE DENE



Denemek için yeşil bayrağa tıklayın.

TOP HAREKET EDİYOR

HAZIRLIK



Bir kukla seç menüsünden Ball kuklası seçilir. İsmi Top olarak değiştirilir.

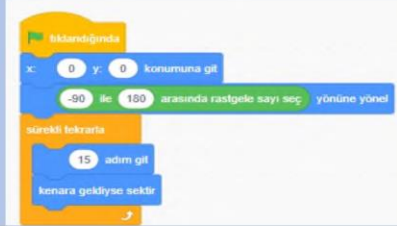
BU KODU EKLE



Top kuklasına yeşil renge değdiği anda topun yönünün tam tersi yönünde dönmesi için yönüne yönel kodu ile -1 ile -180 derece arasında bir sayı seçip farklı yönde gitmesi sağlanır. Ekranda 1 saniye boyunca yeşil yazdırılır.



Yukarıda yapılan işlemlerin kırmızı hokey çubuğunda geçerli olması için benzer işlemler yapılır. Bu kısımda rastgele sayısını 1 ile 180 arasında seçtiyoruz topun daha isabetli gibebilmesi için.



Bu kısımda topun başlangıç konumu belirtilip. Topun farklı yönlerde hareket edebilmesi için yönüne yönel komutu ile ayarlama yapılır. Sürekli tekrarlar kod bloğu içerisinde topun hareket etmesi ve kenara geldiğinde sekmesi sağlanır.

SENDE DENE



Denemek için yeşil bayrağa tıklayın.

KİM KAZANDI?

HAZIRLIK



Bir kukla seç menüsünden çizim araç çubuğu ile yandaki şekildeki gibi dikdörtgen çizilir. İçinin boyama rengi isteğe göre seçilir. Çizilen kuklanın bir kopyası çıkarılır ve kırmızı oyuncu, yeşil oyuncu olarak adlandırılır. Ekranın iki yan kısmına yerleştirilir.



BU KODU EKLE

Kırmızı oyuncu kuklası seçilerek kırmızı oyuncu değişkeni oluşturulur. Sürekli tekrarla kod bloğu altında topun değip değmediği kontrol edilir. Eğer kukla kırmızı hokeyin arkasında ise yeşil oyuncuya puan yazdırılır. Tam tersi durumda ise yeşil oyuncuya puan kazandırılır.

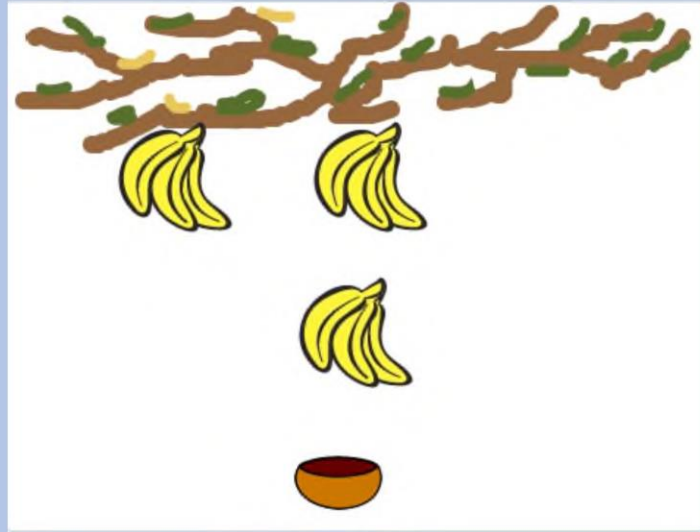
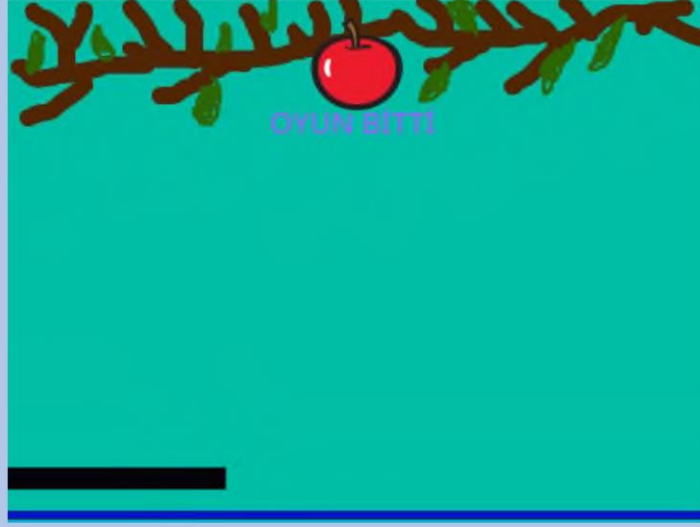
SENDE DENE



Kendine bir rakip bul ve oyuna başla.

SCRATCH OYUN KARTLARI YAPIMI

Kendi oluşturduğumuz dekorda rastgele konumda aşağı düşen elmayı yakalama oyunu.





AĞAÇ DEKORU OLUŞTURMA



ELMA DÜŞÜYOR



ELMA TOPLAMA ÇUBUK SEPETİ



ELMA TOPLAMA



ELMA YERE DÜŞÜNCE OYUN BİTER

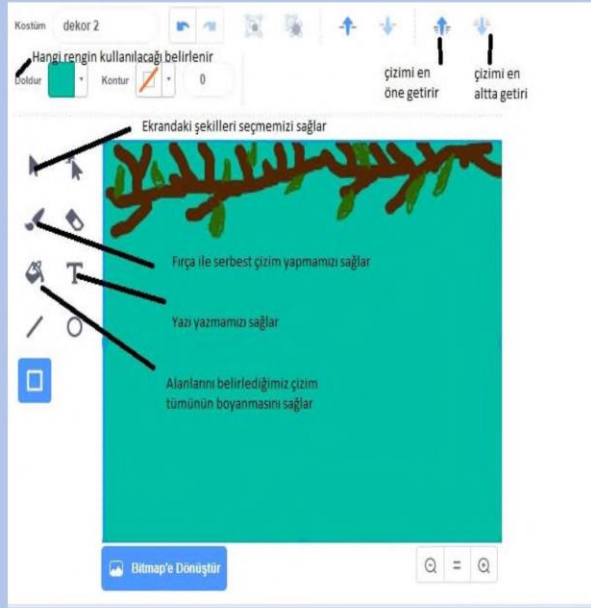
AĞAÇ DEKORU OLUŞTURMA

HAZIRLIK



Bu kısımda bir dekor seç seçilir ve çizim seçilir.

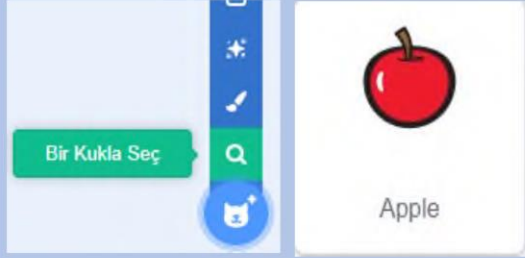
DEKOR DÜZENLEME



Yandaki şekildeki gibi çizim araçlarını kullanarak bir ağaç dalları resmi çizersiniz.

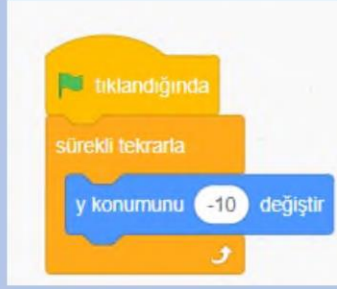
ELMA DÜŞÜYÖR

HAZIRLIK



Bir kukla seç menüsünden apple kuklası seçilir.

BU KODU EKLE



Elmanın sürekli olarak aşağı düşmesi için sürekli tekrarla kodu altında y konumu -10 olarak değiştirilir.

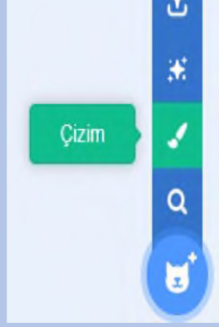
SENDE DENE



Denemek için yeşil bayrağa tıklayın.

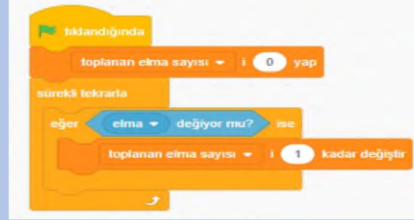
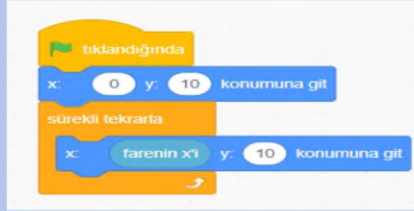
ELMA TOPLAMA ÇUBUK SEPETİ

HAZIRLIK



Bir kukla seç menüsünden çizim araç çubuğu ile yandaki şekildeki gibi dikdörtgen çizilir. İçinin boyama rengi isteğe göre seçilir.

BU KODU EKLE



Bu kod bloğunda faremizin hareketi ile dikdörtgen şeklinde çizdiğimiz kuklamızın hareketinin aynı olması sağlanır. Faremiz sağa gittiği anda kuklamız sağa gider sola gittiği anda sola gider. X farenin x i komutu ile fare imlecinin olduğu koordinat girilmiş olur. Y konumu 10 yaparak kuklanın y konumunda sabit olarak kalması sağlanır.

Bu kod bloğunda elma toplama çubuğuna, elma değiştiği anda toplam elma sayısını 1 artırma işlemi yapılır. İlk başta bir önceki oyundan kalan rakamın sıfırlanması için toplanan elma sayısını sıfır yaparız.

SENDE DENE



Denemek için yeşil bayrağa tıklayın.

ELMA TOPLAMA

HAZIRLIK



Elma kuklası seçilir ve aşağıdaki kodlar eklenir.



BU KODU EKLE

Bu kısımda gizlenmiş olan elma kuklamızın tekrardan gösterilmesi sağlanır. Dikdörtgen olarak çizilmiş olan toplama sepetimize elma dokunduğunda elma kuklasının kaybolup tekrardan yere inmek üzere konumuna git kod bloğu kullanılır. Burada x konumunu rasgele bir sayı üretilerek elmanın farklı konumlarda çıkması sağlanır. Konumu 132 konumu girilerek ağaçtan düşme efekti ayarlanır.

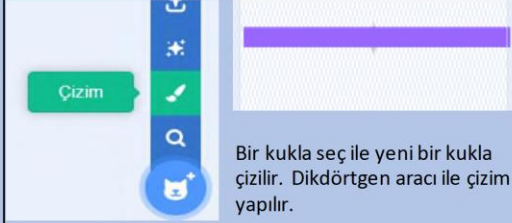
SENDE DENE



Denemek için yeşil bayrağa tıklayın.

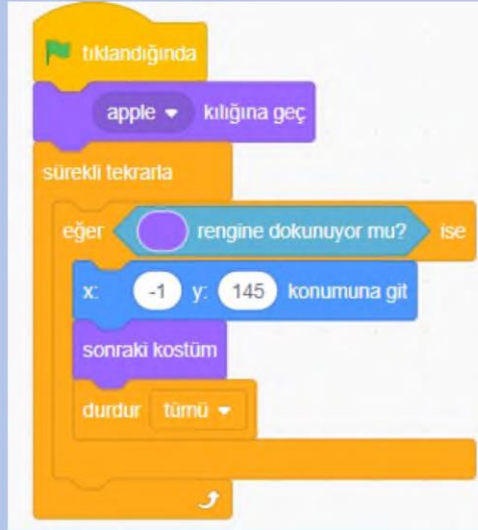
ELMA YERE DÜŞÜNCE OYUN BİTER

HAZIRLIK



Kostümler sekmesinden elma kuklası seçilir. Kopyasını çıkar seçeneği ile yeni kukla kostümü oluşturulur. Bu oluşturulan kostüme oyun bitti yazısı yazılır. Elma kuklasına aşağıdaki kodlar eklenir.

BU KODU EKLE



Uygulama başladığında elmanın ilk kostümüne geçmesi sağlanır. Kukla rengi seçilir ve bu renge değdiği anda ağacın olduğu konuma gönderilir. Ve kostüm olarak oyun bitti kostümü seçilir. Tümü durdur komutu ile uygulama durdurulur.

SENDE DENE



Oyun bitti. Uygulama yapabilirsiniz.

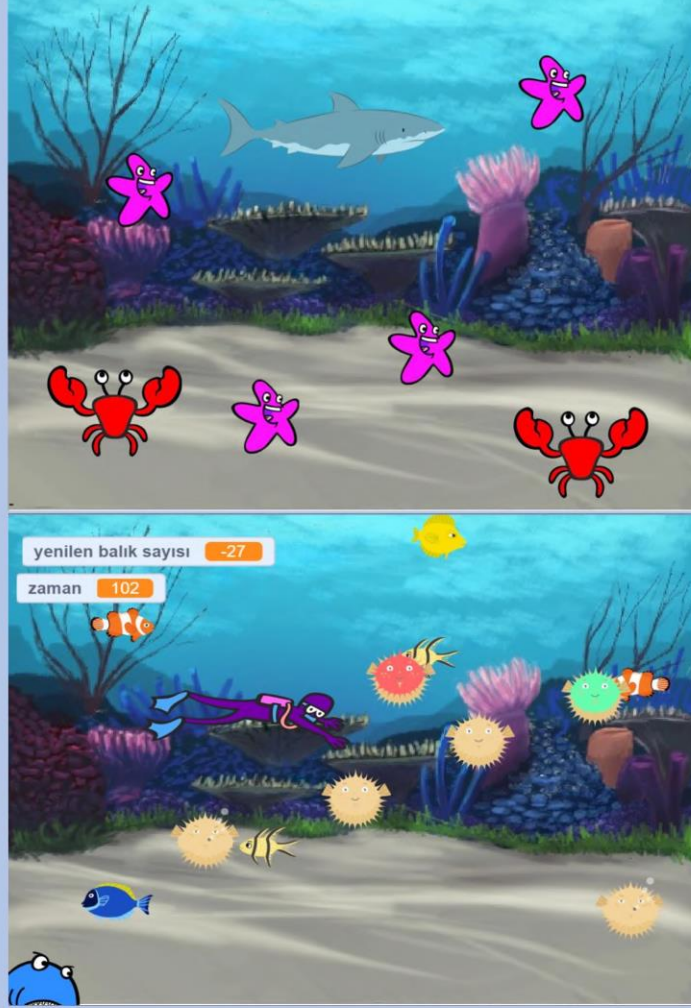
İPUCU



Mor şekilde ürettiğimiz kukla zemini bütünüyle kaplamalıdır.

SCRATCH OYUN KARTLARI YAPIMI

BALIK AVLAMA OYUNU YAPALIM.





KÖPEK BALIĞI HAREKET EDİYOR



OKYANUSTA YEM BALIKLARI YÜZÜYOR



KÖPEK BALIĞI KÜÇÜK BALIKLARI YİYOR



ZAMANINDA BALIKLARI YAKALAMALISIN



BALON BALIKLARINI HAREKET ETTİR



BALON BALIKLARINI SAKIN YEME



BALON BALIKLARI OYUNU ZORLAŞTIRIYOR



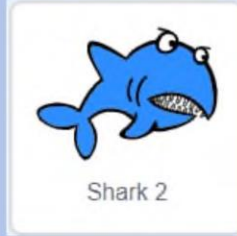
BALIK ADAM İLE DEĞİŞKENLERLE KAYBETME-
KAZANMA



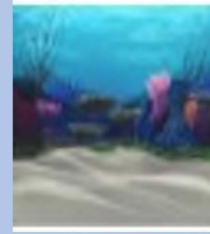
DEĞİŞKENLER DE EKSİ PUANA DÜŞMEYİ ENGELLEME

KÖPEK BALIĞI HAREKET EDİYOR

HAZIRLIK



Kukla ekle
menüsünden Shark2
eklenir.



Dekor seç
menüsünden su altı
balığından herhangi
bir resim eklenir.

BU KODU EKLE



Köpek balığının boyutunu ayarlaması yapılır.Sürekli tekrarla komutu ile fare imleci ile aynı yöne dönmesi için fare imlecinin yönüne doğru yönel kodu kullanılır.Farenin gittiği yer doğru hareket etmesi için fare imlecine git komutu kullanılır.

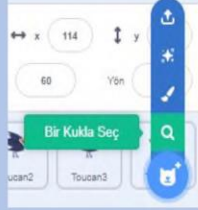
SENDE DENE



Denemek için yeşil bayrağa tıklayın.

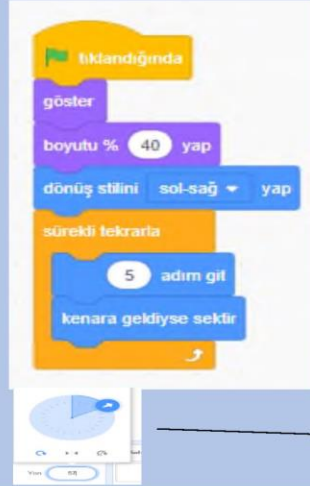
OKYANUSTA YEM BALIKLARI YÜZÜYOR

HAZIRLIK



Fish kuklası seçimi yapılır. Kostümler sekmesinden diğer karakterler silinir. Aşağıdaki kodlama işlemleri yapılır.

BU KODU EKLE



Balıklar bir sonraki bölümde değineceğimiz değme kısmında bir gizlenme meydana gelecektir. Tüm balıkların ekranda oyunun başlangıcında gözükebilmesi için göster komutu kullanılır. Boyutları ayarlanır. Sürekli tekrarla döngüsü için de balıkların ekranda rastgele hareket etmesi için gerekli kodlar yazılır.

Balıkların ekranda rastgele hareket edebilmeleri için yönünü farklı yerlere ayarlanmalıdır.

İPUCU



Birkaç tane daha balık türü eklenip oyun daha canlı hale getirilebilir. Alt sınır olarak 6 balık eklenir. Aynı kodlar bu balıklara da eklenir.

KÖPEK BALIĞI KÜÇÜK BALIKLARI YİYOR

HAZIRLIK



Fish kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz. Bu kısımda yaptığımız kod işlemini eklemiş olduğumuz diğer küçük balıklara da eklemeyi unutmayın.

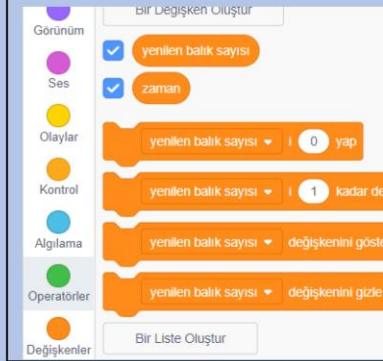


BU KODU EKLE

Sürekli tekrarla komutu ile alt kısımda yapılan işlemlerin devamlı olarak kontrol edilmesi sağlanır. Koşul kodu ile köpek balığının balıklara değip değmeme işlemi kontrol edilir. Yenilen balık sayısı adı altında bir değişken oluşturulur ve köpek balığının her dokunmasında bu sayının bir artması sağlanır. Balığın köpek balığı tarafından yeniliyormuş görüntüsü vermek için gizle komutu tıklanır. 3 saniye bekleme süresi gözetilir ve balığın tekrardan görüntülenmesi sağlanır.

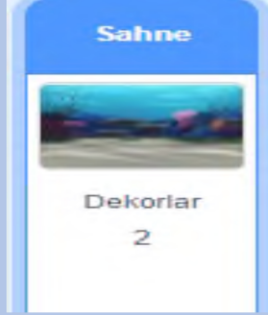
İPUCU

Değişken tanımlamak için Ekranın sol kısmından değişkenler seçilir. Bir değişken oluştur seçeneği tıklanır. Değişkenin ismi oluşturulur. **Değişken:** Girdiğimiz değerleri alan veya programın çalışmasıyla bazı değerlerin atandığı veri tutuculardır. Burada bizde tüm balıkların yenilme sayısını yenilen balık değişkeninde tutmuş oluyoruz. İleriki konularda zaman değişkeni oluşturup süreyi de bu değişkende tutacağız.



ZAMANINDA BALIKLARI YAKALAMALISIN

HAZIRLIK



Resim ikonuna tıklayarak sahne kod kısmına geçilir. Sahne kod kısmı açılarak aşağıdaki kod blokları eklenir.

BU KODU EKLE



Bir önceki uygulamada değişken konusu işlenmiştir. Bu uygulamada ise zaman değişkeni oluşturulur. Zaman değişkeninin amacı arka planda süre oluşturmaktır. Zaman değişkeninin oyuna girdiğimiz anda 0 olması sağlanır. Arka planda sayıların artması için zaman 1 kadar değiştir bloğu kullanılır. Bu kodun saniye cinsinde ilerlemesi 1 saniye bekle komutu konularak sürenin saniye cinsinden ilerlemesi sağlanır.

SENDE DENE



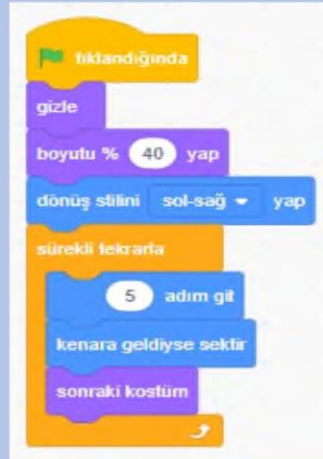
Denemek için yeşil bayrağa tıklayın.

BALON BALIKLARINI HAREKET ETTİR

HAZIRLIK



Pufferfish kuklasının seçimi yapılır. Aşağıdaki kod işlemleri bu kuklaya uygulanır.



BU KODU EKLE

Bir bölüm önce yem balıklarını hareket ettirme kodları balon balıkları içinde kullanılabilir. Yandaki kod blokları sıralanarak balon balığının hareketi sağlanır.

SENDE DENE



Denemek için yeşil bayrağa tıklayın.

DİKKAT!!!



Pufferfish kuklalarından en az 4 kopya oluşturup kodları bu kuklalara da uygulayınız. Oyunu daha zor hale getirmek isterseniz sayıları artırabilirsiniz.

BALON BALIKLARINI SAKIN YEME

HAZIRLIK



Pufferfish

Pufferfish kuklasının seçimi yapılır. Aşağıdaki kod işlemleri bu kuklaya uygulanır.



BU KODU EKLE

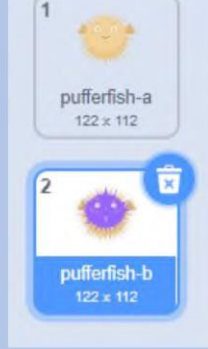
Bu kısımda balon balıklarının köpek balığı deđdiği anda ekrandan kaybolup önceden oluşturmuş olduğumuz yenilen balık sayısı puanını -5 kadar düşürölür. Gizle ve göster arasına 1 saniye bekle konularak balığın tekrardan ekran da belirerek oyuna devamlılık sağlanır.

İPUCU

Oluşturmuş olduğunuz diđer kuklalara da bu kodları eklemeyi unutmayınız. Puan indirilmesi ile ilgili sayıları zor veya daha kolay olması bakımından istediğiniz bir rakam verilebilirsiniz.

BALON BALIKLARI OYUNU ZORLAŖTIRIYOR

HAZIRLIK



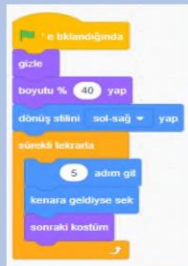
Diğer Pufferfish kuklalarından ayrı olarak yeni bir kukla oluşturulur ve kostümler kısmından balığın ikinci sıradaki kostümü farklı bir renge boyanır. Yukarıdaki balon balıklarını hareket ettir kod blokları bu kısımda eklenir. Ve aşağıdaki kod blokları ile devam edilir.

BU KODU EKLE



Bu kısımda balon balıklarının köpek balığı değdiği anda ekrandan kaybolup önceden oluşturmuş olduğumuz yenilen balık sayısı puanını -5 kadar düşürölür. Gizle ve göster arasına 1 saniye bekle konularak balığın tekrardan ekran da belirerek oyuna devamlılık sağlanır.

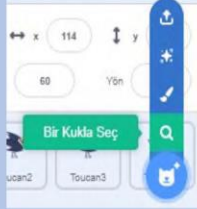
İPUCU



Bu kısımda en az 3 tane balon balığı eklenir. Puan indirilmesi ile ilgili sayıları zor veya daha kolay olması bakımından istediğiniz bir rakam verilebilir. Balon balığının hareket kodlarını yazmayı unutmayınız.

BALIK ADAM İLE DEĞİŞKENLERLE KAYBETME-KAZANMA

HAZIRLIK



Bu kısımda kukla seçimi yaparız.

Kuklamızın diğer kostümleri seçilir ve diver-2 kostümüne kazandın. Diver-3 kostümüne kaybettin yazılır.

BU KODU EKLE



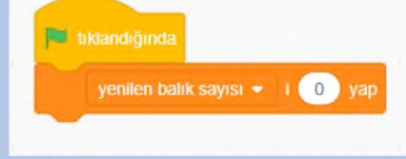
Buradaki kod bloğu ile balık adamın gizle komutu ile ilk başta ekranda görünmemesi sağlanır. 1.resimde yenilen balık sayısı üst kısımlardaki uygulamalarda değişkenlik göstermektedir. Bu kısımda ise yapmış olduğumuz değişikliklerin oyunu kazanma ve kaybetme için kullanmaktayız. Yenilen balık sayısı 60 a ulaştığında oyunu kazandınız yazdırıp oyunu durduruyoruz.2.resimde ise zamanın arka planda ilerleyerek 100 saniye olduğu anda ekrana oyunu kaybettiniz yazdırıp oyun durdurulur.

SENDE DENE



Ali'nin hareket etmesini kontrol edebilirsiniz.

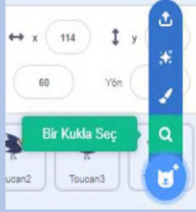
HATIRLATMA



Yenilen balık sayısını oyun başladığında tekrardan sıfır yapınız.

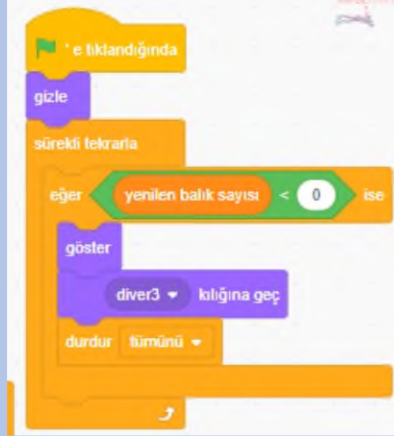
DEĞİŞKENLER DE EKSİ PUANA DÜŞMEYİ ENGELLEME

HAZIRLIK



Bu kısımda kukla seçimi yaparız.

BU KODU EKLE



Bu kod bloğunda yenilen balık sayısının puanı 0'dan küçük olduğu zamanda oyunun durmasını sağlayan kod bloğu yazılır.

SENDE DENE



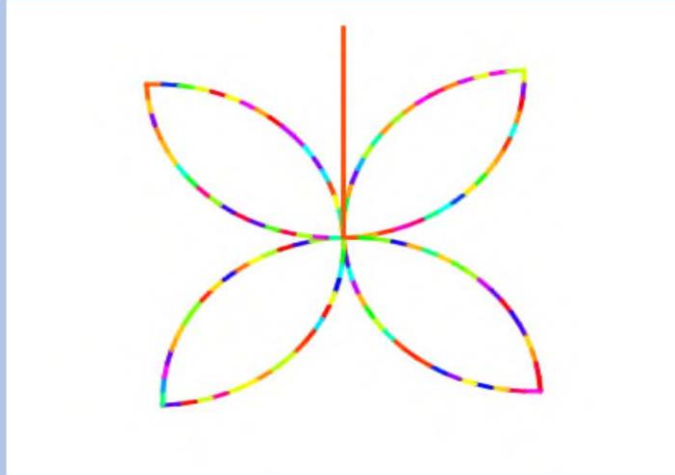
Ali'nin hareket etmesini kontrol edebilirsin.

ÖNERİ

Oyuna çeşitli değişkenler ekleyerek geliştirebilirsiniz.

SCRATCH OYUN KARTLARI YAPIMI

KELEBEK MOTİFİ ÇİZELİM





KALEMİ ÇİZİM İÇİN HAZIRLAMA



KELEBEĞİ ÇİZ KOMUTU BUTONU YAPMA



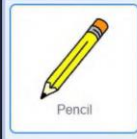
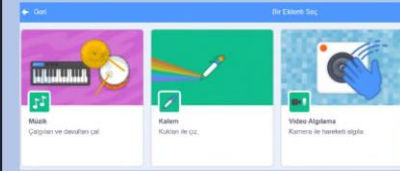
BUTONA TIKLANILDIĞINDA KELEBEĞİN BİR KANADINI
ÇİZME



KELEBEĞİN DÖRT KANADININ VE GÖVDESİNİN
ÇİZİLMESİ

KALEMİ ÇİZİM İÇİN HAZIRLAMA

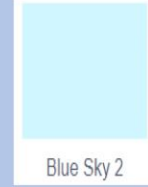
HAZIRLIK



Kukla ekle
menüsünden
Pencil kuklası
eklenir.



Kod bölümüne kalem
nesnesini eklemek için
ekleni ekle menüsüne
tıkılır. Kalem
eklentisi eklenir.



Dekor olarak
blue sky2
eklenir.

BU KODU EKLE



Bu kod bloğunda ekrandaki bütün şekiller silinip kalem hazır konuma getirme işlemi yapılır.

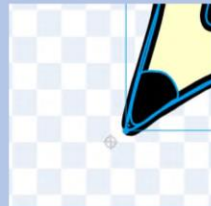
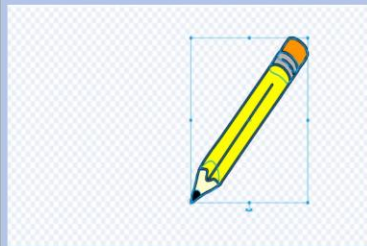
Tümünü sil: Bu komut ile ekranda daha önce çalıştırdığında kalan bütün şekillerin silinmesi için kullanılır.

Kalemi kaldır- Kalemi Bastır: Kalemi bastır ile kalem ekrana şekil çizmeye başlar. Bazı durumlarda ekranda şekil çiziminin yapılamaması istenir bu durumda ise kalemi kaldır komutu kullanılır.

Kalem kalınlığı: Kalem kalınlığı ile çizilecek çizginin boyutu daha geniş hale getirilir.

Konum olarak 0,0 konumu seçilir ve yön olarak 0 yönü verilir.

DİKKAT

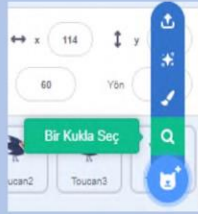


Kalem kuklasının kostümler sekmesinden kalem ucu merkez olarak ayarlanmalıdır.

Bu kısımda kullanılan yeşil bayrağa tıklandığında kodu daha sonra haber sal olarak değişecektir. Sadece deneme amaçlı kullanılmıştır.

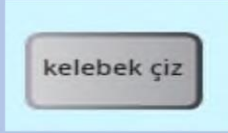
KELEBEĞİ ÇİZ KOMUTU BUTONU YAPMA

HAZIRLIK



Button3 kuklası seçimi yapılır. İç kısmına kelebek çiz yazılır.

BU KODU EKLE



Kelebek çiz butonu seçilerek içine bu kukla tıkladığında kodu eklenir. Bu kukla tıkladığında komutu ile kullanın tıkladığı anda kod bloğunun altında yazılan görev gerçekleşir. Burada yapılmak istenen kukla tıkladığı anda kaleme bu kukla tıklanmıştır haberi göndermektir. Haber sal komutu ile yeni haber diyerek haberimizin ne olduğunu belirtiyoruz.



Dikkat

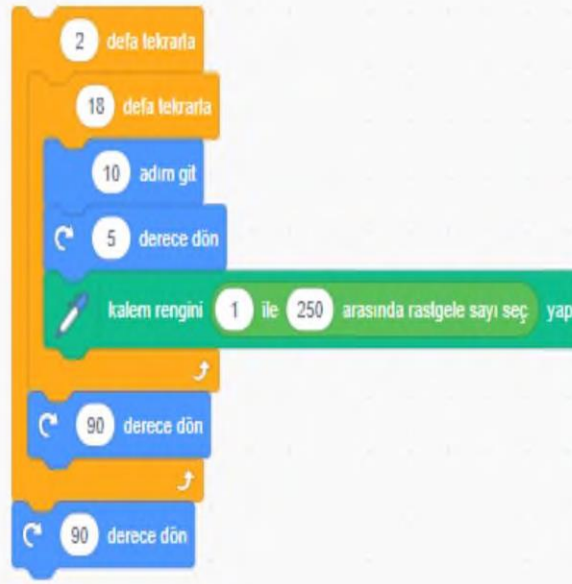
Bu kısımda yayınladığımız haberi başka bir kuklada haber aldığımızda şeklinde kullanılır. Kod bloklarının başındaki yeşil bayrağa tıkladığında komutu yerine kelebek haberi aldığımızda kodu eklenir.

BUTONA TIKLANILDIĞINDA KELEBEĞİN BİR KANADINI ÇİZME

HAZIRLIK



Pencil kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE

Kalem kuklasında üst kısımda yazılan kodun devamı olarak yandaki kod blokları yazılır.

Bu kısımda adım git komutu ile kesikli çizgiler çizilir her kesikli çizgiden sonra renk değişimi yapılır ve 5 derece dönüş yapılır. Bu kısımda kanadın bir tarafı oluşturulur ve 18 çizgiden meydana gelir. Diğer kısımda oluşturulacağı için 2 defa tekrarla döngüsü kullanılır. Kanadın ucunun sivri bir biçimde görünmesi için 90 derece dönüş yaptırılır.

İPUCU

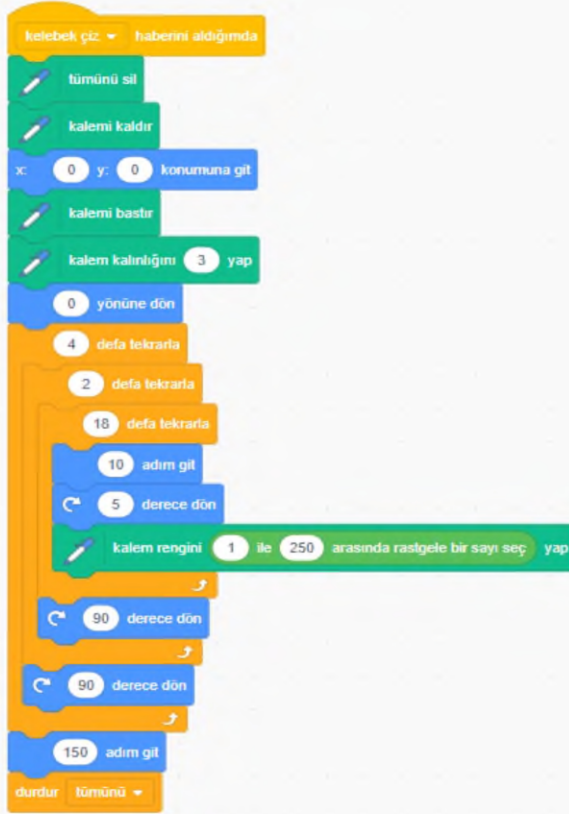
Yukarıdaki gibi birçok tekrarla komutunun bir arada kullanılmasına iç içe döngü denilmektedir. İç içe döngüde dikkat edilmesi gereken önce en içteki döngü tamamlanır daha sonra diğer döngüye geçilir.

KELEBEĞİN DÖRT KANADININ VE GÖVDESİNİN ÇİZİLMESİ

HAZIRLIK



Pencil kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE

Kelebeğin bir kanadı bir önceki bölümde çizilmişti. Bu olay dört defa tekrar edeceğinden dolayı çizilen kanadı dört defa tekrarlanarak kod bloğuna alınır fakat her çizimden sonra 90 derece dönmesi sağlanır. En son aşmada kanatların çiziminden sonra kelebeğin gövdesini oluşturmak için 150 adım git komutu ile oluşturulur. Durdur tümü ile çizim bitirilir.

SENDE DENE



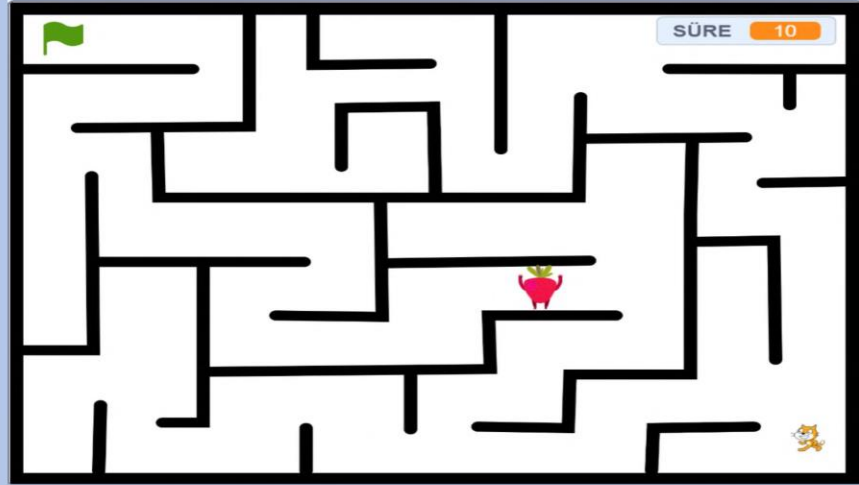
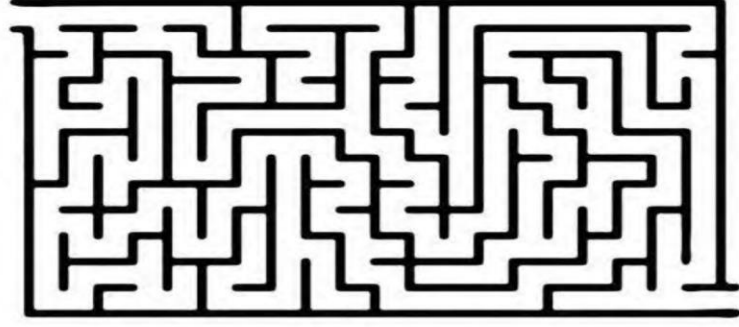
Denemek için yeşil bayrağa tıklayın.

ÖDEV

Sizde İnternette veya bildiğiniz bir markanın logosunu çiziniz.

SCRATCH OYUN KARTLARI YAPIMI

LABİRENT OYUNU YAPALIM.





LABİRENT TE KEDİ HAREKET EDİYOR



DEKORLAR OLUŞTURUYORUZ



LABİRENTTE ÇARPMADAN İLERLE



HEDEFİMİZ YEŞİL BAYRAĞI YAKALAMAK



BAYRAĞI YAKALADIĞIN AN OYUNU KAZANIRSIN



BAYRAĞA BELİRTİLEN SÜREDE ULAŞMALISIN



KARAR SENİN HIZLI MI YAVAŞ MI?



ÇİLEK ÖDÜLÜ



ELMA ÖDÜLÜ



HIZLANDIRMA ÖDÜLÜ



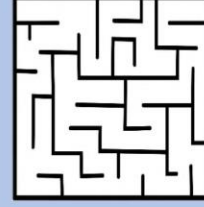
SES EKLEME

LABİRENTTE KEDİ HAREKET EDİYOR

HAZIRLIK

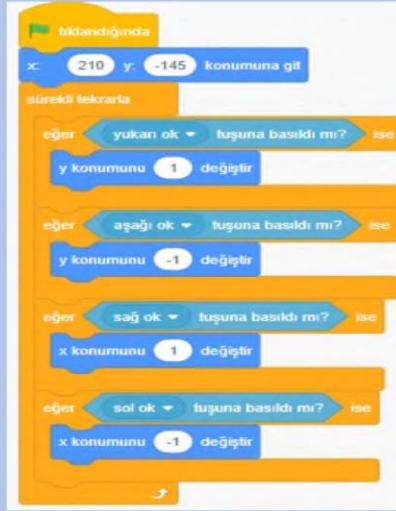


Kukla ekle
menüsünden cat
kuklası eklenir.

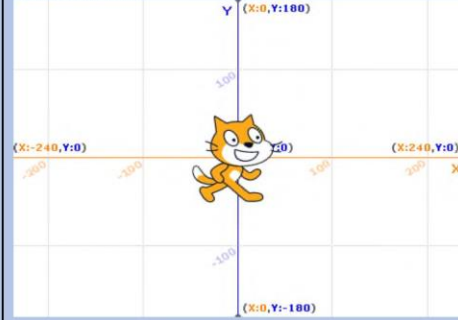


<https://drive.google.com/file/d/1X58nzheZ4q0eMG1btwG8PfGkBz5SShSa/view?usp=sharing>
adresinden labirent
resmi indirilir ve
dekorlar kısmına
eklenir.

BU KODU EKLE



Bu kısımda kedi karakterimizin yukarı- aşağı, sağa-sola şeklinde hareket etmesi sağlanır. Karakterimiz oyunun ilk başlangıcında labirentin başlangıç kısmında olması için konumuna git kodu ile yeri belirlenir.



Konum bilgilerini girerken yukarıdaki grid resmindeki koordinatlar baz alınarak konum değiştir değerleri girilir.

SENDE DENE



Denemek için yeşil bayrağa tıklayın.

DEKORLAR OLUŖTURUYORUZ

HAZIRLIK



Bir dekor seç sekmesinden moon ve stars dekorları seçilir.

BU KODU EKLE



Uygulama yapılacak oyunda kedi yeŖil bayraĖa labirenti aŖarak ulaŖmaya alıŖacaktır. YeŖil bayraĖa ulaŖtıĖı anda tebrikler dekoru gsterilir. Labirentin sonundaki yeŖil bayraĖa belirli bir srede ulaŖımı istenilmektedir. Sre getiĖinde ise sre bitti dekoru çıkmaktadır.

SENDE DENE



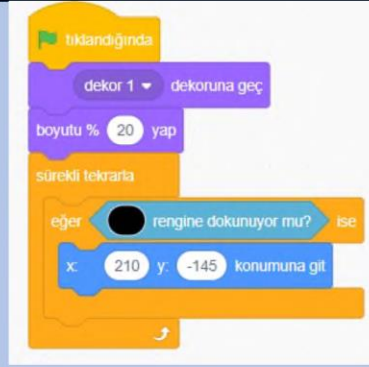
Denemek iin yeŖil bayraĖa tıklayın.

LABİRENTTE ÇARPMADAN İLERLE

HAZIRLIK



Kedi kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE

Bu kod bloğunda labirent dekoruna geçiş sağlanır. Kedinin boyutu küçültülür. Rengine dokunuyor mu komutu labirentin üzerindeki renk seçilir bu renge değdiği anda kukla başlangıç noktasına geri gönderilir.

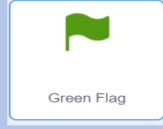
SENDE DENE



Denemek için yeşil bayrağa tıklayın.

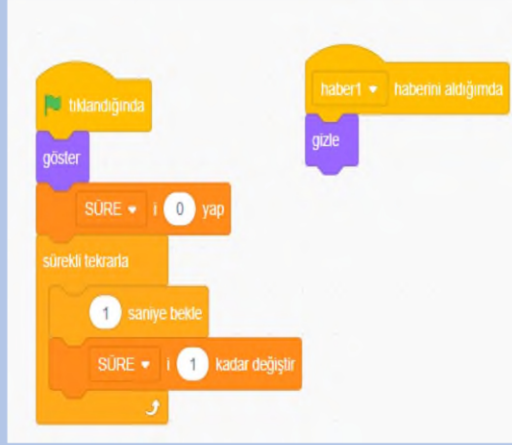
HEDEFİMİZ YEŞİL BAYRAĞI YAKALAMAK

HAZIRLIK



Green Flag kuklasını ekliyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.

BU KODU EKLE



Kuklanın oyun bitiminde gizlendiğinden ötürü bayrağa tıklandığında ekrana çıkması sağlanır. Sürenin bir önceki oyundaki sayısı sıfırlanır. Sürenin 1 saniye şeklinde artması için gerekli komutlar yazılır. Oyun için belirlenen sürenin bitiminde bayrağın gizlenmesi sağlanır.

SENDE DENE



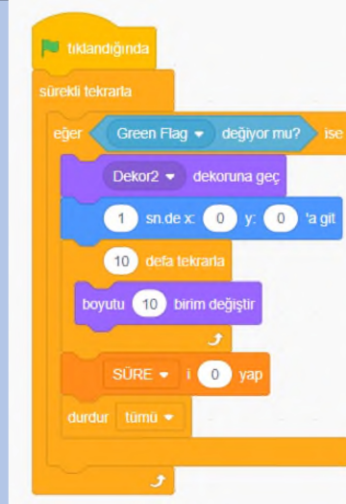
Denemek için yeşil bayrağa tıklayın.

BAYRAĞI YAKALADIĞIN AN OYUNU KAZANIRSIN

HAZIRLIK



Kedi kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE

Yeşil bayrağa kuklamız değdiği anda dekoru kazandın dekoru olan ikinci dekora geçitilir. Ekrandaki kuklamızı x ve y noktalarında 0,0 konumuna getirilir. Kuklanın boyutunun büyüdüğü bir animasyon yapılır. Süre değişkeni sıfırlanır. Oyunun durması sağlanır.

SENDE DENE



Denemek için yeşil bayrağa tıklayın.

BAYRAĞA BELİRTİLEN SÜREDE ULAŞMALISIN

HAZIRLIK



Kedi kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE

Sürenin sonunda kaybolan karakterimizin oyun başladığında tekrardan ekranda görüntülenmesi sağlanır. Oyun bitirme süresi olarak 50 sn yazılır. Süre 50 sn olduğunda süre bitti dekoruna geçilir. Bayrak kuklasının da gizlenmesi için haber salınır. Kuklanın ekranda gizlenmesi sağlanır ve tüm oyun durdurulur.

İPUCU

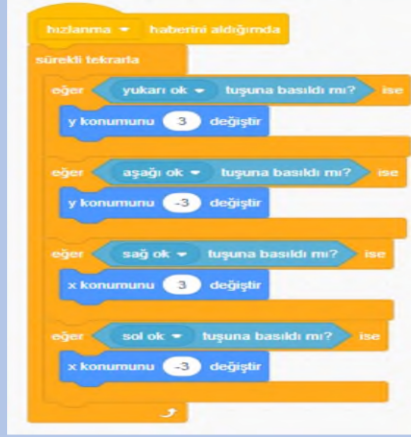
Süre kısmını kullanıcı kolay veya zorluk derecesine göre en az 50 sn olmak üzere çoğaltılabilir.

KARAR SENİN HIZLI MI YAVAŞ MI?

HAZIRLIK



Kedi kuklasının seçimini yapıyoruz. Aşağıdaki kod işlemlerini bu kuklaya uyguluyoruz.



BU KODU EKLE

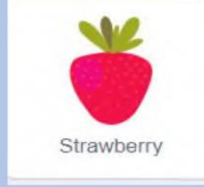
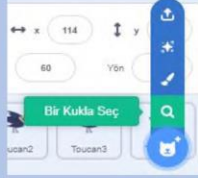
Oyunda seçimli hızlanma ödülü sonucunda kuklanın hızlı hareket etmesi için gerekli kod bloğu yazılır.

İPUCU

Hızlandırma haberini sal komutu hızlandırma işlemini yapacak kuklada yapılacaktır.

ÇİLEK ÖDÜLÜ

HAZIRLIK



Strawberry kuklası eklenir.İsmi çilek olarak değiştirilir.

BU KODU EKLE



1.resimdeki kod bloğu ile çilek kuklamızın gizlenip boyut ayarı %30 olarak belirlenir. Sürenin 10 sn olması ile birlikte ekranda görünmeye başlar ve 15 sn olduğunda ise ekranda kaybolması sağlanır. Çilek kedi kuklasına değdiği anda ödül olarak geçen süre azaltılır ve gizlenmesi sağlanır.

SENDE DENE



Oyunun çalışmasını kontrol edebilirsin.

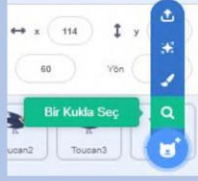
HATIRLATMA



Çilek kuklasının labirentin neresinde çıkmasını isteniyorsa oraya konumlandırılmalıdır.

ELMA ÖDÜLÜ

HAZIRLIK



Apple kuklası eklenir. İsmi elma olarak değiştirilir.

BU KODU EKLE



1.resimdeki kod bloğu ile çilek kuklamızın gizlenip boyut ayarı %30 olarak belirlenir. Sürenin 30 sn olması ile birlikte ekranda görünmeye başlar ve 40 sn olduğunda ise ekranda kaybolması sağlanır. Çilek kedi kuklasına değdiği anda ödül olarak geçen süre azaltılır ve gizlenmesi sağlanır.

SENDE DENE



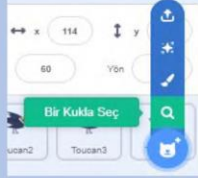
Oyunun çalışmasını kontrol edebilirsin.

HATIRLATMA

Elma kuklasının labirentin çilek kuklasından daha sonraki bir adıma konulması gerekmektedir. Çilek kuklası 10.sn ortaya çıkarken elma kuklası 30 sn ortaya çıkacaktır.

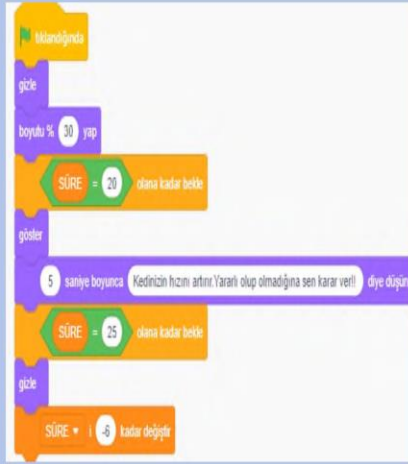
HIZLANDIRMA ÖDÜLÜ

HAZIRLIK



Lightning kuklası eklenir. İsmi hızlandırma olarak değiştirilir.

BU KODU EKLE



1.resimdeki kod bloğu ile çilek kuklamızın gizlenip boyut ayarı %30 olarak belirlenir. Sürenin 20 sn olması ile birlikte ekranda görünmeye başlar ve 25 sn olduğunda ise ekranda kaybolması sağlanır. Bu ödülümüzde kullanıcıya istek sunulur kedi kuklasının hızının artıp artmayacağı sorulur ve ekranda 5 sn boyunca bekletilir. Eyer kullanıcı bu hızlandırıcıyı kullanmak istemez ise süre kaybı yaşamaması için süreye 6 sn ekleme yapılır.2.resimde ise kedi ile hızlandırıcı kukla değişiyorsa hızlanma haberi salınır ve kedi kuklasının hızı artar.

