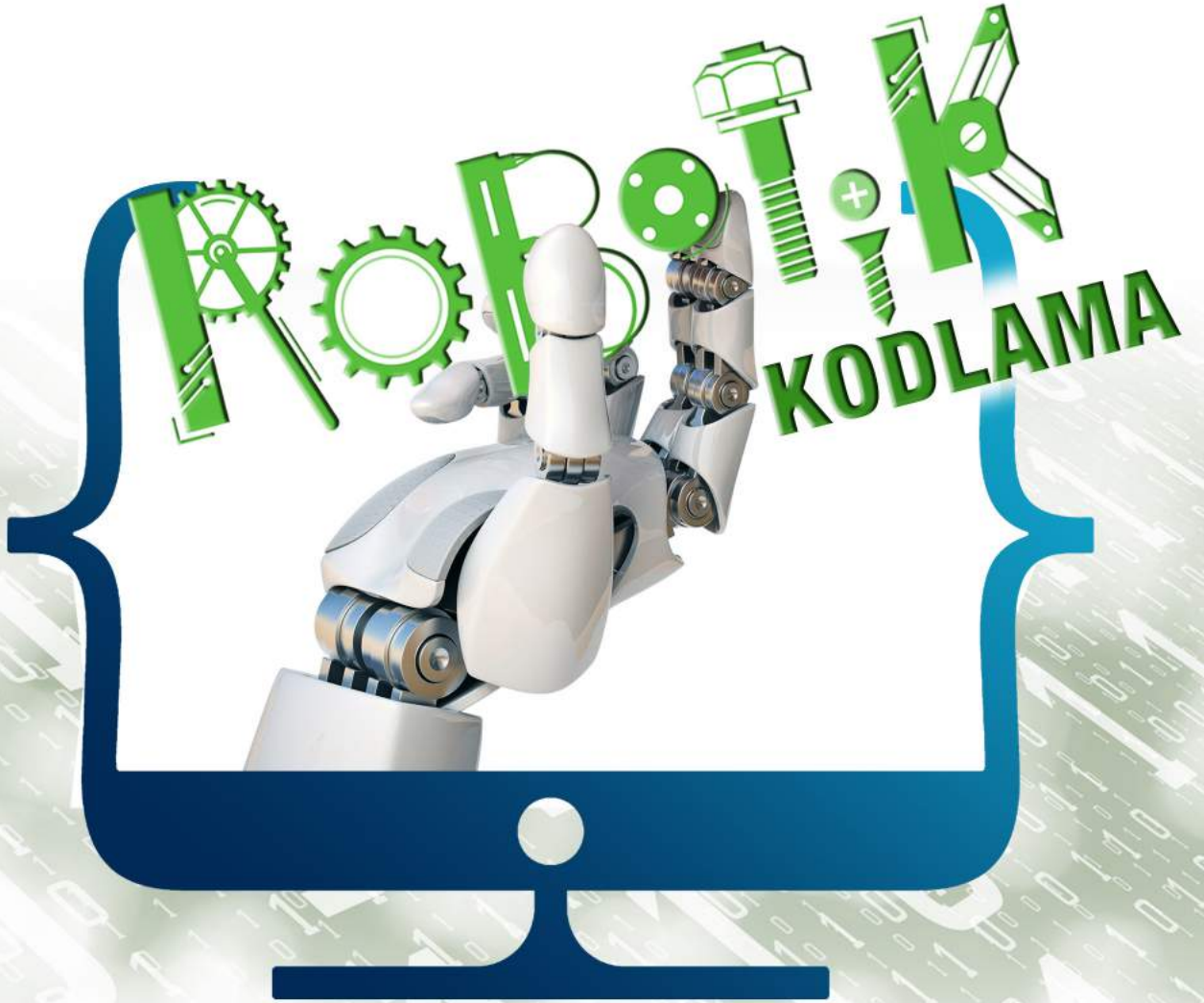


ROBOTİK VE KODLAMA EĞİTİMİ

KODLAMA EĞİTİMİ TEMEL DÜZEY





Birleşmiş Milletler
Eğitim, Bilim ve Kültür
Kurumu

UNESCO
Türkiye
Milli Komisyonu

Millî Eğitim Bakanlığı
Öğretmen Yetiştirme ve Geliştirme
Genel Müdürlüğü Adına
Doç. Dr. Adnan BOYACI
Genel Müdür

Editör
Sibel AKBİYİK
Daire Başkanı

Yazarlar
Dr. Murat ALTUN
Murat KOÇALI
Beytullah ÇİÇEK
Gökhan ARIKAN
Murat DURMAZ
Nihat BAL
Resul BÜTÜNER
Serhat Kağan ŞAHİN
Sinan DURĞUN

Grafik Tasarım
Serkan AKYÖRÜK

Kapak Tasarım
Melih BAKA ÇAKMAKLI



ISBN: 978-975-11-5359-3

Millî Eğitim Bakanlığı Öğretmen Yetiştirme ve Geliştirme Genel Müdürlüğü'nün yazılı izni olmadan bu kitap içeriğinin bir kısmı veya tamamı yeniden üretilemez, çoğaltılamaz, dağıtılamaz.