SENDE DENE



Bayrağa tıklayarak artık oyunu oynayabilirsiniz.

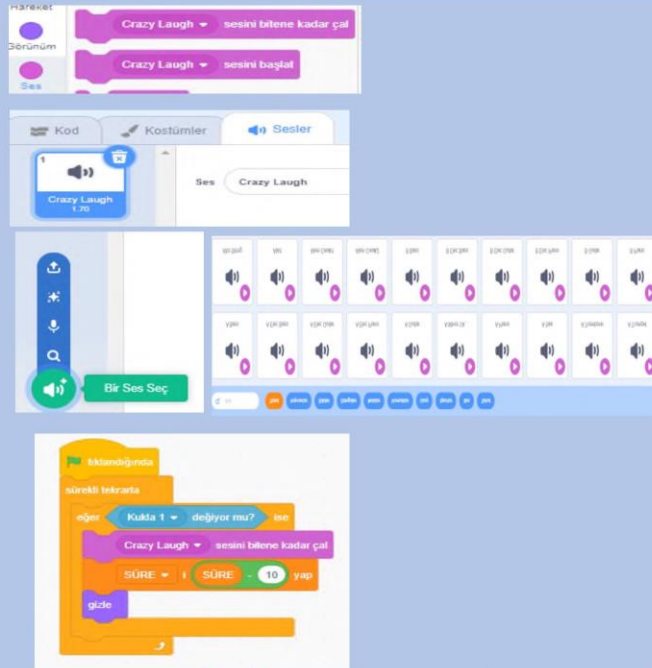
SES EKLEME

HAZIRLIK



Bu kısımda örnek olarak çilek kuklası seçilecektir. Aynı işlemler elma ve yıldırım kuklasına da uygulanacaktır.

BU KODU EKLE



Kuklaya ses eklemek için ses menüsünden sesini bitene kadar çal kod bloğu eklenir. Buradaki ses kod bloğundan seçilebileceği gibi kuklanın sesler sekmesinden, bir ses seç denilerek scratch uygulamasında bulunan ses eklenir. Bu uygulama için Crazy Laugh sesini seçiyoruz. Kukla1 in çileğe değdiği anda bu sesin çıkması sağlanır.

İPUCU

Sesler kısmında istenirse mikروفon yardımı ile ses kaydı yapılabilir veya kendi istediğiniz bir sesi mp3,waw uzantılı olarak eklenebilir.

SENDE DENE



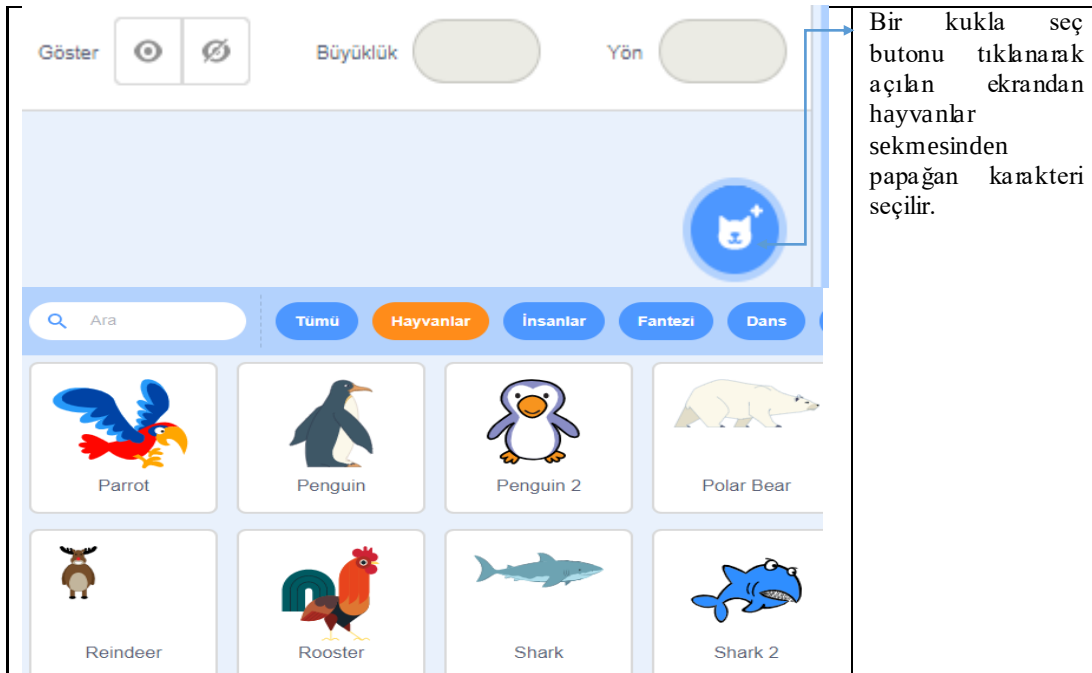
EK 2 GUTÜD Modeli İçin Hazırlanan Ders Planları

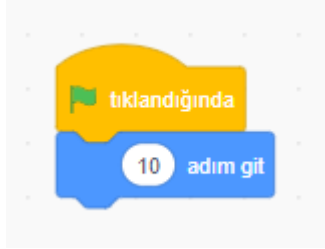
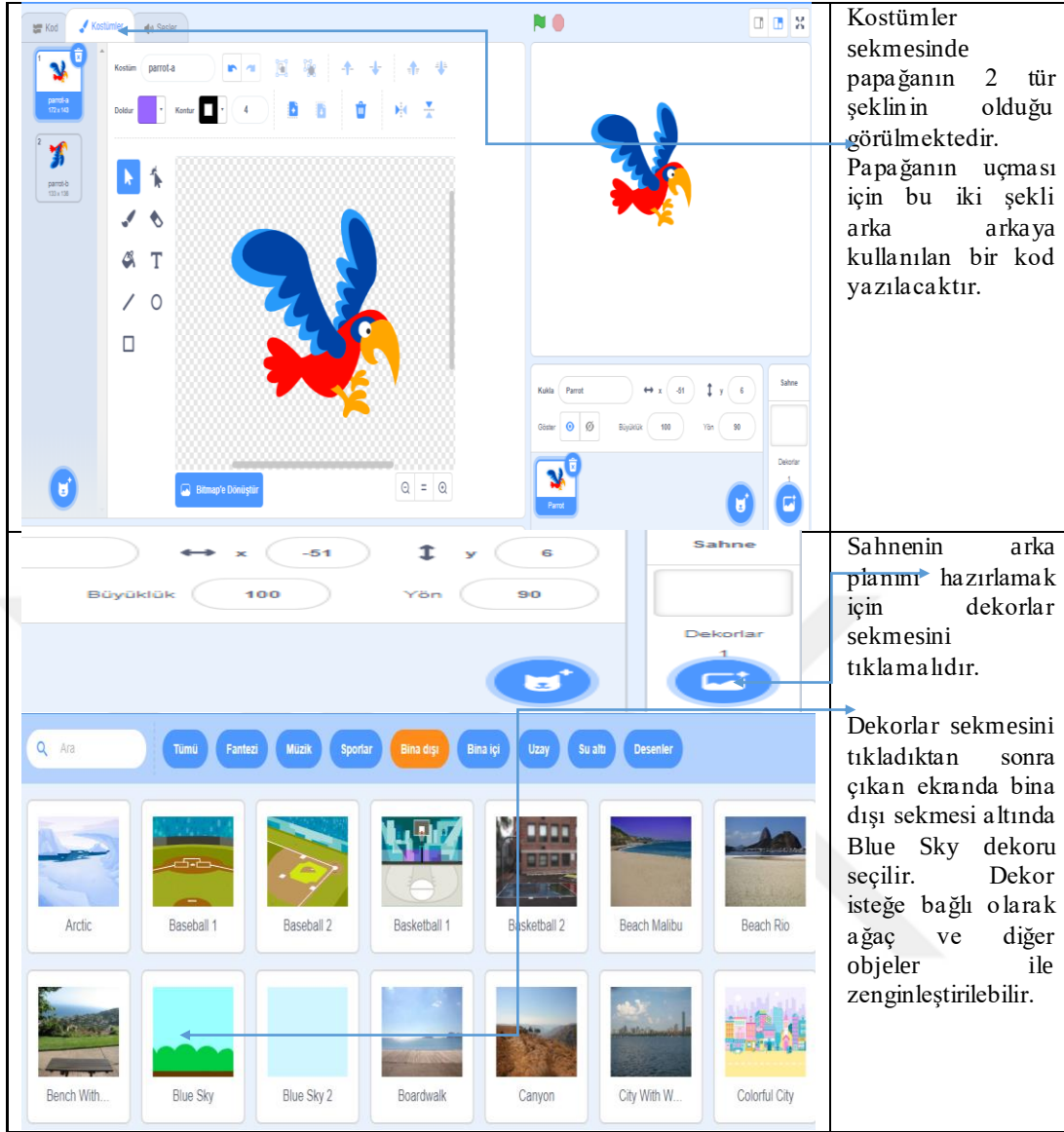
1.Adım: Gözle ve Uygula

1.1 Gözle: Papağının Kanat Çırparak Uçmasını Gösteren Animasyon

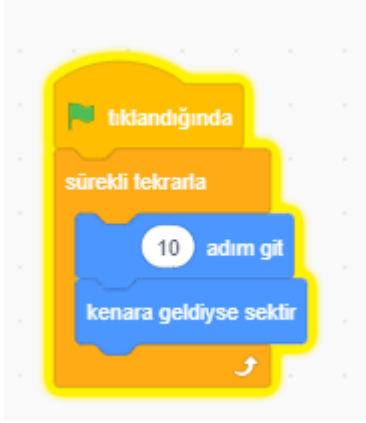
Animasyon resimlerin ya da hareketsiz nesnelerin arka arkaya hızlı bir şekilde gösterilmesiyle karakterlerin hareket ediyormuş gibi görünmesini sağlayan bir gösterim şeklidir. İlk animasyonlar birkaç kâğıda işlenen resimlerin çizilmesi ve kağıtların hızlıca geçirilmesi veya bir çemberin içine yerleştirilip döndürülmesi ile yapılıyordu. Çizgi filmlerde birer animasyondur. Scratch programı ile de kuklaların şekillerini değiştirerek bizde animasyon oluşturabiliriz. Scratch programının kostümler sekmesinde karakterlere birden fazla şekil çizmemize ve hareket ettirmemize olanak vermektedir.

Bu uygulamada bir papağan karakterlerinin kostüm değiştirerek sahnenin sağından soluna doğru uçuşmasını sağlayan bir program yazılacaktır. Bu kısımda öğretmen kendisi yapıp öğrencilerin program yapımını takip etmelerini isteyecektir. İlk aşamada sahneye gerekli karakterlerin seçilerek tasarım kısmının oluşturulması sağlanmalıdır. Tasarım kısmının nasıl oluşturulacağı aşağıdaki şekillerde belirtilmiştir.

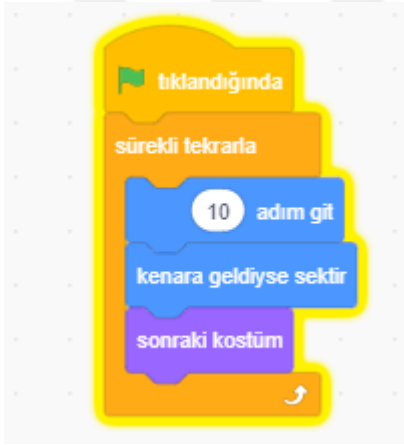




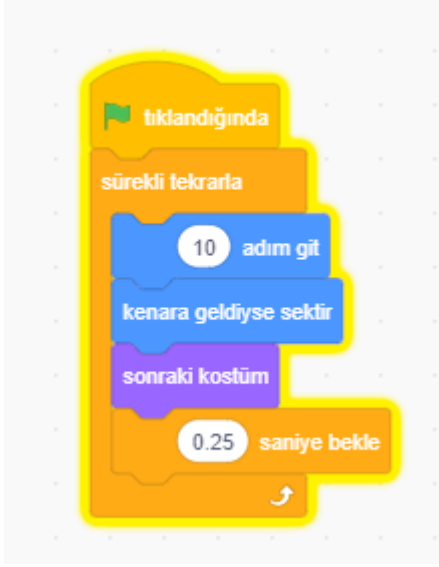
Sahneye ilk olarak yukarıdaki kod bloğu oluşturulur. Burada öğrencilere bu üç blok ile yeşil bayrağa tıkladığımızda papağan sadece sol tarafa 10 adım gider ve burada bekler. Papağan kuklasının devamlı olarak hareket etmesini sağlamak için sürekli tekrarla bloğunu kullanırız. Papağanın ekranda hareket ederken kenara gelip durmaması için kenara geldiyse sek kod bloğunu kullanarak hareket sağlandığı söylenir.



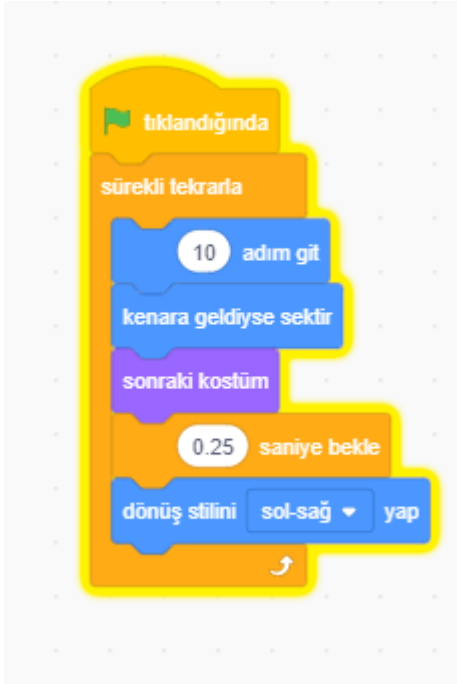
Sonrasında ise papağan karakterinin kostüm değiştirmesi için sonraki kostüm seçeneği seçilir. Bu bölümde öğretmen öğrencilere kuklanın kostümler sekmesini açar ve papağanın iki tür kostümü olduğunu gösterir. İstenirse buraya birden fazla kostüm eklenebileceği söylenir. Sonraki kostüm bloğu eklenir ve tekrar çalıştırılır.



Burada papağan karakterinin sonraki kostüm denilince çok hızlı hareket ettiği bunun ise animasyonda iyi gözükmediği belirtilir. Bu durumlarda saniye bekle komutunun kullanılıp hareketlerin yavaşlatılacağı belirtilir. Saniye süresinin ise kullanılacak programa göre değişebileceğine değinilir.



Kod bloğu tekrar çalıştırılıp öğrencilere papağan karakterinin aşağı yukarı şeklinde devamlı döndüğünü, bunun engellenmesi içinde dönüş stilini sağ-sol yap bloğunu kullanılması gerektiği söylenir. Bu bloğun içindeki döndürme seçeneğinin karakterin dönmeden sabit olarak gitmesini sağladığı tüm yönlerle denildiğinde aşağı yukarı olarak hareket ettiği sağ-sol yap denildiğinde ise karakterin sağ ve sol şeklinde yönünü çevirdiği söylenmelidir.

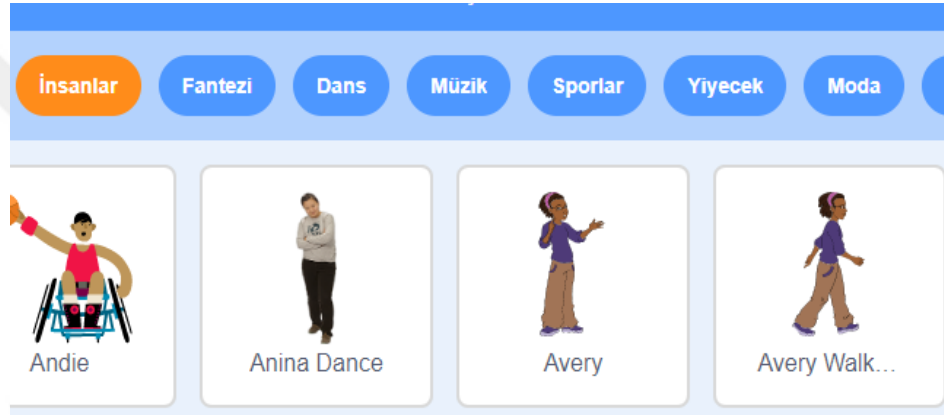


Dönüş stilini sol-sağ yap bloğu eklendikten sonra program çalıştırılıp papağan karakterinin sola sağa doğru hareket edip kanat çırpıdığı gösterilmelidir.

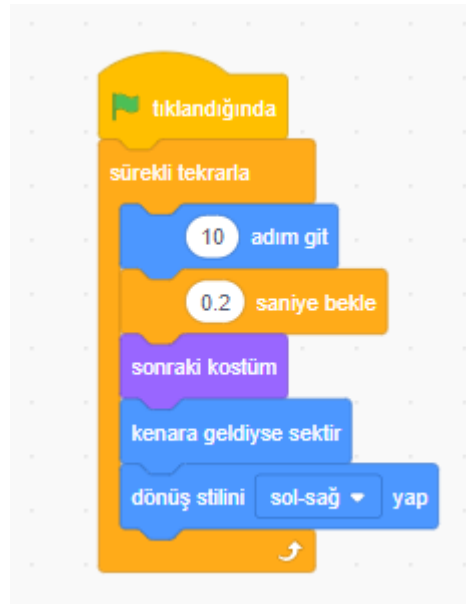
1.1 Uygula: Sağa-Sola Doğru Yürüten İnsan Figürü Yapımı

Bu kısımda öğretmen öğrencilerden sahneye yürüyen bir insan kuklası koyarak bu insan figürünün sahnenin sağına ve soluna doğru yürüyormuş gibi görünüp dolaşmasını isteyecektir. Bu kısımda öğretmen ilk başta öğrencilerin uygulamayı yapmalarını beklemelidir. Öğrencilerin zorlandığı kısımlarda yardımcı olmalıdır fakat tüm çözümü söylememelidir.

İlk olarak sahneye avery walking-d karakteri eklenir. Burada karakterin dört adet kostümü olduğu söylenir.



Sonrasında ise karakterin yeşil bayrağa tıklanığında sağa sola doğru hareket etmesi için aşağıdaki kod blokları eklenir.

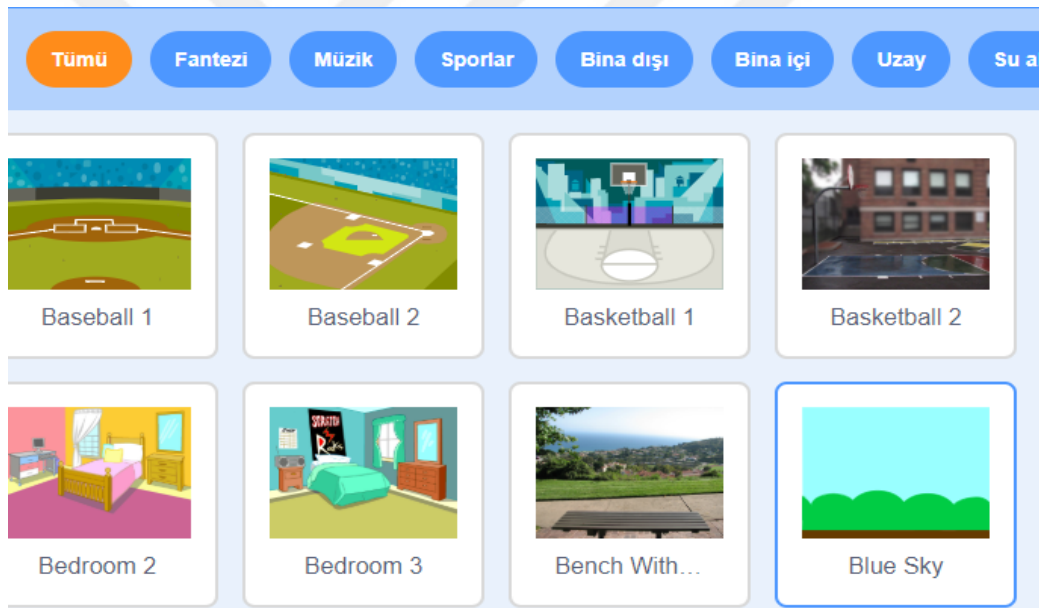


Öğrencilerinde bu işlemleri yapması istenir ve kuklanın sağa - sola doğru yürümesi sağlanır.

1.2Gözle: Ekranda Fare İmlecini Takip Ederek Uçan Kedi Animasyonu Yapalım

Scratch programında fare imlecimizin programa eklediğimiz kuklalar tarafından takip edilmesini sağlamak için fare imlecine git kod bloğu kullanılmaktadır. Burada dikkat edilmesi bloğunun sürekli tekrarla altında çalıştırmamız gerekmektedir. Örnek verecek olursak yeşil bayrağa tıkla dediğimizde sadece fare imlecine git komutunu eklersek komut bir kere okunur ve bir defa fare imlecine gider ve kalır.

Bu etkinlikte ekranda fare imlecini takip eden bir kedi animasyonu yapılacaktır. İlk olarak sahne seçimi yapılır. Burada sahne olarak Blue sky seçilmiştir.

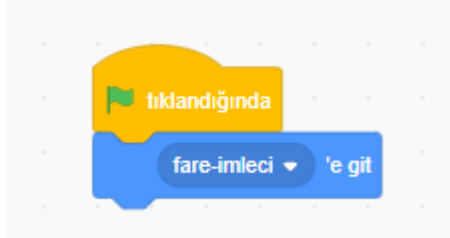


Kukla seçimine geçilir. Kukla olarak Cat Flying seçimi yapılır.



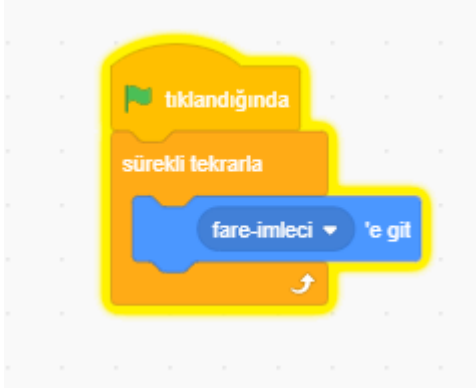
Sahne ve dekor ayarlamalarından sonra kedi kuklası seçilerek kod bloğu oluşturulmaya başlanır.

Kedi kuklasına yeşil bayrağa tıklandığında kod bloğu eklenir. Kedi kuklasının fare imlecinin olduğu yere gitmesini sağlamak için fare imlecine git kod bloğunun kullanılacağı gösterilir ve bu kod bloğunun iki kısımdan oluştuğu diğer kısmının ise rastgele git özelliğidir. Bu özellik ile kuklaların ekranda rastgele bir yere doğru hareket etmesi sağlanmaktadır. Bu bölüme kadar olan kod blokları çalıştırılır.



Burada yeşil bayrağa tıklanır kedinin fare imlecinin bulunduğu konuma geldiği fakat daha sonrasında farenin hareket etmesiyle kedinin hareket etmesi sağlanmadığı görülmüştür. Bu olayın sebebinin sadece bir kere işlemin gerçekleştiğinden kaynaklandığından bahsedilir. Programda ise bu olayın devamlı olarak istenildiği için sürekli tekrarla komutunu kullanarak fare imlecini devamlı olarak kedinin takip etmesi sağlanır.

Dikkat: Rehber öğretmen bu kısımda scratch programında ve diğer yazılım programlarında bir veya birden fazla işlem satırını, bir koşula bağlı olarak, belirli sayıda veya bir koşul sağlandığı sürece tekrarlayarak çalıştıran kalıplara döngü denildiğini söylemelidir. Scratch programında birçok döngünün olduğunu bunları zamanla kullanacağımız söylenmelidir.



1.2 Uygula: Ekranda Rastgele Hareket Eden Balon

Bu kısımda öğretmen öğrencilerden bir balon kuklası alıp bu balon kuklasının ekran da belirli bir hızda ekran da hareket etmesini isteyecektir. Öğretmen bu kısımda uygulamanın öğrenciler tarafından yapılması için belirli bir süre vermelidir. Öğrencilere takıldığı noktalarda yardımcı olabilir ama çözümün hepsi verilmemelidir.



Ekran yandaki şekildeki balonun eklenmesi sağlanır.

Balon eklendikten sonra kod kısmına gelinir ve aşağıdaki işlemlere devam edilir.



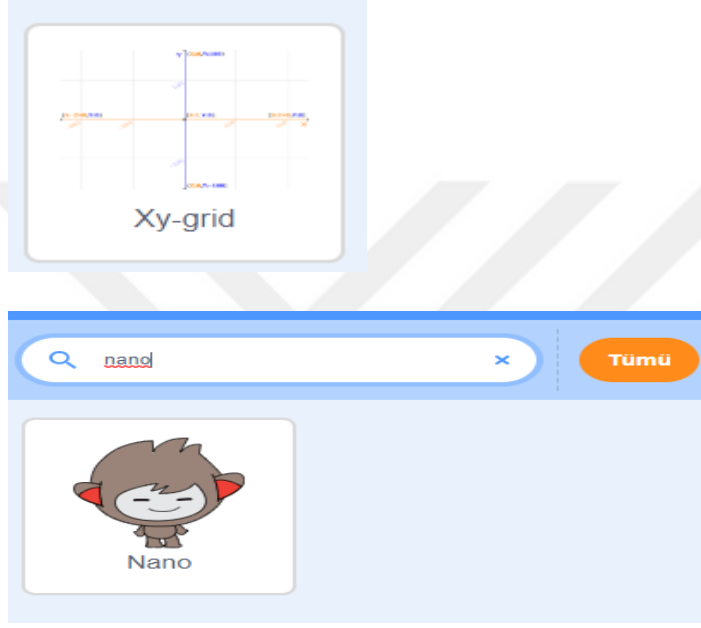
Tıklandığında kod sekmesi konularak kod bloklarının çalışması sağlanır. Sürekli tekrarlar kod bloğu ile içerisinde bulunan kodların devamlı bir biçimde çalışmasına olanak verilir. Rastgele git kod bloğu konularak ekrandaki balonun ekranda rastgele gezinmesi sağlanır. Bu bölümde öğretmen öğrencilerin kendi

yapması için bekler ve bazı ipuçları ile sadece yardımcı olur.

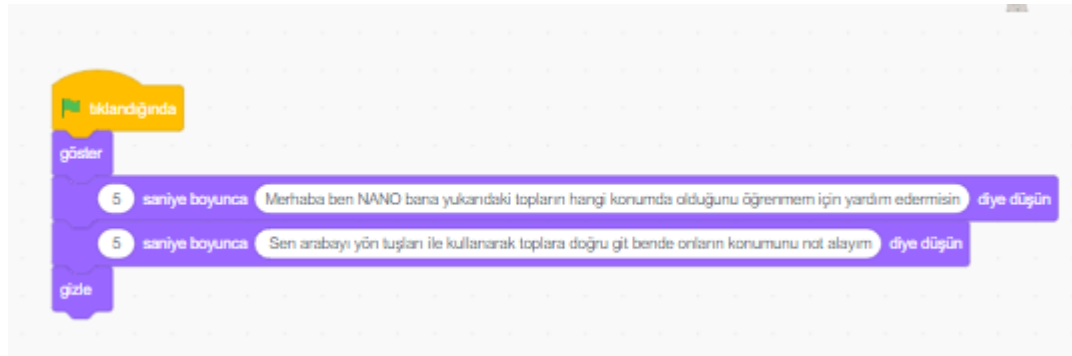
1.3Gözele: Konum Bulmaca Oyunu Yapma

Bu oyunda amacımız öğrencilerin koordinat düzlemi hakkında bilgi almaları sağlamaktır. Ekranda bulunan araba kuklasının mavi, sarı, yeşil ve kırmızı toplara değdiğinde arabanın bulunduğu konumu söylemesi istenir.

İlk önce sahne seçilir. Sahnenin koordinat sistemi ile ilgili olması için xy-grid seçilir.



Oyunu oluşturmak için Nano kuklasını seçiyoruz. Ve aşağıdaki kod bloklarını oluşturmaya başlıyoruz.

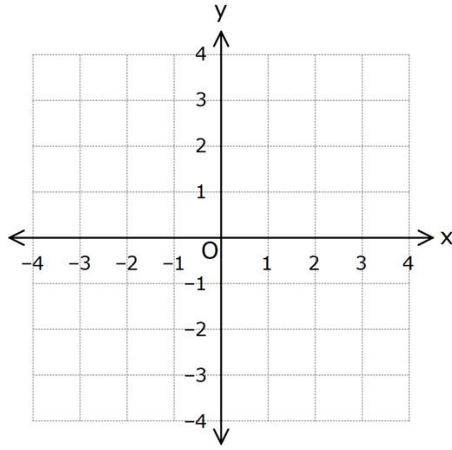


Burada kuklanın 5 saniye boyunca oyunda olacağını anlatmasını sağlıyoruz.

” Merhaba ben NANO bana yukarıdaki topların hangi konumda olduğunu öğrenmem için yardım eder misin?”

“Sen arabayı yön tuşları ile kullanarak toplara doğru git bende onların konumunu not alayım”

Şeklinde uyarı cümleleri eklenmiştir. Kuklamız oyunun başında ekranda gözükecek sonrasında gizlenecektir. Burada gizle komutunu kullanmamızın sebebi, cümleleri oluşturduktan sonra kuklanın ekrandan kaybolması, göster kod bloğunun kullanılmasının sebebi ise, kuklanın yeşil bayrağa basılınca oyunun tekrar başlangıcında sahnede gözükmesi için kullanılmaktadır.



Dikkat: Rehber öğretmen öğrencilere ilk başta koordinat sisteminin ne olduğu noktasında bilgi vermelidir bunun için yan taraftaki şekle benzer bir çizimi tahtaya çizmelidir. Öğrencilere scratch programında sahne, yatay eksen (x)

$240+240=480$ pixel, dikey eksen (y) ise $180+180=360$ pixel bir dikdörtgen çalışma alanı sunar. Sahnenin tam ortasında $x=0$ ve $y=0$ koordinatlarını içerir. Örnek verecek olursak herhangi bir kuklayı 1.bölümde olmasını istersek x ve y sayılarına pozitif sayılar girilmelidir.

Koordinat sistemi açıklamasından sonra rehber öğretmen tarafından sahneye araba kuklası eklenir. Arabanın sağa-sola yukarı-aşağı hareket etmesi için kod blokları oluşturulur. Örnek olarak yukarı ok tuşuna basılınca komutu ile kuklanın y konumunda ve pozitif yönünde yani yukarı gidebilmesi için 10 yazılır. Aşağı yönlü gidebilmesi içinde -10 yazılır. Diğer konumda da bu uygulanır.

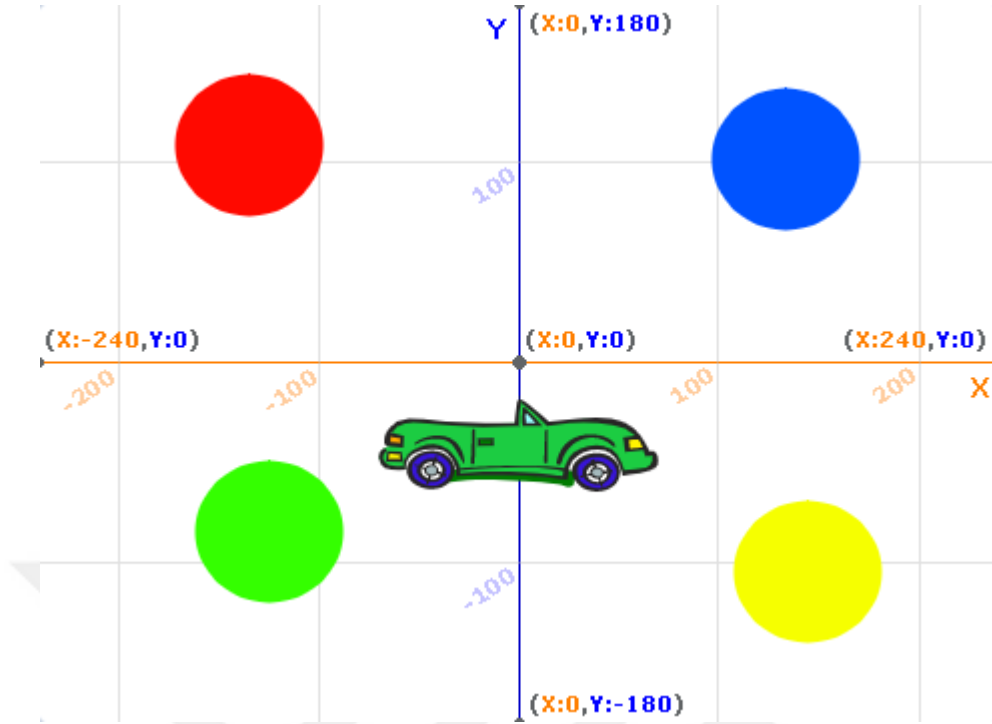


Bu kısımda rehber öğretmen yukarı ok tuşuna basıldığında y konumunda hareket edileceği ve pozitif yönde gidileceği için 10 yazıldığı, aşağı ok tuşuna basıldığında y konumunda ve negatif yönde gidileceği için -10 yazıldığı vurgulanmalıdır. Bu işlemin x koordinatı içinde geçerli olduğu vurgulanmalıdır.

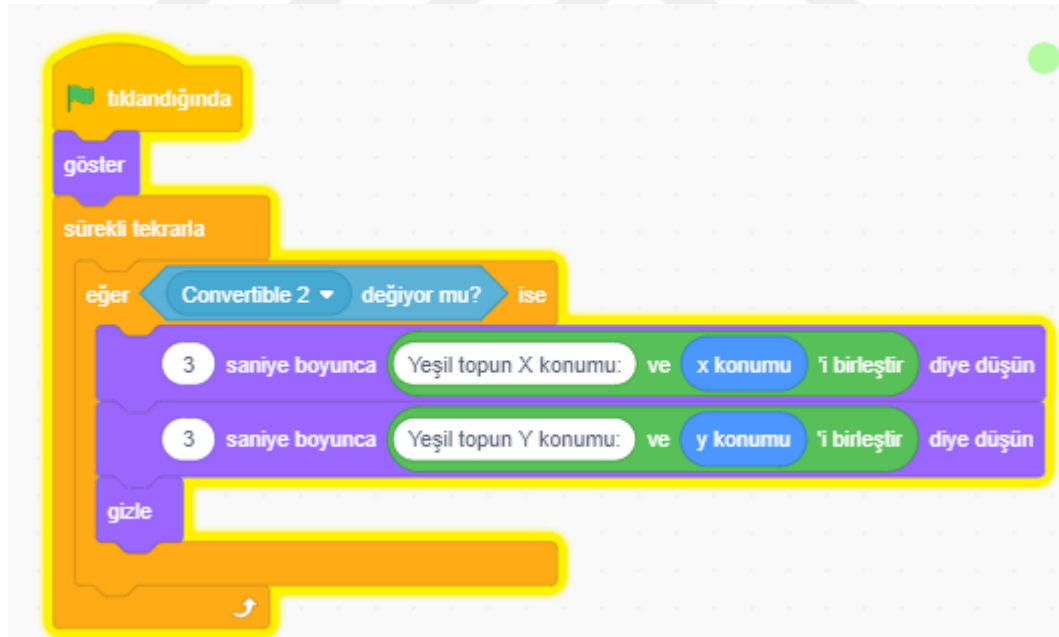
Arabamızın ilk oluştuğunda hiçbir yere değmemesi için 0,0 konumu için aşağıdaki kod bloğu yazılır.



Sahnemize 4 adet top kuklası eklenir. Ve aşağıdaki konuma getirilir.



Yeşil top kuklası seçilerek kod blokları yazılır.



Burada “eğer” kod bloğu bir şart ifadenin gelip gelmediğini sorgular. Başka bir deyişle şart ifadenin sonucu “doğru” veya “yanlış” değer üretir. Bu örnekte de convertible2 (araba) yeşil topa değip değmediği kontrol edilmektedir.” Değişiyor mu “komutu ile sahnede bulunan tüm kuklaların, değme işlemi ve fare ile temas işlemi kontrol edilebilir. “Birleştir “komutu ile yazdığımız yazıları veya içine

koyduğumuz, kod bloğundaki koordinat sistemindeki yazıyı birleştirme işlemi sağlanır. Kuklamızın gizle komutuyla kaybolması sağlanır.

Bu kod bloklarının aynısını diğer top bloklarına da eklenir.

1.3 Uygula: Koordinat Düzleminde Bölgeleri Öğrenme Animasyonu

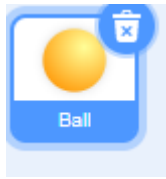
Bu uygulama da amaç ekranda serbest bir biçimde hareket eden bir topun koordinat sistemleri üzerinde gezmesi ve bölgeler üzerine gelince hangi bölgede olduğunu yazmasıdır. Bu kısımda rehber öğretmen bir önceki konuda koordinat sistemlerinin bölgelerini tekrardan hatırlatır ve öğrencilerin diğer kısımlarını yapması için belirli bir süre tanır. Gerekli yerlerde yardımcı olabilir ama çözümün tamamı verilmez.

Animasyonda x ve y koordinatlarının hangi bölgelerde – veya + olduğunu ekranda hareket eden bir top vasıtasıyla gösterilir.

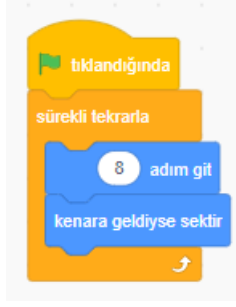
Animasyon için dekor seçimi yapılır ve sahnenin koordinat sistemi ile ilgili olması için xy-grid seçilir.



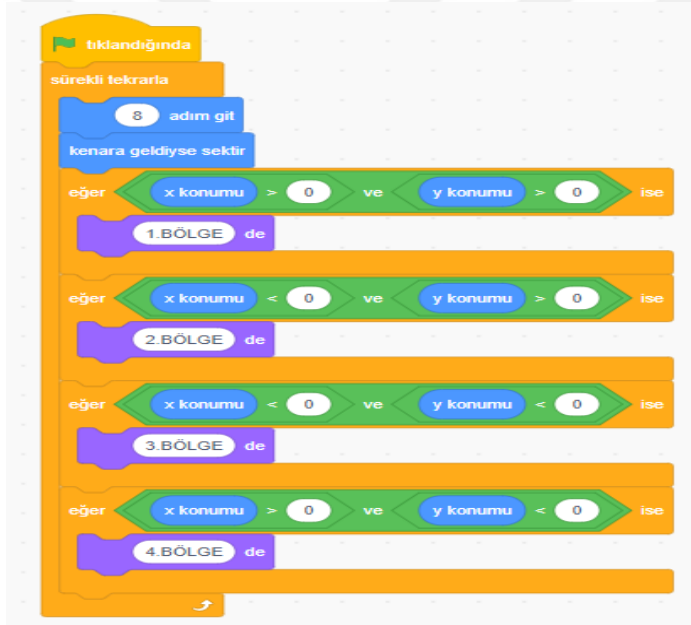
Bir tane top kuklası seçilir ve sahneye eklenir.

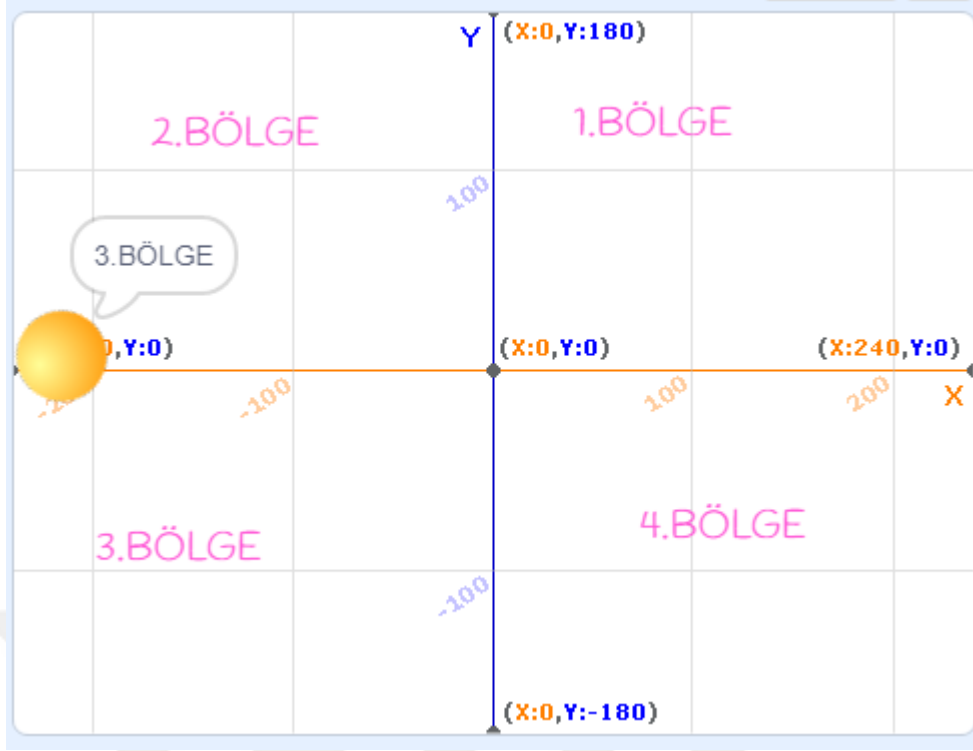


Topun sahnede bölgeler arasında hareket etmesi için top kuklası seçilir ve aşağıdaki kod blokları oluşturulur.



Topun hangi bölgede olduğunu ekrana yazdırmak için topun x konumunun 0'dan büyük ve y konumunun da yine 0'dan büyük olması durumunda ekrana I. bölge de yazısı yazılacaktır. Burada “ve” komutu ile her iki şartında sağlandığı durumda komut çalışmaya başlar. Diğer bölgeler içinde aynı kod bloğu uygulanır.





1.Tasarla ve Üret

1.1 Tasarla: Kuş Yakalama Oyunu

Bu bölümde öğrencilerden fare ile hareket eden kuklanın sahnede uçan kuşları yakalama oyunu yapmaları istenecektir. Ekrana eklenen istenilen sayıdaki kuşun seçimi yapılan herhangi bir kukla tarafından yakalanması istenilmektedir. Bu oyun için öğrencilerden ilk önce tasarım ve kodlama aşamalarının nasıl olacağı konusunda düşünceleri ve bunu yazmaları istenir. Bu sırada öğretmen kesinlikle çözümü ve yapılacak aşamaları söylememelidir, sadece rehberlik edebilir.

Örnek olarak

- hangi kuklalar eklenmeli
- kuklaların hareketli görünmesi için gereken kod bloklarının neler olduğu belirlenmesi
- kuklaların birbirine değdiği anda hangi kod bloklarının kullanılacağına karar verme

-kuklanın ekranda fare imlecini takip edebilmesi için hangi kod bloklarının kullanılması gerektiği

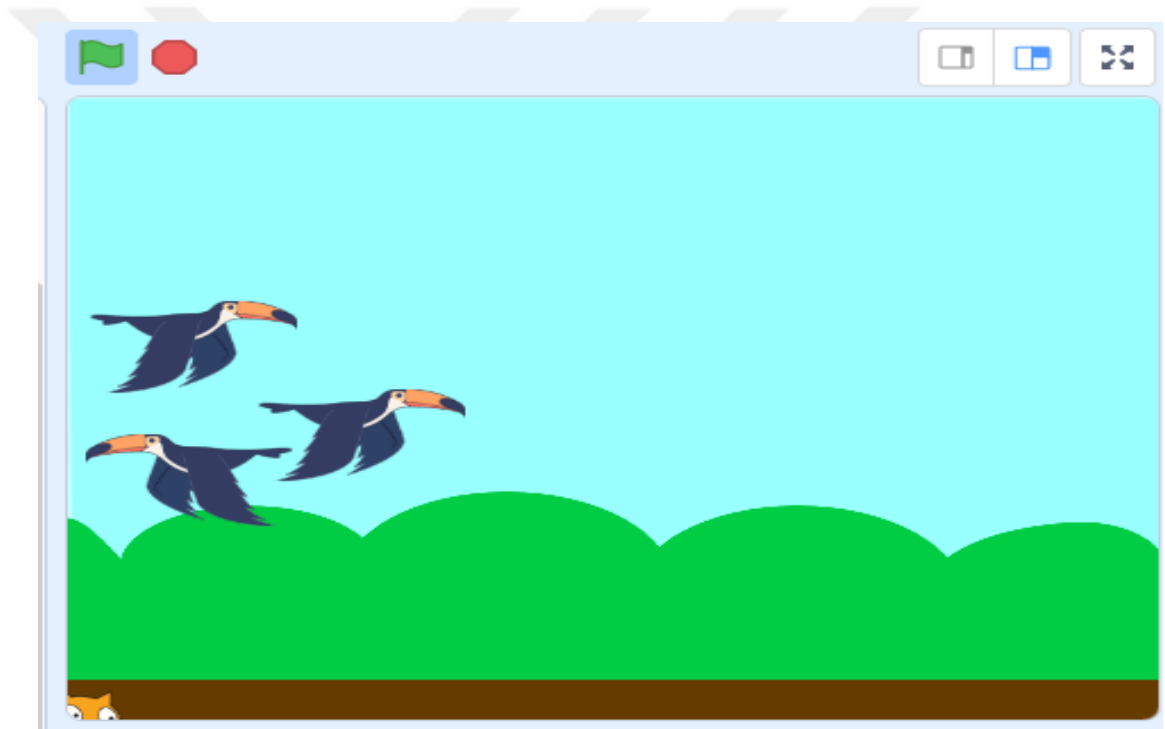
-ekrandaki kuşların ilk başlangıçta hepsinin ekranda gözükmesi ve daha sonra gizlenmesi

Gibi adımların oluşturulması ve bir algoritma oluşturulması

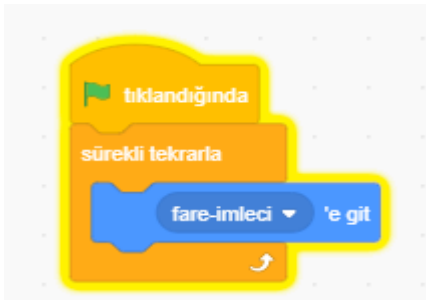
1.1üret: Kuş Yakalama Oyunu

Öğrenciler tasarımlarını ve algoritmalarını oluşturduktan sonra scratch programında uygulamaları istenir.

Aşağıdaki örneğe benzer bir örnek oluşması beklenir.



Kedi hareketi için kod:



Kuşlar için kod



1.2 Tasarla: Binalar Arasından Uçan Kedi

Bu bölümde öğrencilerden ekranda kayar gibi hareket eden binaların arasında uçan kedi yapmaları istenmektedir. Binalar yükseklik olarak azalıp artmaktadır. Kedinin fare imlecini takip etmesi ve binalardan kaçması beklenmektedir. Binalardan herhangi birine değdiğinde de oyunun bitmesi amaçlanmaktadır. Burada öğrencilerin ilk başta kukla ve dekorasyon aşamasının nasıl olacağı ve hangi kod bloklarını kullanacağını yazmaları istenir.

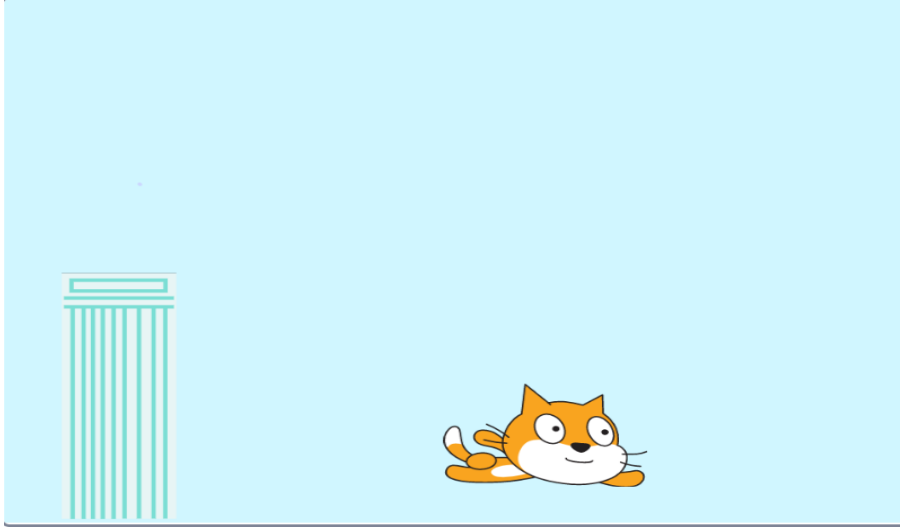
Örnek olarak

- Kullanacak kuklalar Cat Flying, Buildings
- Binaların nasıl ekranda kayar halde görünebileceğini düşünmeleri gerekir
- Binaların az katlı çok katlı olarak nasıl geleceği
- Kayan binalarla kedinin değdiği anda hangi kod bloklarının kullanılacağına karar vermeleri beklenir.

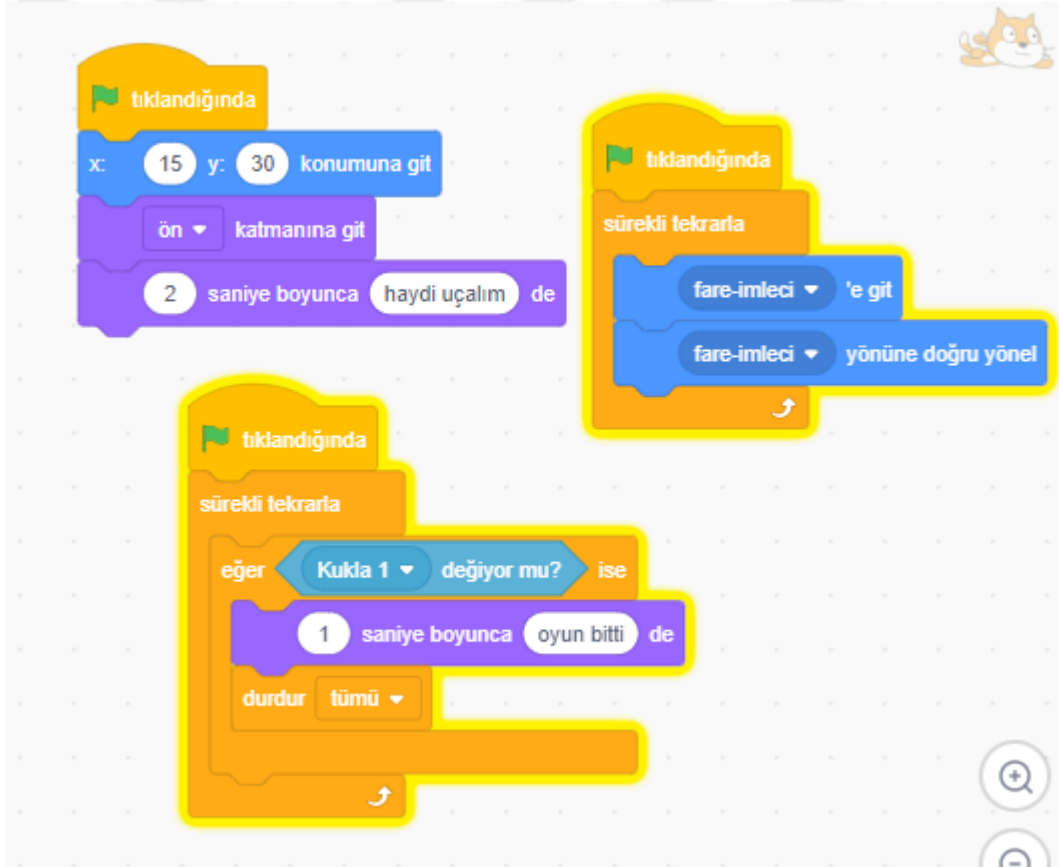
1.2 Üret: Binalar Arasından Uçan Kedi

Öğrenciler tasarımlarını ve algoritmalarını oluşturduktan sonra scratch programında uygulamaları istenir.

Aşağıdaki örneğe benzer bir örnek oluşması beklenir.



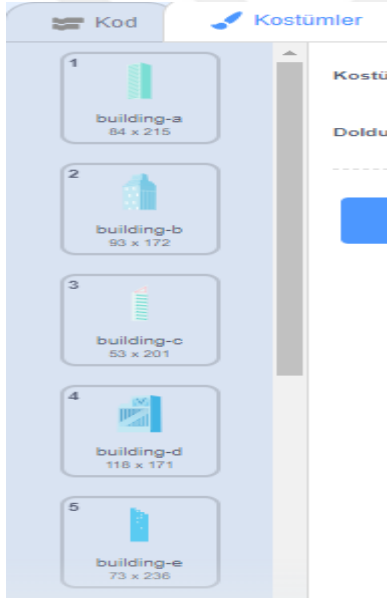
Uçan kedi için yazılmış kodlar



Binalar için yazılması gereken kodlar



Ayrıca binaların boyutları kostümlerden ayarlanabilir isteğe göre



1.3 Tasarla: Arkadaşının Sınavdan Aldığı Notu Söyleyince Dans Eden Kız

Bu bölümde öğrencilerden ekranda 2 adet insan kuklası bulunması istenecektir. Burada bir kukla diğer bir kuklaya koşarak yaklaştığında duracak ve sınavdan 100 aldığını söyleyecektir bunu duyan diğer kukla sevinip dans etmeye başlayacaktır.

Örnek olarak

-Ali karakterinin Ayşe karakterine yürümesi için hangi kod bloklarını kullanılmalıdır

-Ali'nin Ayşeye değdiği anda konuşmanın gerçekleşmesi için hangi adımların olması gerekir

-Konuşma metinlerinin sırasının ayarlanması için

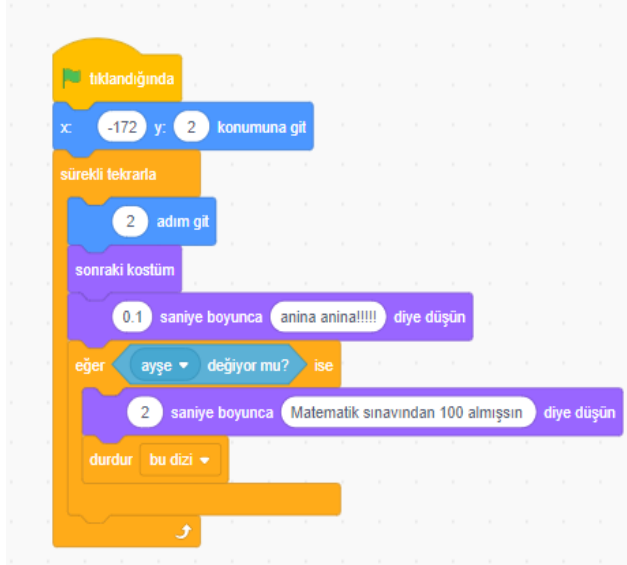
1.3 Üret: Arkadaşının Sınavdan Aldığı Notu Söyleyince Dans Eden Kız



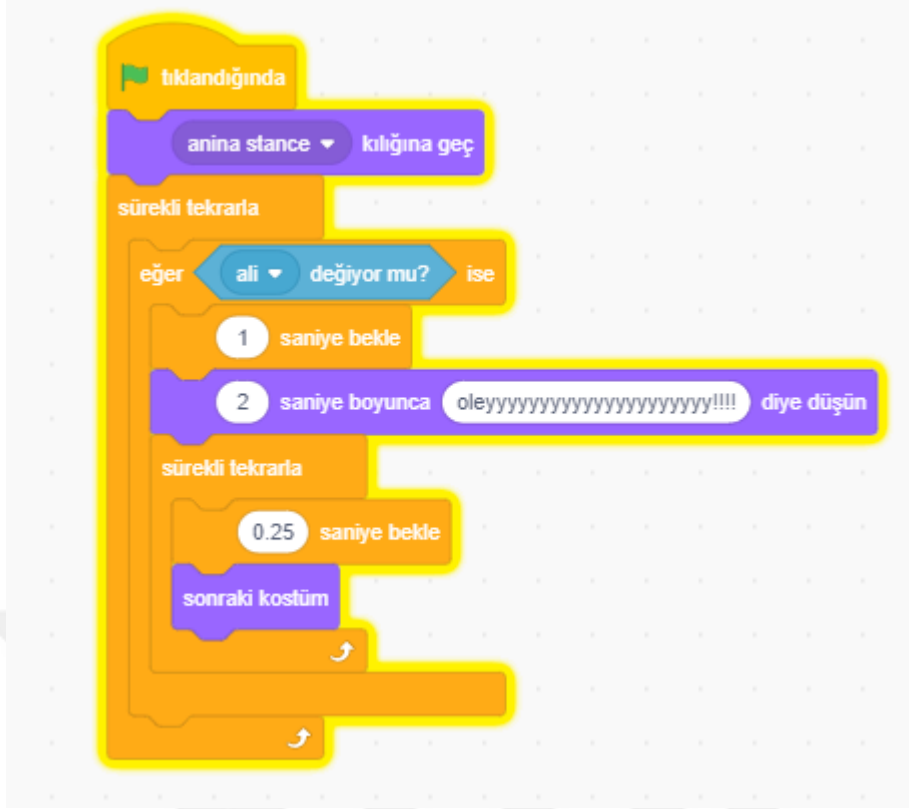
Sahnemize iki adet insan kuklası yüklüyoruz. Bu kuklaları tercih etme sebebimiz bir tanesinin yürüme kılıklarını içermesi diğerinin ise dans etme kılıklarını içermesidir.

Anina olan kuklanın ismini Ayşe Jaime isimli kuklayı Ali ismi ile değiştiriyoruz.

Ali kuklamızı tıklayarak aşağıdaki kodu yazıyoruz.



Sonrasında ise Ayşe kuklasını tıklıyoruz ve aşağıdaki kodu yazıyoruz.



Öğrencilerinde buna benzer bir animasyon çıkarmaları beklenir.

1.1 Değerlendir

A) Rehber öğretmen aşağıdaki soruları sınıfa sorarak, öğrencilerin soruları tartışmaları sağlar.

1-Scratch uygulamasında bazı işlemlerin devamlı olarak programda çalışmasını istemekteyiz bu amaçla döngüleri kullanmaktayız. Döngüler günlük hayatımızda hangi aşamalarda karşılaşmaktayız. Tartışınız.

2-Scratch uygulamasında en sık kullandığımız konum yerleştirme işleminde x ve y koordinatları ile hangi tarafın negatif hangi tarafın pozitif olduğu konularında zorlandınız mı? Yönleri ayarlamak için hangi yöntemleri kullandınız.

B) Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).

- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- o Karakterler arası geçiş yapmayı anlamakta zorlandım.

- o Kodların hangisinin önce hangisinin sonra kullanılacağını anlamakta zorlandım.

- o Karakterleri oluştururken zorlandım.

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?

- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?

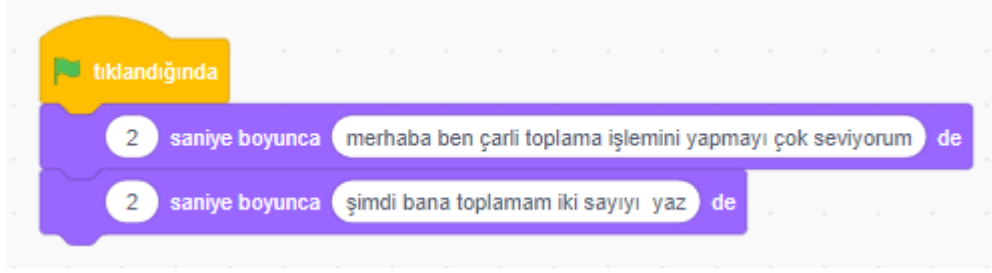
- Grup arkadaşınızdan ne öğrendiniz?

2.Adım: Gözle ve Uygula

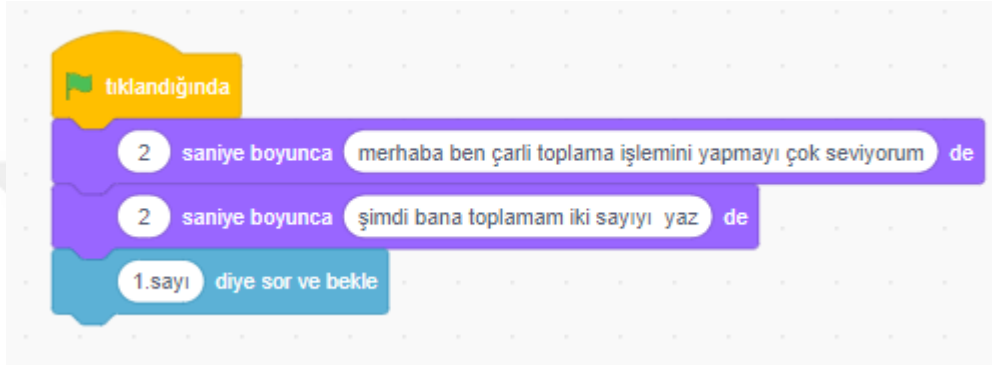
2.1 Gözle: İki Sayıyı Toplama İşlemi

Scratch programında toplama işlemi yapabilmek için operatörler ve değişkenler sekmeleri kullanılır. Değişken: Girdiğimiz değerleri alan veya programın çalışmasıyla bazı değerlerin atandığı veri tutuculardır. Değişkenleri kullanarak bellekte yer açarız ve yapmış olduğumuz işlemleri bellekte tutmalarını sağlar. Rehber öğretmen bir değişken sadece bir değer tutulabileceğini vurgulamalıdır. Değişkenlerin daha iyi anlaşılabilmesi için toplama işlemi yapan bir programda inceleyelim.

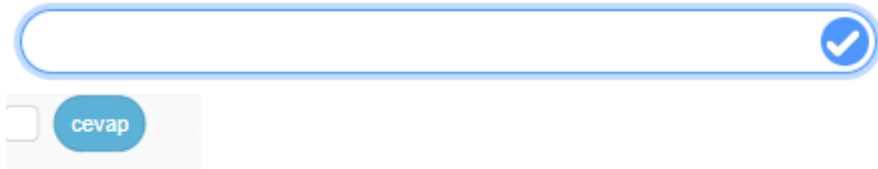
İlk olarak scratch programında bir adet kukla ekleyelim. (Günlük yaşamdan örnekler)



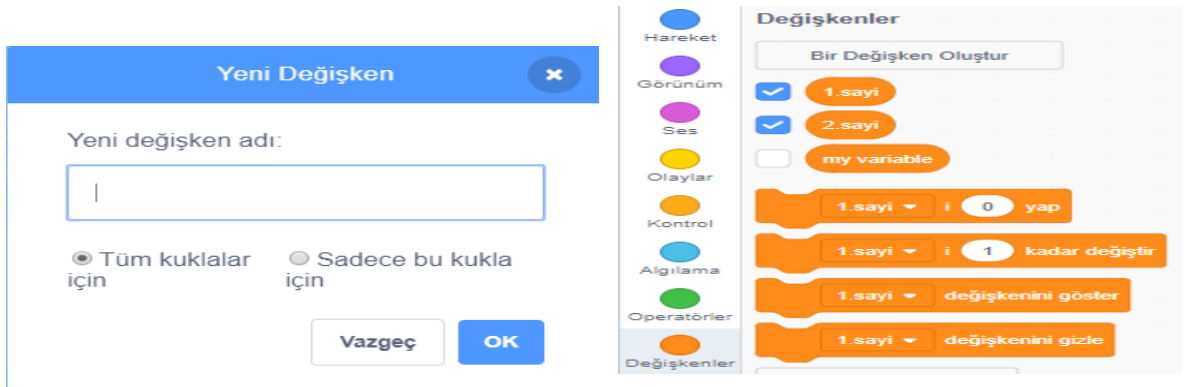
Kuklaya yukarıdaki konuşma metinleri eklenir ve programın nasıl çalışacağı hakkında bilgi verilir.



İkinci adım olarak “diye sor ve bekle” komutu eklenir. Bu komut ile kullanıcıdan cevap almak istediğimiz durumlarda kullanılır.



Ekranda çıkan bölüme gerekli olan sayı yazılmasını sağlar ve bunu cevap olarak tutar. Burada verilen cevabın belleğe kaydedilebilmesi için bir adet değişken oluşturulur.

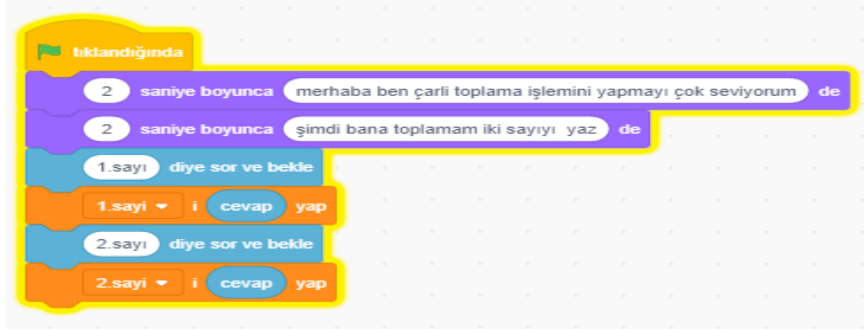


Buradan “Bir Değişken Oluştur” seçeneği seçilir.

Yeni deęiřken adı kısmına deęiřkene verilecek isim yazılır. Bu bölüme 1.sayı yazılır ve ok tuřuna basılır.



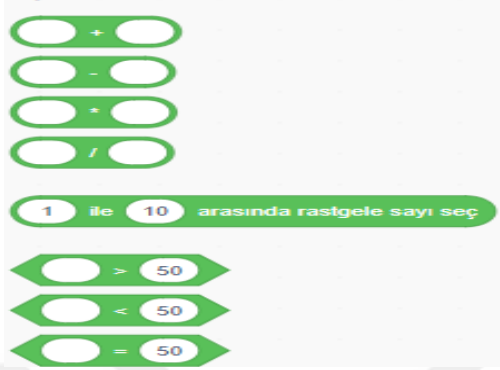
1.sayıya girilen cevabı 1.sayı deęiřkeninin iine aktarıyoruz. Bellekte kullanılmak iin bekletilir.



Aynı iřlemleri de 2.sayı iinde uygulanır.



Toplama işlemini ekrana yazdırmak için metin kutusu içine birleştir kod bloğu eklenir. Toplama işlemini yapabilmek için operatörler kısmında + işlemi alınır. Operatörler kısmında toplama, çıkarma, çarpma, bölme, rastgele sayı üretme, büyük, küçük işlemleri de yapılır.



Burada toplama operatörü seçilir ve 1.sayı değişkeni ve 2.sayı değişkeni seçilerek değişkenlerin içindeki değerlerin toplanması sağlanır.

1.sayı 25
2.sayı 36



2.1 Uygula: İki Sayıyı Arasında Çıkarma İşlemi

Bu kısımda öğretmen öğrencilerden kullanıcıdan istenen iki sayı arasında çıkarma işlemini yapmalarını isteyecektir. Rehber öğretmen bu kısımda öğrencilere yapması gereken işlemi söyler ve belirli bir süre verir. Bu kısımda bazı noktalarda rehber öğretmen yardımcı olabilir fakat cevabın tümü verilmez.

Ekrana bir tane kukla eklenir. Konuşma metinleri ile ilgili kodlar yazılır.

1.sayı 65
2.sayı 25

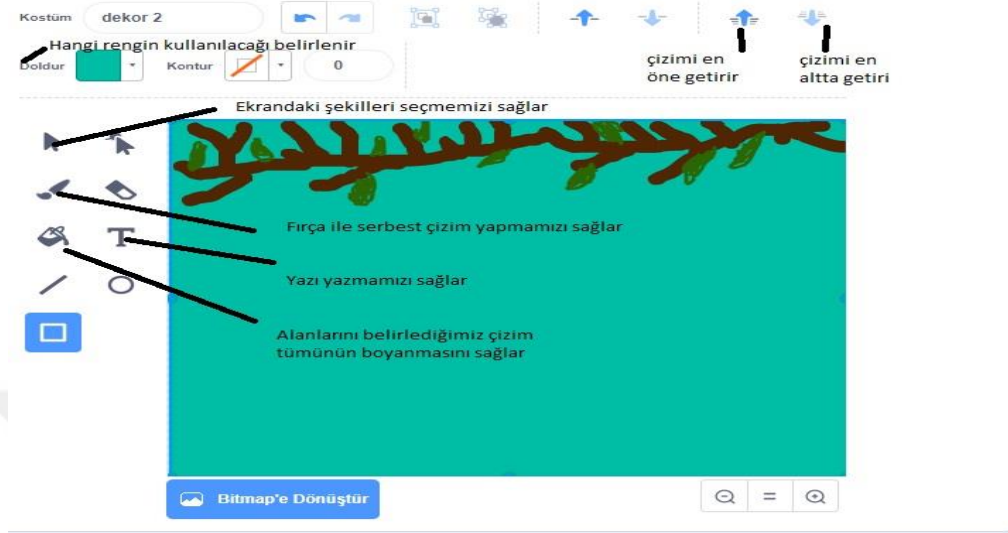


Bu kısımda operatörlerden  kullanarak çıkarma işlemi gerçekleştirilir.

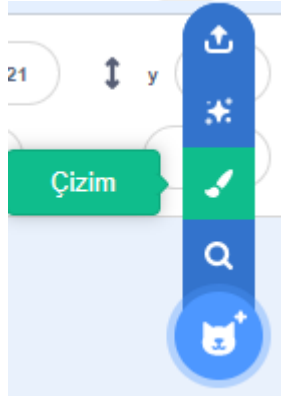
2.2 Gözle: Kendi Oluşturduğumuz Dekorda Rastgele Konumda Aşağı Düşen Elmayı Yakalama Oyunu

Bu uygulamamızda yukarıdan rastgele konumda (x konumu devamlı olarak değişen) bir elma kuklası ile aşağıda bulunan bir adet çubuk sepet ile düşen elmaları yakalamaya çalışacaktır. Elma, çubuk sepete değdiği anda üst kısma çıkarak tekrar düşme işlemi sağlayacaktır. Alt kısımda bulunan kırmızı engelle değdiği anda oyun bitecektir. Toplanan elma sayısı bir değişkende tutularak yazılacaktır.

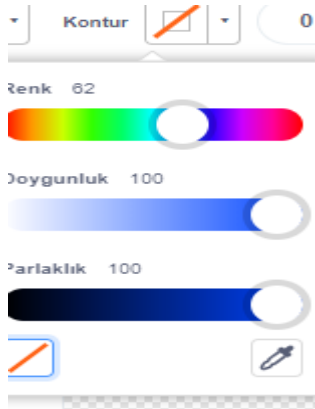
Bunun için ilk önce dekorlardan kendi çizimimizle bir dekor oluşturulur. Dekor sekmesi bu arada anlatılır.



Yukarıdaki gibi bir şekil oluşturulur ve dekor sekmesi tanıtılır.



Bir kukla ekle seçeneği ardından çizim seçilir ve bir adet çubuk çizilir. Dikdörtgen sekmesi kullanılarak çizim yapılır.



Burada dikkat edilmesi gereken çizilen dikdörtgenin konturu ile dolgu boyasının aynı renkte olmasıdır. Diğer bir seçenek ise kenar çizgisinin yandaki şekildeki gibi gizlenmelidir.



Çizilen çubuk sepet isimlendirilir. Çubuk sepet kuklası seçilerek kod blokları yazılır.

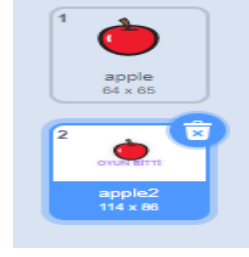
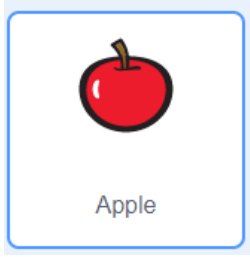


Bu kısımda sepetin hangi konumda olacağı belli edilir.

Bu kısımda öğretmen öğrencilere *sepetin y konumunda sabit kalıp x konumunda nasıl hareket ettirileceği* sorulur.

Sorunun cevapları beklendikten sonra yandaki kod bloğu yazılarak açıklama yapılır.

-Kukla seç menüsünden apple yazılarak elma kuklası seçilir. Apple kuklasının adı elma olarak değiştirilir. Kostümler sekmesinden elmanın bir kopyası çıkarılarak ikinci kostüme oyun bitti yazılır.



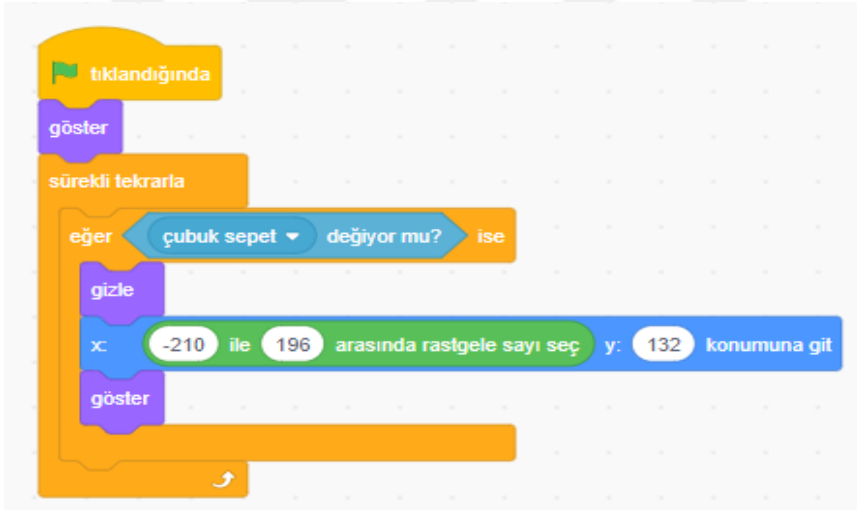
Elma

kuklası seçilir ve kod blokları oluşturulur.

Elma kuklasının yukarıdan aşağı doğru hareket etmesi için yandaki kod bloğu yazılır.



Bu kısımda öğretmen öğrencilere *elmanın çubuk sepete değdiği anda tekrardan ağaç dalında bitip yeniden aşağı düşmüş gibi gözükmelerini isteyen kod bloklarını açıklar.*

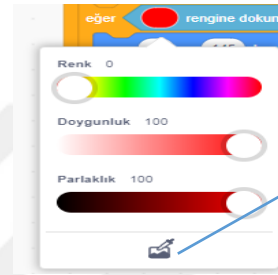


Sahneye bir adet kırmızı zemin kuklası eklenir.



Elma kuklası seçilerek, bu kısımda elma oyunun oynandığı varsaydığımızda ekran da hala oyun bitti kuklasının durduğu bunun oyun başlangıcında değişmesi gerektiği için kılık değişikliği yapılır. Elmanın kırmızı zemine dokunup dokunmadığı kontrol ettirilir, değiştiği anda dal figürüne yollanır ve oyun bitti kostümü getirilerek oyun durdurulur.

Rengi aynı zemin rengi olarak seçmek damlası kullanılır.



için renk

-Bu kısımda öğretmen *rengine dokunuyor mu kodu yerine başka hangi kodu kullanabileceğimizi sorar.*



Cevap:

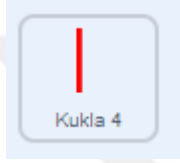
Çubuk sepet kuklası seçilerek toplanan elma sayısı yazdırılır.



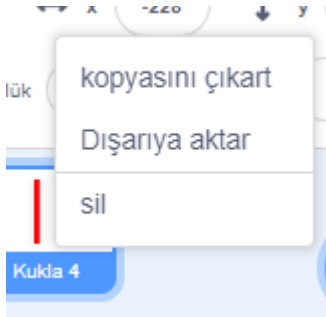
2.2.Uygula: Gol Atma Oyunu

Bu uygulamada öğretmen öğrencilerden ekrana 2 adet çubuk kuklası çizmesi istenir. Bunların her ikisinin de farklı renklerde olması istenir. Bu oyunda çubuklar 2 kullanıcı tarafından aşağı yukarı hareket ederek topun çubuğa değdiği anda karşı tarafa gönderilmesi ve topun değdiği çubuğun renginin söylenmesi sağlanır. İki tarafa çizilen siyah çizgilere değdiği anda karşı kısımdaki oyuncunun puanı bir artırılır. Bu kısımda öğretmen bu uygulamayı öğrencilerden yapmasını ister ve sadece çok zorlandıkları yerlerde rehberlik edebilir.

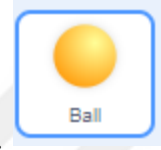
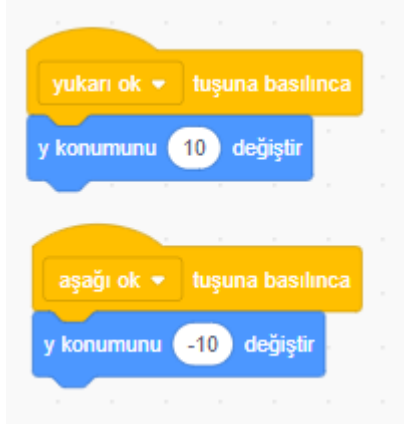
İlk olarak oyunda bir adet çubuk çizilir. Rengi kırmızı olarak ayarlanır.



Ve kuklaya aşağı yukarı hareket edebilmesi için kodlar yazılır.



Kırmızı çubuğa gelerek kopyasını çıkartarak rengini yeşile çevirerek, kod bloğu bu sefer yukarı aşağı şeklinde değiştirilir.



Oyunun hareketli kuklası olarak top kuklası eklenir.

Kuklanın kod kısmına ekranda topun sağa sola rastgele hareket etmesini ve oyunun ilk başında topun hangi konumda bulunacağı ayaları yapılır.



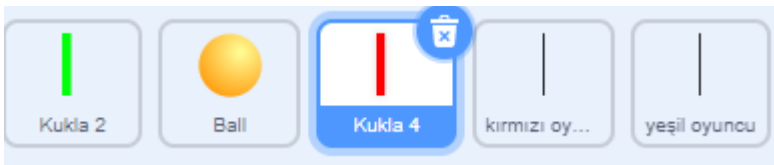
Ekranda hareket eden topun çubuğa değdiği anda yönü değiştirip tekrardan karşı tarafa göndermek için yönüne yönel komutu içinde rastgele sayı seç komutu ile topu karşı tarafa gönderir. Topun ters yönde gitmesi için -1 ile -180 arası sayılar seçilebilir. Topun yeşil topa değdiği anda 1 saniye boyunca yeşil yazdırırız.



Kodun aynısı kırmızı çubuğa da uygulanır. Burada dikkat edilmesi gereken rastgele sayı seç komutunda sayı olarak 1 ile 180 arasında sayı seçtirmemiz gerekiyor. Sayının sabit kalması durumunda topun puka girmesine neden oluyor.



Kukla ekle menüsünden çizim seçilir ve bir adet siyah ekranın yan kısmını tam olarak kaplayacak şekilde çizilir.

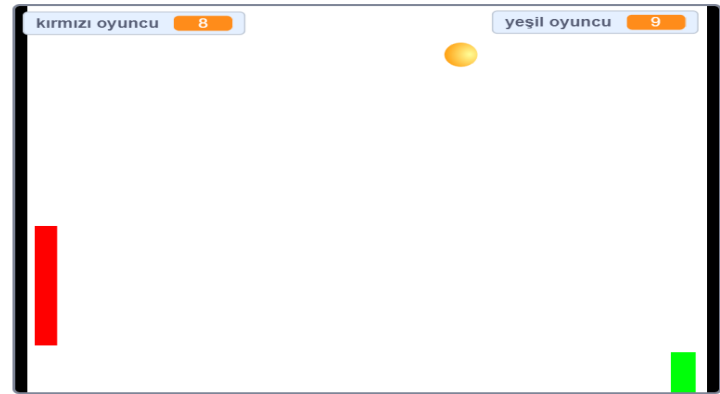


Bir kopyası çıkarılır. Bir tanesine kırmızı oyuncu değerine yeşil oyuncu olarak adlandırılır.

Kırmızı oyuncu kuklasına;



Yeşil oyuncu kuklasına;

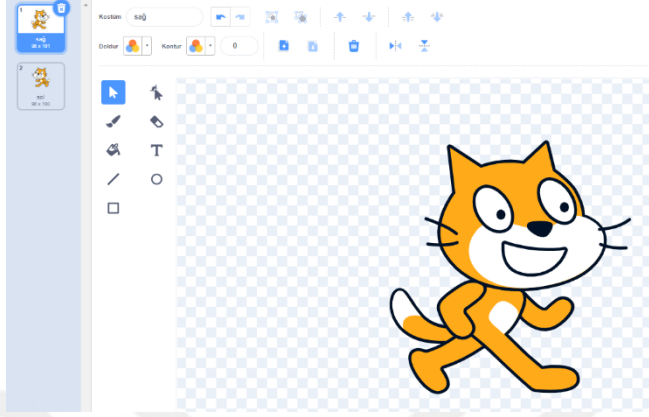


2.3 Gözle: Kedi Fareyi Yakalama Oyunu

Bu uygulamada yön tuşları ile hareket eden kedi kuklası ile sahnede serbest biçimde dolanan fareyi yakalama oyunu yapılacaktır. Burada bir zamanlayıcı arka

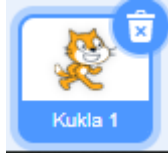
planda çalışıp kedi fareyi yakaladığında kaç saniye geçmişse o saniye yi söyleyip ekrana yazdıran bir oyun yapılacaktır.

Uygulamaya ilk olarak kedi kuklası eklenerek başlanır.



Kedi kuklası karakterine sağ, sol biçimde iki tane kostüm eklenir ve isimleri yazılır.

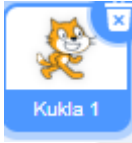
Kedi kuklası aşağı, yukarı, sağa, sola doğru hareket edebilmesi için gerekli kodlar yazılır.



Sonrasında fare kuklası eklenerek farenin ekranda serbest biçimde hareket etmesi için gerekli kod blokları yazılır.



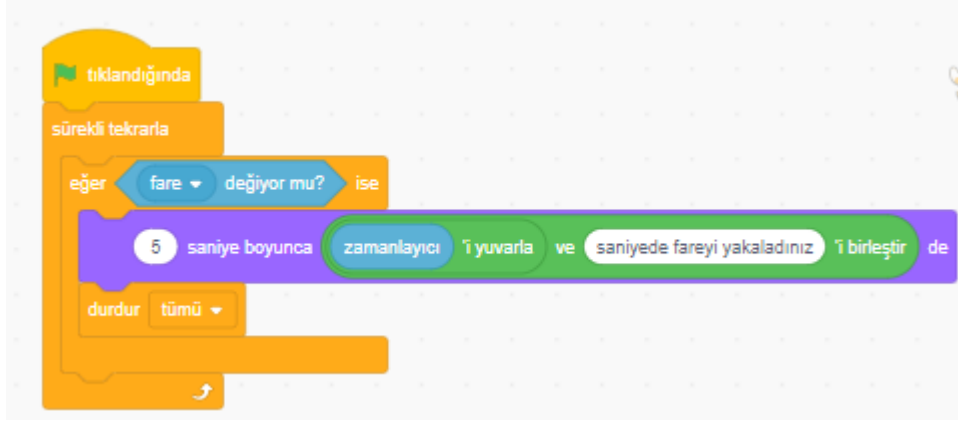
Farenin büyüklüğü olabildiğince küçük seçilir.



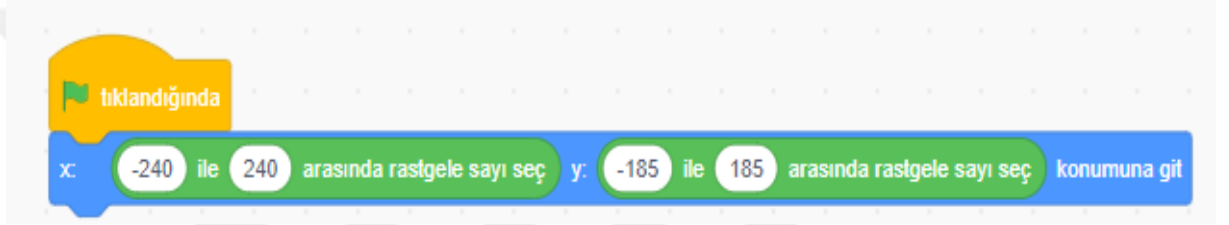
Kedi kuklası seçilir. Kedinin fareye değdiği andaki kod blokları yazılır.



Daha sonra ekrana süre yazılabilmesi için “zamanlayıcı” kod bloğu kullanılır. Zamanlayıcı kod bloğu arka planda saniye türünden zamanı tutar. Zamanı tam sayı halinde yazdırmak için “yuvarla” komutu kullanılır. Ekrana yazdırmak için saniye ve yazıları birleştirmek için “birleştir” komutu kullanılır. Oyunun değdiği anda bitmesi için tümünü durdur komutu kullanılır.



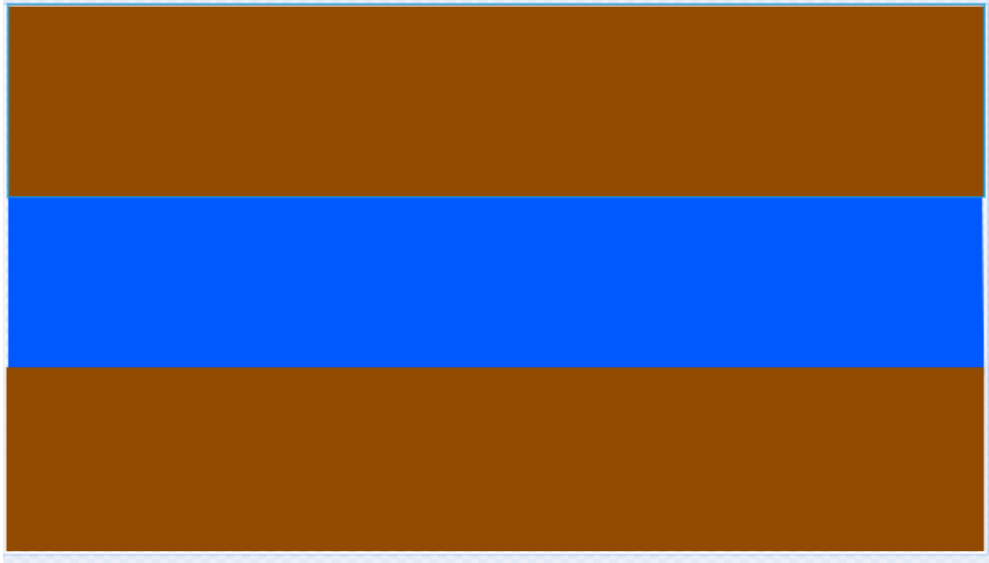
Kedinin her yeşil bayrağa tıklanıldığında farklı yerlerde çıkması için aşağıdaki kod bloğu yazılır.



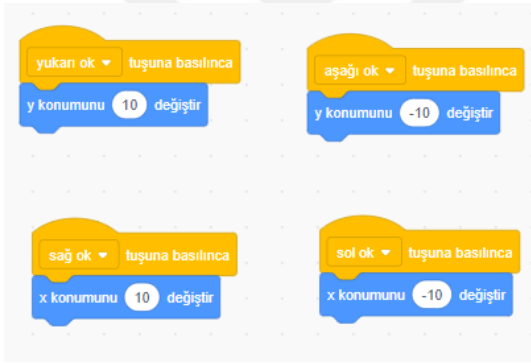
2.3 Uygula: Nehir'i Geçme Uygulaması

Bu uygulamada çizilen nehir temalı dekordan bir adet böceğin karşı tarafa geçip çileği yakalaması istenmektedir. Böcek Nehir'e değdiği zaman başlangıç noktasına gidecektir. Nehir'in bir ucundan bir ucuna hareket halinde olan sal bulunmalıdır. Böcek bu salın üstünden karşıya geçip çileği yakalamalıdır.

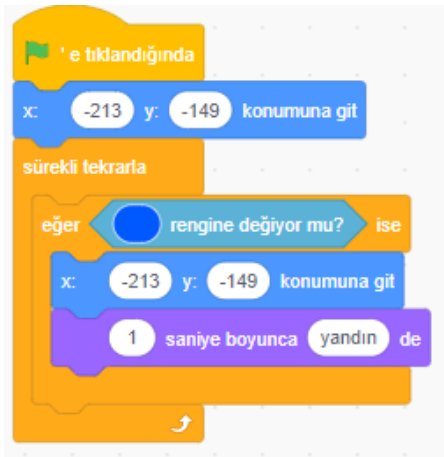
Uygulamaya dekor çizme işlemi ile başlanır. Aşağıdaki dekora benzer bir dekor çizilir.



- Ekranı bir adet böcek kuklası eklenir. Aşağıdaki kod blokları yazılır.



Bu kısımda böceğin yukarı aşağı sağa sola gidebilmesi için gerekli kodlar yazılır. Bu kısımda böceğin hızı isteğe göre ayarlanabilir.

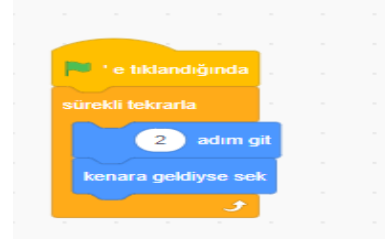


Bu kısımda böcek kuklasının başlangıç konumu ayarlanır, mavi renge (Nehir'e) değiştiği anda başlangıç noktasına gitmesi sağlanır.



Uygulamada bir adet sal kuklası çizimi yapılır. Çizim yapılırken kare çizgisinin tam olarak ortalandığına

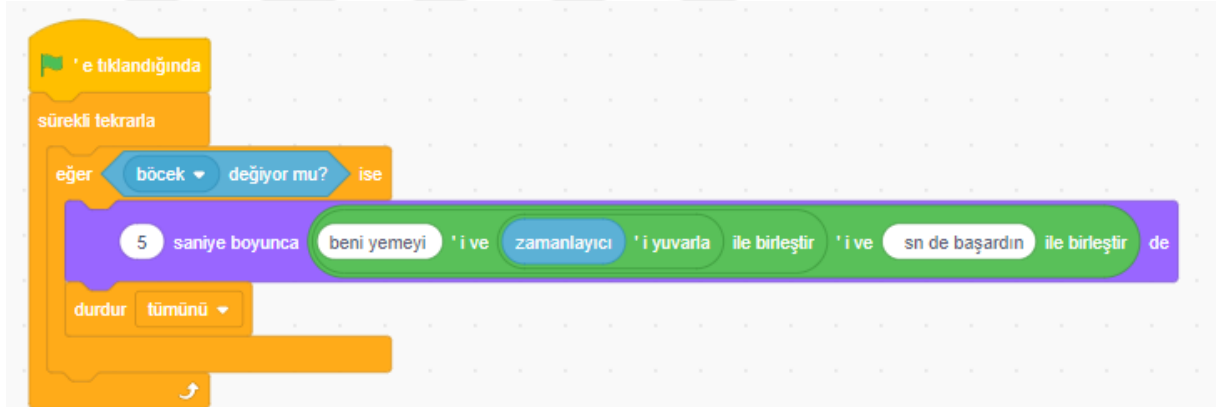
emin olunmalıdır. Yoksa salın sağa sola doğru kaydığı görülecektir.



Salın nehirde sağa-sola doğru hareket etmesi için gerekli kod bloğu yazılır.



Uygulamaya çilek kuklası eklenir. Böceğin değdiği anda zamanı söylemesi için gerekli kod bloğu eklenir.



2.Tasarla ve Üret

2.1 Tasarla: Balık Avlama Oyunu

Bu bölümde öğrencilerden balık yakalama oyunu istenecektir. Bu oyunu yaparken dikkat edilecek hususlar şunlardır.

-Ekranda birden fazla balık kuklası olacaktır. Bu balık kuklaları ekranda serbest bir biçimde sağa sola ve aşağı yukarı şekilde akvaryumda gezer gibi görüntü verilecektir.

-Bir adet köpek balığı kuklası eklenip bu kukla fare ile hareket etmesi sağlanacaktır. Bu kuklanın görevi balıkları yemek tir. Balıklara değdiği anda balıklar kaybolacaktır. Fakat belirli bir süre sonra balık tekrardan ekrana görünür halde çıkacaktır.

-Kukla ekle menüsünde denizde bulunan canlılardan herhangi biri eklenir ve bu canlılar oyunda yenilmemesi gereken kukla olarak kullanılır. Fare ile hareket ettirdiğimiz kukla bu kuklalara değdiği zaman puan menüsünden puan kaybedecek. Kaybedilecek puanı öğrencinin seçimine bırakılmıştır. Bu kuklalarda yenildiği anda ekranda kaybolacak ve belirli bir süre sonra tekrardan ekranda belirecektir.

-Oyunda puan sistemi uygulanacaktır. Bu puan sisteminde her yenilen balık için puan verilecek, her yenilen deniz canlısı için ise puan düşürülecektir.

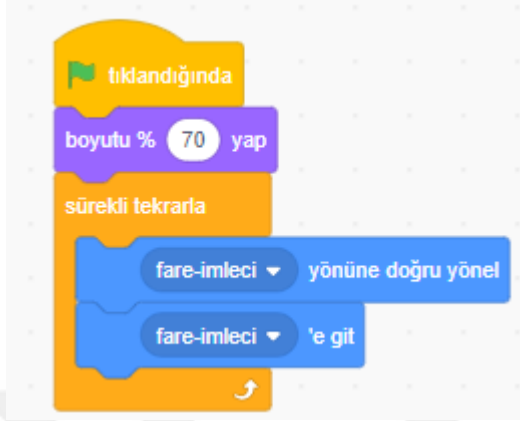
-Oyuna zaman bloğu eklenecektir. Oyunun hangi saniyede biteceği öğrenciler tarafından belirlenecektir. Bu süre oyun içinde eksi puan getiren kuklaların ortaya çıkması içinde kullanılacaktır. Bu kuklalar ekrana çıkacak ve diğer eksi getiren kuklalardan farklı olarak renk değiştirme animasyonu verilecek, bir daha kaybolmayacaktır.

-Örneğin 10.saniyede 1 kukla 20.saniyede başka bir kukla çıkararak oyunu zorlaştıracaktır.

-Oyunun sonunda herhangi bir ekrana eklenmiş kukla belirli bir puana ulaşıldığında veya belirli bir süre aşıldığında veya – puana düştüğünde kaybettiniz veya kazandınız yazması istenmektedir.

2.1 Üret: Balık Avlama Oyunu

Oyunumuza ilk olarak köpek balığı kuklası seçilerek başlanır. Bu kısımda köpek balığı kuklasının fareyi takip etmesi ve boyutu ile ilgili kodlar yazılır.



Dekor kısmı seçilir ve su altı dekoru seçimi yapılır.

Dekor kısmına gidilerek kodlama menüsü seçilir.

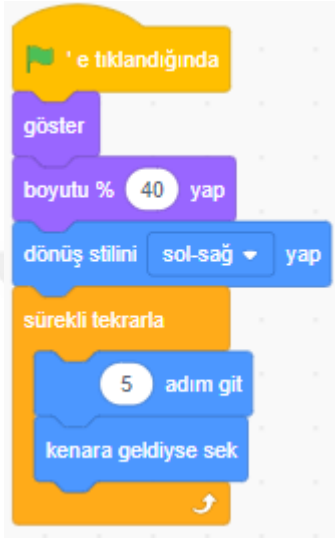
Aşağıdaki kodlar yazılır.



Ekranda yenildiğinde puan kazandıracak balıkların eklenmesi



Yandaki şekildeki gibi bir balık kuklası seçilir isim olarak herhangi bir değişiklik yapılmaz. Balık kuklası için aşağıdaki kod blokları eklenir.



Bu kısımda balığın boyutu ayarlanır. Ekranda serbest bir bicide dolanması sağlanır.

Balıkların ekranda sabit olarak gitmemesi için yön sekmesinden derece 128 olarak ayarlanır. Kendi belirlediğiniz bir sayıda yapılabilir.



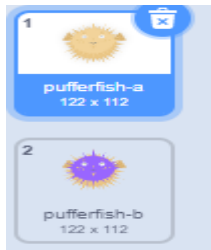
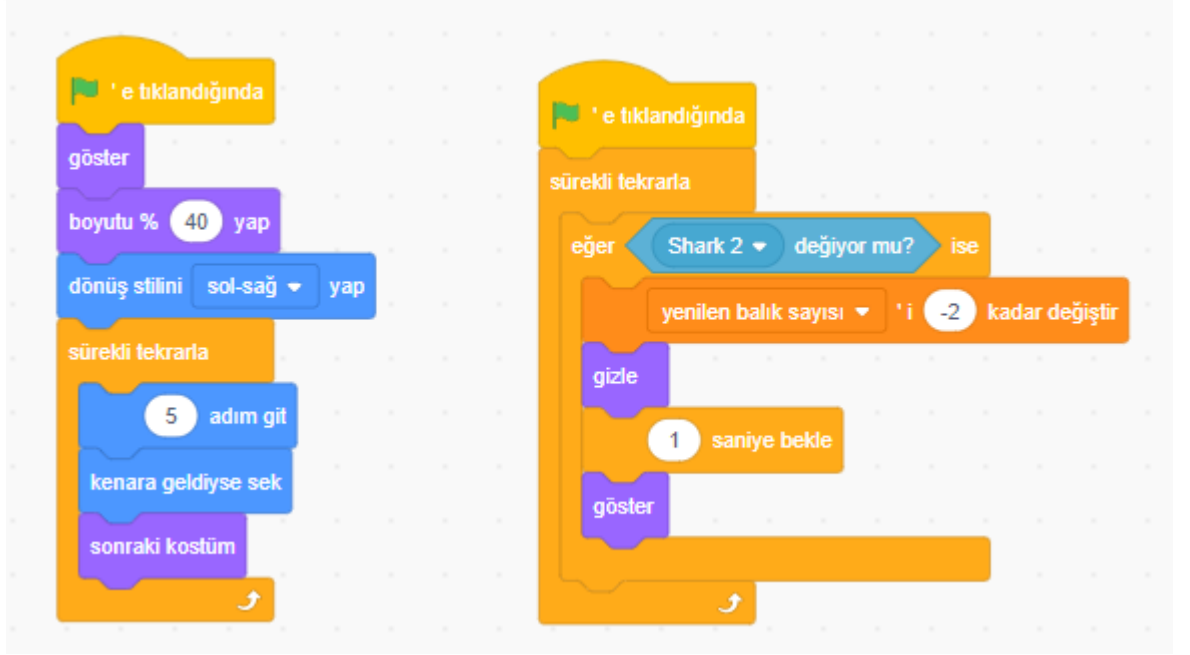
Bu kısımda köpek balığının balığa değdiği anda kaybolması ve 3 saniye sonra tekrardan ortaya çıkmasını

sağlayan kod bloğu yazılır. Yenilen balık sayısı değışkeni ise 1 kadar artırılır.

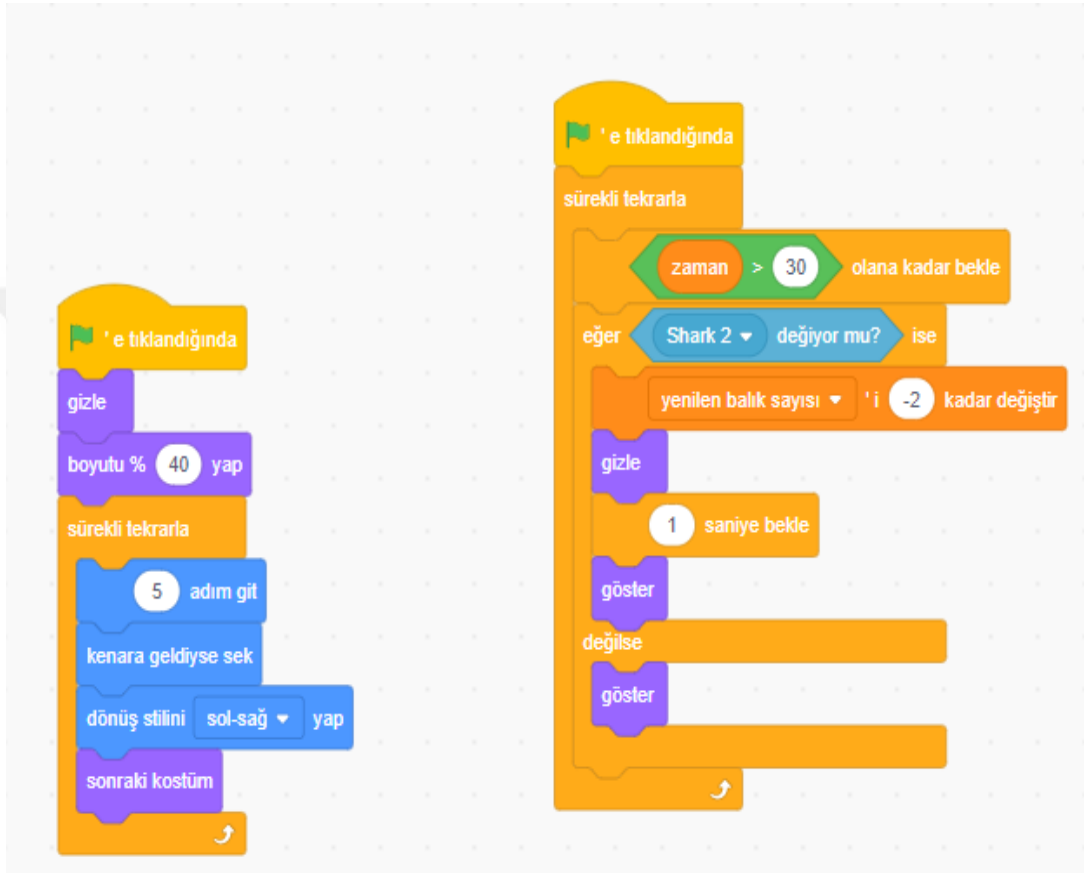
Yukarıdaki işlemler için en az altı tane balık kuklası eklenir ve aynı kodlar bu kuklalara da eklenir.



Ekrana yandaki pufferfish kuklası eklenir. Bu kukla köpek balığının yememesi gereken kukla olarak eklenir. Aşağıdaki kodlar eklenir. Bu kuklada sayı olarak en az üç tane ayarlanır.



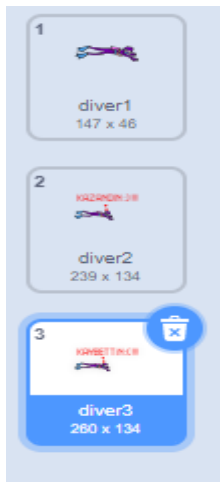
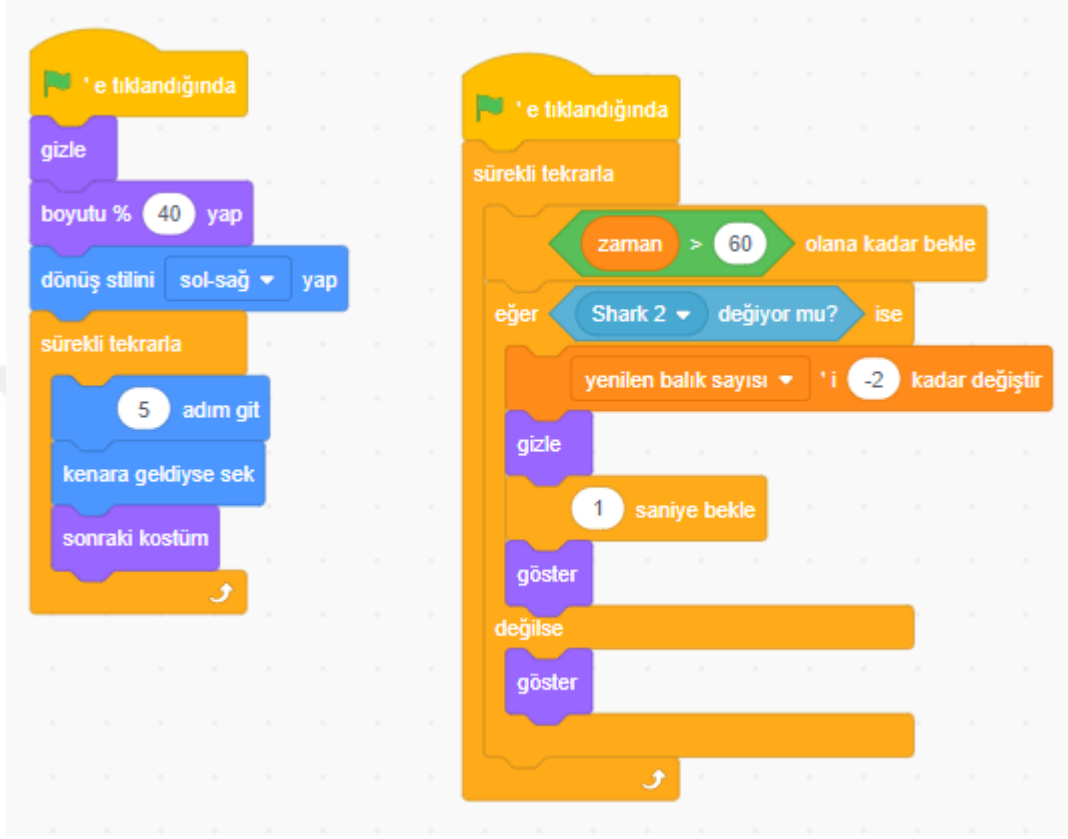
Bu kısımda belirlediğimiz bazı pufferfish kuklaların belirli bir zaman sonra çıkması sağlanacaktır. Bunun için pufferfish kuklasını mor renge boyanır. Bu kukladan 2 tane farklı zamanda çıkması için ekleme yapılır.



Yukarıdaki kod kısmında kuklanın sağa sola serbest bir biçimde dolaşması için ayar yapılır. Saniyenin 30 dan büyük olmasına kadar beklenir. Saniye 30 olduktan sonra kuklanın çıkması sağlanır. Köpek balığı kuklasının bu kuklayı yemesi durumunda gizlenip gözükmesi için gerekli kod blokları oluşturulur.



Yandaki şekilde yeniden kukla oluşturulabilir. Kullanıcın kendi kararına göre ekleme yapılabilir. Aşağıdaki kodlar eklenir.



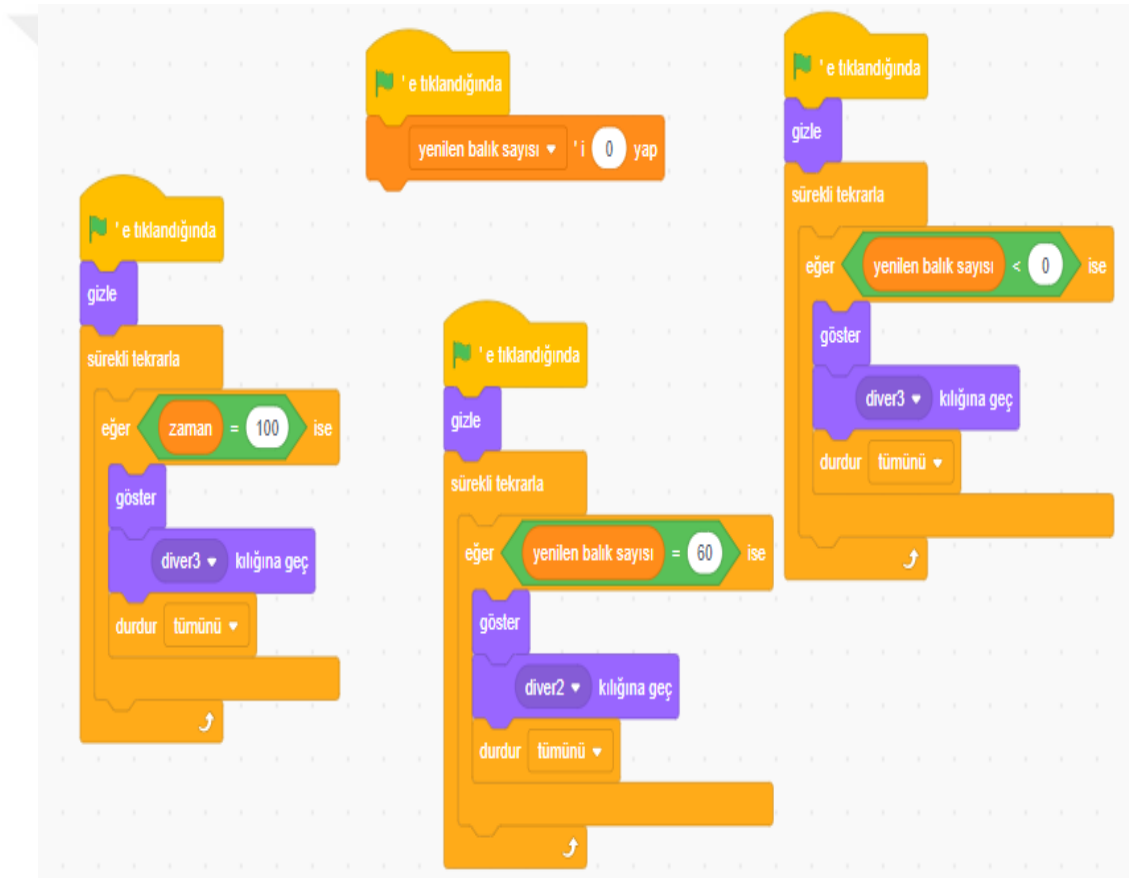
Yandaki kukla seçilir ve kopyalanır.2 kısma kazandın yazılır.3.kısma kaybettin yazılır.

-Bu kuklamızda yenilen balık sayısı sıfırlanır.

-Zaman 100 olduğunda kaybettin kuklasına geçilir ve oyun durdurulur.

-Yenilen balık sayısı 60 olduğunda kazandın yazan kuklaya geçer ve oyunu durdurur.

-Yenilen balık sayısı sıfırın altına düştüğünde oyunu kaybettin kuklasına geçer ve oyunu durdurur.



1.2 Değerlendirir

Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini

değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).

- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- o Değişkenlerin nasıl çalıştığı konusunu anlamakta zorlandım.

- o Zamanlama sistemini oluştururken zorlandım.

- o Sahnelerin çizimde araçların kullanımı noktasında zorlandım.

- o Puan verme işlemlerinin hangi kod işleminden sonra yapılacağı tercihi noktasında zorlandım.

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?

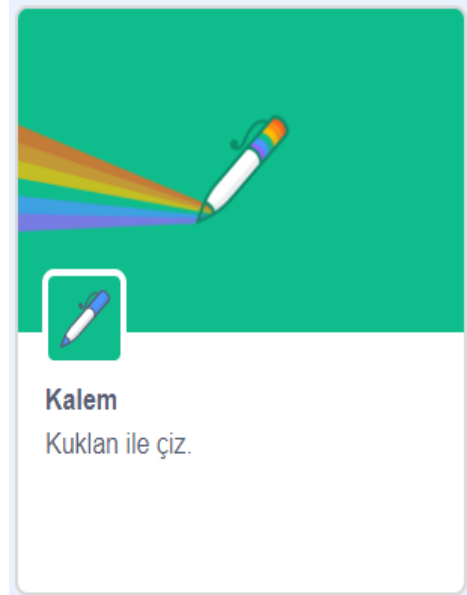
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?

- Grup arkadaşınızdan ne öğrendiniz?

3.1 Gözle: Kalem ile Kare, Üçgen Ve Renkli Kesikli Çizgi Çizme

Scratch programı ile kalem kuklasını veya başka kuklaları kullanarak şekillerin çiziliyormuş gibi gösterebilmekteyiz. Bunu için “eklenti ekle” kısmından kalem eklentisini programa ekliyoruz.

Not: Kalem eklenti kısmını her scratch açılışında eklememiz gerekmektedir.

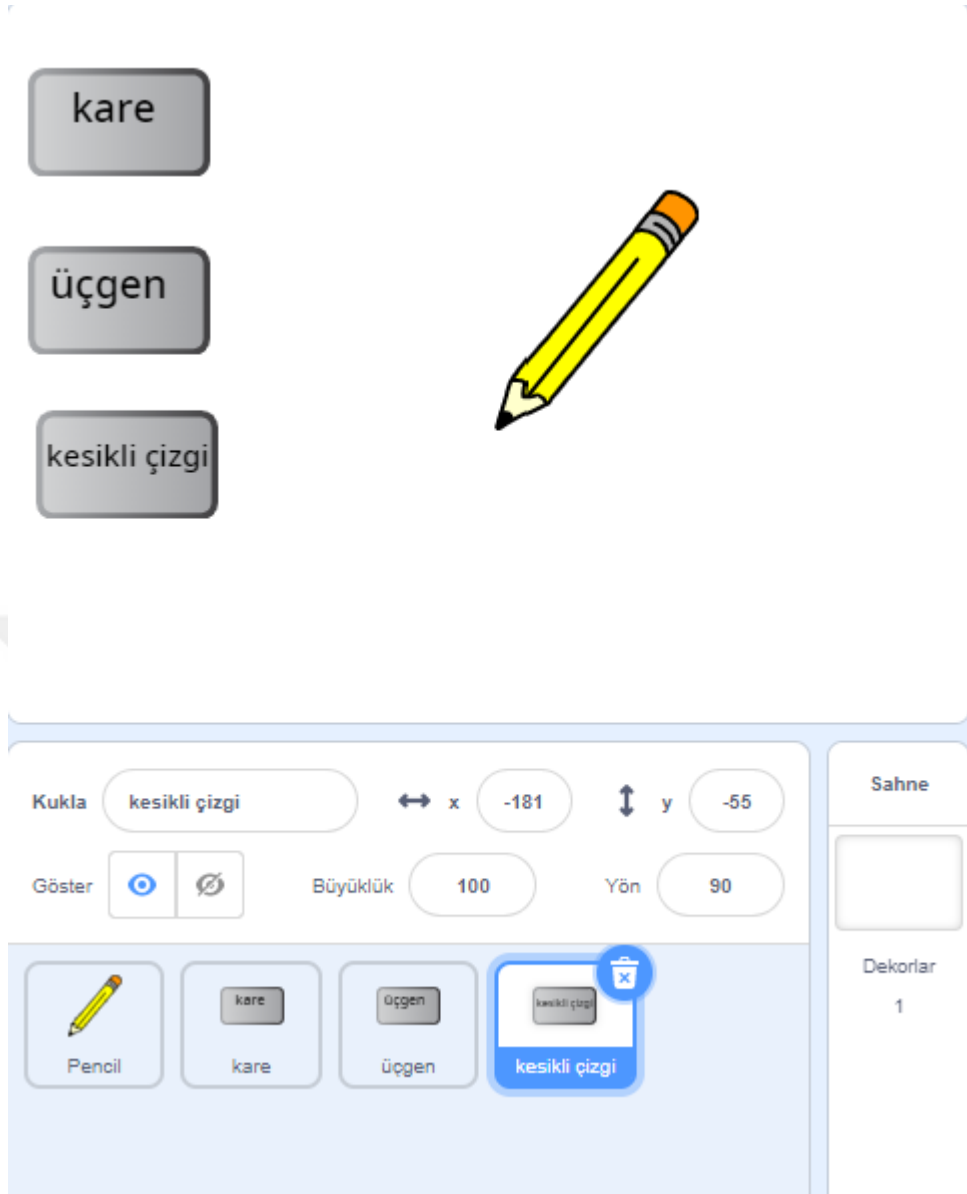




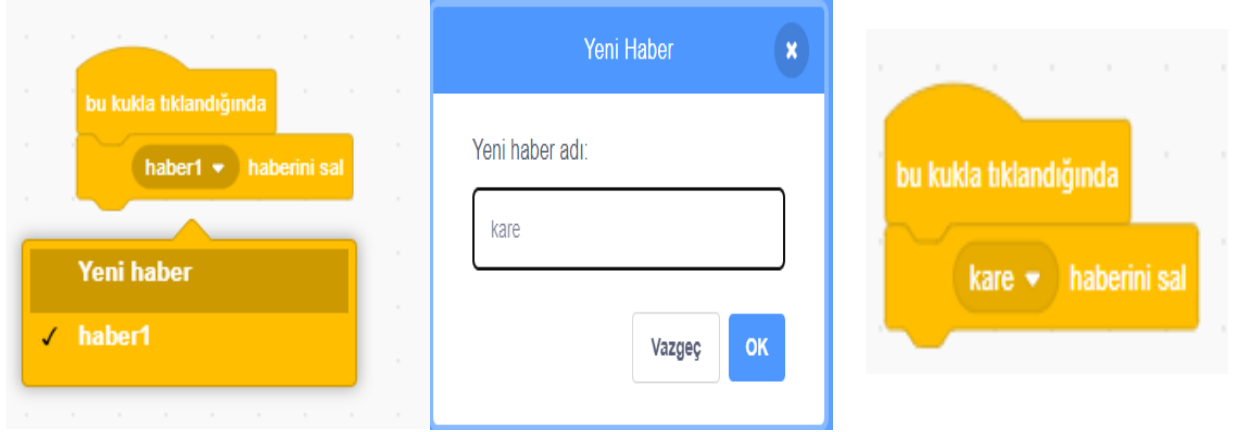
Kukla seç menüsünden pencil yazarak kalem kuklası eklenir. Burada rehber öğretmen tarafından çizimin istenirse başka kuklalar ile yapılabileceği de vurgulanır.



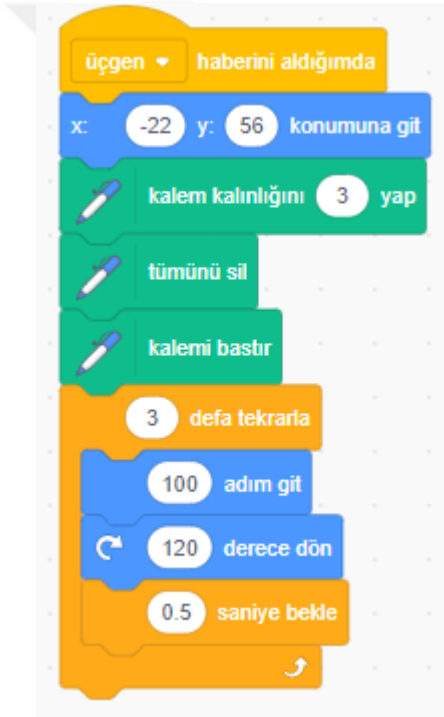
Bu kısımdan sonra kukla seç menüsünden üç tane buton eklenir. Kare, üçgen, kesikli çizgi olarak isimlendirilir.



Bu kısımda yapılacak işlem her bir butona basıldığında üzerindeki işlemi gerçekleştirmektir. Butonun basılıp basılmadığını kontrol etmek için “bu butona basıldığında” kod bloğu kullanılır. Basılan butonun hangisi olduğunu kalem kuklasına bildirmek için “haber sal” komutu kullanılır.

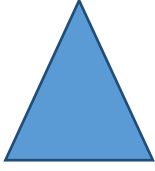


Bu işlem her buton için uygulanır. Üçgen butonu için üçgen haberi sal, kesikli çizgi için kesikli çizgi haberi sal şeklinde.



Üçgen haberi geldiğinde kalem kuklasının konumu belirtilir. Kalem kalınlığı sayısal olarak belirtilir. Ekranda daha önce çizilen herhangi bir şekil varsa onların silinmesi sağlanır. Çizgi komutlarında ilk önce kalem basılı olmadığından dolayı “kalemi bastır” komutu kullanılır ve kalemin çizmesi sağlanır.

Rehber öğretmen döngü kullanılarak üçgen çizmeyi anlatmadan önce eşkenar üçgen hakkında bilgilendirme yapar.



Aynı doğru üzerinde olmayan üç noktayı birleştiren doğru parçalarından meydana gelen geometrik şekle üçgen denir. Üçgenlerin iç açıları toplamı 180 derecedir. Bu uygulamada çizeceğimiz üçgen eşkenar üçgendir. Eşkenar üçgen bütün kenarları ve açıları eşit olan üçgene eşit kenar üçgen denir. Eşkenar üçgenin bir iç açısı $180/3=60$ derecedir. Eşkenar üçgenin bir dış açısı ise $180-60=120$ derecedir.

Üçgenin üç kenarı olduğundan dolayı çizgi çizme ve dönme işlemlerimizi 3 defa yapacağımızdan dolayı döngüyü 3 olarak ayarlıyoruz.

Not: Rehber öğretmen bu kısımda dönme işlemlerini dış açılara göre yaptığımızı vurgulamalıdır.

Rehber öğretmen üçgen çizimini anlattıktan sonra karenin ne olduğunu sınıfa sorar ve biraz tartışıldıktan sonra aşağıdaki açıklamayı yapar.

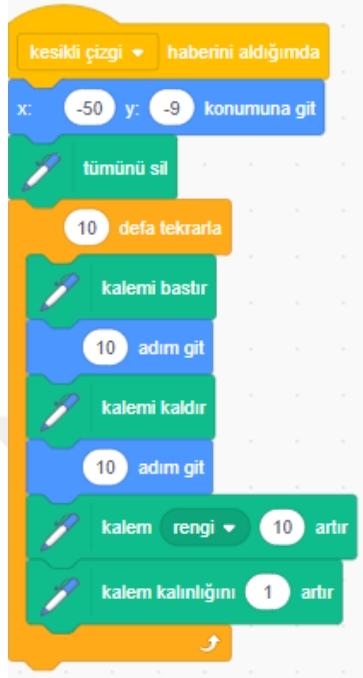


Kare, bütün kenarları ve açıları birbirine eşit olan düzgün dörtgendir. Karenin bir iç açısı 90 derecedir.

Rehber öğretmen açıklamaları yaptıktan sonra kare çizimi içinde aynı adımlar uygulanır.



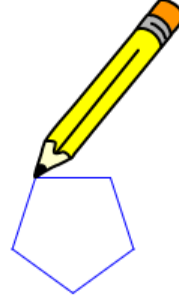
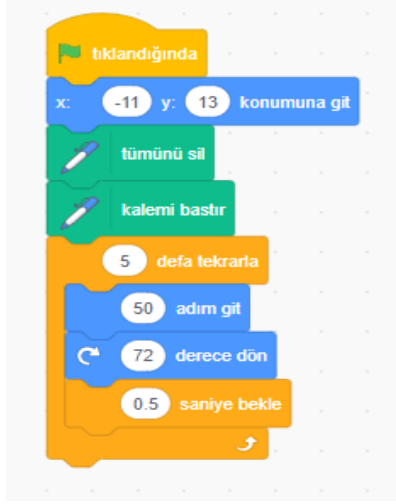
Kesikli doğru çizgi için aynı işlemler yapılır. Çizgilerin kesikli olabilmesi için kalemi bastır ve kalemi kaldır işlemleri uygulanır. Rengini değiştirmek için rengi artır komutları kullanılır.



Not: Kalem kuklası ile çizim yapıldığında kalemin orta kısmı ile çizim yapıldığı görülecektir. Burada rehber öğretmen bunun merkez noktasından kaynaklı olduğunu kalemin merkez noktasının kalemin ucuna alınması gerektiği söylenmelidir.

3.1 Uygulama: Kalem ile Düzgün Beşgen Çizme

Bu kısımda rehber öğretmen öğrencilerden kalem kuklasını kullanarak düzgün beşgen çizmelerini ister. Düzgün beşgenin kaç kenarlı olduğu ve iç ve dış açılarının kaç derece olacağını araştırmaları istenir. Aşağıdaki uygulama gibi bir kod yapısı olabilir.



3.1 Tasarla: Kelebek Motifi Çizme

Bu bölümde öğrencilerden kalem uygulamasını kullanarak kelebek motifi çizmelerini istenir. Bu uygulamayı yaparken şu hususlara uyması beklenir.

-Kelebek çiz butonuna basıldığında çizim başlamalıdır.

-Öğrencilerin kelebek motiflerini internetten veya kitaplardan inceleyerek araştırmaları istenir.

Kelebek kanatlarının 4 ayrı biçimde görünüyormuş gibi çizilmeli

Kelebek kanatlarının rengarenk olması sağlanmalı

Kanatlarının arasında kelebeğin gövdesi geçmeli

Kanatlarının içinin dolu biçimde olmasına gerek yoktur sadece çizgisel olarak belirtilmelidir.

Öğrencilerin öncelikle kelebek motifini çizmek için kaç derece dönüş yapacakları ve hangi döngüleri kullanmalarını belirlemesi gerekir. Gerekirse oluşturacakları çiçek motifini karakalem ile çizebilirler.

Örnek

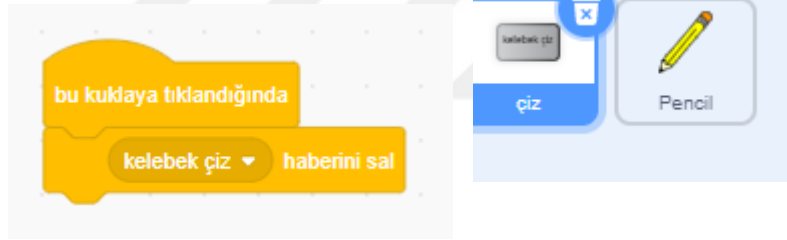
- i- Kelebeğin bir kanadının bir kenarını çizmek için kaç defa tekrar yapılmalı

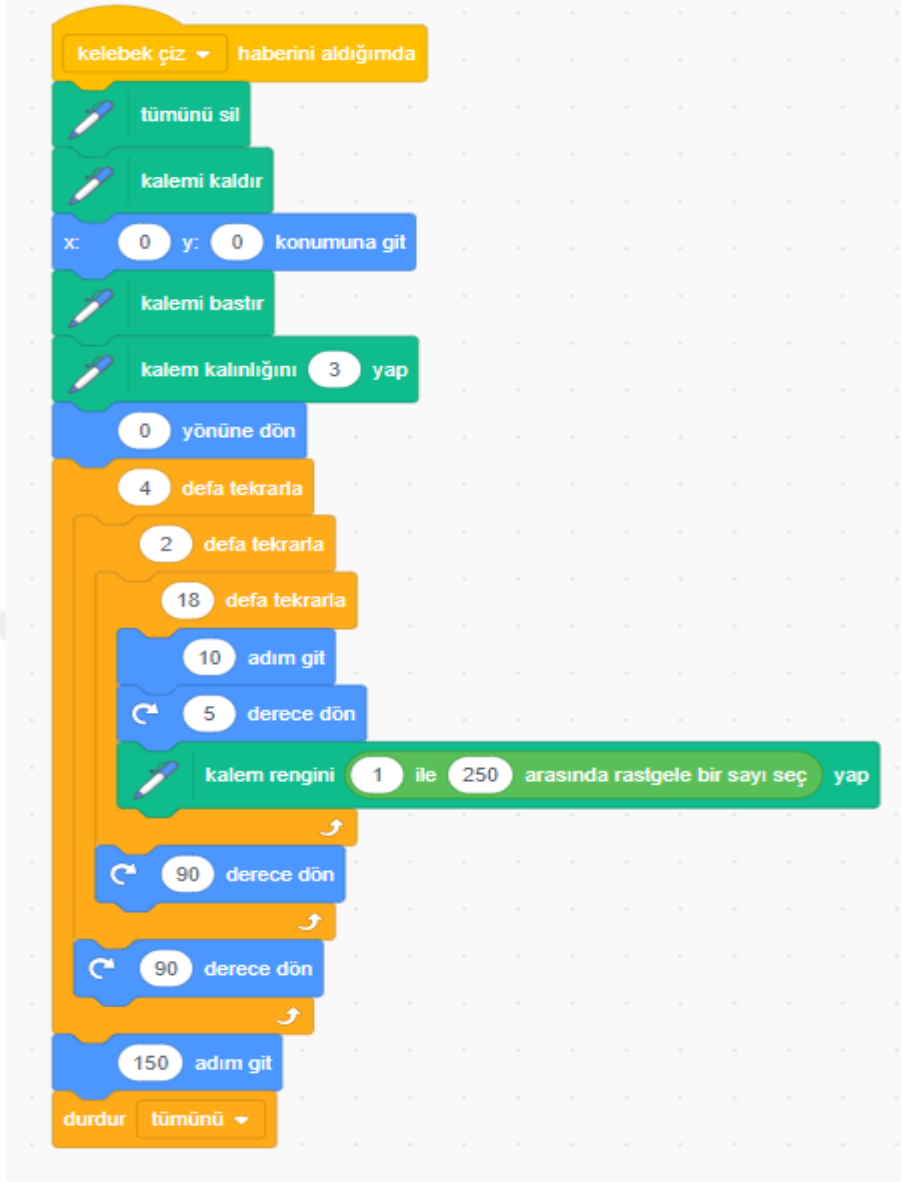
- ii- Kelebek kanadının ucundaki ovallık vermek için kaç derece dönüş yapılmalı
- iii- Kelebek kanadının aynısını karşı bloğa oluşturmak için ne yapılmalı
- iv- Kelebek kanadının ikincisi oluşturulurken kaç derecelik dönüş yapılmalı
- v- Hangi döngüler iç içe kullanılmalı
- vi- İç kısımdaki birleştirme motifi için kaç derece dönüş yapılmalı

Sorularına cevap arayarak motif çizilmesini sağlar.

2.1 Üret: Kelebek Motifi

Öğrenciler çözüme yönelik planlarını yaptıktan sonra bilgisayar başında istenilen görevler yerine getirilir. Öğrencilere rehber öğretmen tarafından etkinliklerini tamamlamak için yeterli süre verilir. Öğrencilerin aşağıdaki kod bloğuna benzer bir içerik üretmeleri beklenir. Gerektiği noktada rehber öğretmen yardımcı olabilir. Fakat hiç biçimde tam bir çözüm verilmemelidir.





1.3 Değerlendir

Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).

• Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- o Değişkenlerin nasıl çalıştığı konusunu anlamakta zorlandım.
- o Zamanlama sistemini oluştururken zorlandım.
- o Sahnelerin çizimde araçların kullanımı noktasında zorlandım.
- o Puan verme işlemlerinin hangi kod işleminden sonra yapılacağı tercihi noktasında zorlandım.

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

4.1Gözle: Zıplayarak Gezegen Değiştiren Ördek

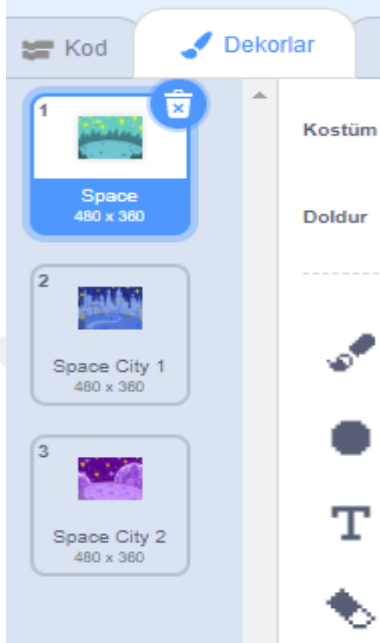
Scratch programında sahnenin arka planına dekor denilmektedir. Dekor ile yaptığımız yazılımın mekanını belli edilmektedir. Yapacağımız animasyon veya oyunlar ev içindeyse evin içerisini andıran bir resim seçilir veya uzay mekânında olaksa uzayla ilgili ya resim indirilir ya da dekorlar kısmından çizilir. Bu kısımda bilgisayara yüklediğimiz bir resmi seçebilir veya kendimiz dekor çizebiliriz, diğer bir seçecek olarak da scratch programındaki dekorlardan kullanabiliriz.



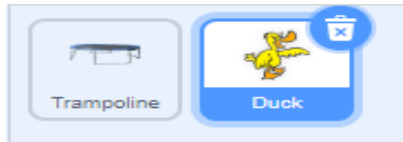
Bu kısımda yapılacak uygulamada ise ekranda bir adet trambolin bulunacak ve ördek kuklası bu tramboline yöne tuşları ile hareket ederken değdiği anda yukarı

zıplayacak ve sahnenin de deęiřimi saęlanacaktır. Ördek zıpladıęı zaman başka bir yere mekâna geiyormuř gibi gözükmesi istenmektedir.

İlk olarak dekorlar kısmından uzay ile ilgili scratch programında bulunan 3 tane dekor seçilir ve eklenir.



Ekrana bir adet tranbolin kuklası ve bir adet ördek kuklası eklenir.



Ördek Kuklasının kod kısmına gelinir ve hareket ile ilgili kod blokları yazılır.



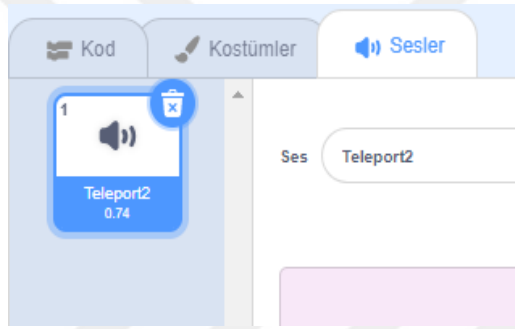
Trampoline ördek kuklası değdiği anda x ve y konumunu değiştirerek ördek kuklasının zıplamasını sağlarız ve dekorun değişmesi için haber sal komutunu kullanırız.

Bu bölümde karakterimize gezegenler arasında dolaşırken ses ekliyoruz.

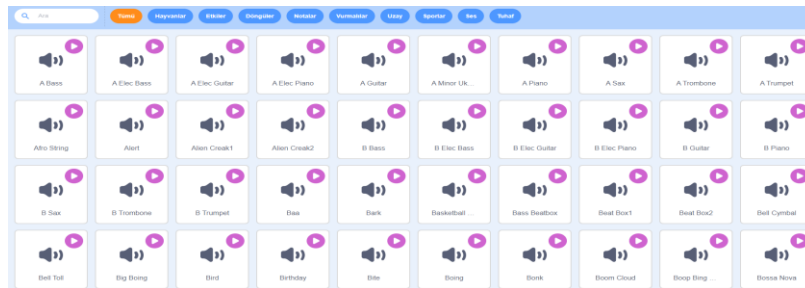
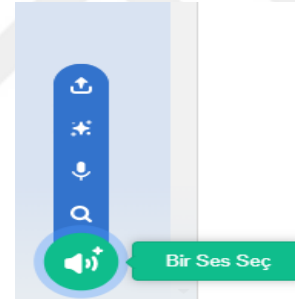


Bu kısımda ses hangi kod bloğunda çalınacak ise o kısma ses bitene kadar çal komutu

eklenir. Çalınacak sesler bu komuttan seçilebileceği gibi sesler menüsünden bir seç denilerek eklenebilir. Mikrofon kullanılarak da ses kaydı eklenebilir.

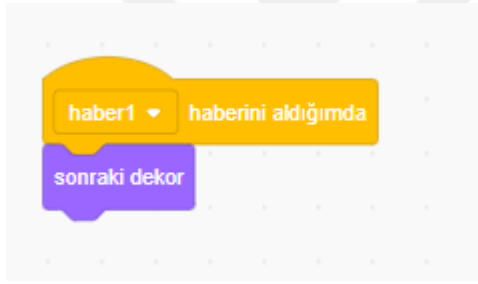


Dışarıdan herhangi bir waw dosyası da eklenebilir. Bir ses seç denilerek scratch içerisinde bulunan birçok ses eklenebilir.





Dekorlar kısmının kod bölümüne gelerek haber geldiğinde dekorun değişmesi için sonraki dekor komutu kullanılır.



4.2Uygula: Mevsimleri Keşfet

Bu uygulamada öğretmen öğrencilerden ekrana 4 adet buton çizmelerini ve bunların yaz, sonbahar, kış ve ilkbahar isimlerini içeren, bu butonlara tıklandığında arka planında bu mevsimlerle alakalı olarak değişmesi istenmektedir.

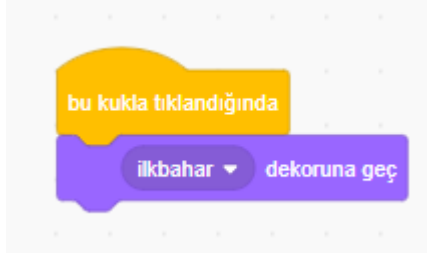
Rehber öğretmen öğrencilere internetten resim indirme kısmında yardımcı olabilir fakat uygulamanın tamamı söylenmemelidir.

Uygulama için ilk olarak internet tarayıcımızdan dört adet mevsimlerle ilgili resim indirilir ve bunlar sırası ile dekorlara yüklenir.

Kukla çiz menüsünden buton şeklinde dikdörtgen şekiller çizilir.



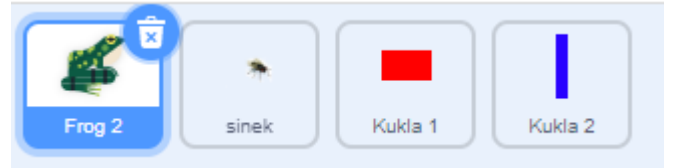
Her bir kuklaya sahne değişimi için dekor değiştir kodları eklenir. Burada bu kukla tıklandığında komutu kullanılır.



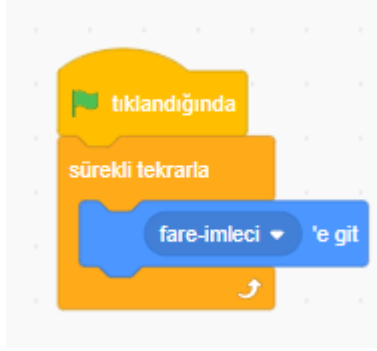
4.2 Gözle: Sinek Yakalama Oyunu

Bu uygulamada fare ile hareket eden kurbağa kuklası ekranın rastgele noktalarında çıkan sinek kuklasını yemesi istenmektedir. Her bir yenilen sinekte aşağı kısımda bulunan dikdörtgen şeklindeki kukla mavi dikdörtgene doğru yavaşır ve yeşil dikdörtgene değene kadar bu işleme devam edilir. Sonunda “evdeki bütün sinekleri temizledin” mesajı verilir.

Uygulamada kullanılacak kuklalar eklenir ve çizilir.



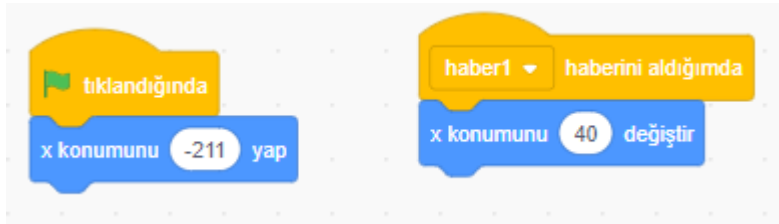
Kurbağa kuklasına kod kısmına fare ile hareket edebilmesi için kod bloğu yazılır.



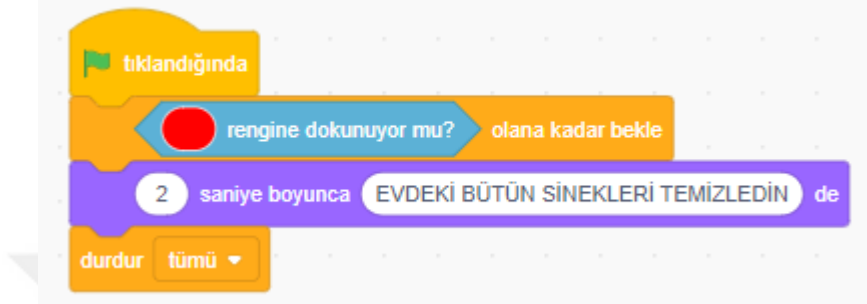
Sinek kuklasına giderek ekranda belli sürelerde rastgele konumlarda çıkması için gerekli komutlar yazılır. Kurbağa kuklası sineğe değdiği anda kırmızı kutunun harekete geçebilmesi içinde haber sal komutu kullanılır.



Ekranda ilerleme görevinde olan kırmızı kutu kuklaya hareket etmesi için kod blokları yazılır. Kurbağanın sineğe değdiği anda gönderilen haber geldiğinde kırmızı kutunun ilerlemesi için x konumu değiştirilir.



Hedef olarak gösterilen mavi kutu kuklasına kırmızı rengin değene kadar hiçbir kodu çalıştırmayıp sadece kırmızı rengin değdiği anda alttaki kodların çalışması için “olana kadar bekle” kod bloğu kullanılır. “Olana kadar bekle” kod bloğu ile bir işlem oluncaya kadar beklenir ve işlem olduktan sonra alttaki kodlar çalışmaya devam eder. Kırmızı kutu her bir sinek yenildiği anda ilerler ve mavi kutuya geldiğinde mesaj yazdırılır.



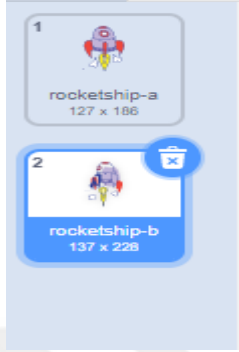
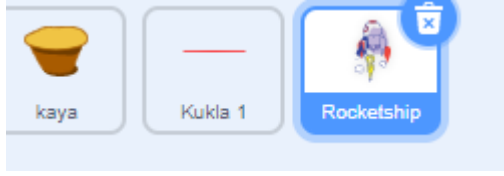
4.2 Uygulama: Engelden Kaçan Uzay Aracı

Bu bölümde öğretmen öğrencilerden ekranda klavyenin sağ sol tuşu ile hareket eden bir uzay aracı kuklası eklenmesi ve ekranda kayan yolda yukarıdan aşağı doğru gelen kaya parçalarına çarpmadan yolda gitmesini sağlayan oyun yapılması istenmektedir. Beyazlı zemine çarptığı zaman oyun yine bitecektir.

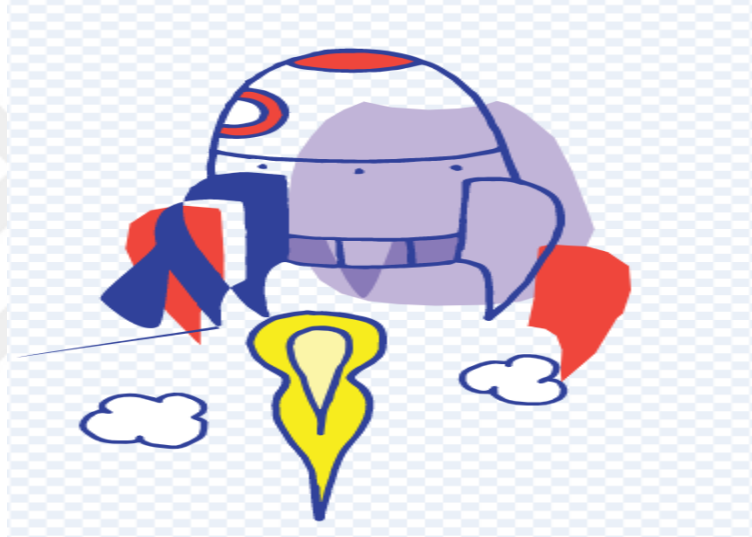
İlk olarak gerekli olarak sahne çizimi yapılacaktır. Burada ekranın kayıyormuş gibi gözükmesi için yol çizgilerini her bir sahnede aşağıya doğru kaydırılır.



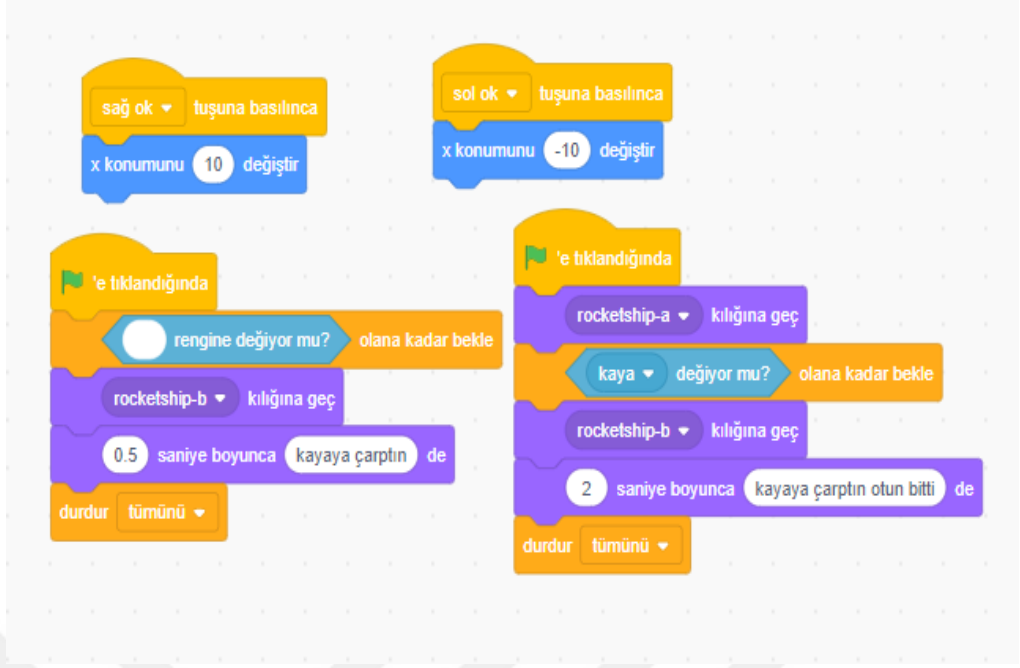
Sahne çizimden sonra ekrana gerekli kukla eklemesi yapılır.



Roket kuklası seçilir ve kostümlerden roket kuklası parçalı bir biçime getirilir.



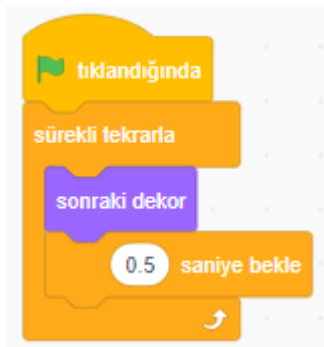
Roket kuklası seçilir. Roketin sağa sola gitme kotları eklenir. Beyaz zemine değdiği anda oyunun bitmesi istenir. Yukarıdan gelen kaya parçalarının arabaya değdiği anda oyunun bitmesi mesajı yazılır.

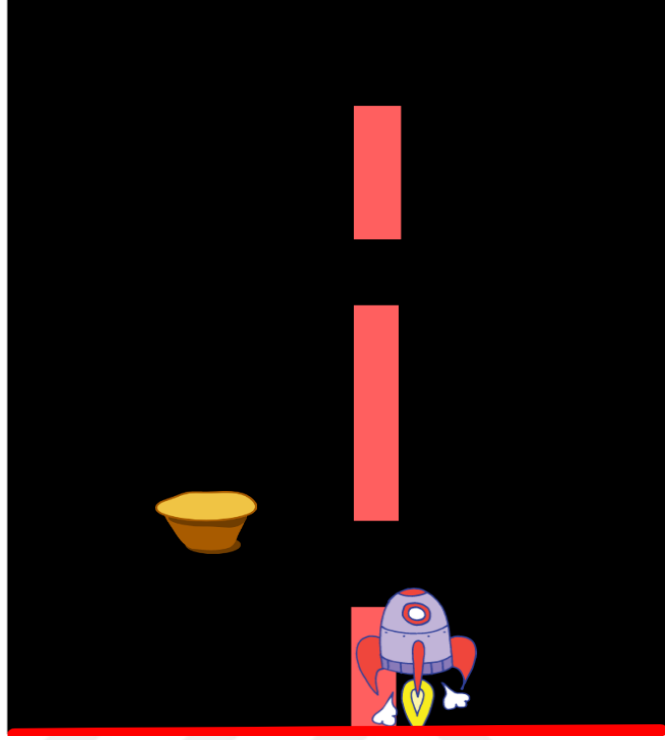


Kaya kuklası seçilir, kırmızı çizgiye değdiği anda gizlenip gösterilerek farklı konuma gitmesi sağlanır. Kayanın boyutu ve yukarıdan aşağı doğru gelmesini sağlayan diğer bir kod bloku oluşturulur.



Ekranda çizgilerin hareket etmesini sağlayan kod blokları oluşturulur.





4.1Tasarla: Labirent Oyunu

Bu bölümde öğretmen öğrencilerden labirent yapma oyunu isteyecektir. Oyun yapılırken dikkat edilecek hususlar şunlardır.

-Oyunun dekor bölümü üç sahneden oluşacaktır. Bunlar labirent haritası (internetten labirent diye görseller kısmından bulunabilir.) Süre bitti yazan dekor, tebrikler yazan dekor şeklinde hazırlanır. Bu kısımda rehber öğretmen örnek bir labirent resmi öğrencilere verebilir.

<https://drive.google.com/file/d/1X58nzheZ4q0eMG1btwG8PfGkBz5SShSa/view?usp=sharing>

-Ekranı bir adet bitiş için bayrak veya herhangi kukla eklenir labirentin sonuna gelince oyunun bitiş bu logo sayesinde bitirilecektir.

-Oyunda labirentler arasında hareket edebilecek bir kukla eklenir. Bu kukla yön tuşları ile hareket edecek olup ekrandaki labirent çizgilerinin arasındaki boşluklardan hareket edecektir. Bu çizgilerden herhangi birine değdiğinde kukla başlangıç noktası ile belirtilen kısma dönecektir.

-Ekranada bir adet aktif biçimde süre olup devamlı olarak sayaç şeklinde devam edecektir. Bu süre bizim belirlediğimiz örnek olarak 50 sn aştığı anda oyunu bitirip süre bitti dekorunu açacaktır.

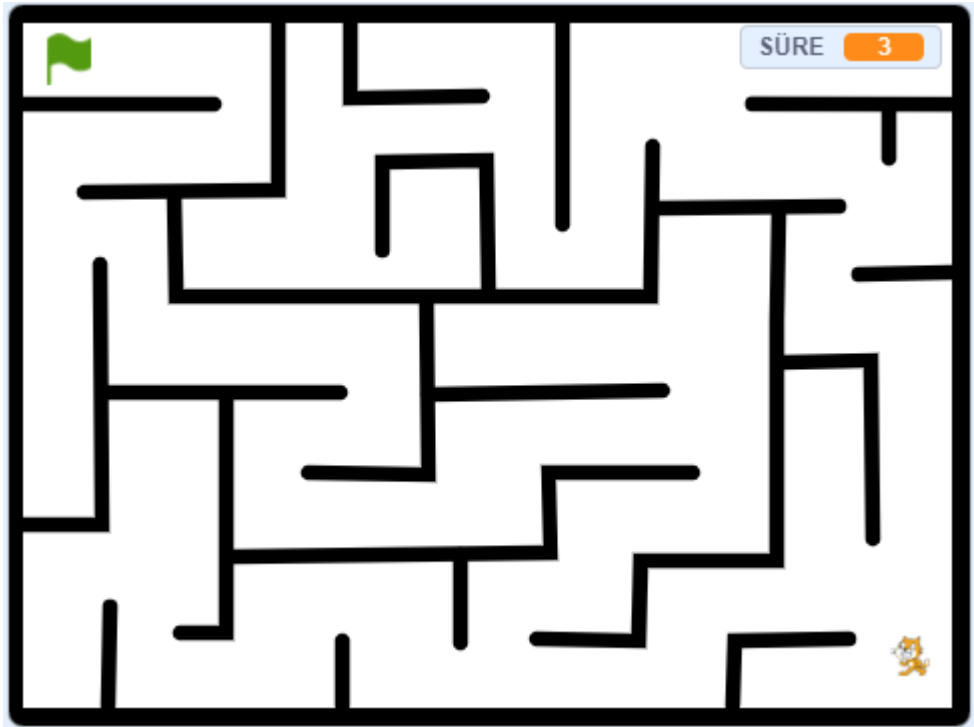
-Labirentin belli noktalarında bonus kuklaları çıkıp bu kuklalar yenildiği vakit süre kısmında belirlenen süre kadar azaltma yapıp oyunun daha geç bitmesi için kullanıcıya avantaj sağlayacaktır. Bu çıkan bonuslar aynı anda çıkmayıp labirentin belirli yollarında çıkacaktır. Buradaki süre yapımcı tarafından hesaplanacaktır. Çıkan bonus ise ekranada belirli bir süre bekleyip daha sonra yok olacaktır. Her bir bonus alındığı anda ekranada başarıyı temsil eden bir ses çalınır.

Öğrenciler yukarıdaki maddelerde oyun hakkında bilgi sahibi olup oyunun algoritmasını oluşturmaları istenir. Oyunun süre bitişleri ve diğer aşamalarında kullanacağı kod blokları hakkında düşünmeleri için belirli süre verilir.

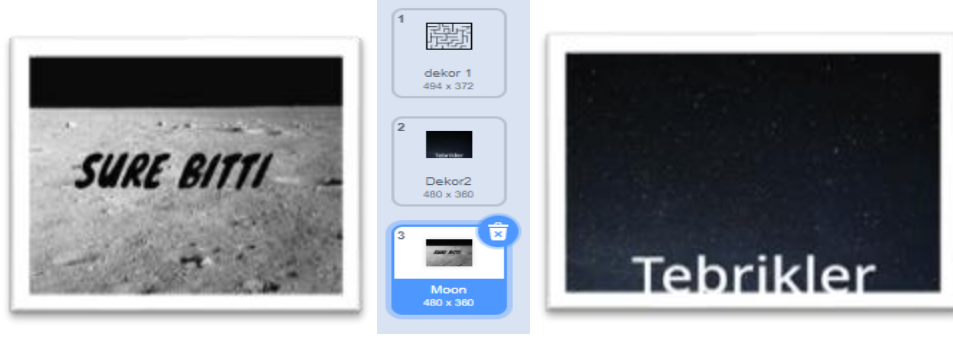
4.1Üret: Labirent Oyunu

Öğrenciler tasarımlarını ve algoritmalarını oluşturduktan sonra scratch programında uygulamaları istenir.

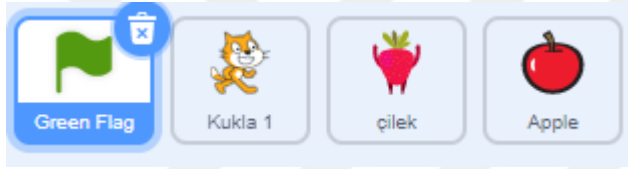
Aşağıdaki örneğe benzer bir örnek oluşması beklenir. Öğretmen sadece rehberlik görevi yapmalıdır.



Ekrana gerekli dekorlar yüklenir ve çizilir.



Daha sonrasında gerekli kukla eklemeleri yapılır.

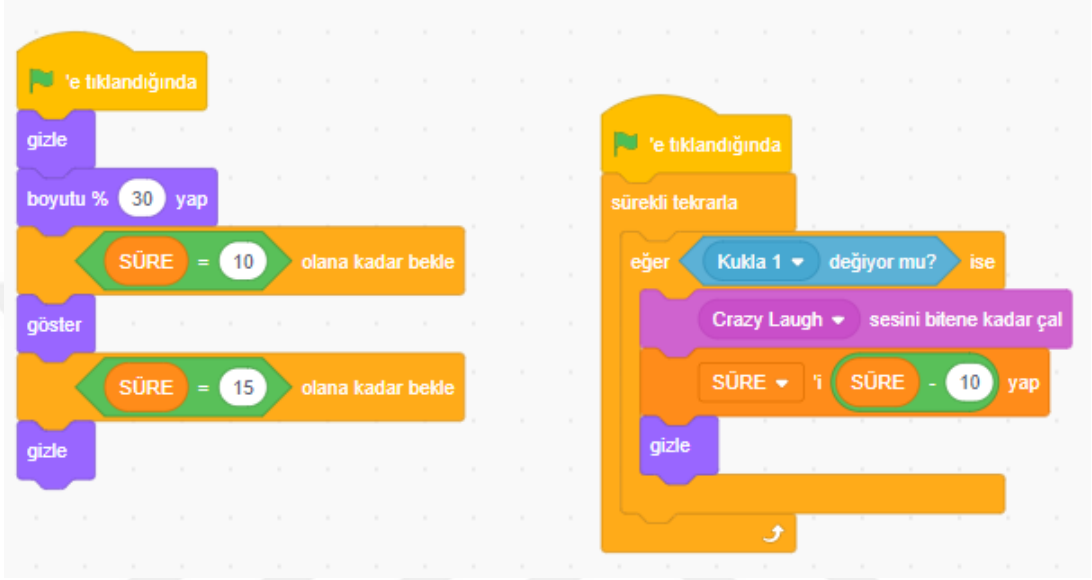


İlk başta yeşil bayrak kuklası kod bloğu tıklanır ve süre ayarlamaları burada yapılır. Oyunun bitiminde gelen haber ile gizlenmesi sağlanır.

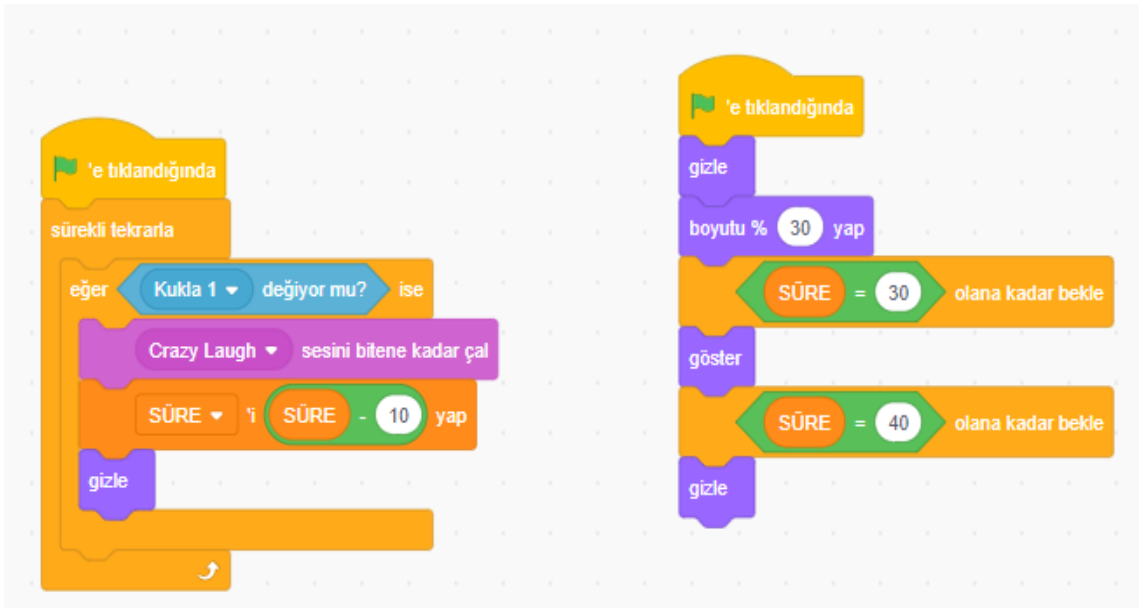


Bonus kodlarına geçilir ve bonus için belirlemiş olduğumuz çilek ve elma kodlarına aynı yazım yapılır. Bu kısımda boyut ayarlama ve belli sürelerde ekrana çıkıp kaybolması ile ilgili kodlar yazılır. Diğer bonuslar da dikkat edilmesi gereken çıkacak olan sürenin farklı olmasıdır. Elma kuklasında 10 ila 20 sn ise

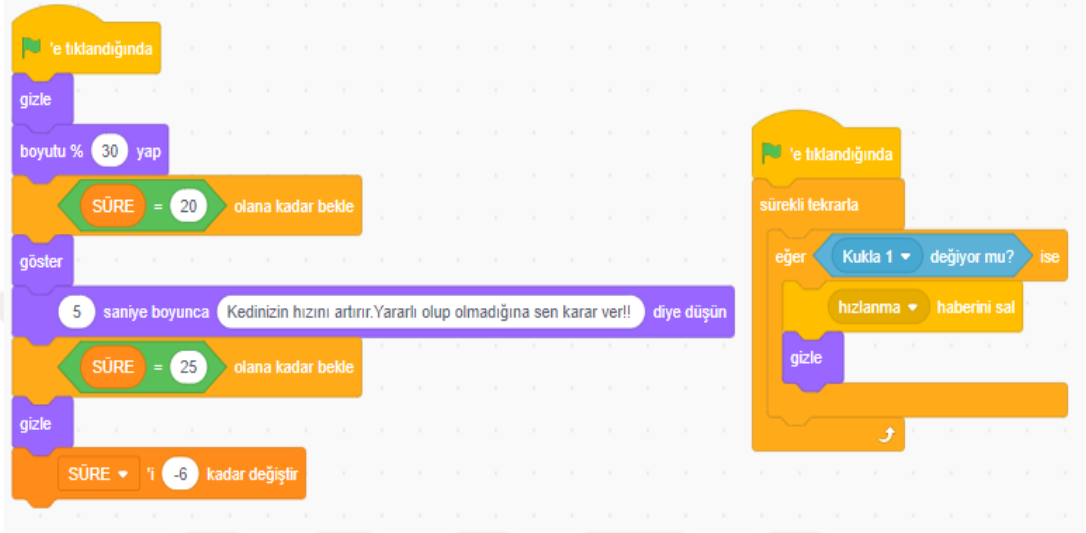
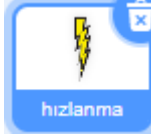
diğer kuklalarda farklı süreler kullanılır. Diğer kısımda ise kedi kuklası değdiği zaman sürenin azalması ve oyunun daha geç bitmesini sağlar.



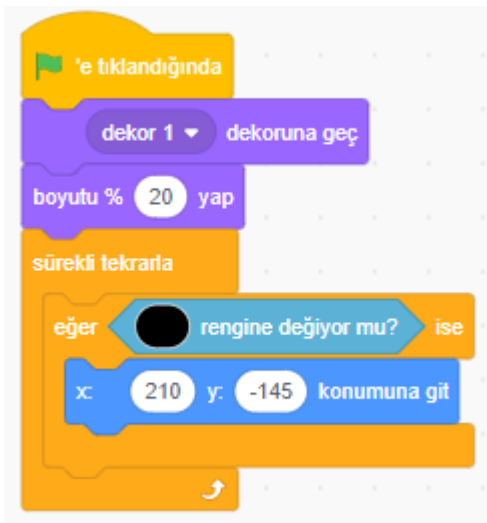
Elma kuklası için gerekli kodlar yazılır.



Şimşek kuklası seçilir ve ismi hızlandırma olarak değiştirilir. Şimşek kuklası çıktığında kedi kuklasının hızlanması için gerekli kod blokları yazılır.



Kedi kuklası seçilir. Labirentin siyah çizgilerine değdiği anda başlangıç noktasına gitmesini gerektiren kod blokları yazılır.



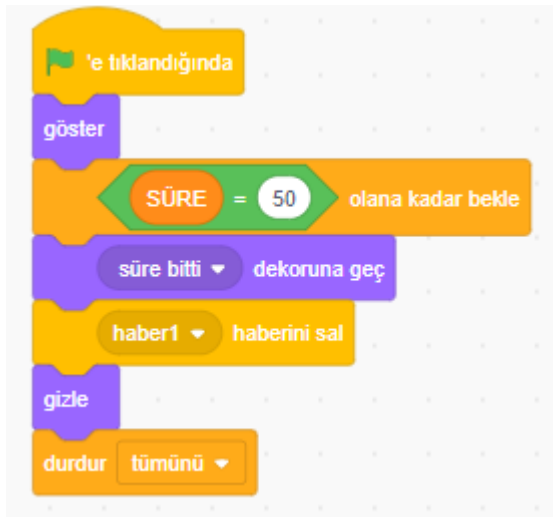
Kedi kuklasının labirentte hareket etmesini sağlayan kod blokları eklenir.



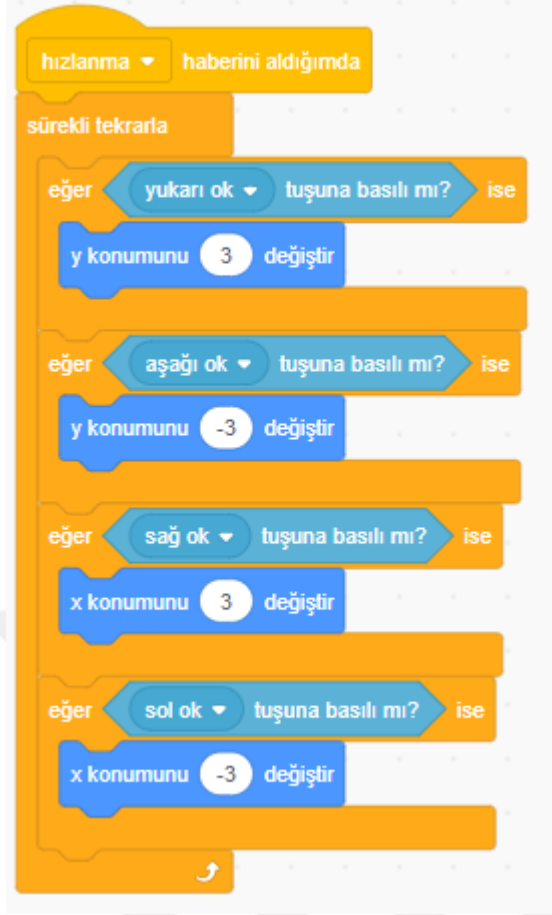
Kedi kuklasının yeşil bayrağa değdiği anda oyunun bittiğini söyleyen ekrana geçip, animasyon yaptıran kod blokları yazılır.



Sürenin bitmesi ile oyunun bittiğini gösteren ekrana geçmesini sağlayan kod bloğu yazılır.



Kedinin hızlanma kuklasını yakaladığı anda hızının artması için gerekli kod blokları yazılır.



1.3 Değerlendir

Bu bölümde hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede öğrenciler, problem çözme, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendisini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

Karşılaştığınız problemleri tanımlayınız (problemi kendi cümleleri ile ifade etme).

- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.

- o Bonus kodlarını oluştururken hangi aşamalarda zorlandınız.

- o kukla veya sahne kodlamaları konusunda zorlandım.

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?

- Grup arkadaşınızla fikir ayrılığına düřtüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?



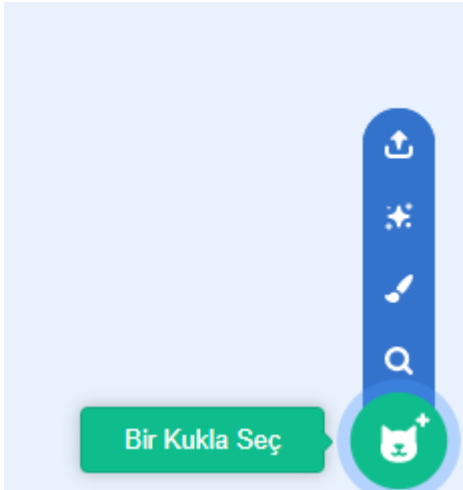
EK 3 Doğrudan Öğretim Modeli Ders Hazırlık Planı

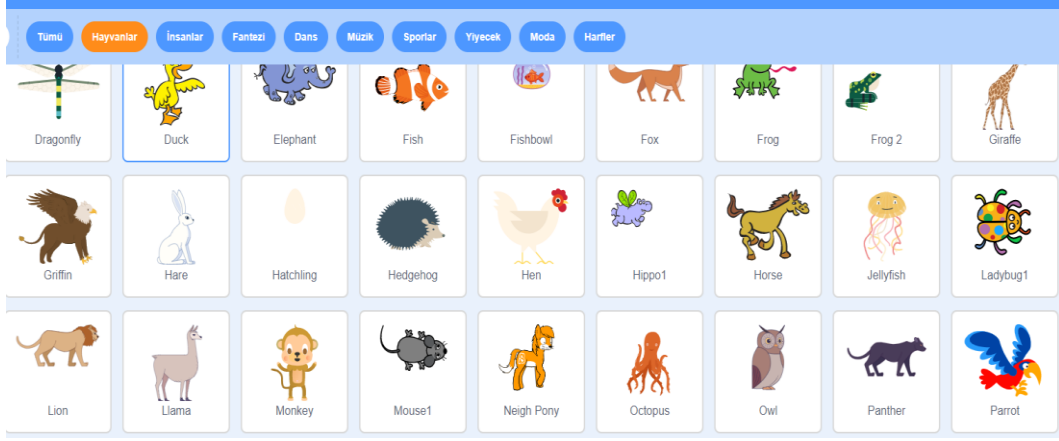
Ders1: Papağanın Kanat Çırparak Uçmasını Gösteren Animasyon Yapımı

Animasyon resimlerin ya da hareketsiz nesnelerin arka arkaya hızlı bir şekilde gösterilmesiyle karakterlerin hareket ediyormuş gibi görünmesini sağlayan bir gösterim şeklidir. İlk animasyonlar birkaç kâğıda işlenen resimlerin çizilmesi ve kağıtların hızlıca geçirilmesi veya bir çemberin içine yerleştirilip döndürülmesi ile yapılıyordu. Çizgi filmlerde birer animasyondur. Scratch programı ile de kuklaların şekillerini değiştirerek bizde animasyon oluşturabiliriz. Scratch programının kostümler sekmesinde karakterlere birden fazla şekil çizmemize ve hareket ettirmemize olanak vermektedir.

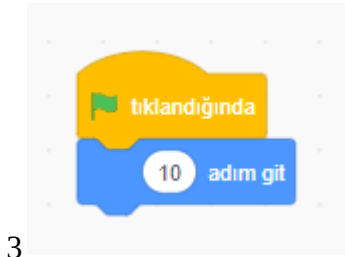
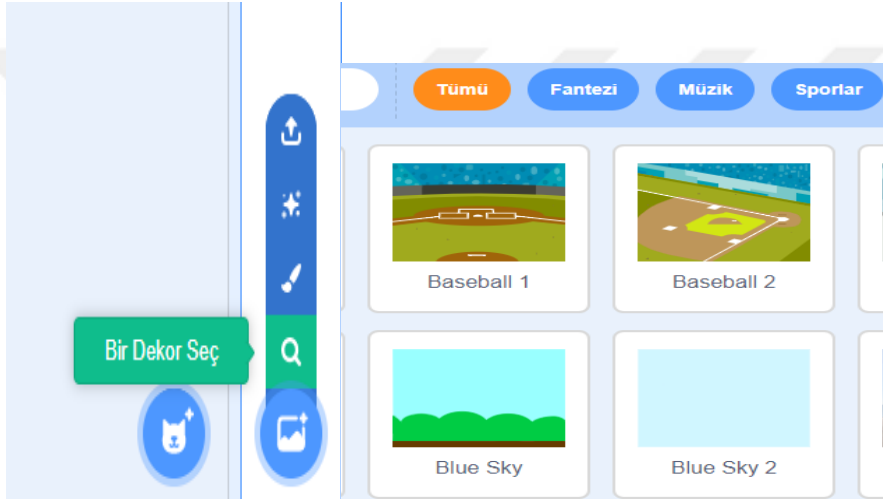
Bu uygulamada bir papağan karakterlerinin kostüm değiştirerek sahnenin sağından soluna doğru uçuşmasını sağlayan bir program yazılacaktır. Bu kısımda öğretmen kendisi yapıp öğrencilerin program yapımını takip etmelerini isteyecektir. İlk aşamada sahneye gerekli karakterlerin seçilerek tasarım kısmının oluşturulması sağlanmalıdır. Tasarım kısmının nasıl oluşturulacağı aşağıdaki şekillerde belirtilmiştir.

1-Öğrencilerden kukla ekle menüsüne girip papağana benzer bir karakter seçmeleri istenir.

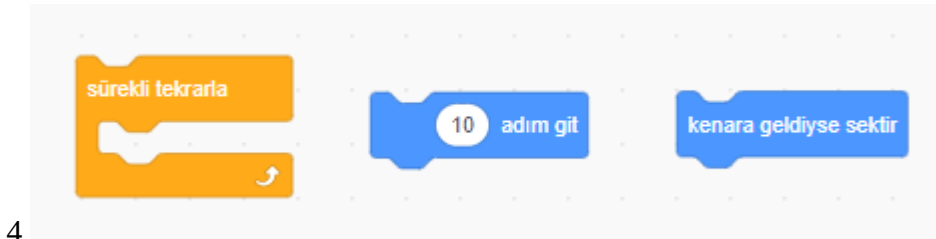




2-Dekor seç menüsünden herhangi bir dekor seçilir.



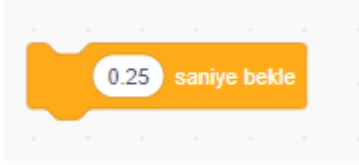
3 Şimdi ekrana tıklandığında kod bloğunun altına 10 adım git kod bloğunu koyarak papağanın hareket etmesini deneyelim.



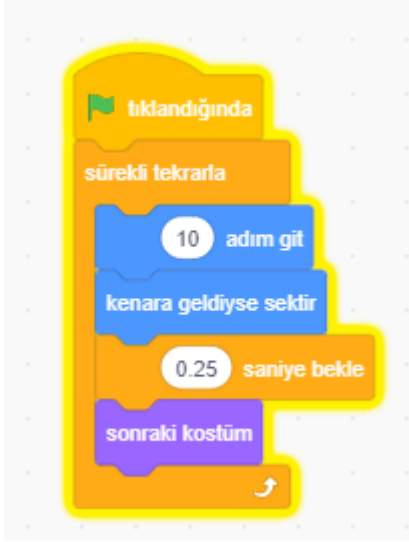
4

Yukarıdaki kod bloklarını kullanarak papağan karakterimizin ekranda devamlı olarak 10 adım gidip kenara değdiği anda ise geri gelmesi için kenara geldiyse sektir komutunu deneyelim.

5.papağan karakterimizin yavaş hareket edebilmesi için 0.25 sn bekle komutunu yazalım.



6.papağanın uçuyormuş gibi görünmesi içinde sonraki kostüm hareketini ekleyelim.



Papağanımız şimdi sağa sola doğru hareket edebiliyor.

Sen yapsaydın neleri değiştirmek isterdin? Öğrendiklerimizle projeyi değiştirebilirsin.

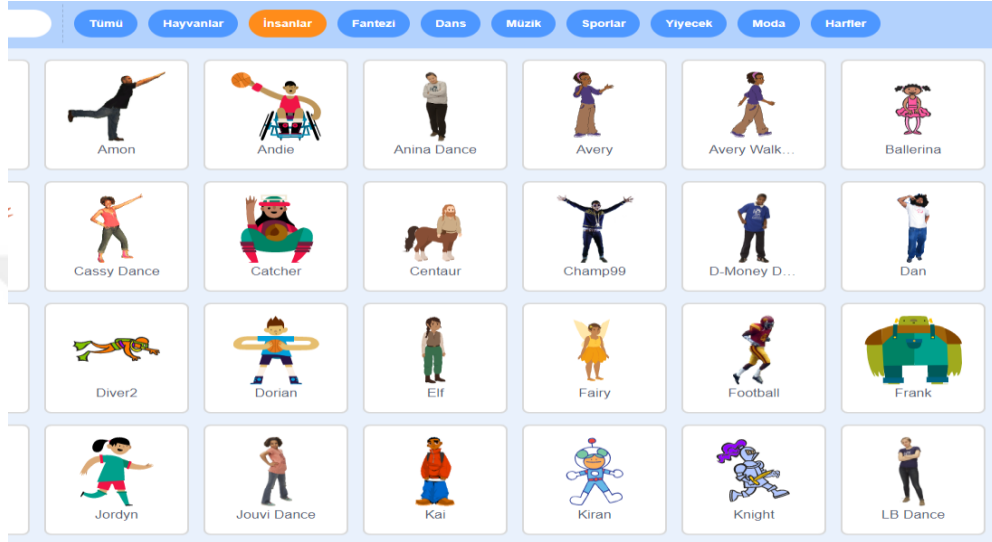
*kuklayı değiştirebilirsin

*adım sayısını değiştirebilirsin

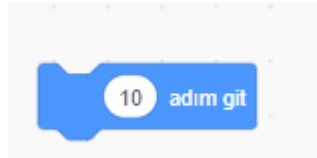
2-Sağa-Sola Doğru Yürüyen İnsan Figürü Yapımı

Bu kısımda öğretmen öğrencilerden sahneye yürüyen bir insan kuklası koyarak bu insan figürünün sahnenin sağına ve soluna doğru yürüyormuş gibi görünüp dolaşmasını isteyecektir.

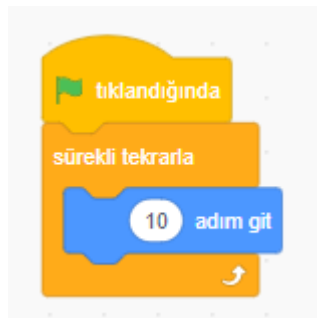
1-Ekrana herhangi bir insan kuklası ekleyelim.



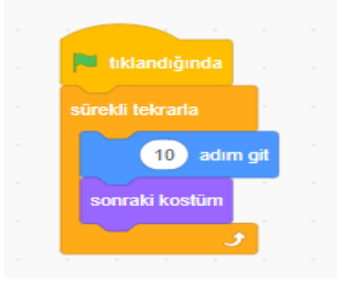
2-insan figürünün hareket edebilmesi için 10 adım git kod bloğunu ekleyelim.



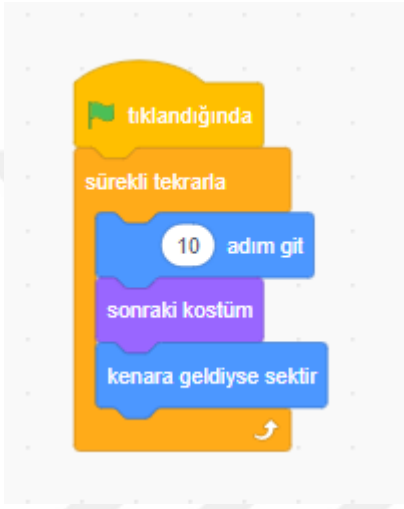
3-Yürüme işleminin sürekli olabilmesi için sürekli tekrarla kod bloğunu ekliyoruz.



4-İnsan figürünün yürüyormuş gibi gözükmesi için sonraki kostüm seçeneği eklenir.



5-Kenara geldiği anda kuklanın geri dönmesi için kenara geldiyse sektir komutunu ekleriz.



6-Kenara gelip sağa sola sektiği anda yönünün de değişmesi için dönüş stilini sol sağ yap deriz.



7-Tüm bu işlemlerden sonra figürümüz sağa sola doğru hareket etmeye başlar.

Sen yapsaydın neleri değiştirmek isterdin? Öğrendiklerimizle projeyi değiştirebilirsin.

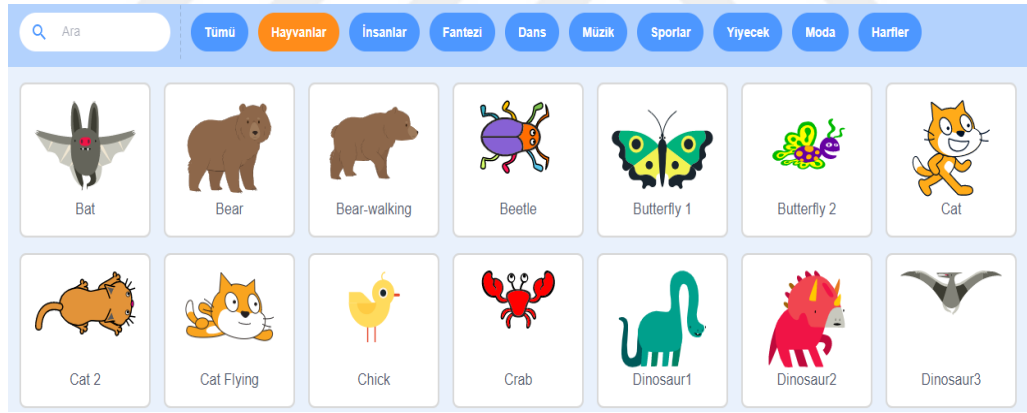
*kuklayı değiştirebilirsin

*adım sayısını değiştirebilirsin

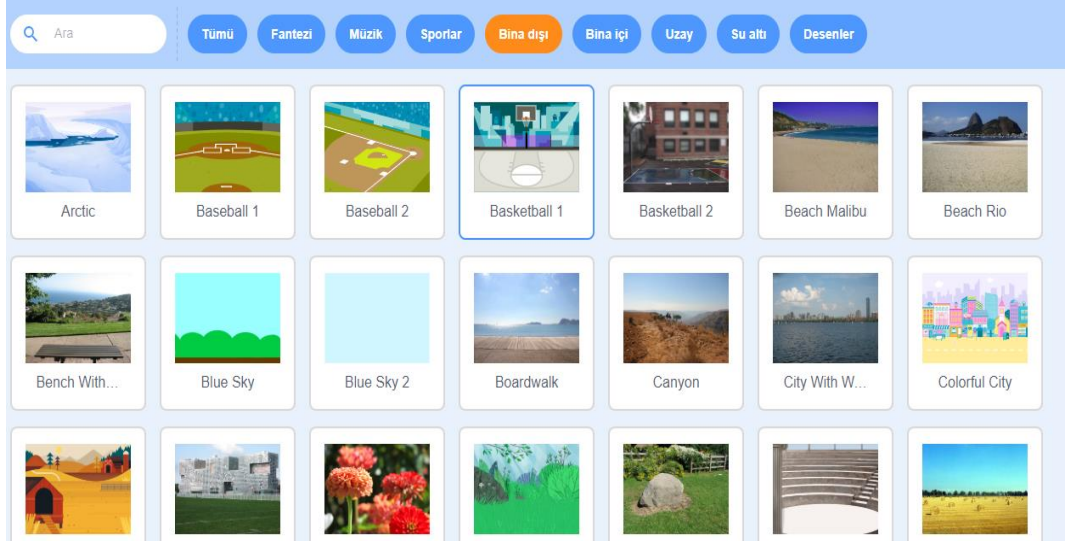
3- Ekranda Fare İmlecini Takip Ederek Uçan Kedi Animasyonu Yapma

Scratch programında fare imlecimizin programa eklediğimiz kuklalar tarafından takip edilmesini sağlamak için fare imlecine git kod bloğu kullanılmaktadır. Burada dikkat edilmesi bloğunun sürekli tekrarla altında çalıştırmamız gerekmektedir. Örnek verecek olursak yeşil bayrağa tıkla dediğimizde sadece fare imlecine git komutunu eklersek komut bir kere okunur ve bir defa fare imlecine gider ve kalır.

1-Ekrana hayvan kuklalarından herhangi bir kedi figürü eklenir.



2-Dekorlar kısmından dekor seçimi yapılır. İstenilen herhangi bir dekor seçilir.

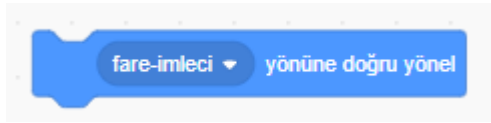


3-Kedinin fareyi takip edebilmesi için bu kısımda fare imlecine git komutu kullanılır. Bu komut ile ekrandaki rastgele konuma gitme seçeneği de



ayarlanabilir.

4-Eklenen figürün fare ile aynı yönde olabilmesi için ise fare imleci yönüne doğru yönel komutu kullanılır.



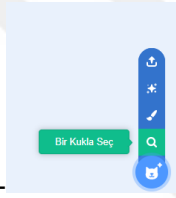
5-Son olarak yukarıdaki verilen bütün kodları birleştirerek kuklanın fareyi takip etmesini deneyiniz.



6-

Sürekli komutu altına kod bloklarını koyarak işlemlerin devamlı olarak yapılması sağlanır.

4-Ekranda Rastgele Hareket Eden Balon Animasyonu



1-

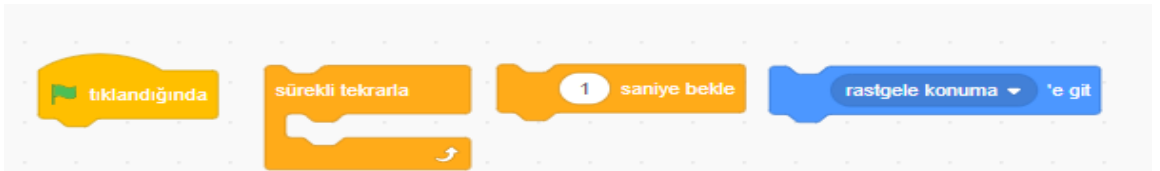
Bir kukla seç menüsünden balon şeklinde bir kukla seçilir.



2- Örnek olarak yukarıdaki Balloon1 kuklası seçilir.

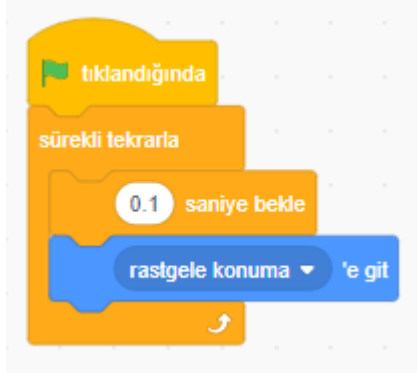
3-Arka plan beyaz olarak bırakılabilir veya istenilen bir dekor seçilebilir.

4-



Yukarıdaki kod blokları scratch programında tahtada gösterilerek öğrencilere 5 dk süre verilir. Öğrencilerin balonun sağa sola hareket etmesi için gerekli kod

bloklarını yazmaları istenir. Sınıftan seçilen birkaç öğrenciye söz hakkı verilir ve yaptığı yerleşim düzenini açıklaması istenir.

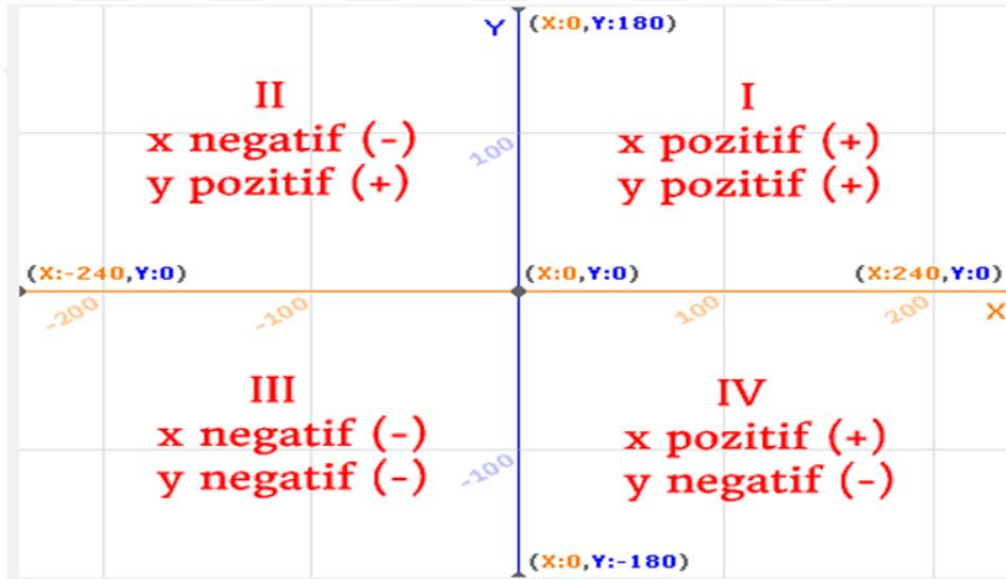


5-

Scratch programında yukarıdaki kod blokları uygulanır. Öğrencilere gösterilir.

5- Konum Bulmaca Oyunu Yapma

Bu oyunda amaç öğrencilerin koordinat düzlemi hakkında bilgi almaları sağlamaktır. Ekranda bulunan araba kuklasının mavi, sarı, yeşil ve kırmızı toplara değdiğinde arabanın bulunduğu konumu söylemesi istenir.



1-

Öğretmen ekrana koordinat sistemini gösteren şekli öğrencilere gösterir ve Y eksenin pozitif ve negatif yönlerin nereler olduğu hakkında bilgilendirilir.

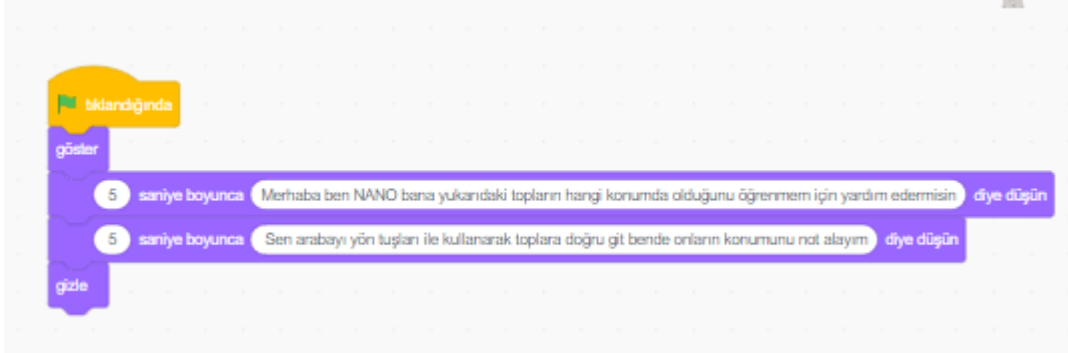
2-Öğrencilerin bir kâğıda veya paint ekranını kullanarak koordinat sistemi çizmeleri istenir. Bu sistemde (x:120, y: -100), (x: -150, y: -50), (x:140, y:60), (x:25, y:52) olduğu koordinat noktalarını göstermeleri istenir.

3-Scracht programının açılıp öğretmenin ardından öğrencilerin aşağıdaki uygulamaları yapmaları istenir.

4- İlk önce sahne seçilir. Sahnenin koordinat sistemi ile ilgili olması için xy-grid seçilir.



Oyunu oluşturmak için Nano kuklasını seçiyoruz. Ve aşağıdaki kod bloklarını oluşturmaya başlıyoruz.



Burada kuklanın 5 saniye boyunca oyunda olacağını anlatmasını sağlıyoruz.”

Merhaba ben NANO bana yukarıdaki topların hangi konumda olduğunu öğrenmem için yardım eder misin?”

“Sen arabayı yön tuşları ile kullanarak toplara doğru git bende onların konumunu not alayım”

Şeklinde uyarı cümleleri eklenmiştir. Kuklamız oyunun başında ekranda gözükecek ve daha sonra gizlenecektir. Burada gizle komutunu kullanmamızın

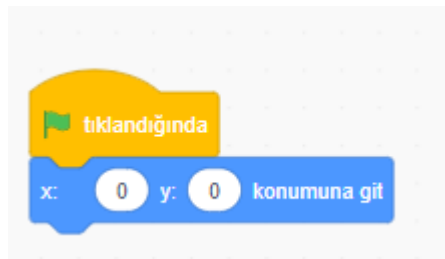
sebebi cümleleri oluşturduktan sonra kuklanın ekrandan kaybolması göster kod bloğunun kullanılmasının sebebi ise kuklanın yeşil bayrağa basılınca oyunun tekrar başlangıcında sahnede gözükmesi için kullanılmaktadır.

Sahneye araba kuklası eklenir.

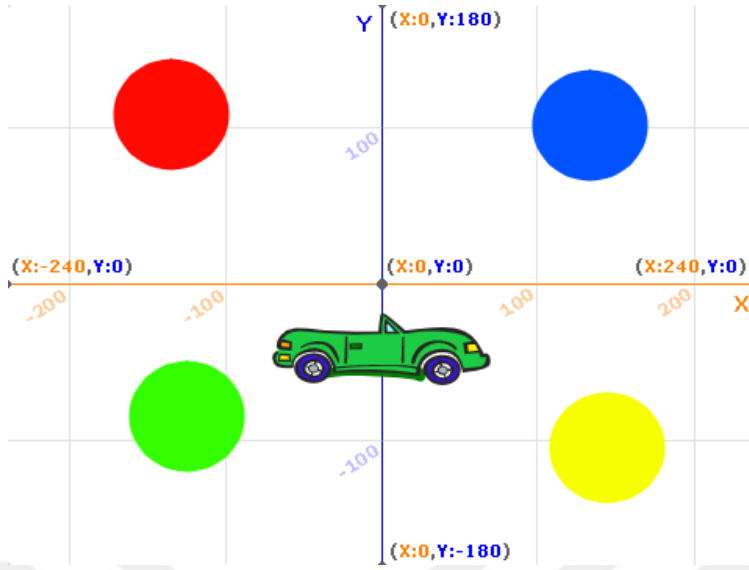
Arabanın sağa-sola yukarı-aşağı hareket etmesi için kod blokları oluşturulur. Örnek olarak yukarı ok tuşuna basılınca komutu ile kuklanın y konumunda ve pozitif yönünde yani yukarı gidebilmesi için 10 yazılır. Aşağı yönlü gidebilmesi içinde -10 yazılır. Diğer konumda da bu uygulanır.



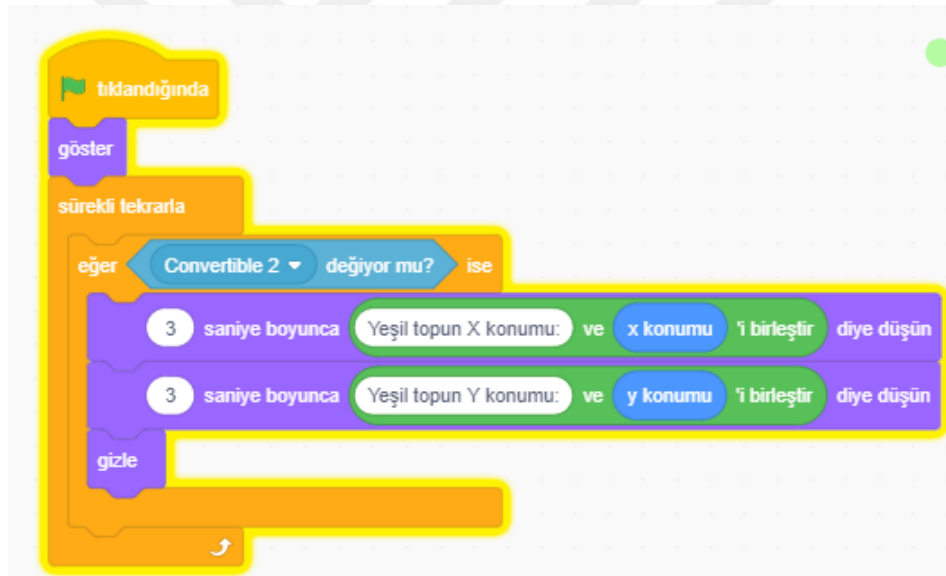
Arabamızın ilk oluştuğunda hiçbir yere değmemesi için 0,0 konumu için aşağıdaki kod bloğu yazılır.



Sahnemize 4 adet top kuklası eklenir. Ve aşağıdaki konuma getirilir.



Yeşil top kuklası seçilerek kod blokları yazılır.



Burada “eğer” kod bloğu bir şart ifadenin gelip gelmediğini sorgular. Başka bir deyişle şart ifadenin sonucu “doğru” veya “yanlış” değer üretir. Bu örnekte de convertible2 (araba) yeşil topa değip değmediği kontrol edilmektedir.” Değiyor mu “komutu ile sahnede bulunan tüm kuklaların değme işlemi ve mouse temas işlemi kontrol edilebilir. “Birleştir komutu ile yazdığımız yazıları veya içine koyduğumuz kod bloğundaki koordinat sistemindeki yazıyı birleştirme işlemi sağladık. Ve kuklamızın gizle komutuyla kaybolmasını sağladık.

Bu kod bloklarının aynısını diğer top bloklarımıza da ekliyoruz.

6-Koordinat Düzleminde Bölgeleri Öğrenme Animasyonu

Bu uygulama da amaç ekranda serbest bir biçimde hareket eden bir topun koordinat sistemlerini tanıtmayı amaçlanmaktadır.

Öğretmen öğrencilere bir önceki oyunda öğrenmiş oldukları koordinat sistemlerini tekrar eder ve aşağıdaki hatırlatmayı yapar.

Animasyonda x ve y koordinatlarının hangi bölgelerde – veya + olduğunu ekranda hareket eden bir top vasıtasıyla gösterilir.

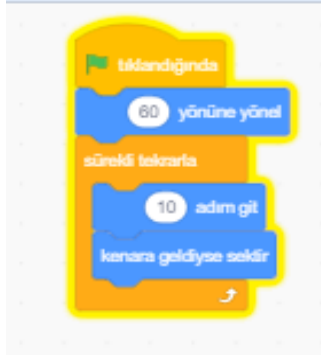
Daha sonra uygulama için dekor seçimi yapılır ve sahnenin koordinat sistemi ile ilgili olması için xy-grid seçilir.Bu kısımda öğretmen ve öğrenciler aynı anda uygulamaları yapar.



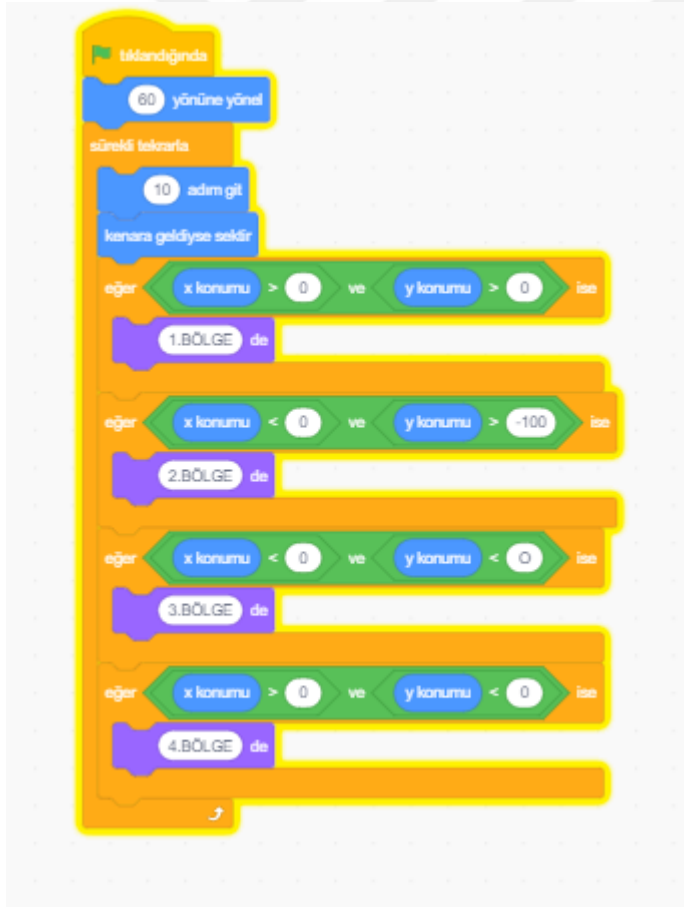
Bir tane top kuklası seçilir ve sahneye eklenir.

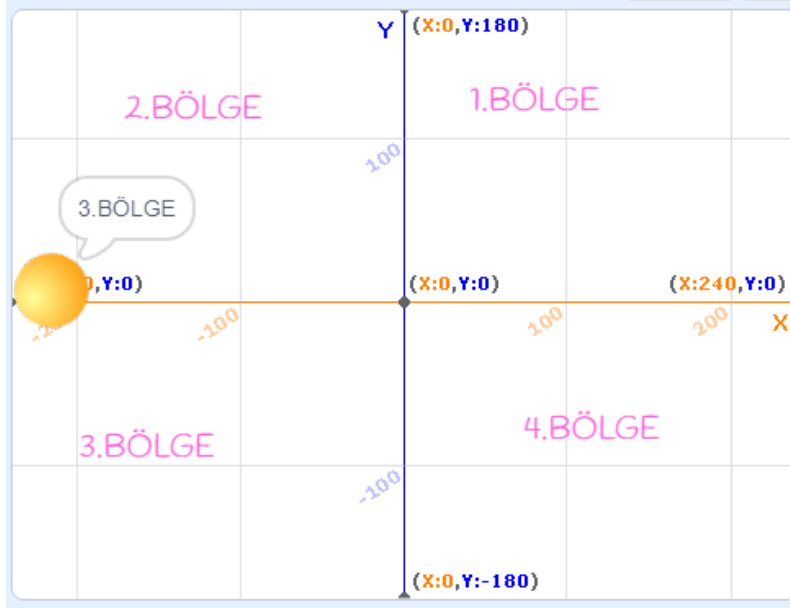


Topun sahnede bölgeler arasında hareket etmesi için top kuklası seçilir ve aşağıdaki kod blokları oluşturulur.



Topun Bölgede olduğunu ekrana yazdırmak için topun x konumunun 0'dan büyük ve konumunun da yine 0'dan büyük olması durumunda ekrana I. bölge de yazısı yazılacaktır. Burada “ve” komutu ile her iki şartında sağlandığı durumda komut çalışmaya başlar. Diğer bölgeler içinde aynı kod bloğu uygulanır.





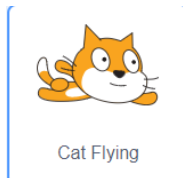
7-Kuş Yakalama Oyunu

Bu bölümde öğrencilerden fare ile hareket eden kuklanın sahnede uçan kuşları yakalama oyunu yapmaları istenecektir. Ekrana eklenen istenilen sayıdaki kuşun seçimi yapılan herhangi bir kukla tarafından yakalanması istenilmektedir.

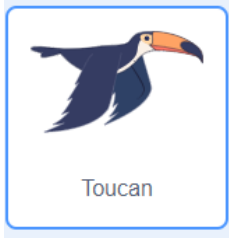
Öğretmen öğrencilere kuşların uçabileceği bir dekor eklemesi istenir. Örnek olarak Blue Sky dekorunu kullanabileceklerini söyler.



Daha sonra ekrana cat flying kuklası eklenir.



Kedinin yiyeceđi kuş karakteri sahneye eklenir. Örnek olarak Toucan kuklası eklenir. Öğrencilere uçan animasyonunu gösterebilen başka kuklalar da eklenebileceđi söylenir.



Kedi kuklası seçilerek öğrencilere aşağıdaki kodlar ile kedinin ekranda fareyi takip ederek ilerlemesi için gerekli kod düzenlemesi istenir ve belirli bir süre verilir.



Kod blođu kediye eklenerek sürekli tekrarla kod blođunun altındaki fare imlecine git komutunu çalıştırdıđı söylenir. Kedinin fareyi hareket ettirildiđinde o yöne doğru gittiđi anlatılır.

Uçan kuşların kodlama aşamasına geçilir ve öğrencilere aşağıdaki sorular sorulur.

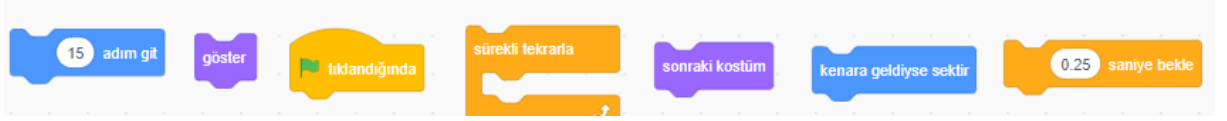
1-Kedinin kuşlara deđdiđi anda ekranda kuşların kaybolması için hangi kod blođunu kullanılır?

Cevap: gizle komutu

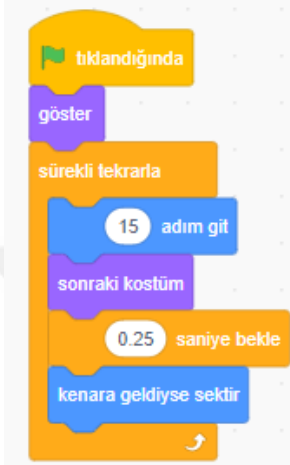
2-Kuşların uçar gibi gözükmesi için hangi kod blokları kullanılır?

Cevap: Sürekli kod blođu altında sonraki kostüm

Öğrencilerden cevaplar alındıktan sonra scratch ekranı açılarak aynı sırada aşağıdaki kod şemaları verilir ve denemeleri için süre verilir.



Doğru kodlama blokları gösterilir.



Kedinin kuşlara değdiği anda kuşların kaybolması için değiyor mu komutu kullanılır. Değdiği anda kuşların kaybolması için gizle seçeneği ile kuşlar sahneden kaybolur.

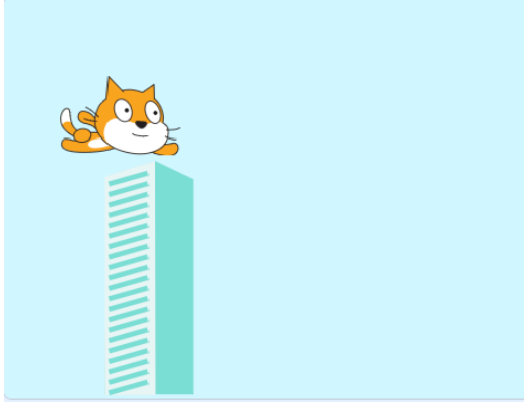


Kuşların bütün kodları gösterilir.

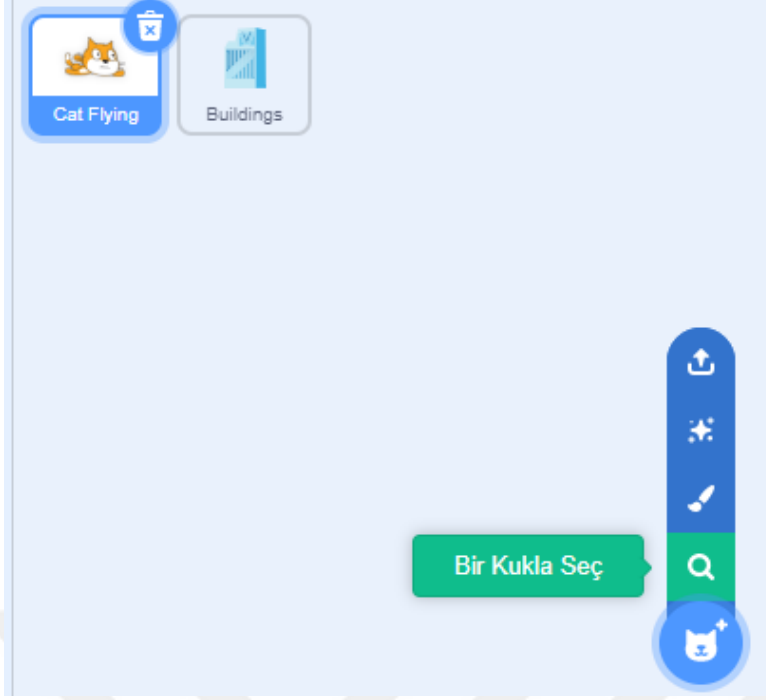


8- Binalar Arasından Uçan Kedi

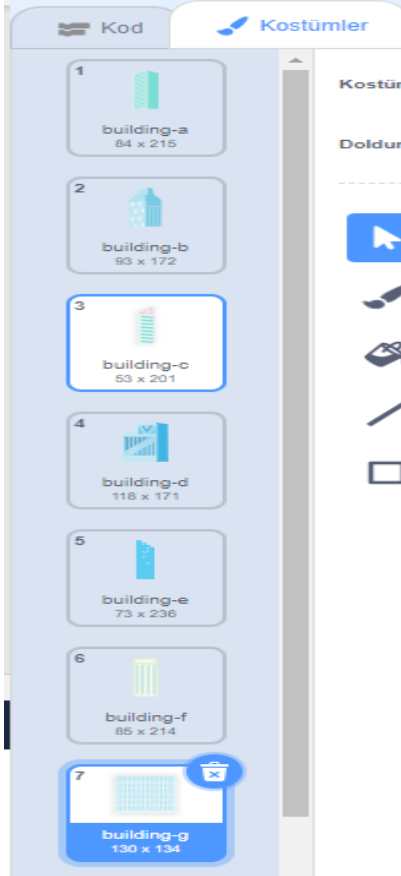
Bu bölümde öğrencilerden ekranda kayan binalar arasında uçan kedi yapmaları istenmektedir. Kedinin fare imlecini takip etmesi ve binalardan kaçması beklenmektedir. Binalardan herhangi birine değdiğinde de oyunun bitmesi amaçlanmaktadır.



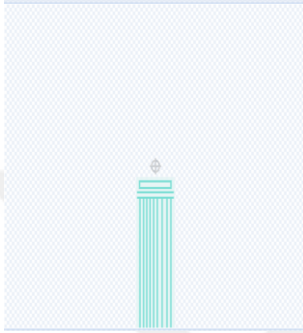
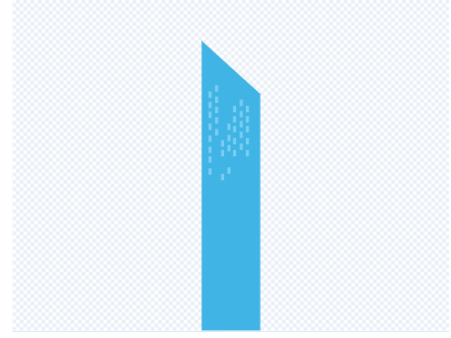
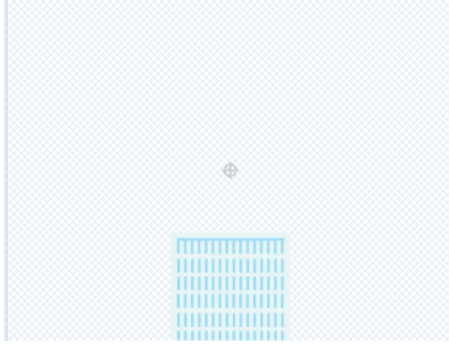
Bir kukla seç ile ekrana cat flying kuklası ve buildings kuklaları eklenir.



Buildings kuklasında kostümler sekmesi açılır ve öğrencilere gösterilir. Bu kısımda öğrencilere kostümler sekmesinde birçok gökdelen olduğu ve oyunumuzda buradaki gökdelenlerin geçişinin kullanılacağı söylenir.

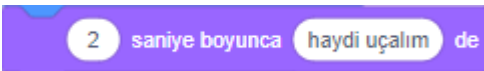


Ve gökdelenleri ekranın belirli noktalarında çıkması için ayarlamalar yapılır.

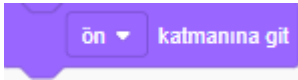


Kedi kuklasına geçilir ve öğrencilere aşağıdaki sorular yöneltilir.

1-Kedi kuklasına oyunun başlangıcında belirli bir sürede haydi uçalım yazdıran kod bloğu nedir?

Cevap= 

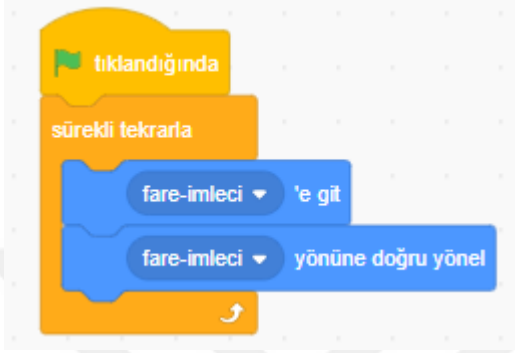
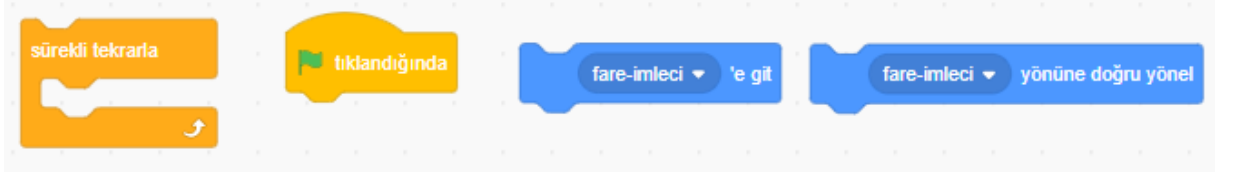
2-Kedi kuklasının en öndeki karakter olması için hangi kod bloğu kullanılır?



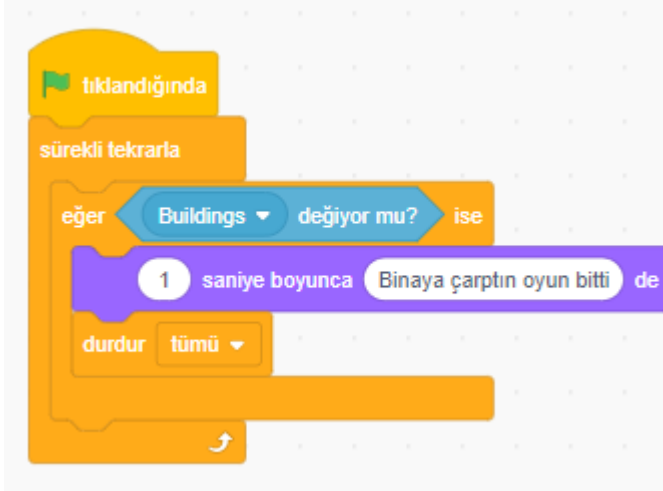
Kedinin başlangıç noktasında duracağı ve konuşacağı kod bloğu yazılır.



Aşağıdaki karışık haldeki kod blokları verilir. Öğrencilerden fareyi takip eden kod bloklarını oluşturan doğru düzeni yapmaları istenir.



Doğru bir biçimde kodların nasıl olacağı sınıfta söz hakkı verilerek tartışılır. Ve doğru kod bloğu yazılır.



Yukarıdaki kod bloğu oluşturularak öğrencilere bu kod bloğu ile binalara dokunulduğu anda 1 saniye boyunca binaya çarptın yazdırılır. Tümü durdur komutu ile uygulama durdurulur.

Buildings kuklası seçilir. Öğrencilere kostümler sekmesinde bulunana binaların sağdan sola doğru kaymasını sağlayan kod bloğunu yazmaları istenir ve belirli bir süre verilir.



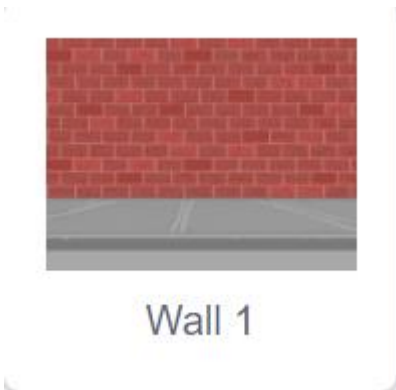
Buradaki kodlar yazılır ve öğretmen tarafından açıklamalar yapılır. X ve y konumunda ilk başta nerede çıkacağı belirtilir. Kostüm değişikliği yapılır. Sağdan sola binaların kayıyormuş gibi gözükmesi için x konumunu değiştir seçeneği 100 defa tekrarlatılır.

Öğrencilere aynı işlemi yapmaları söylenir.

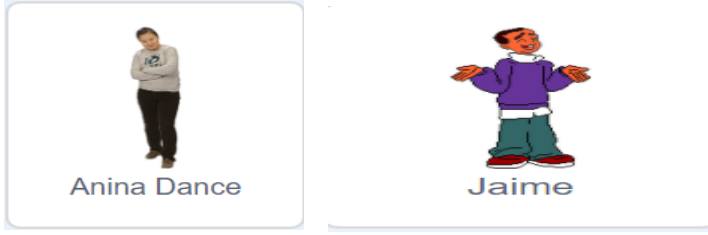
9- Arkadaşının Sınavdan Aldığı Notu Söyleyince Dans Eden Kız

Bu bölümde öğrencilerden ekranda 2 adet insan kuklası bulunması istenecektir. Burada bir kukla diğer bir kuklaya koşarak yaklaştığında duracak ve sınavdan 100 aldığını söyleyecektir bunu duyan diğer kukla sevinip dans etmeye başlayacaktır.

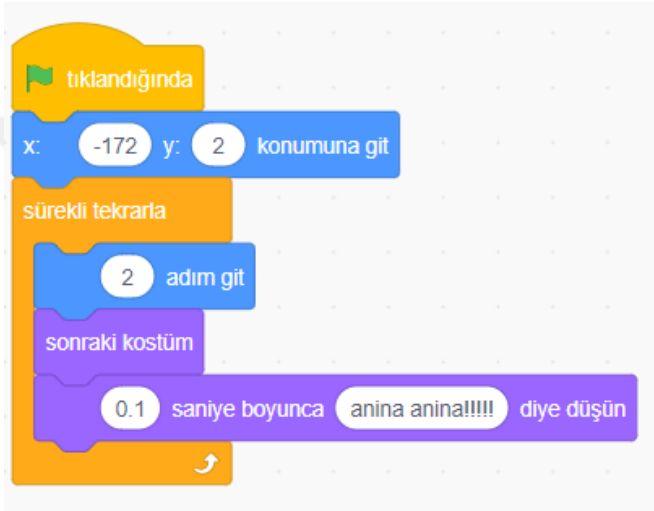
Wall dekoru seçilir.



Daha sonra ekrana yürüyen ve dans eden figürleri içeren insan kuklaları eklemeleri istenir. Burada kuklaların adının değiştirebileceği söylenir.



Jaime kuklası seçilir ve kodları oluşturulmaya başlanır.



Bu kısımdaki kodlar sırası ile yazılarak öğrencilere aşağıdaki sorular sorularak ekleme yapılır.

1-Jaime animasyonun başında ekranın sol kısmında durması için hangi kod bloğu yazılmalıdır?

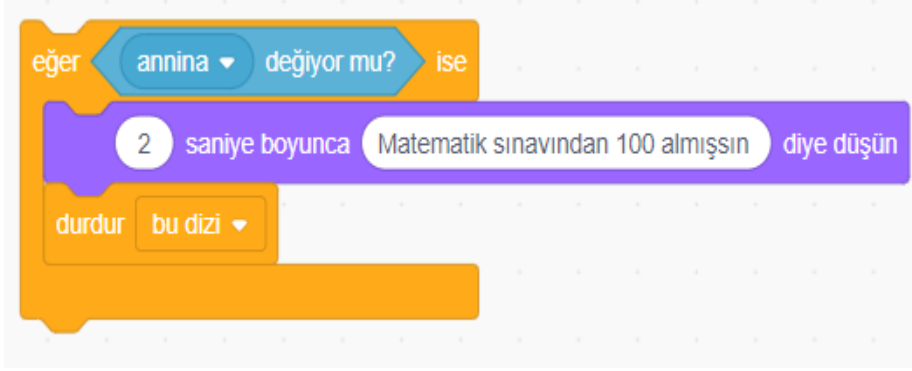
2-Yaptığımız işlemlerin sürekli olarak tekrarlanması için hangi kod bloğu eklenmelidir?

3-Kuklanın hareket etmesi için hangi kod bloğu eklenmelidir?

4-Jaime kuklasının yürüyormuş gibi gözükmesi için hangi kod bloğunu kullanırsınız?

5-Kuklanın yürürken bir anda da anina anina diye düşünmesini hangi kod bloğu ile yaparsınız?

Bu kod blokları yazıldıktan sonra öğrencilerden Jaime kuklasının Annina kulasına değiştiği anda matematik sınavından 100 almışsın deyip durmasını sağlayan kod bloğunu oluşturunuz denilip belirli bir süre verilir.

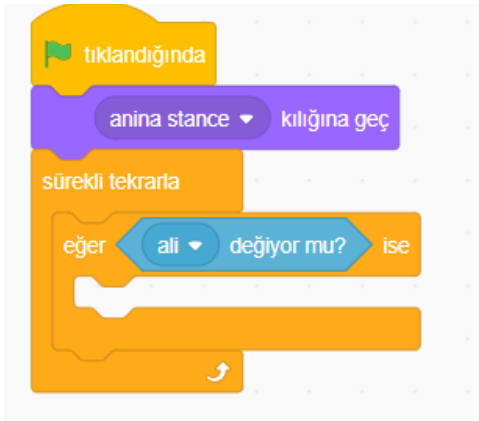


Annina kuklası seçilir ve sırasıyla aşağıdaki sorular sorularak kod bloğu oluşturulur.

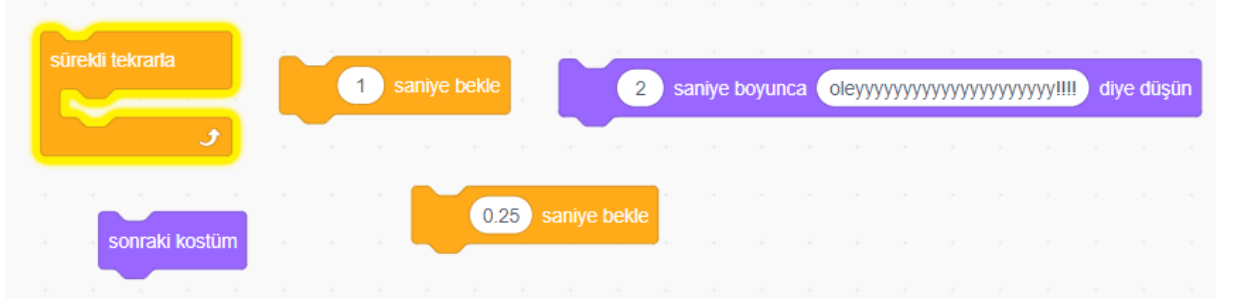
1-Anina nın animasyonun başında anina stance kılığında olması için hangi kod bloğu yazılır?

2-Dans etme ve diğer ilmelerin sürekli olarak tekrarlanması için hangi kod bloğu kullanılır?

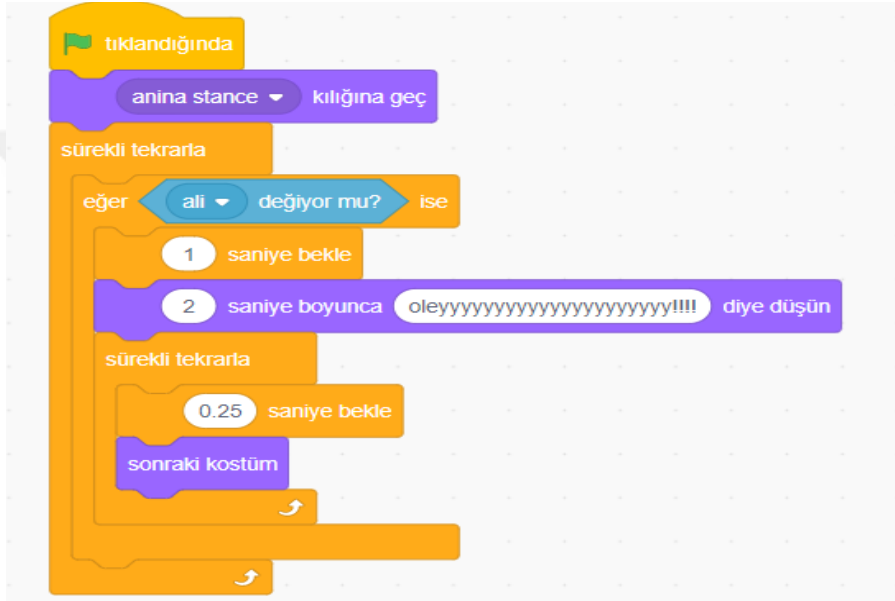
3-Jaime kuklasının Annina değiştiğini belirten kod bloğu hangisidir?



4-Aninanın oley diye düşünüp dans etmeye başlaması için aşağıdaki kod bloklarını düzenleyiniz.



Öğrencilere son hali gösterilerek onların da birkaç tane konuşma cümlesi ekleyebilecekleri söylenir.

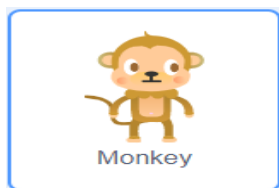


ÖDEV

Bir önceki uygulamada iki kukla arasında konuşma sağlanmıştı. Sizler de kukla sayısını artırarak ve konuşma metnini ve animasyonlarını değiştirerek kuklaların konuşmalarını sağlayınız.

10- İki Sayıyı Toplama İşlemi

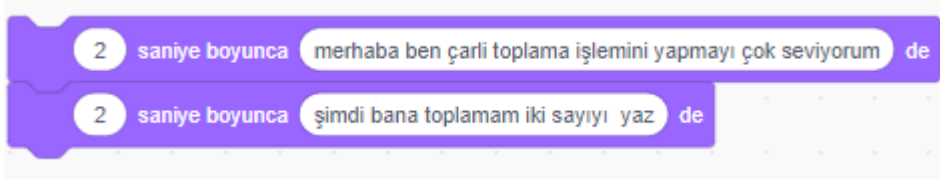
Bu uygulamamızda ekrana eklemiş olduğumuz maymun kuklası ile 2 sayının toplamını gösteren uygulama yapılacaktır. Değişkenler konusu anlaşılması zor bir konu olduğu için bu kısımda öğretmen tüm konuyu adım adım anlatır.



Ekranımıza monkey kuklasını ekliyoruz.

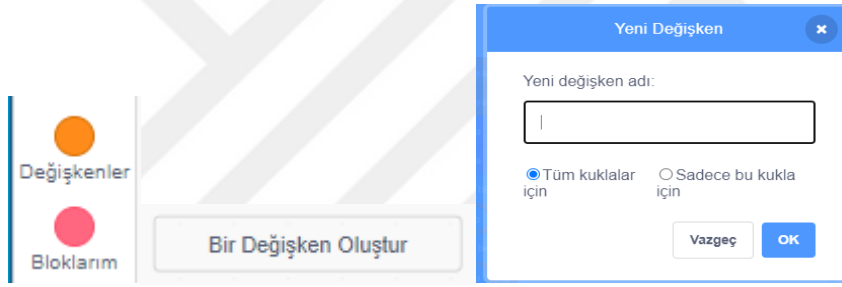
Uygulamamızın nasıl kullanılacağını anlatmak için konuşma balonları eklenir.

Öğrencilerden belirli bir sürede konuşan balonlar eklemeleri istenir.



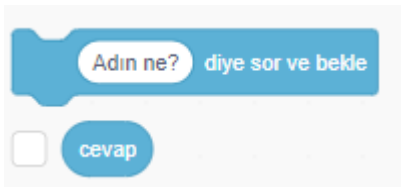
Yazdığımız uygulamalarda, projelerde kimi zaman bir sayıyı, kimi zaman bir ismi, haftanın günlerini, Türkiye'nin illerini, kaydolan kişinin soyadını yapılan işleme ve amaca göre geçici bir süreliğine hafızada tutmak isteyebiliriz ve bu işlemleri yapabilmek için değişkenleri kullanırız.

Değişken eklemek için aşağıdaki işlemleri kullanarak değişken ekleme işlemi yapılır.



Öğrencilerden 1. Sayı ve 2. Sayı değişkenlerini oluşturmaları istenir.

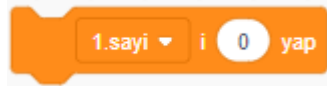
Öğrencilere değişkenleri oluşturduğumuzu ve bu değişkenlere ekrandan almış olduğumuz veri girişini nasıl yapabileceği sorulur. Ve aşağıdaki işlemler öğrencilere anlatılır.

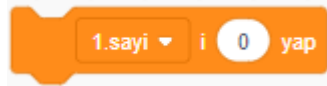


Komutu ile ekrandan kullanıcıdan veri girişi yapılması beklenir ve cevap değişkeninin içinde girilen veriler kayıtlı halde bulunur.



1.sayı diye sor bekle komutu işlendiğinde üstteki ekran çıkar ve kullanıcıdan gerekli olan veriyi ister.



Girilen veriyi  değişkeninin 0 yerine cevap getirilerek kayıt işlemi tamamlanır.

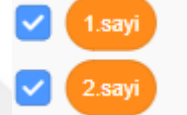
Aynı işlem 2. Sayı veya daha çok sayı içinde yapılır.

Ekrandaki sayıların her seferinde sıfırlanması için 1.sayıyı 0 yap değişken bloku konulur.

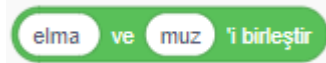


Girilen sayıların toplanması için operatörler sekmesinden  bloku kullanılır. Burada toplama yapmak istediğimiz operatörler birinci ve ikinci

sayıdır. Bunlar değişkenler sekmesi altında bulunmaktadır. Bu kısımda değişkenlerin sahnede gözükmesi isteniyorsa tik blokları seçili hale getirilir.



Yapılmış olan ifadenin ne anlam içerdiğini belirtmek için



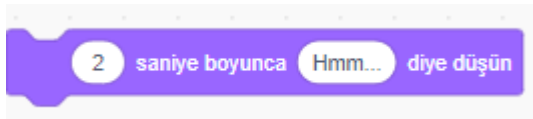
bloğu kullanılır.

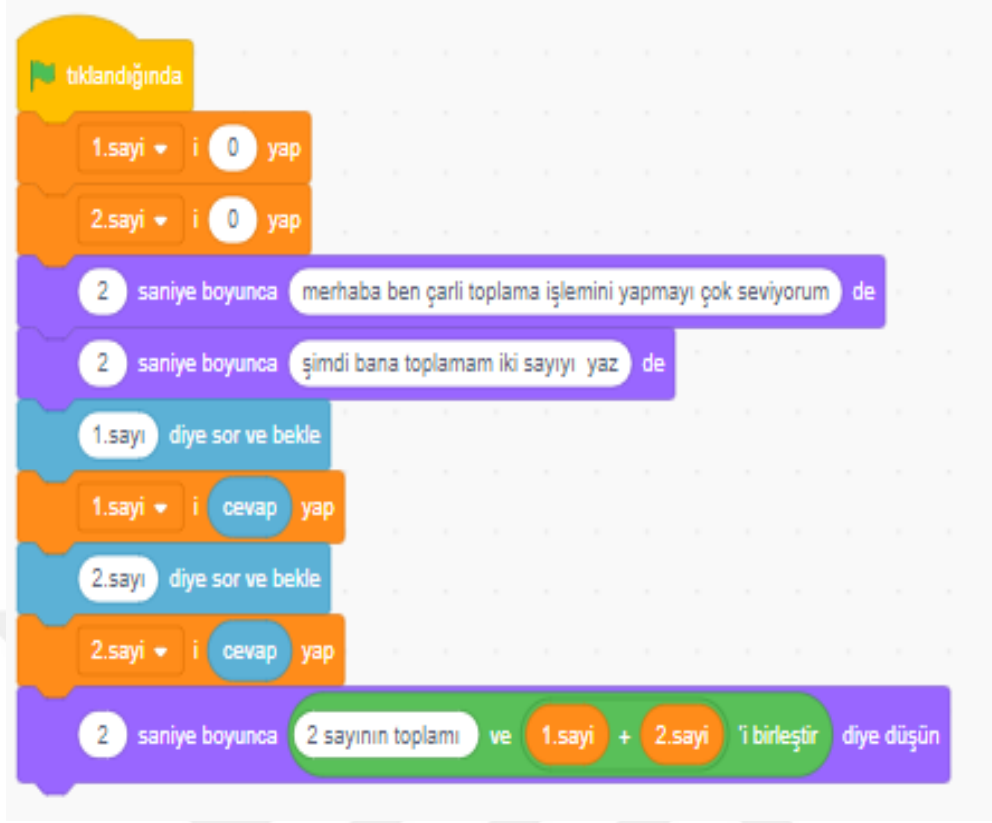


2 sayının toplamının yazılı olarak

gözükmesi için yandaki işlem yapılır.

Yaptığımız toplama işleminin ekranda gözükmesi için diye düşün bloğuna eklenir.





11- İki Sayı Arasında Çıkarma İşlemi

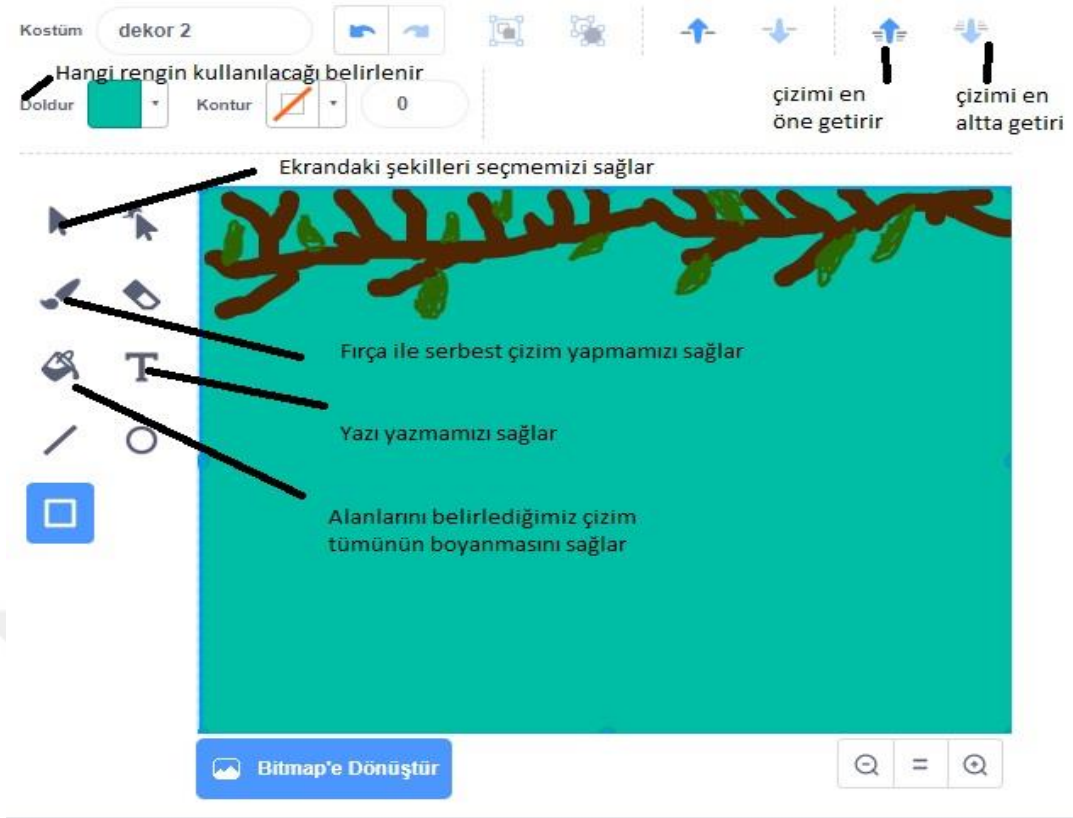
Bu kısımda öğrencilerden ekrana kullanıcı tarafından girilen iki sayının çıkarma işlemi yapması istenir.



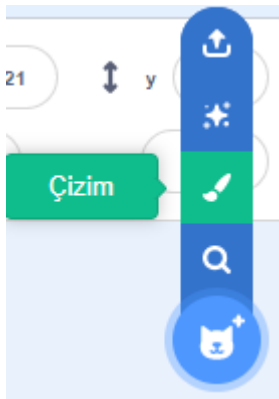
12- Kendi Oluşturduğumuz Dekorda Rastgele Konumda Aşağı Düşen Elmayı Yakalama Oyunu

Bu uygulamamızda yukarıdan rastgele konumda (x konumu devamlı olarak değişen) bir elma kuklası ile aşağıda bulunan bir adet çubuk sepet ile düşen elmaları yakalamaya çalışacaktır. Elma, çubuk sepete değdiği anda üst kısma çıkarak tekrar düşme işlemini sağlayacaktır. Alt kısımda bulunan kırmızı engele değdiği anda oyun bitecektir. Toplanan elma sayısı bir değişkende tutularak yazılacaktır.

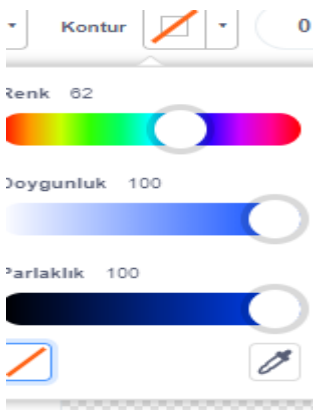
Bunun için ilk önce dekorlardan kendi çizimimizle bir dekor oluşturulur. Dekor sekmesi bu arada anlatılır.



Yukarıdaki gibi bir şekil oluşturulur ve dekor sekmesi tanıtılır.

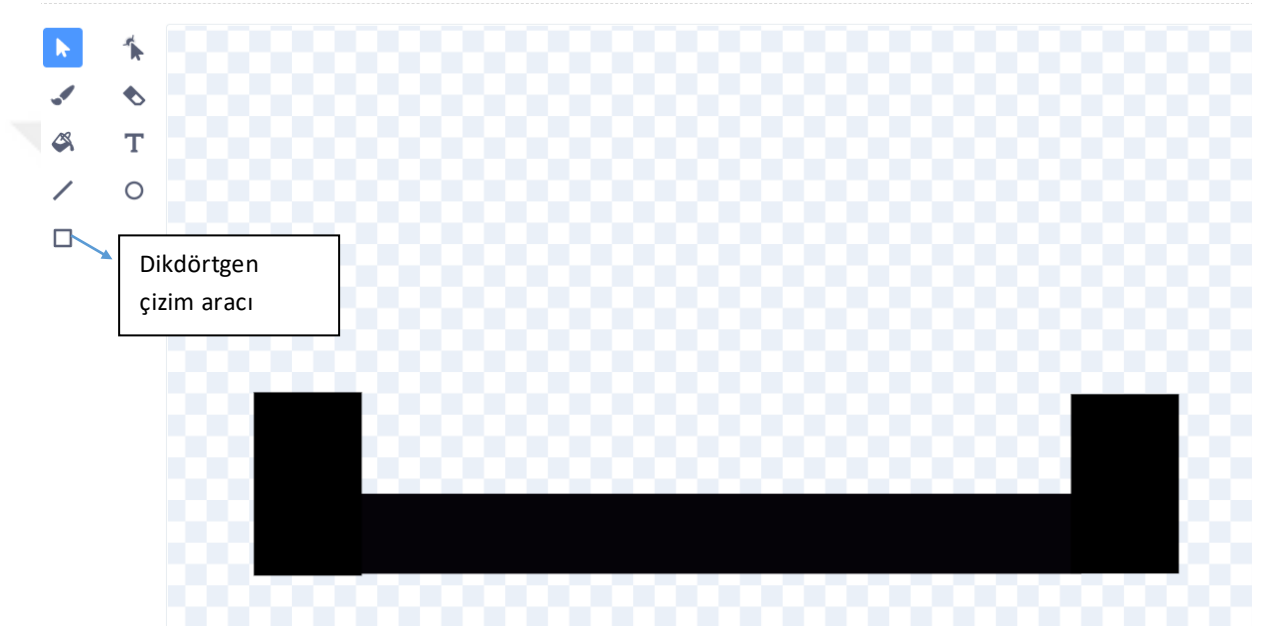


Bir kukla ekle seçeneği ardından çizim seçilir ve bir adet çubuk sepet çizilir. Dikdörtgen sekmesi kullanılarak çizim yapılır.



Burada dikkat edilmesi gereken çizilen dikdörtgenin konturu ile dolgu boyasının aynı renkte olmasıdır. Diğer

bir seçenek ise kenar çizgisinin yandaki şekildeki gibi gizlenmelidir.



Çizilen çubuk sepet isimlendirilir. Çubuk sepet kuklası seçilerek kod blokları yazılır.



Bu kısımda sepetin hangi konumda olacağı belli edilir.

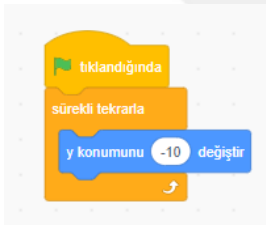
Bu kısımda öğretmen öğrencilere *sepetin y konumunda sabit kalıp x konumunda nasıl hareket ettirileceği* sorulur.

Sorunun cevapları beklendikten sonra yandaki kod bloğu yazılarak açıklama yapılır.

-Kukla seç menüsünden apple yazılarak elma kuklası seçilir. Apple kuklasının adı elma olarak değiştirilir. Kostümler sekmesinden elmanın bir kopyası çıkarılarak ikinci kostüme oyun bitti yazılır.

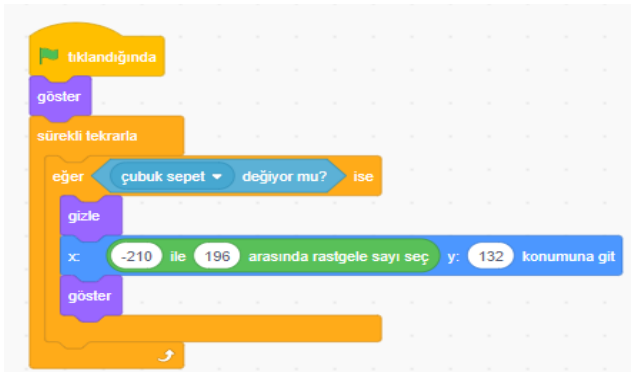


Elma kuklası seçilir ve kod blokları oluşturulur.



Elma kuklasının yukarıdan aşağı doğru hareket etmesi için yandaki kod bloğu yazılır.

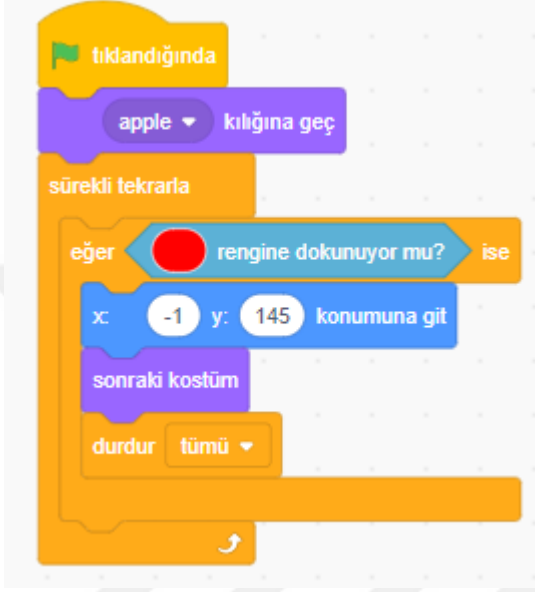
Bu kısımda öğretmen öğrencilerden *elmanın çubuk sepete değdiği anda tekrardan ağaç dalında bitip yeniden aşağı düşmüş gibi gözükmesini isteyen kod bloklarını kendilerinin yapmasını ister ve belirli bir süre tanır.*



Sahneye bir adet kırmızı zemin kuklası eklenir.

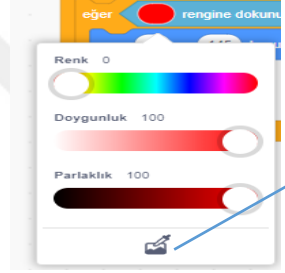


Elma kuklası tekrar seçilerek öğretmenin öğrencilere aşağıda yapılan kodların işlevlerini sorarak sırasıyla yerleştirir.



Bu kısımda elma oyununun oynandığı varsaydığımızda ekran da hala oyun bitti kuklasının durduğu bunun oyun başlangıcında değişmesi gerektiği için kılık değişikliği yapılır. Elmanın kırmızı zemine dokunup dokunmadığı kontrol ettirilir, değdiği anda dal figürüne yollanır ve oyun bitti kostümü getirilerek oyun durdurulur.

Rengi aynı zemin rengi olarak seçmek



için renk damlası kullanılır.

-Bu kısımda öğretmen *rengine dokunuyor mu* kodu yerine *başka hangi kodu kullanabileceğimizi* sorar.

Cevap:



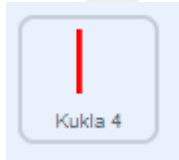
Çubuk sepet kuklası seçilerek toplanan elma sayısı yazdırılır.



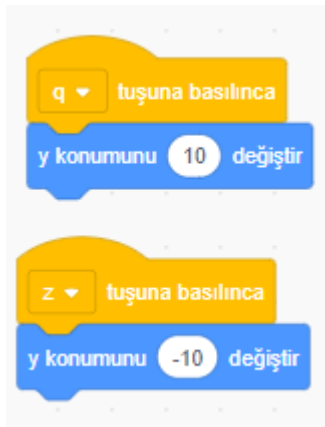
13-Gol Atma Oyunu

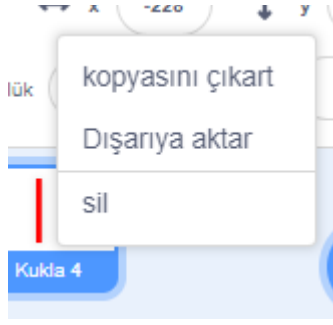
Bu uygulamada öğretmen öğrencilerden ekrana 2 adet çubuk kuklası çizmesi istenir. Bunların her ikisinin de farklı renklerde olması istenir. Bu oyunda çubuklar 2 kullanıcı tarafından aşağı yukarı hareket ederek topun çubuğa değdiği anda karşı tarafa gönderilmesi ve topun değdiği çubuğun renginin söylenmesi sağlanır. İki tarafa çizilen siyah çizgilere değdiği anda karşı kısımdaki oyuncunun puanı bir artırılır.

İlk olarak oyunda bir adet çubuk çizilir. Rengi kırmızı olarak ayarlanır.

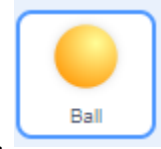
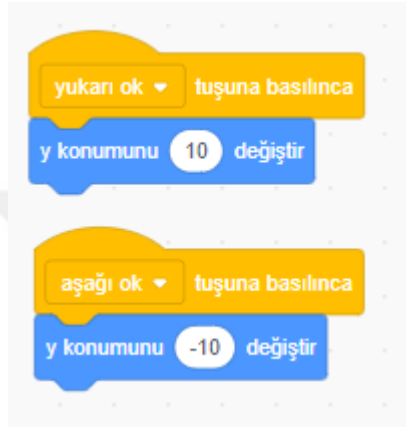


Ve kuklaya aşağı yukarı hareket edebilmesi için kodlar yazılır.





Kırmızı çubuğa gelerek kopyasını çıkartarak rengini yeşile çevirerek, kod bloğu bu sefer yukarı aşağı şeklinde değiştirilir.



Oyunun hareketli kuklası olarak top kuklası eklenir.

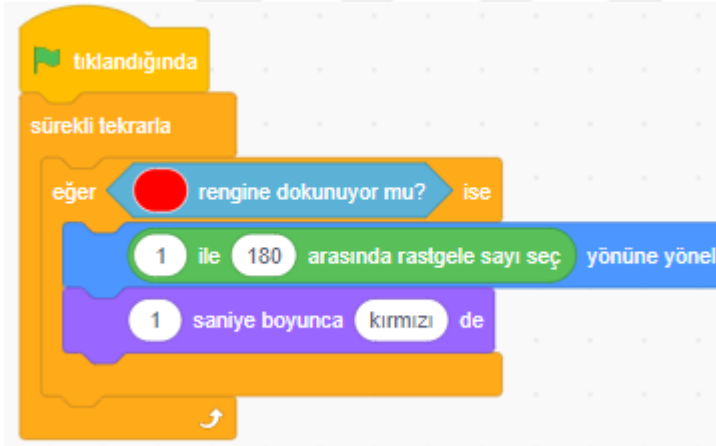
Kuklanın kod kısmına ekranda topun sağa sola rastgele hareket etmesini ve oyunun ilk başında topun hangi konumda bulunacağı ayaları yapılır.



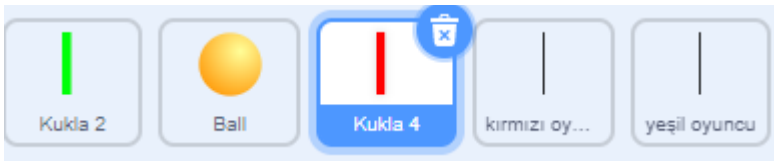
Ekranda hareket eden topun çubuğa değdiği anda yönü değiştirip tekrardan karşı tarafa göndermek için yönüne yönel komutu içinde rastgele sayı seç komutu ile topu karşı tarafa gönderir. Topun ters yönde gitmesi için -1 ile -180 arası sayılar seçilebilir. Topun yeşil topa değdiği anda 1 saniye boyunca yeşil yazdırırız.



Kodun aynısı kırmızı çubuğa da uygulanır. Burada dikkat edilmesi gereken rastgele sayı seç komutunda sayı olarak 1 ile 180 arasında sayı seçtirmemiz gerekiyor. Sayının sabit kalması durumunda topun puka girmesine neden oluyor.



Kukla ekle menüsünden çizim seçilir ve bir adet siyah ekranın yan kısmını tam olarak kaplayacak şekilde çizilir.



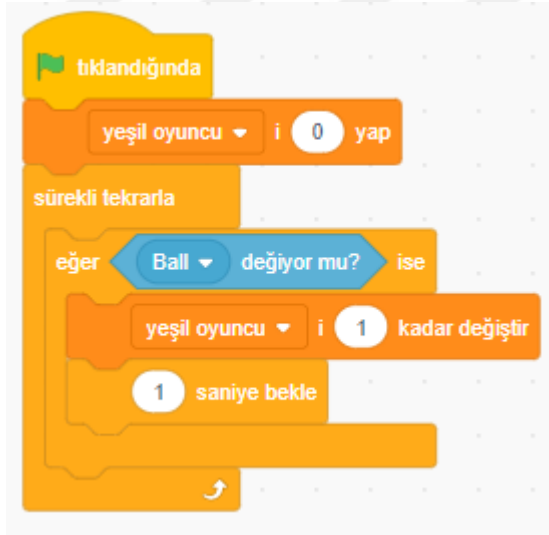
Bir kopyası çıkarılır. Bir tanesine kırmızı oyuncu diğerine yeşil oyuncu olarak adlandırılır.

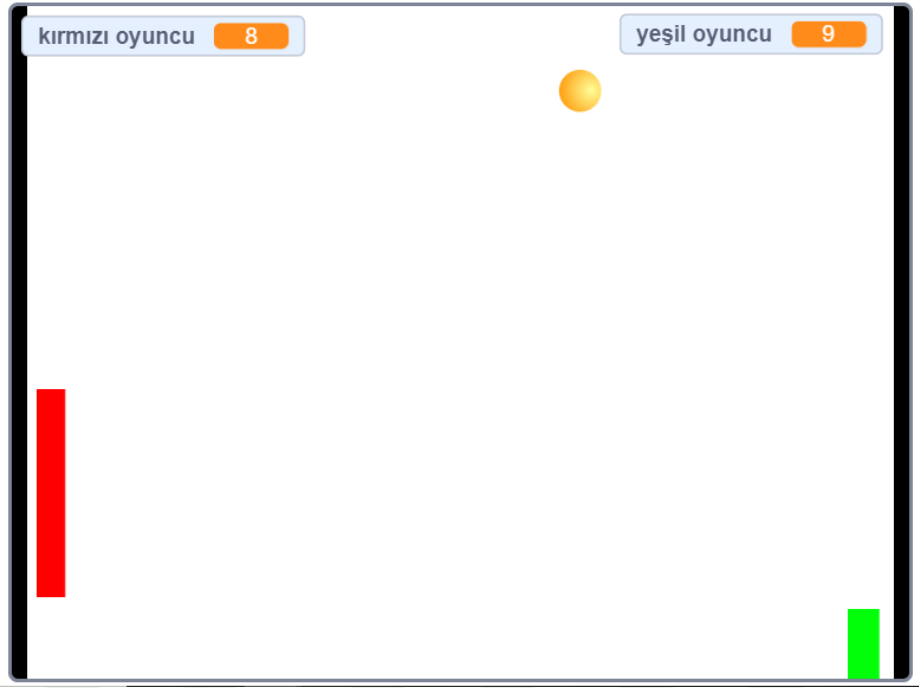
Aşağıdaki kodların nasıl olacağı öğretmen tarafından açıklandıktan sonra belli bir süre beklenir ve öğrencilerin yapması istenir. Yapan öğrenciler olduğunda tahtaya kaldırılır ve nasıl yaptıklarını sınıfa anlatmaları istenir.

Kırmızıoyuncu;



Yeşil oyuncu kuklasına;

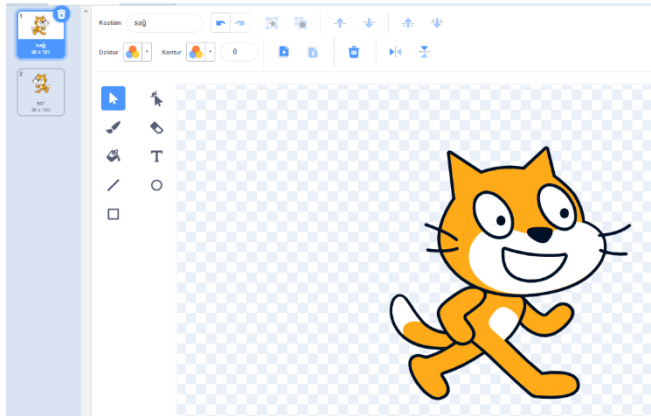




14-Kedi Fareyi Yakalama Oyunu

Bu uygulamada yön tuşları ile hareket eden kedi kuklası ile sahnede serbest biçimde dolanan fareyi yakalama oyunu yapılacaktır. Burada bir zamanlayıcı arka planda çalışıp kedi fareyi yakaladığında kaç saniye geçmişse o saniye yi söyleyip ekrana yazdıran bir oyun yapılacaktır.

Uygulamaya ilk olarak kedi kuklası eklenerek başlanır.



Kedi kuklası karakterine sağ, sol biçimde iki tane kostüm eklenir ve isimleri yazılır.

Kedi kuklası aşağı, yukarı, sağa, sola doğru hareket edebilmesi için gerekli kodlar yazılır.

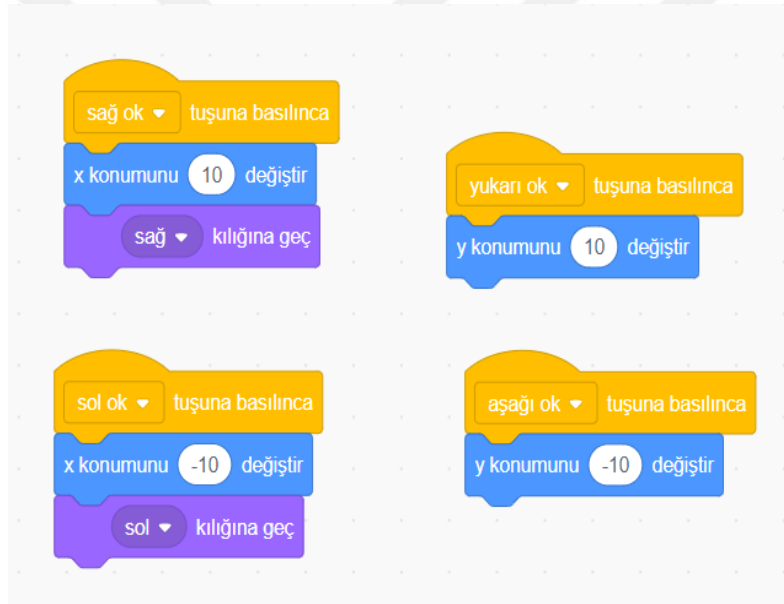
Öğrencilere aşağıdaki sorular sorulur.

1-Kedi kuklasını yukarı ok tuşuna basıldığını programa bildiren kod bloğu nedir?

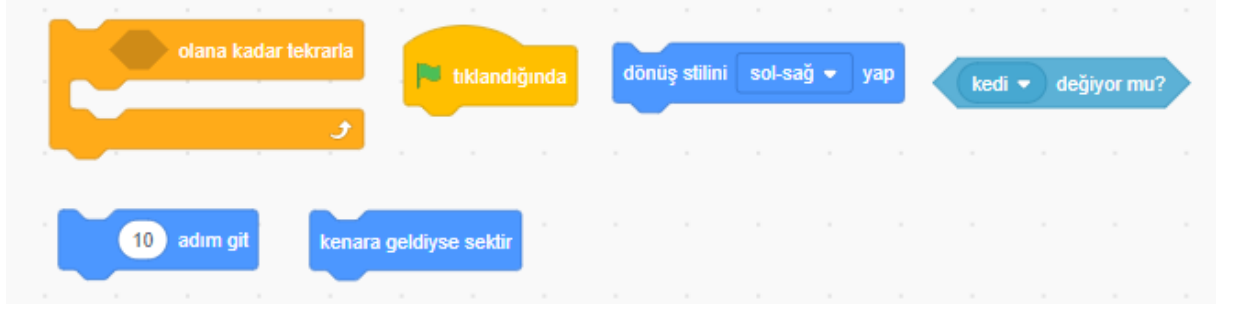
2-Kedi kuklasının yukarı ok tuşuna basıldığında yukarı yönlü hareket etmesi için hangi eksenle ilerlemesi gerekmektedir?

3-Yukarı aşağı sağa sola şeklinde hareket ettirmek için hangi sayıların eklenmesi gerekmektedir.?

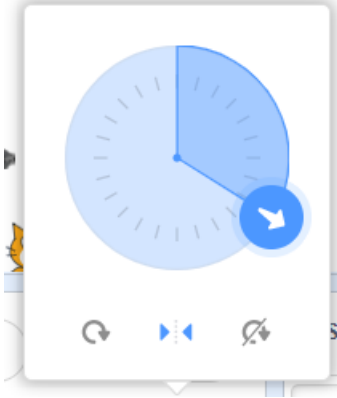
Sorular tartışıldıktan sonra aşağıdaki kod blokları yazılır.



Fare kuklasını seçip kedinin fareye değinceye kadar ekranda serbestçe dolaşması için gerekli kod bloklarını öğrencilerden düzenlemesi istenir. Farenin büyüklük sekmesi isteğe göre küçültülür.



Bu kod blokları yazıldıktan sonra öğrencilere farenin ekranda sabit olarak değil açı olarak da farklı yönlere gitmesi için hangi ayarlamaları yapılması gerektiği sorulur.



Yön ayarlamaları 122 derece olarak ayarlanır.

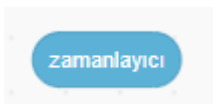
Kedinin fareye deđdiđi andaki kod blokları yazılır.



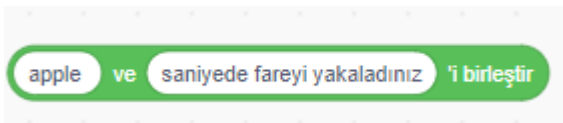
Daha sonra ekrana süre yazılabilmesi için “zamanlayıcı” kod blođu kullanılır. Zamanlayıcı kod blođu arka planda saniye türünden zamanı tutar. Zamanı tam sayı halinde yazdırmak için “yuvarla” komutu kullanılır. Ekrana yazdırmak için saniye ve yazıları birleřtirmek için “birleřtir” komutu kullanılır. Oyunun deđdiđi anda bitmesi için tümünü durdur komutu kullanılır.



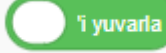
Bu kısımda öğretmen tarafından öğrencilere ařađıdaki komutların ne iře yaradıđı anlatılır.



Süre ölçer olarak kullanılır. Ekranda görünmesi istenirse yanındaki kutucuk iřaretli hale getirilir.



Bu operatör ile yazıları veya diđer kod bloklarını birleřtirme iřleminde kullanılır.



Verilen bir sayıyı en yakın tam sayıya dönüştürür.

Kedi kuklası seçilerek; kedinin farklı oyunun başlangıç aşamasında farklı yerlerde çıkması için aşağıdaki kod blokları yazılır.

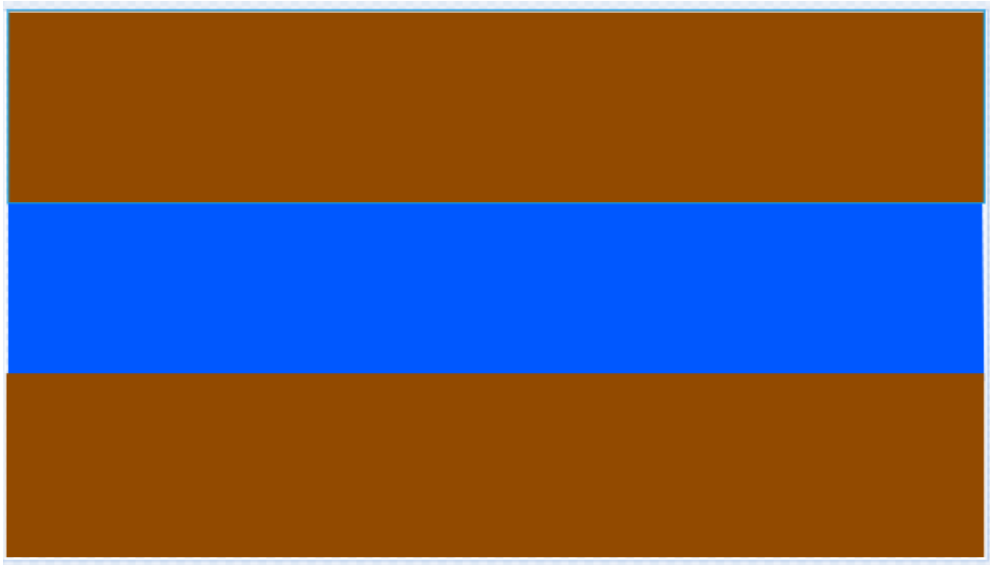


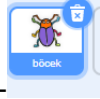
15-Nehir'i Geçme Uygulaması

Bu uygulamada çizilen nehir temalı dekordan bir adet böceğin karşı tarafa geçip çileği yakalaması istenmektedir. Böcek Nehir'e değdiği zaman başlangıç noktasına gidecektir. Nehir'in bir ucundan bir ucuna hareket halinde olan sal bulunmalıdır. Böcek bu salın üstünden karşıya geçip çileği yakalamalıdır.

Bu uygulamada öğretmen öğrencilerin uygulamayı kendileri yapmaları için belirli bir süre verir sonrasında adım adım anlatarak gösterip-yaptırır.

Uygulamaya dekor çizme işlemi ile başlanır. Aşağıdaki dekora benzer bir dekor çizilir.





- Ekranı bir adet böcek kuklası eklenir. Aşağıdaki kod blokları yazılır.



Bu kısımda böceğin yukarı aşağı sağa sola gidebilmesi için gerekli kodlar yazılır. Bu kısımda böceğin hızı isteğe göre ayarlanabilir.

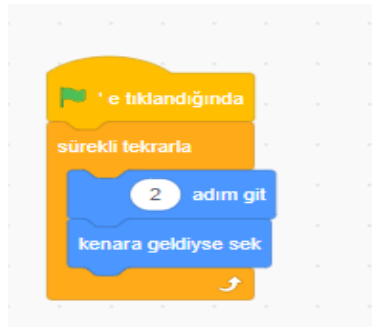
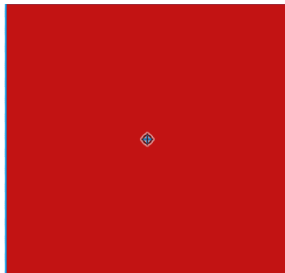


Bu kısımda böcek kuklasının başlangıç konumu ayarlanır, mavi renge (Nehir'e) değiştiği anda başlangıç noktasına gitmesi sağlanır.

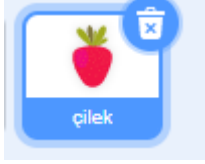


Uygulamada bir adet sal kuklası çizimi yapılır. Çizim yapılırken kare çizgisinin tam olarak ortalandığına emin olunmalıdır. Yoksa

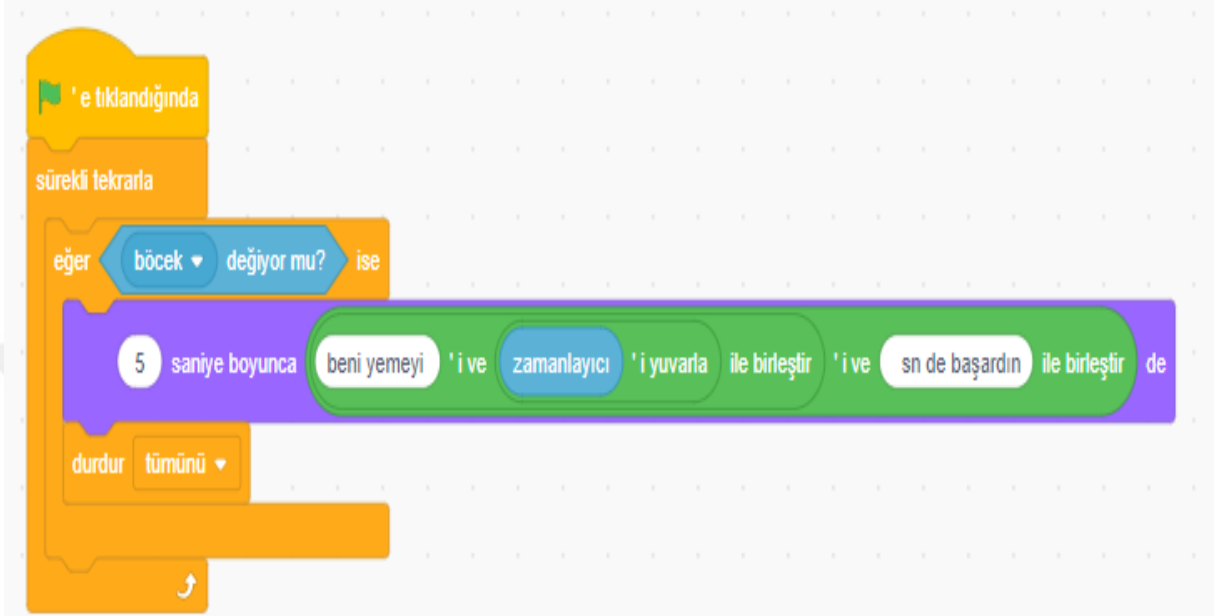
salın sağa sola doğru kaydığı görülecektir.



Salın nehirde sağa-sola doğru hareket etmesi için gerekli kod bloğu yazılır.



Uygulamaya çilek kuklası eklenir. Böceğin değdiği anda zamanı söylemesi için gerekli kod bloğu eklenir.



16-Balık Avlama Oyunu

Bu bölümde öğrencilerden balık yakalama oyunu istenecektir. Bu oyunu yaparken dikkat edilecek hususlar şunlardır.

-Ekranda birden fazla balık kuklası olacaktır. Bu balık kuklaları ekranda serbest bir biçimde sağa sola ve aşağı yukarı şekilde akvaryumda gezer gibi görüntü verilecektir.

-Bir adet köpek balığı kuklası eklenip bu kukla fare ile hareket etmesi sağlanacaktır. Bu kuklanın görevi balıkları yemek tir. Balıklara değdiği anda balıklar kaybolacaktır. Fakat belirli bir süre sonra balık tekrardan ekrana görünür halde çıkacaktır.

-Kukla ekle menüsünde denizde bulunan canlılardan herhangi biri eklenir ve bu canlılar oyunda yenmemesi gereken kukla olarak kullanılır. Fare ile hareket ettirdiğimiz kukla bu kuklalara değdiği zaman puan menüsünden puan

kaybedecek. Kaybedilecek puanı öğrencinin seçimine bırakılmıştır. Bu kuklalarda yenildiği anda ekranda kaybolacak ve belirli bir süre sonra tekrardan ekranda belirecektir.

-Oyunda puan sistemi uygulanacaktır. Bu puan sisteminde her yenilen balık için puan verilecek, her yenilen deniz canlısı için ise puan düşürülecektir.

-Oyuna bir adet süre eklenecektir. Oyunun hangi saniyede biteceği öğrenciler tarafından belirlenecektir. Bu süre oyun içinde eksi puan getiren kuklaların ortaya çıkması içinde kullanılacaktır. Bu kuklalar ekrana çıkacak ve bir daha kaybolmayacaktır.

-Örneğin 10.saniyede 1 kukla 20.saniyede başka bir kukla çıkararak oyunu zorlaştıracaktır.

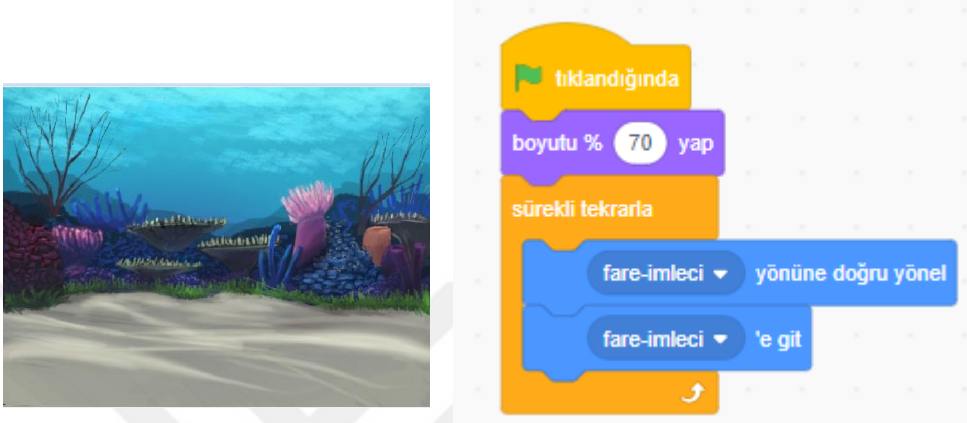
-Oyunun sonunda herhangi bir ekrana eklenmiş kukla belirli bir puana ulaşıldığında veya belirli bir süre aşıldığında kaybettiniz veya kazandınız demesi istenmektedir.



Bu uygulamamızda öğretmen öğrencilerden ilk önce kendisini takip etmelerini sonrasında ise uygulamayı kendilerinin yapmasını istemektedir. Öğretmen

uygulamayı yaptıktan sonra bütün öğrencilere belirli bir zaman tanır. Yapılan yanlışları veya yapılamayan yönler tüm sınıf ile soru- cevap şeklinde cevaplanarak işlenir.

Oyunumuza ilk olarak köpek balığı kuklası seçilerek başlanır. Bu kısımda köpek balığı kuklasının fareyi takip etmesi ve boyutu ile ilgili kodlar yazılır.



Dekor kısmı seçilir ve su altı dekoru seçimi yapılır.

Dekor kısmına gidilerek kodlama menüsü seçilir.

Aşağıdaki kodlar yazılır.

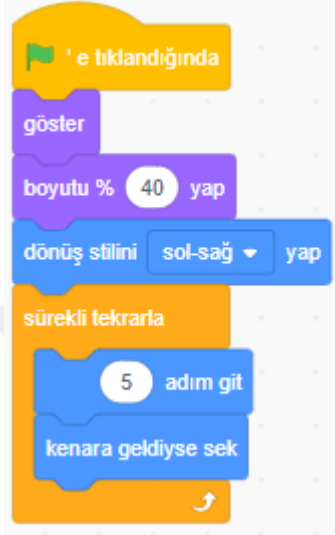
Zaman menüsü eklenmesi için bir adet zaman değişkeni oluşturulup kodları ekrandaki gibi yazılır. Burada zamanın 1 saniye olarak değişmesi sağlanmıştır.





Ekranda yenildiğinde puan kazandıracak balıkların eklenmesi

Yandaki şekildeki gibi bir balık kuklası seçilir isim olarak herhangi bir değişiklik yapılmaz. Balık kuklası için aşağıdaki kod blokları eklenir.



Bu kısımda balığın boyutu ayarlanır. Ekranda serbest bir biçimde dolanması sağlanır.

Balıkların ekranda sabit olarak gitmemesi için yön sekmesinden derece 128 olarak ayarlanır. Kendi belirlediğiniz bir sayıda yapılabilir.

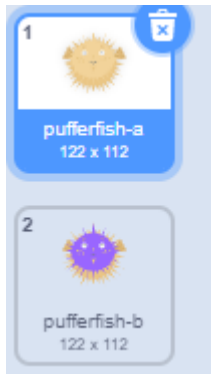
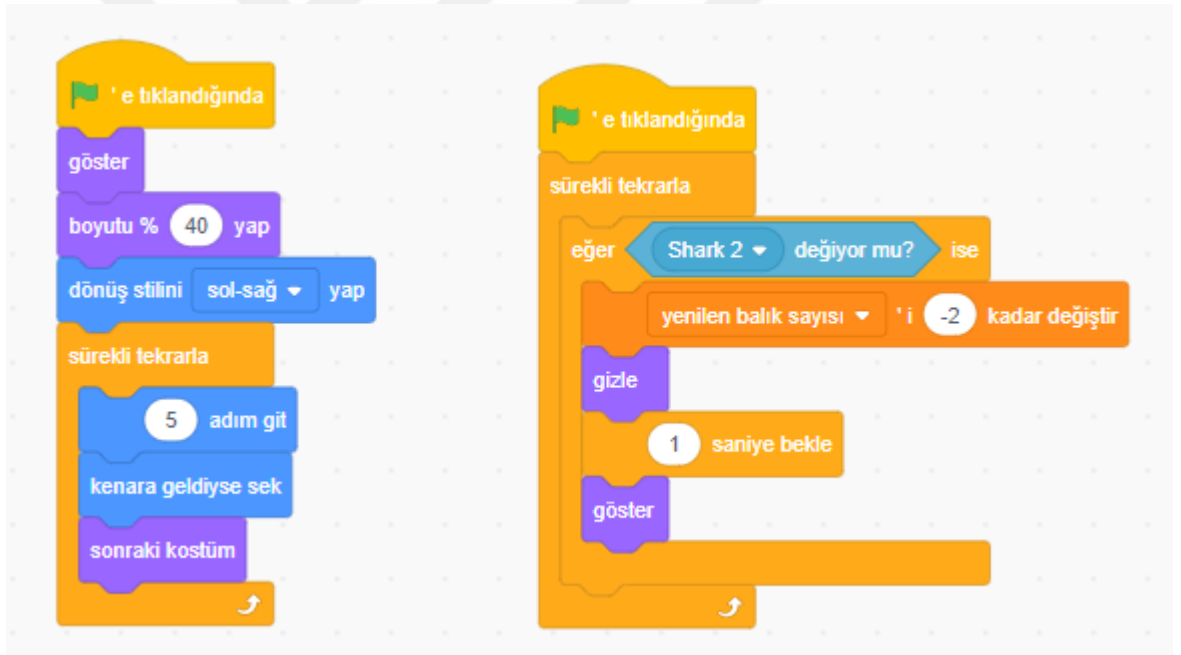


Bu kısımda köpek balığının balığa değdiği anda kaybolması ve 3 saniye sonra tekrardan ortaya çıkmasını sağlayan kod bloğu yazılır. Yenilen balık sayısı değişkeni ise 1 kadar artırılır.

Yukarıdaki işlemler için en az altı tane balık kuklası eklenir ve aynı kodlar bu kuklalara da eklenir.

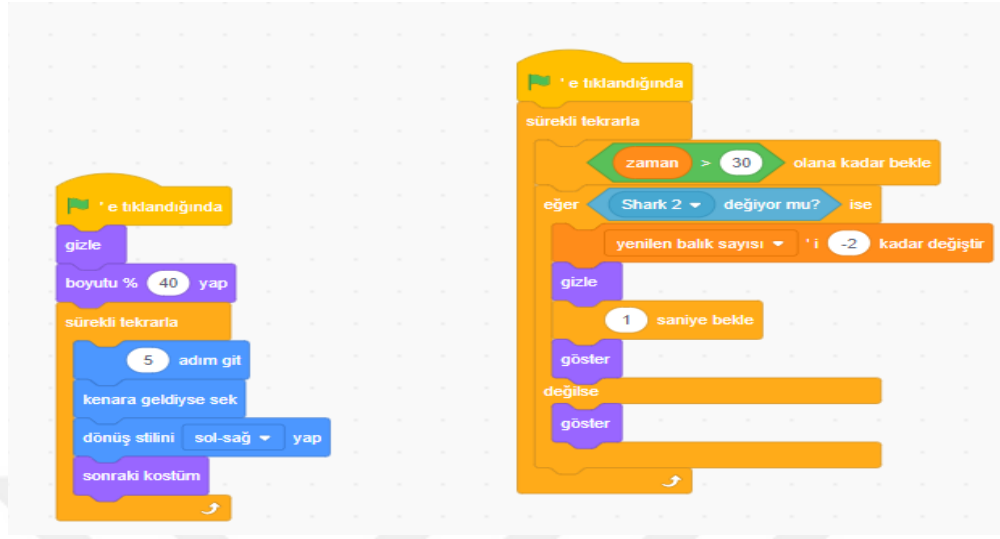


Ekrana yandaki pufferfish kuklası eklenir. Bu kukla köpek balığının yememesi gereken kukla olarak eklenir. Aşağıdaki kodlar eklenir. Bu kuklada sayı olarak en az üç tane ayarlanır.



Bu kısımda belirlediğimiz bazı pufferfish kuklaların belirli bir zaman sonra çıkması sağlanacaktır. Bunun için pufferfish

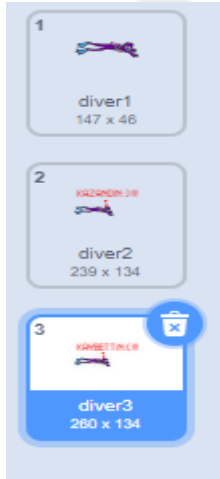
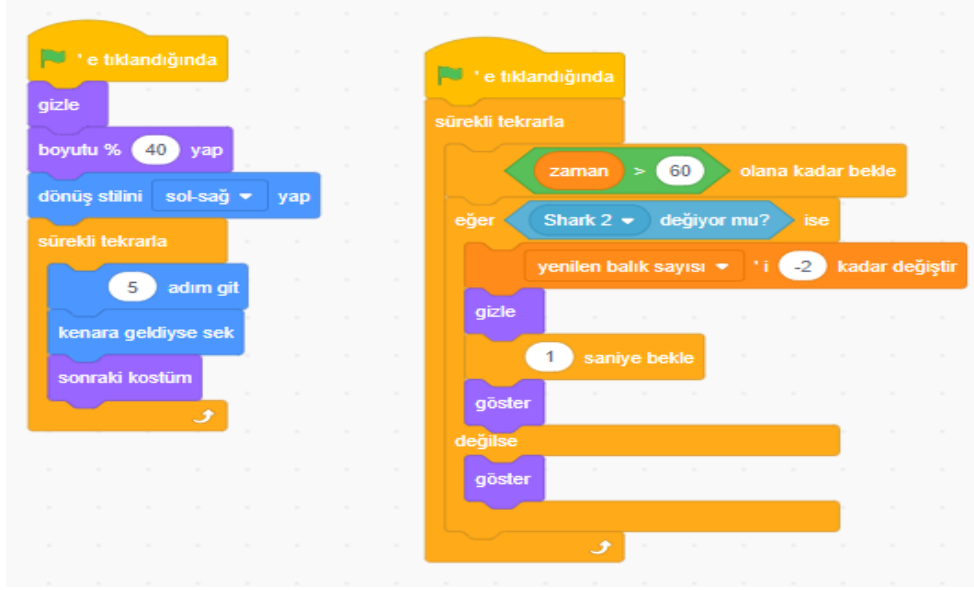
kuklasını mor renge boyanır. Bu kukladan 2 tane farklı zamanda çıkması için ekleme yapılır.



Yukarıdaki kod kısmında kuklanın sağa sola serbest bir biçimde dolaşması için ayar yapılır. Saniyenin 30 dan büyük olmasına kadar beklenir. Saniye 30 olduktan sonra kuklanın çıkması sağlanır. Köpek balığı kuklasının bu kuklayı yemesi durumunda gizlenip gözükmesi için gerekli kod blokları oluşturulur.



Yandaki şekilde yeniden kukla oluşturulabilir. Kullanıcın kendi kararına göre ekleme yapılabilir. Aşağıdaki kodlar eklenir.



Yandaki kukla seçilir ve kopyalanır.2 kısma kazandın yazılır.3.kısma kaybettin yazılır.

-Bu kuklamızda yenilen balık sayısı sıfırlanır.

-Zaman 100 olduğunda kaybettin kuklasına geçilir ve oyun durdurulur.

-Yenilen balık sayısı 60 olduğunda kazandın yazan kuklaya geçer ve oyunu durdurur.

-Yenilen balık sayısı sıfırın altına düştüğünde oyunu kaybettin kuklasına geçer ve oyunu durdurur.



Ödev: Oyuna başka kuklalar ekleyerek geliştiriniz.

17-Kalem ile Kare, Üçgen Ve Renkli Kesikli Çizgi Çizme

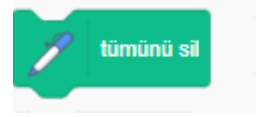
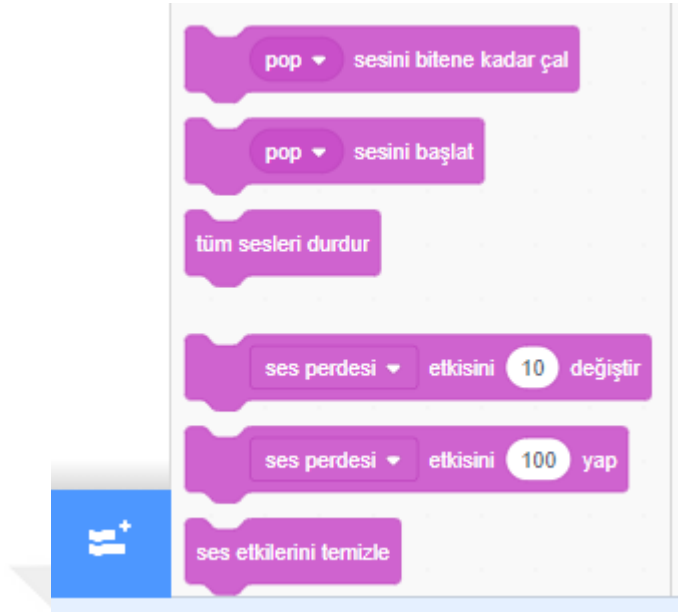
Rehber öğretmen dersin başında öğrencilere geometrik şekillerin neler olduğunu sorar.

Hangi geometrik şekilleri bildiklerini ve bunların nasıl çizildiklerini sorar.

Bu maddeler tartışıldıktan sonra geometrik şekillerin scratch ile çizilebileceği söylenir ve bunun kalem eklentisi ile yapılabileceği öğrencilere anlatılır.

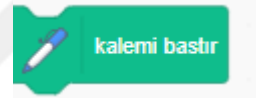
Öğretmen öğrencilere bu derste scratch ile ekrana kare, üçgen ve kesikli çizmeyi öğreneceğiz

Kalem eklentisini eklemek için eklentiler kısmından kalem seçilir. Öğretmen öğrencilere bunun nereden seçileceğini gösterir.



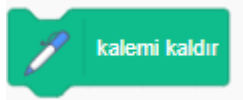
Buradaki

tümünü sil komutu ile sahnede daha önceden çizilmiş şekillerin tümü silinir.

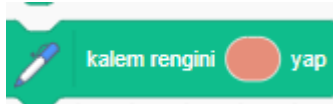


Kalemi

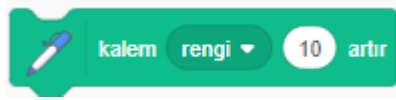
bastır komutu ile program ilk açıldığında kalem kuklası elimizde havada duruyor gibi düşünebiliriz. Kalem kuklasının çizebilmesi için kalemi bastır komutu kullanılması gerekir.



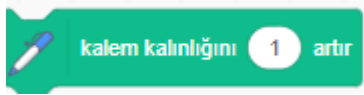
Kalemi kaldır komutu ile bazı durumlarda kalemin çizmeyi bırakması gerekebilir. Bu gibi durumlarda bu kod bloğu kullanılır.



Kalem rengini ayarlamak için kullanılır.



Kalem rengini belirtilen sayıda artırarak farklı renklerin oluşması sağlanır.



Kalem kalınlığını belirtilen sayıda artırır.

Yukarıdaki örnekler rehber öğretmen tarafından söylendikten sonra öğrencilerin serbest olarak bir şeyler çizmesi için belirli bir süre tanınır.

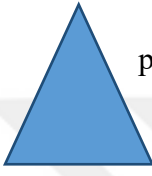
Öğrencilere rehber öğretmen tarafından üçgenin, karenin ve kesikli çizginin ne olduğu sorulur.

Bunların cevabı sınıfta tartışıldıktan sonra rehber öğretmen tarafından açıklama



yapılır.

Kare, bütün kenarları ve açıları birbirine eşit olan düzgün dörtgendir. Karenin bir iç açısı 90 derecedir.

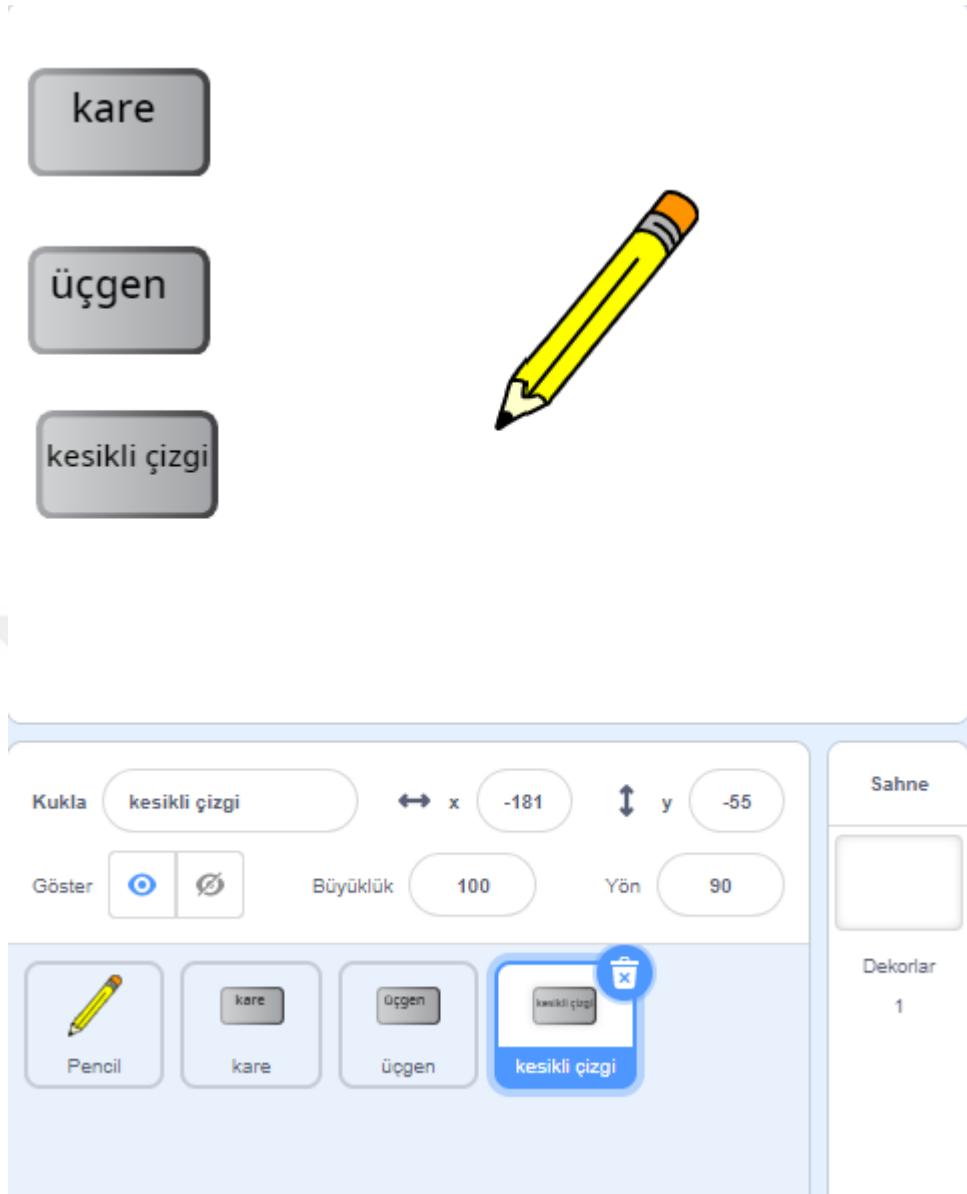


Aynı doğru üzerinde olmayan üç noktayı birleştiren doğru parçalarından meydana gelen geometrik şekle üçgen denir. Üçgenlerin iç açıları toplamı 180 derecedir. Bu uygulamada çizeceğimiz üçgen eşkenar üçgendir. Eşkenar üçgen bütün kenarları ve açıları eşit olan üçgene eşit kenar üçgen denir. Eşkenar üçgenin bir iç açısı $180/3=60$ derecedir. Eşkenar üçgenin bir dış açısı ise $180-60=120$ derecedir.



Kesikli çizgi aralarında birer boşluk olan çizgiler bütünüdür.

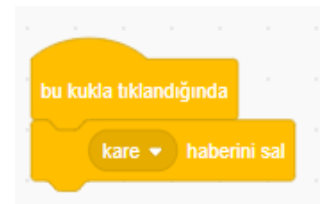
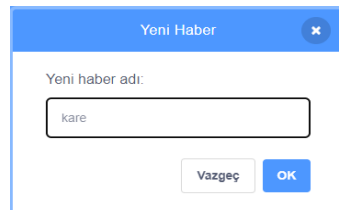
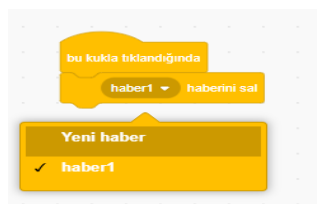
Öğretmen öğrencilere daha önceki konuları hatırlamalarını ister ve butonu nereden ekleyebileceğimizi sorar.



Buton ekleme işlemleri yapılır ve isimlendirilir.

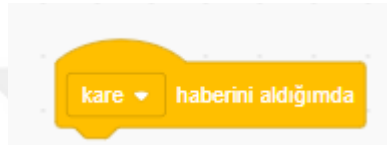
Öğretmen öğrencilere bu kısımda yapılacak işlem her bir butona basıldığında üzerindeki işlemi gerçekleştirmektir. Butonun basılıp basılmadığını kontrol etmek için “bu butona basıldığında” kod bloğu kullanılır. Basılan butonun hangisi olduğunu kalem kuklasına bildirmek için “haber sal” komutu kullanılır açıklamasını yapar.

-Kare butonu içine aşağıdaki haber salma işlemini yapar.



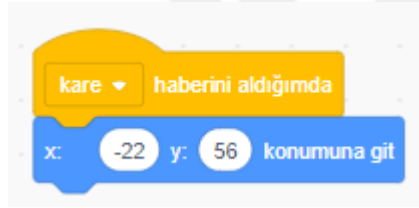
-Rehber öğretmen öğrencilerden aynı işlemi üçgen ve kesikli çizgiler içinde yapmasını ister ve belirli bir süre bekler.

Kalem kuklası seçilir. Öğretmen tarafından kare çizimi anlatılır.

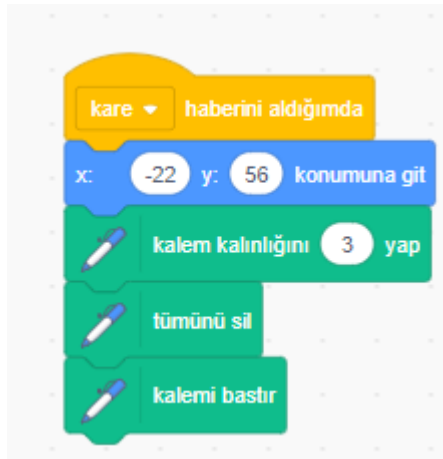


İlk kod olarak kare haberini aldığımda kodu eklenir.

Kare butonuna kullanıcı bastıktan sonra bu kod bloğunun çalışacağı vurgulanır.



Karenin ilk olarak hangi koordinatlarda başlayacağı belirtilir.



Öğretmen dersin başında anlatmış olduğu kısımlardan öğrencilere aşağıdaki soruları sorar.

Kalem kalınlığını hangi kod bloğu ile artırılabilir?

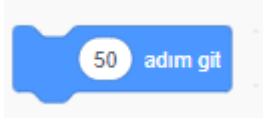
Ekrandaki daha önceden yapılmış çizimleri silmek için hangi kod bloğu kullanılır?

Kalem şekil çizme emri verildiğinde hemen ekrana çizim çizer mi? Bunun için hangi işlemi yapmamız gerekir?

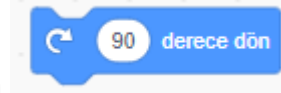
Soruları sorularak yandaki kod bloğu oluşturulur.

Öğrencilere daha önce öğrenmiş olduğunuz döngü konusunu hatırlayalım denir ve sınıfa döngü işleminin ne olduğu sorusu yöneltilir.

-Bu kısımda ilk başta ekrana çizgi çizmek için kalem komutunun



komutu ile ekrana çizgi çizileceği söylenir.

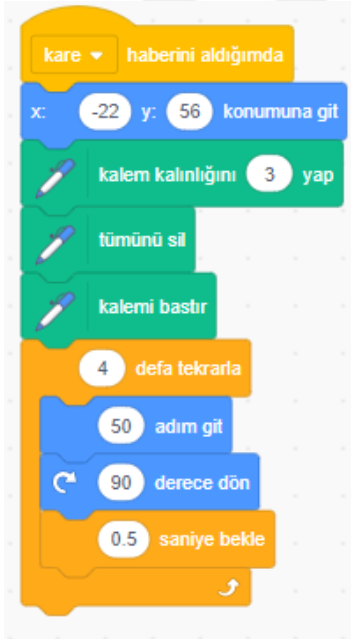


-Karenin dış açısı 90 derece olduğu için komutu kullanılır.

-Karenin dört kenarı bulunduğu için bu işlemlerin dört defa tekrar edeceği söylenir.

İşlemin görsellik olarak daha iyi gözükmesi için 0,5 saniye bekle komutu eklenir.

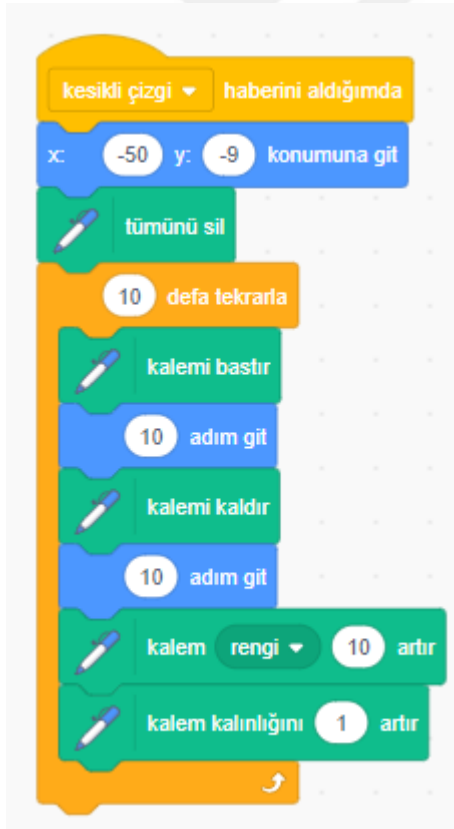
Açıklamaları yapılarak kod bloğu oluşturulur.



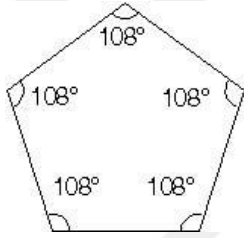
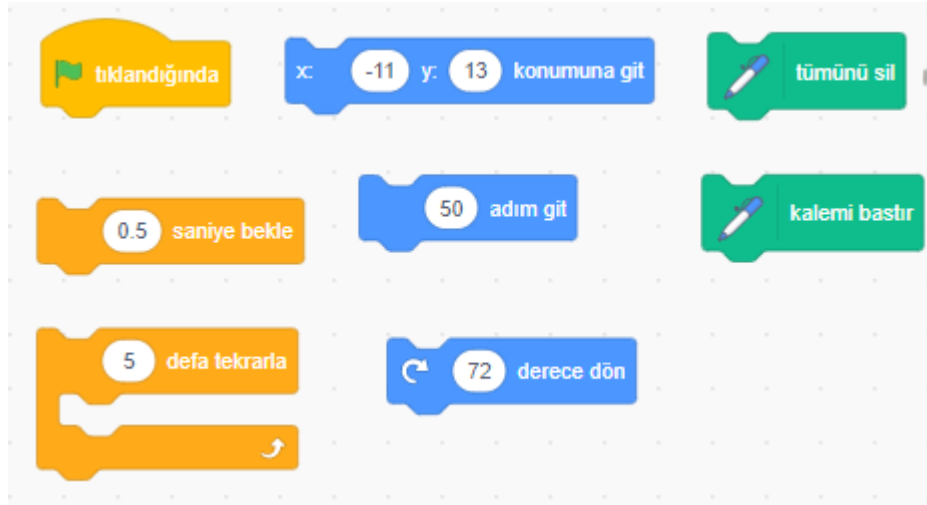
Üçgen çizimi için öğretmen öğrencilere belirli bir süre verir ve üçgen çiziminin de öğrenciler tarafından oluşturmasını bekler. Belirli bir süre sonra aşındaki kod bloğunu oluşturur.



Öğretmen öğrencilere kesikli çizginin ne olduğunu hatırlamalarını ister ve bunun nasıl yapılacağını kendilerinin denemelerini ister. Aşağıdaki kod bloğunu oluşturur.



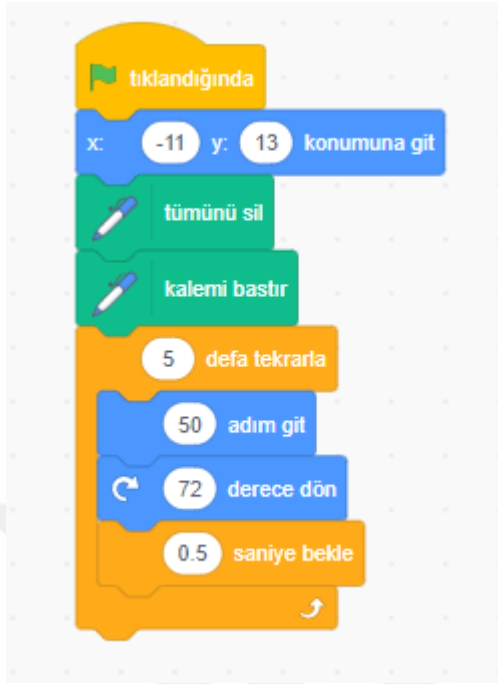
18-Kalem ile Düzgün Beşgen Çizme



Öğretmen öğrencilere beşgenin beş ayrı doğru parçasından oluşan kapalı geometrik şekil olduğunu söyler. Ve beşgenin bir iç açısının 108 derece olduğunu söyler.

Öğretmen tahtadan düzgün beşgen modeli açar ve yukarıda karışık olarak verilen kod bloklarını öğrencilere verir.

Öğrencilerden doğru kod sıralamasını oluşturmalarını ister.



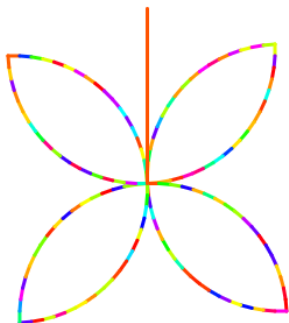
19-Kalem ile Kelebek Çizme



söylenir.

Öğretmen tahtadan herhangi bir kelebek resmi açar ve öğrencilere bu kelebek resmini scratch ortamında nasıl çizilebileceği sorulur.

Bunun birkaç biçimde yapılabileceği belirtilir. Kukla çiz, sahne çiz ekranları ile yapılabileceği burada yapılmak istenenin animasyonlu bir biçimde kelebek motifi çizmek istendiği

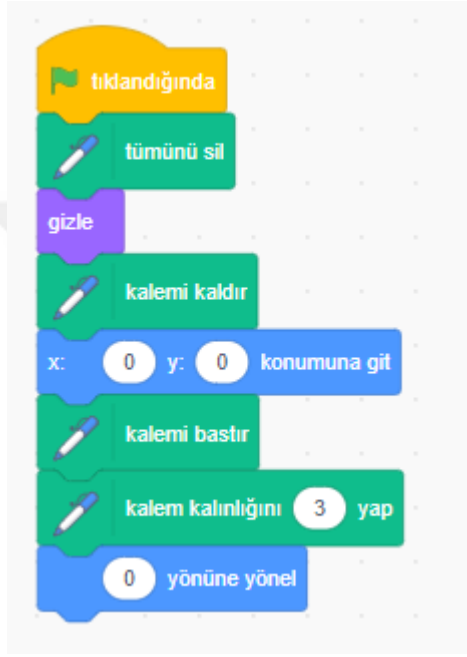


Öğrencilere yandaki kelebek motifinin çizilmesi istendiği söylenir. Bunun nasıl çizileceği hangi kod bloklarının kullanılması

gerektiği sınıfta tartışılır. Öğrencilere belirli bir süre tanınır ve kelebeğin sadece bir kanadını çizmeleri istenir.

Çizen öğrencilerden bunu nasıl yaptığı açıklanması istenir ve sınıfla beraber çizilir.

Öğretmen tarafından da eksik kalan kısımlarda aşağıdaki açıklamalardan faydalanarak anlatım yapılabilir.



Yeşil bayrağa tıklandığı anda ilk önce daha önceden kalan çizimler tümünü sil kod bloğu ile silinir.

-Bu kısımlarda öğretmen öğrenciler ile soru cevap yapılabilir.

Örnek olarak;

Ekrandaki kuklayı gizlemek istiyorum bunun için hangi kod bloğunu kullanmam gerekir?

Gizle seçeneği ile sahnedeki kukla gizlenir. Kalem kaldır butonu ile konuma gitme

aşamalarında kalemin basılı olmaması sağlanır.

Kalemin konumu 0,0 koordinatlarına ayarlanır.

Kalem basılı hale getirilir.

Kalem kalınlığı belirtilir.

Yön başlangıcı olarak ta sıfır ayarlanır.



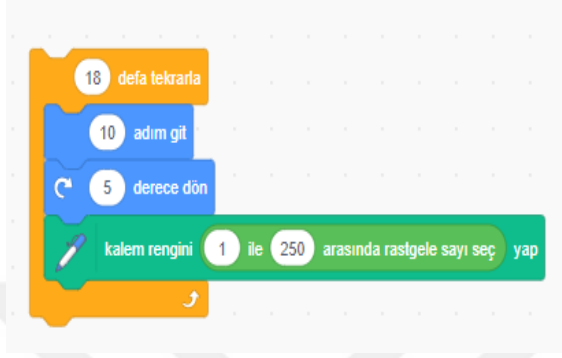
Bu kısımda öğretmen öğrencilere ilk önce kelebek kanatların çiziminde adım adım gitmemiz gerektiğini belirtir ve kelebek kanadının yandaki görünen kısmını gösterir ve bu kısmının yapılabilmesi için hangi kod bloklarını kullanacağı sorulur.

Örnek sorular;

-Kaç çizgiden oluşacak

-Çizgilerin uzunluklarını hangi kod bloğu ile yapabiliriz

-Kalem renginin çeşitli olması için hangi kod bloğu kullanılır



Yandaki kod bloğu oluşturularak kelebek kanadının bir kısmı

oluşturulur.



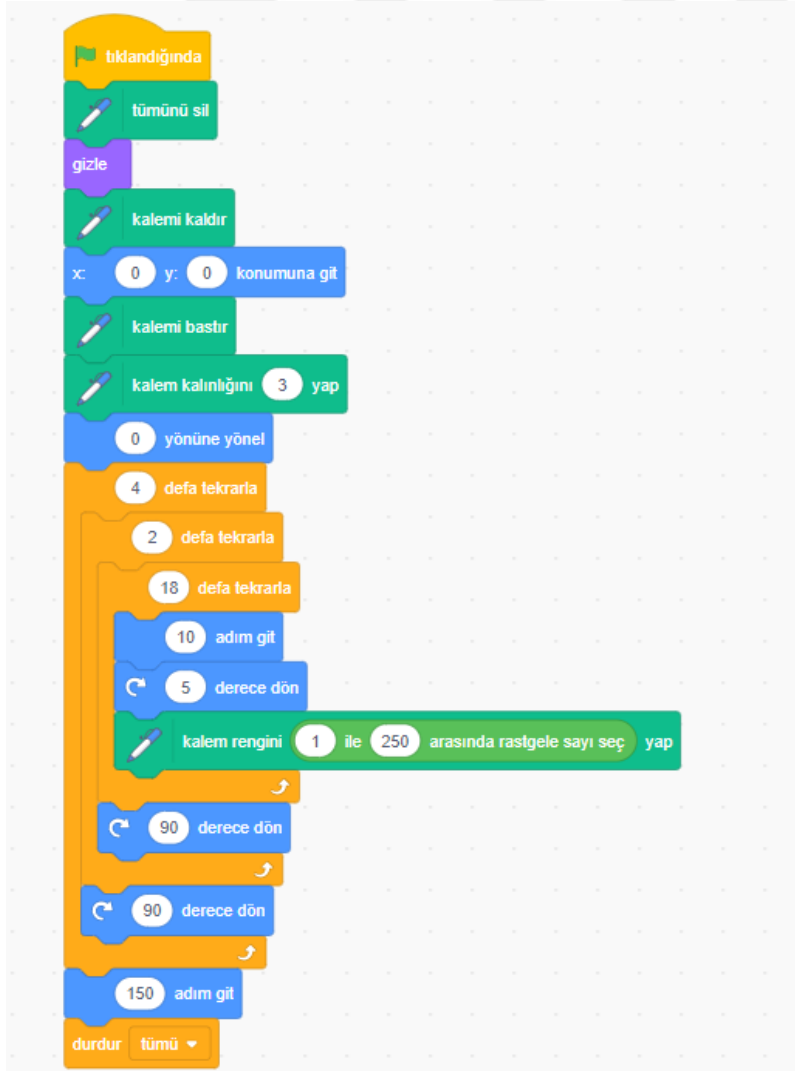
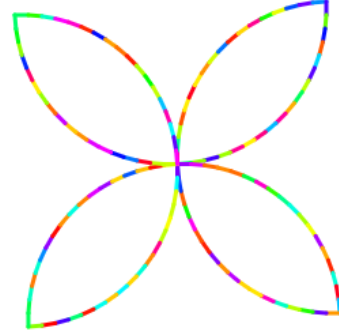
Kanadın diğer kısmının yapılabilmesi için 90 derece dönülmesi ve aynı işlemin tekrar yapılması gerektiği söylenir.

-Bu kısımda öğrencilere tekrar işlemi olduğunda hangi yöntemle başvurduğumuz sorulur

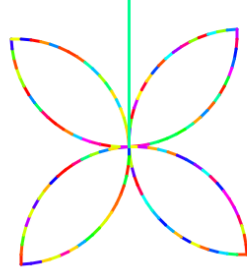


Bu kısım da olan kod blokları yazılır ve ekranda tek kanat çizimi yapılır.

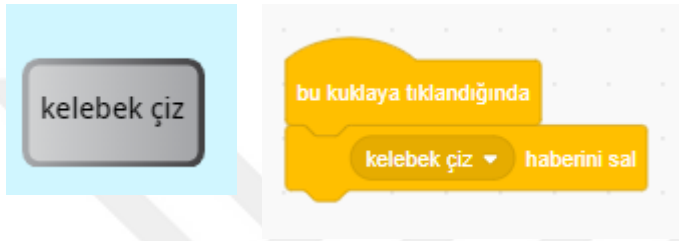
Kelebeğin dört kanadı olacağından dolayı bu işlemi dört defa yapacağımız ve bu işlem içinde iç içe döngülerin kullanacağı söylenir. VE program yeniden çalıştırılır.



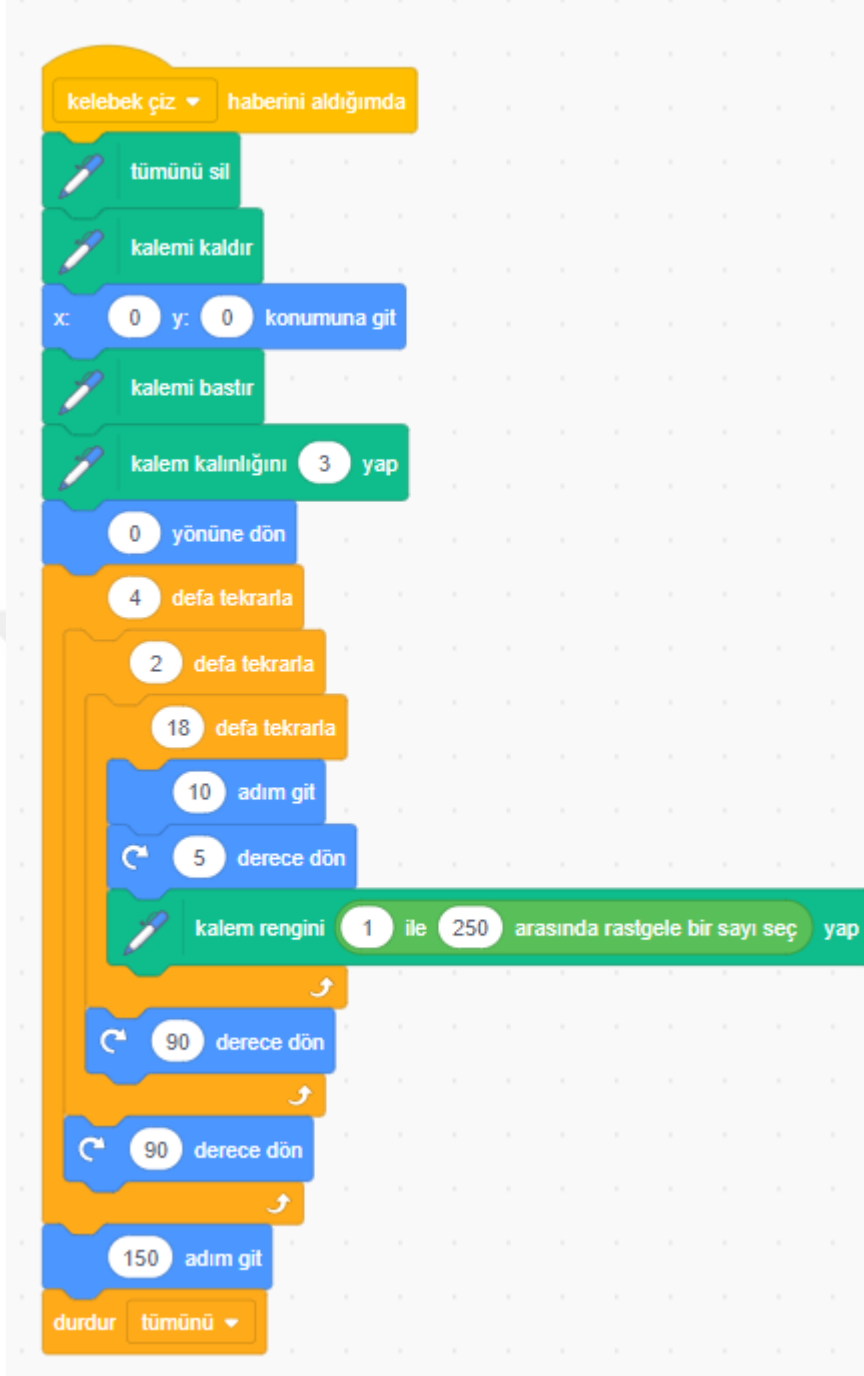
Kelebeğin gövdesi çizilmesi içinde 150 adım git komutu kullanılır. Durdur tümü ile çizim bitirilir.

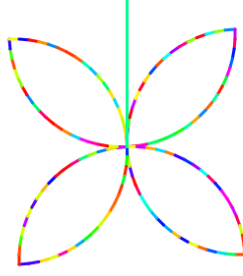
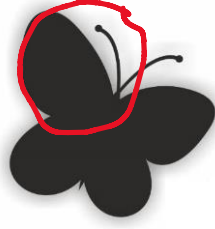


Ekrana bir adet yandaki şekilde buton eklenir. Butona aşağıdaki kod bloğu eklenir.



Yeşil bayrağa tıklandığı kod bloğu yerine kelebek çiz kod bloğu eklenir.





ÖDEV:

Yapmış olduğumuz

kelebeğin gövde kısmına yandaki şeklideki gibi antenlerini sizler ekleyiniz.

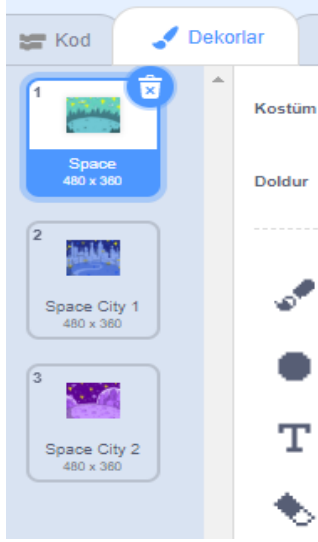
Sizlerde kendi belilediğiniz bir bitkinin veya hayvanın şeklini oluşturunuz.

20- Zıplayarak Gezegen Değiştiren Ördek

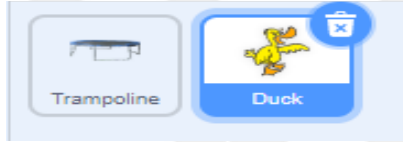
Bu uygulamada ekranda bir adet trambolin bulunacak ve ördek kuklası bu tramboline yön tuşları ile hareket ederken değdiği anda yukarı zıplayacak ve sahnenin de değişimi sağlanacaktır. Ördek zıpladığı zaman başka bir yere mekâna geçiyormuş gibi gözükmesi istenmektedir.



İlk olarak dekorlar kısmından uzay ile ilgili scratch programında bulunan 3 tane dekor seçilir ve eklenir.



Ekrana bir adet trambolin kuklası ve bir adet ördek kuklası eklenir.



Ördek kuklası seçilir ve öğrencilerin önceki örneklerden öğrendikleri ile yön tuşları ile hareket ettirmesi istenir.



Öğrencilere aşağıdaki soru sorulur ve sınıf ortamında fikir yürütülür.

1-Ördeğin tramboline değdiği anı belirlemek için hangi kod blokları kullanılır?

Daha sonra aşağıdaki açıklama öğrencilere öğretmen tarafından yapılır.

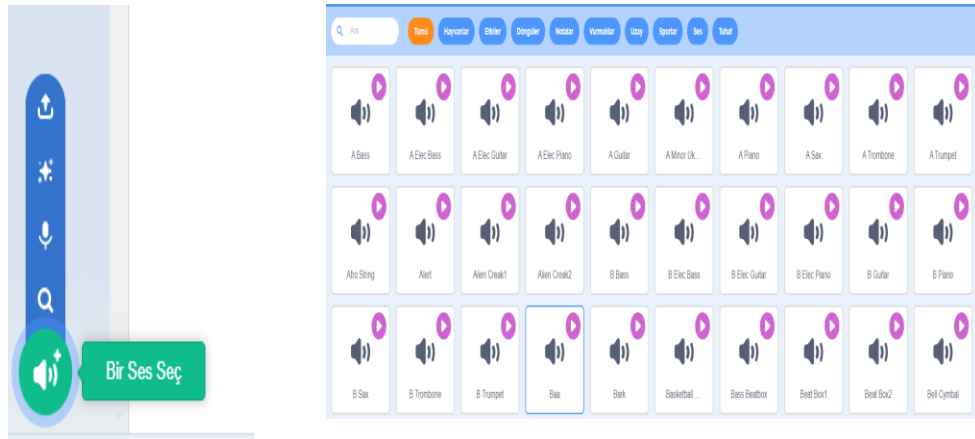
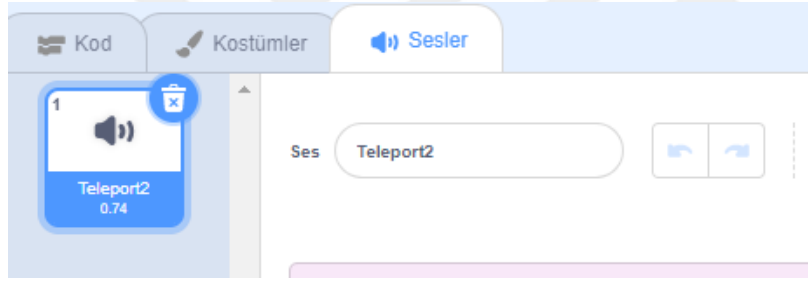
Trampoline ördek kuklası değdiği anda y konumunu değiştirerek ördek kuklasının zıplamasını sağlarız ve dekorun değişmesini belirtmek için haber sal komutunu kullanırız. Haber sal komutu ile dekorlar kısmına ördeğin trampoline değdiği ve dekor değiştirmesi gerektiği komutu gönderilir.

Öğretmen tarafından öğrencilere dekorlara ve kuklalara ses eklenebileceği söylenir. Ses eklemek için

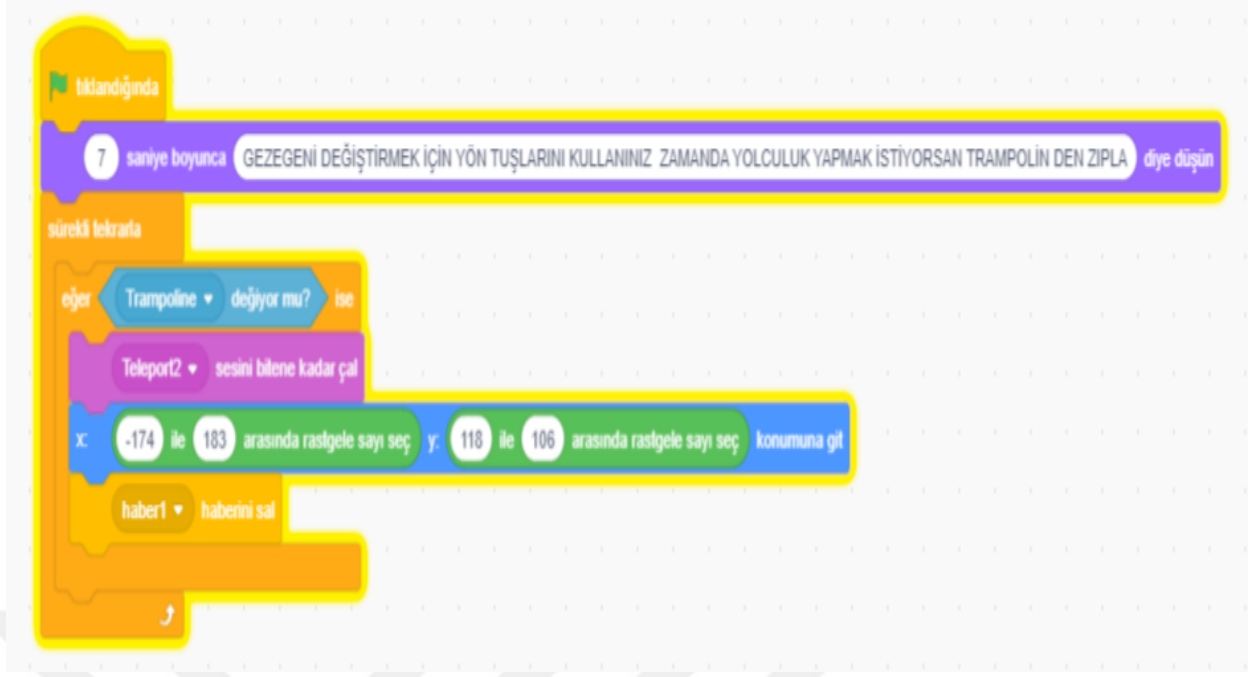


Sesini bitene kadar çal komutu kullanılır. Ses seçme işlemi için bu kısımdan kodun yan kısmından seçilebileceği gibi

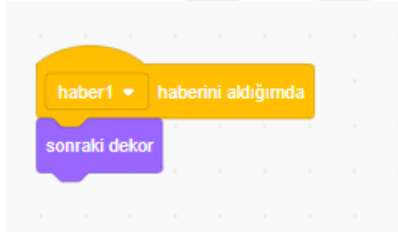
ses menüsünden de değiştirilebilir.



Ses kısmında bir ses seç denilerek scratch içerisinde bulunan seslerden eklenebilir. Sesinizi mikrofon ile kayıt alınabilir ve ses olarak kullanılır. Bilgisayarda bulunan herhangi bir waw, mp3 kukla sesi olarak kullanılabilir. Aynı zamanda bu seslerin üzerinde işlemlerde yapılabilir. Bu kısımda gezegen den gezegene geçerken teleport2 seçimini yaparız.



Dekorlar kısmının kod bölümüne gelerek haber geldiğinde dekorun değişmesi için sonraki dekor komutu kullanılır.

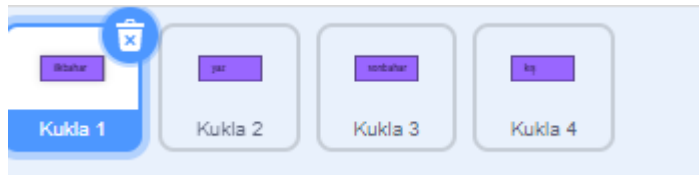


21- Mevsimleri Keşfet

Bu uygulamada öğretmen öğrencilerden ekrana 4 adet buton çizmelerini ve bunların yaz, sonbahar, kış ve ilkbahar isimlerini içeren, bu butonlara tıklandığında arka planında bu mevsimlerle alakalı olarak değişmesi istenmektedir.

Uygulama için ilk olarak internet tarayıcımızdan dört adet mevsimlerle ilgili resim indirilir ve bunlar sırası ile dekorlara yüklenir.

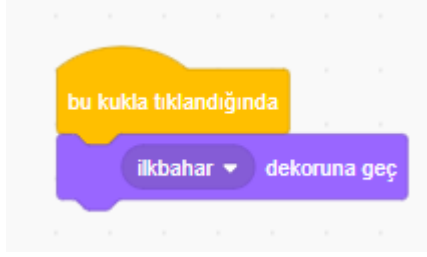
Kukla çiz menüsünden buton şeklinde dikdörtgen şekiller çizilir.





-Bu kısımda öğrenciler ikiyeşerli grup oluşturulur ve hangi kod bloğunun konulacağı konusunda uğraşmaları istenir.

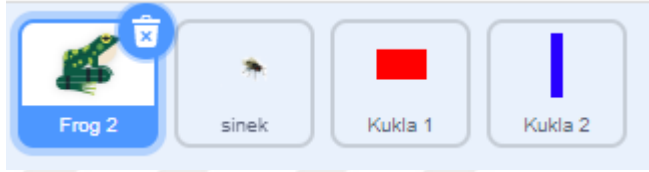
Her bir kuklaya sahne değişimi için dekor değiştir kodları eklenir. Burada bu kukla tıklandığında komutu kullanılır.



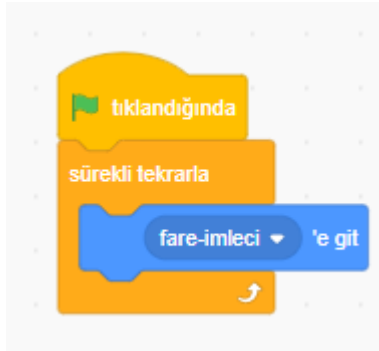
22-Sinek Yakalama Oyunu

Bu uygulamada fare ile hareket eden kurbağa kuklası ekranın rastgele noktalarında çıkan sinek kuklasını yemesi istenmektedir. Her bir yenilen sinekte aşağı kısımda bulunan dikdörtgen şeklindeki kukla mavi dikdörtgene doğru yavaşır ve yeşil dikdörtgene değene kadar bu işleme devam edilir. Sonunda “evdeki bütün sinekleri temizledin” mesajı verilir.

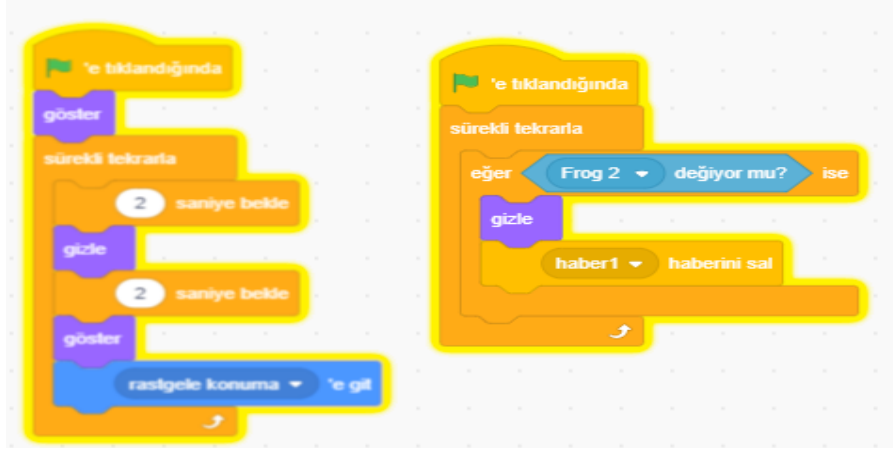
Uygulamada kullanılacak kuklalar eklenir ve çizilir.



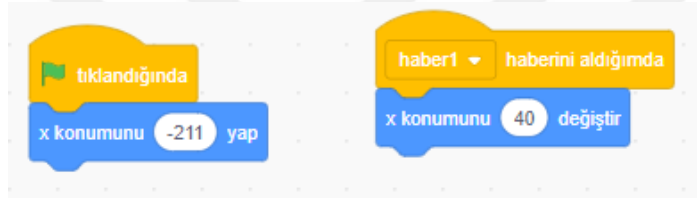
Kurbağa kuklasına kod kısmına fare ile hareket edebilmesi için kod bloğu yazılır.



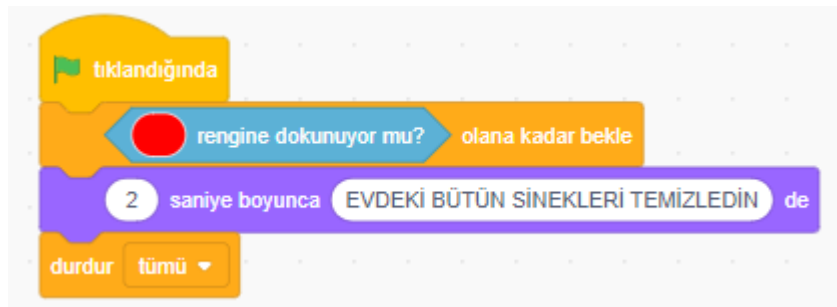
Sinek kuklasına giderek ekranda belli sürelerde rastgele konumlarda çıkması için gerekli komutlar yazılır. Kurbağa kuklası sineğe değdiği anda kırmızı kutunun harekete geçebilmesi içinde haber sal komutu kullanılır.



Ekranda ilerleme görevinde olan kırmızı kutu kuklaya hareket etmesi için kod blokları yazılır. Kurbağanın sineğe değdiği anda gönderilen haber geldiğinde kırmızı kutunun ilerlemesi için x konumu değiştirilir.



Hedef olarak gösterilen mavi kutu kuklasına kırmızı rengin değene kadar hiçbir kodu çalıştırmayıp sadece kırmızı rengin değdiği anda alttaki kodların çalışması için “olana kadar bekle” kod bloğu kullanılır. “Olana kadar bekle” kod bloğu ile bir işlem oluncaya kadar beklenir ve işlem olduktan sonra alttaki kodlar çalışmaya devam eder. Kırmızı kutu her bir sinek yenildiği anda ilerler ve mavi kutuya geldiğinde mesaj yazdırılır.

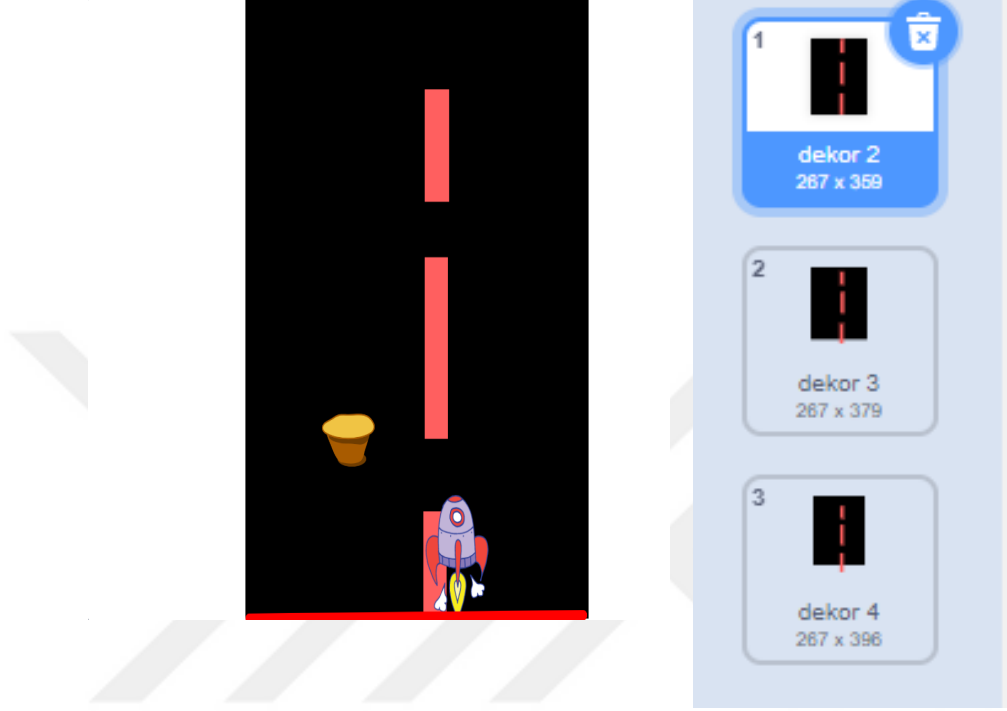


23- Engelden Kaçan Uzay Aracı

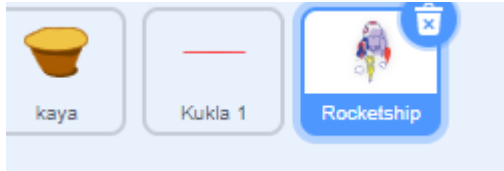
Bu bölümde öğretmen öğrencilerden ekranda klavyenin sağ sol tuşu ile hareket eden bir uzay aracı kuklası eklenmesi ve ekranda kayan yolda

yukarıdan aşağı doğru gelen kaya parçalarına çarpmadan yolda gitmesini sağlayan oyun yapılması istenmektedir. Bu zeminde çarptığı zaman oyun yine bitecektir.

İlk olarak gerekli olarak sahne çizimi yapılacaktır. Burada ekranın kayıyormuş gibi gözükmesi için yol çizgilerini her bir sahnede aşağıya doğru kaydırılır.



Sahne çizimden sonra ekrana gerekli kukla eklemesi yapılır.

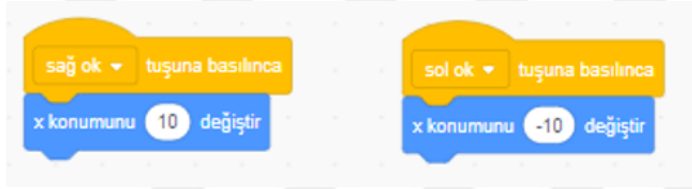


Bu kısımdan itibaren ders soru cevap şeklinde işlenir. Aşağıdaki sorular sorulur gerekli kodları yazması istenir ve belirli bir süre tanınır. Ve daha sonra doğru kod sınıfta tartışılır ve eklenir.

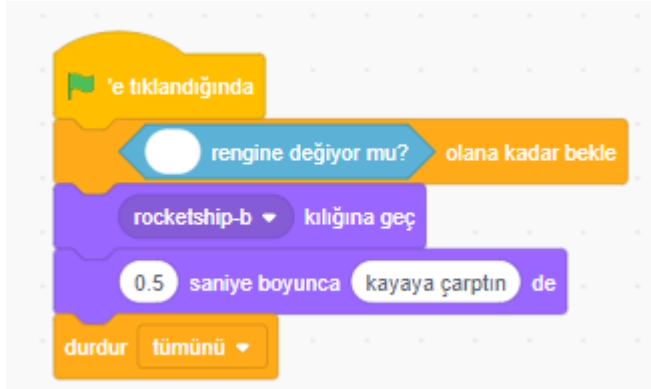
Roket kuklası seçilir ve kostümlerden roket kuklası parçalı bir biçime getirilir.



1-Roket kuklasının sağa-sola gitmesi için gerekli kodları yazınız.



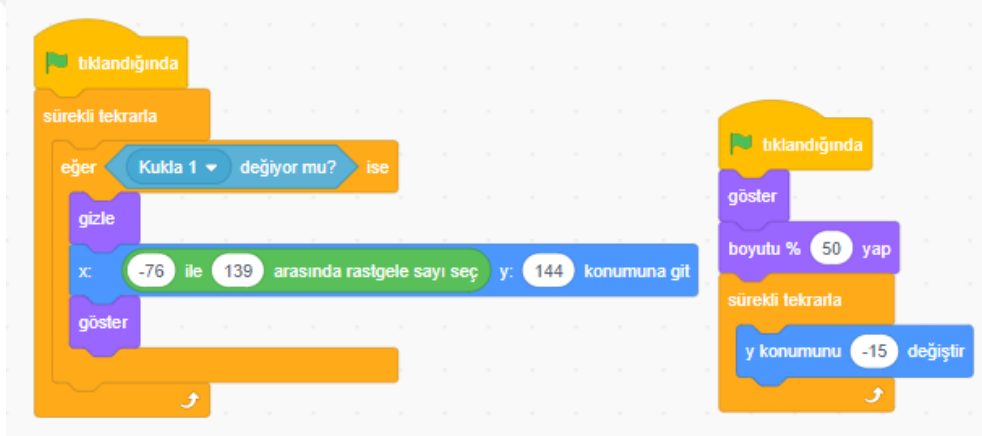
2-Araba siyah yoldan çıktığı anda “kenara çarptın oyun bitti” yazdırıp oyunun durmasını sağlayan kodları yazınız.



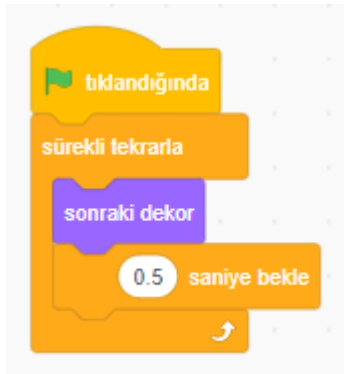
3-Kaya kuklasının araba kuklasına çarptığında “kayaya çarptın oyun bitti” yazdırıp oyunun durmasını sağlayan kod bloklarını yazınız.



4-Kaya kuklasının boyutunu %50 yapıp ekran üzerinde devamlı olarak yukarıdan aşağıya doğru akmasını sağlayan kod bloğunu yazınız.



5-Dekorlar kısmında yol çizgilerinin hareket ediyormuş gibi görünmesini sağlayan kod bloklarını oluşturunuz.



24- Labirent Oyunu

Bu bölümde öğretmen öğrencilerden labirent yapma oyunu isteyecektir. Oyun yapılırken dikkat edilecek hususlar şunlardır.

-Oyunun dekor bölümü üç sahneden oluşacaktır. Bunlar labirent haritası (internetten labirent diye görseller kısmından bulunabilir. Süre bitti yazan dekor, tebrikler yazan dekor şeklinde hazırlanır.) Bu kısımda öğretmen hazır olarak bir labirent resmini öğrencilere verebilir.

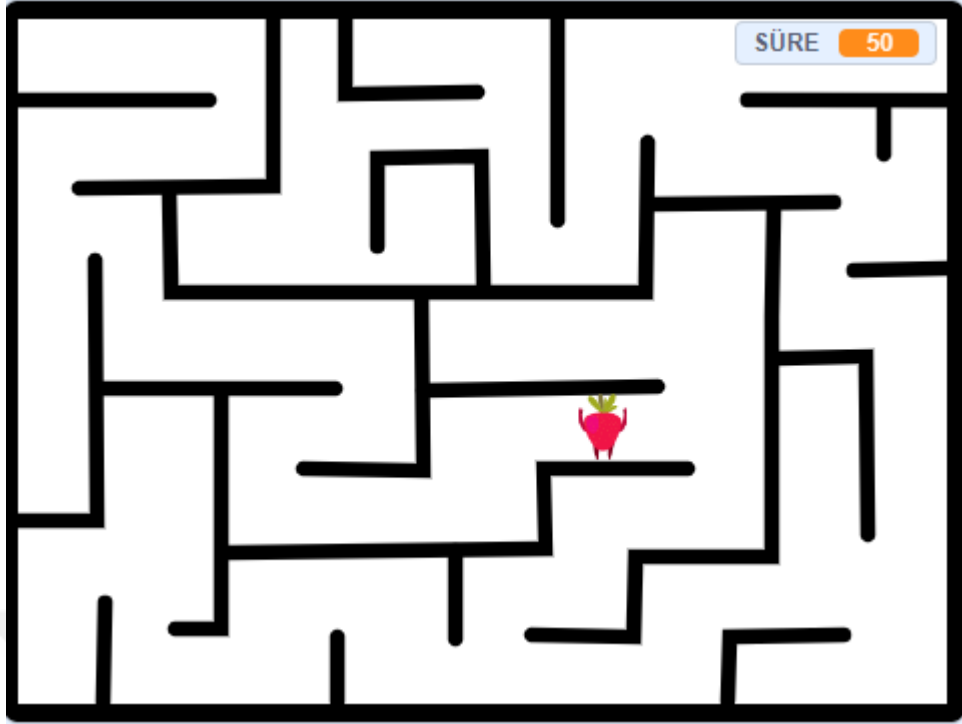
<https://drive.google.com/file/d/1X58nzheZ4q0eMG1btwG8PfGkBz5SShSa/view?usp=sharing>

-Ekranı bir adet bitiş için bayrak veya herhangi kukla eklenir labirentin sonuna gelince oyunun bitışı bu logo sayesinde bitirilecektir.

-Oyunda labirentler arasında hareket edebilecek bir kukla eklenir. Bu kukla yön tuşları ile hareket edecek olup ekrandaki labirent çizgilerinin arasındaki boşluklardan hareket edecektir. Bu çizgilerden herhangi birine değdiğinde kukla başlangıç noktası ile belirtilen kısma dönecektir.

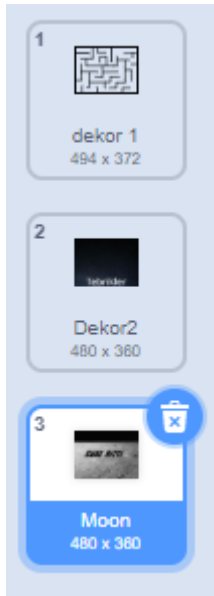
-Ekranı bir adet aktif biçimde süre olup devamlı olarak sayaç şeklinde devam edecektir. Bu süre bizim belirlediğimiz örnek olarak 50 sn aştığı anda oyunu bitirip süre bitti dekorunu açacaktır.

-Labirentin belli noktalarında bonus kuklaları çıkıp bu kuklalar yenildiği vakit süre kısmında belirlenen süre kadar azaltma yapıp oyunun daha geç bitmesi için kullanıcıya avantaj sağlayacaktır. Bu çıkan bonuslar aynı anda çıkmayıp labirentin belirli yollarında çıkacaktır. Buradaki süre yapımcı tarafından hesaplanacaktır. Çıkan bonus ise ekranda belirli bir süre bekleyip daha sonra yok olacaktır. Her bir bonus alındığında ise başarıyı simgeleyen bir ses eklenecektir.

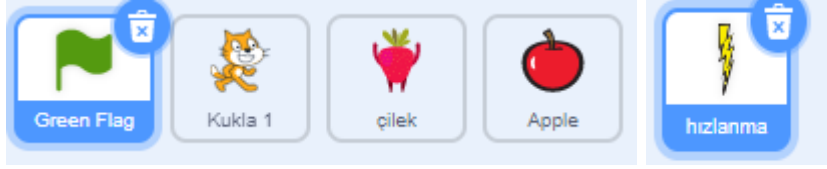


Bu kısımda öğretmen scratch konusunda son uygulama olmasından dolayı yukarıdaki koşulları içeren metni yansıtır ve öğrencilere kuklaları ve ekleyeceği dekorları gösterir ve geri kalan kod kısmını öğrencilerden yapması istenir ve 1 ders saati süre verilir. 2.dersimizde öğretmen öğrencilerin yapımını anlatır ve öğrencilerin yapmış olduğu yanlışları düzeltmeleri istenir.

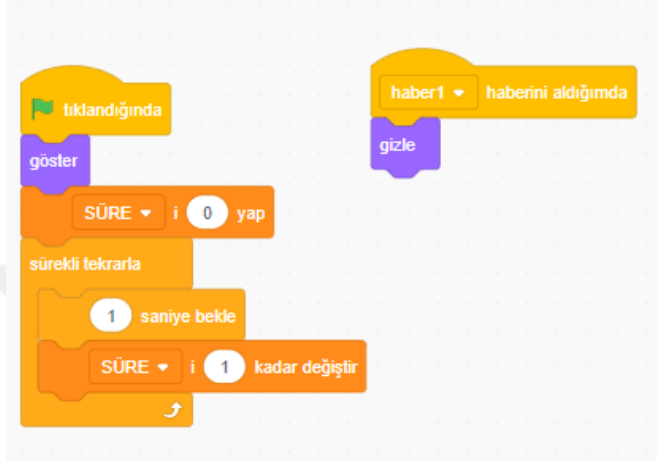
Ekrana gerekli dekorlar yüklenir ve çizilir.



Daha sonrasında gerekli kukla eklemeleri yapılır.

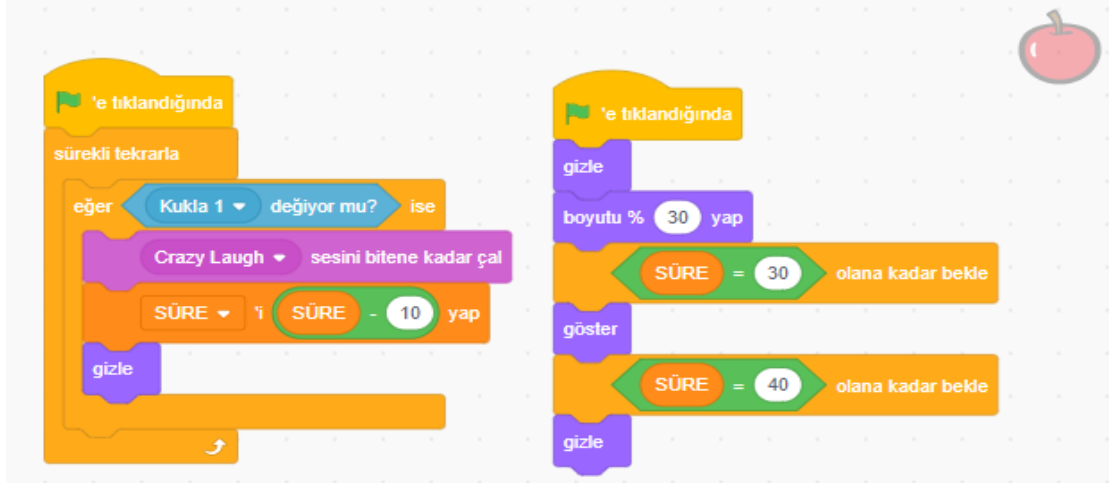


İlk başta yeşil bayrak kuklası kod bloğu tıklanır ve süre ayarlamaları burada yapılır. Oyunun bitiminde gelen haber ile gizlenmesi sağlanır.

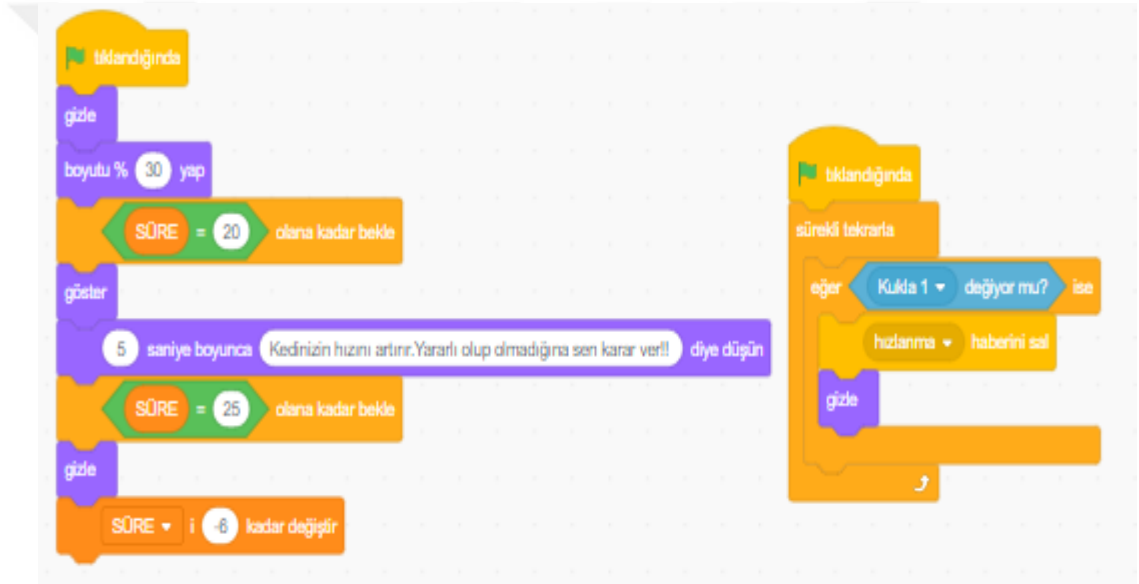


Bonus kodlarına geçilir ve bonus için belirlemiş olduğumuz çilek, elma kodlarına aynı yazım yapılır. Bu kısımda boyut ayarlama ve belli sürelerde ekrana çıkıp kaybolması ile ilgili kodlar yazılır. Diğer bonuslar da dikkat edilmesi gereken çıkacak olan sürenin farklı olmasıdır. Elma kuklasında 10 ila 20 sn ise diğer kuklalarda farklı süreler kullanılır. Diğer kısımda ise kedi kuklası değdiği zaman sürenin azalması ve oyunun daha geç bitmesini sağlar.





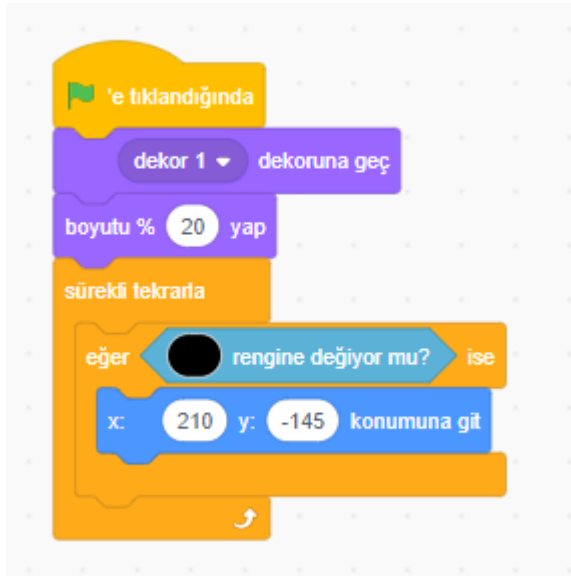
Yıldırım kulası seçilir ve gerekli kod blokları yazılır.



Kedi kuklası seçilir ve hareket etmesi için gerekli kodlar yazılır.



Daha sonrasında kedinin siyah çizgilere değdiği zaman başlangıç noktasına gitmesini sağlayan kod bloğu yazılır.



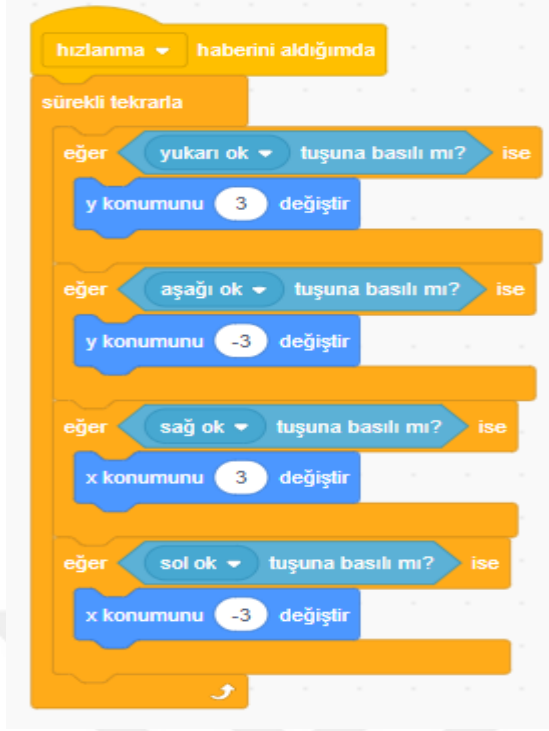
Yeşil bayrağa ulaştığında oyunun bitti dekoruna geçip ekranda kedi kuklasını büyüten kod bloğu yazılır.



Kedi kuklasına ayrı bir kod bloğu olarak arka planda işlemekte olan sürenin 50 sn olduğunda dekor değiştirip süre bitti yazılı dekorun gelmesi sağlanır. Oyundaki bütün araçlar durdurulur. Haber sal diyerek ekranın başlangıcındaki bayrak animasyonun kaybolması haber gönderilir.



Kedi kuklasının yıldırım kuklasına değdiği anda kedi kuklasının hızlanmasını sağlayan kod bloğu yazılır.



27-Ödev

İnternette bulmuş olduğunuz farklı bir labirent resmini ekleyiniz.

Bonus veren kukla sayısını artırınız.

Hızlandırma kuklası ile yavaşlatma kuklası ekleyiniz.

Süre bitiminde ekrana çıkan bir kuklayı kullanarak konuşma balonu ile biten süreyi yazdırınız.

Oyun devam ettiği süre boyunca arka planda bir ses çalmasını sağlayınız.

Sahne dekoru ekleyerek yeni bölümler ekleyiniz.

EK 4 Etik Kurul İzin



Bolu Abant İzzet Baysal Üniversitesi
Sosyal Bilimlerde İnsan Araştırmaları Etik Kurulu

Erhan DEMİRTAŞ
Bolu Abant İzzet Baysal Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar ve Öğretim Teknolojileri Eğitimi

Sayın Erhan DEMİRTAŞ,

“Gütüd Öğretim Modeli Etkililiğinin Çeşitli Değişkenler Açısından İncelenmesi” adlı İnsan Araştırmaları Etik Kuruluna yapmış olduğunuz başvurunuz (Protokol NO. 2021/316) kurulumuzun 30.07.2021 tarihli ve 2021/08 toplantısında değerlendirilerek etik olarak uygun bulunmuştur. Bilgilerinize sunarız.

Prof. Dr. Hamit COŞKUN (Başkan)	Prof. Dr. Mehmet ERGİT (Üye)
Prof. Dr. Altay EREN (Üye)	Prof. Dr. H. Birol YALÇIN (Üye)
Doç. Dr. Seval ALKOY (Üye)	Doç. Dr. Abdullah DURAKOĞLU (Üye)
Av. Zuhâl Demirci (Üye)	

EK 5 MEB İzin Belgesi



T.C.
BOLU VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : E-39307281-605.01-42589838
Konu : Araştırma İzin Talebi
(Erhan DEMİRTAŞ)

03/02/2022

VALİLİK MAKAMINA

İlgi : a) MEB Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.01.2020 tarih ve 81576613-10.06.02-E.1563890 (2020/2) Sayılı Genelgesi
b) Bolu Abant İzzet Baysal Üniversitesi Rektörlüğünün (Öğrenci İşleri Daire Başkanlığının) 24.01.2022 tarih ve 2200010066 sayılı yazısı.

Bolu Abant İzzet Baysal Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar ve Öğretim Teknolojileri Eğitimi Ana Bilim Dalı Bilgisayar ve Öğretim Teknolojileri Eğitimi tezli yüksek lisans programı öğrencisi Erhan DEMİRTAŞ'ın "**Gutüd Öğretim Modeli Etkililiğinin Çeşitli Değişkenler Açısından İncelenmesi**" konulu tez çalışmasına veri sağlamak üzere Müdürlüğümüze bağlı yazı ekinde belirtilen ortaokullarda öğrenim gören öğrencilere uygulama yapabilme talebine ilişkin ilgi (b) yazı ve ekleri Araştırma ve Değerlendirme İzin Komisyonu tarafından 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 hüküm, esas ve amaçlara aykırılık teşkil etmeyecek şekilde, **denetimi ilgili okul müdürlükleri tarafından gerçekleştirilmek üzere, eğitim öğretim aksatılmadan, gönüllülük esasına göre**, ilgi (a) Genelge doğrultusunda **2021-2022 Eğitim Öğretim Yılında** uygulanması müdürlüğümüzce uygun görülmektedir.

Makamlarınızca da uygun görülmesi halinde olurlarınıza arz ederim.

Dr. Cemil SARICI
İl Millî Eğitim Müdürü

OLUR
Ahmet ATILKAN
Vali a.
Vali Yardımcısı

Adres : Aktaş Mah. Çelik Sokak No:2 Merkez /Bolu

Bu belge güvenli elektronik imza ile imzalanmıştır.

Bu evrak güvenli olarak

teyit edilebilir.