2020

ROBOTİK ve KODLAMA

(Kodlama Eğitimi Temel Düzey)



Doç. Dr. Adnan BOYACI
ÖĞRETMEN YETİŞTİRME VE GELİŞTİRME GENEL MÜDÜRÜ

ÖN SÖZ

Değerli Meslektaşlarım,

Tarihsel süreçte medeniyetler günümüzle kıyaslandığında çok uzun bir dönem içinde bulunduğu coğrafya ile sınırlı kalmış ve anılmıştır. İnşa ettikleri kültür içindeki değerler bilginin sistematik olarak uygulamada karşılık bulmasına ve dönemin teknolojisinin üretilmesine katkı sağlamıştır. Kültürlerin içindeki bilgiye ve bilginin kullanılmasına ilişkin değerler içinde buldukları dönemin öğrenme öğretme süreçlerinde medeniyet iddiasını taşıyan 'eğitim' sistemlerinin temel kılavuzlayıcısı olmuştur.

Özellikle öğrenmeye ve tüketmeye ilişkin coğrafi sınırlılıkların belirsizleştiği günümüzde medeniyet iddiası büyük ölçüde teknolojiyi tüketen değil üreten bireylerin yetiştirildiği, teknolojinin üretiminde; insanlığa, bireylere ve vatanın tüm fertlerine yönelik ahlaki bir sorumlulukla algılandığı bir çerçeveye oturmuştur. Bu çerçevede en temel öncelik bilgiyi ve bilgiye dayalı yenileşmeyi (innovasyonu) üreten eğitim sistemlerinin yeniden inşası üzerinedir. Girdisi, süreçleri ve çıktısı insan olan tek sistem niteliğini taşıyan eğitim sistemlerinin bu öncelik üzerine yeniden yapılandırılması ise insan kıymetlerine ilişkin bir kapasite geliştirme sürecine dayanmaktadır. 'Güçlü Öğretmen Güçlü Gelecek' sloganı ile çıktığımız yolda öğretmen ve okul yöneticilerimizin mesleki ve bireysel gelişimleri Bakanlığımızın stratejik önceliğidir. Bu çerçevede Millî Eğitim Bakanlığı Öğretmen Yetiştirme ve Geliştirme Genel Müdürlüğü olarak tüm öğretmen ve okul yöneticilerimize yönelik bilgi ve bilişim teknolojileri başta olmak üzere bireysel ve mesleki gelişim programları hazırlamaya ve sunmaya ilişkin çabalarımız devam etmektedir. "Robotik ve Kodlama: Kodlama Eğitimi Temel Düzey" öğretmen ve okul yöneticilerimizin konuya ilişkin temel düzeyde bilgi ve beceri kazanmalarına yönelik hazırlanmış ilk kitap özelliğini taşımakta ve dijital beceri ve yetkinlik ihtiyaçlarını gidermek amacıyla hazırlanmıştır. Bu kitabın analitik ve eleştirel düşünme gibi 21. yy. becerileri ile bilgi işlemsel düşünme faaliyetlerinin geliştirilmesine katkı sağlayacağını umuyoruz.

Kodlama Eğitimi kitabında, "bilgi işlemsel düşünme", "programlama", "temel elektronik", "blok temelli kodlama", "kodlama platformları ve eğitsel robotik kitleler" konuları ele alınmıştır. Kodlama; bireylerin problem çözme, bağımsız olarak ve iş birliği içinde çalışma, organizasyon yapma ve sorumluluk alma becerilerini kullanmalarını gerektirir. Bu becerileri kullanabilen bireylerde aynı zamanda akademik beceriler de gelişir. Robotik kodlama bireylere; organizasyon, problem çözme, ekip duygusu ve iş birliği yoluyla öğrenme becerileri kazandırır. Kodlama eğitimi sayesinde öğrencilerimiz, hayatın içinde var olan makinelerin çalışma mantığını kavrar ve benzerlerini oluşturmaya yönelik deneyimler elde ederler.

Kitapta yer alan kuramsal bilgilerin yanında etkinlikler de öğrencilerimizin farklı öğrenme alanlarında ve pek çok kazanıma yönelik uygulamaları, bilgi işlemsel düşünme eşliğinde gerçekleştirmeleri amacıyla hazırlanmıştır. UNESCO Türkiye Millî Komisyonu iş birliğinde hazırlanan kitapta emeği geçen öğretmenlerimize verdikleri katkı ve destekten dolayı çok teşekkür eder, bu çalışmanın öğretmenlerimiz ile öğrencilerimize faydalı olması temennisiyle sevgi ve saygılarımı sunarım.

Doç. Dr. Adnan BOYACI

ÖĞRETMEN YETİŞTİRME VE GELİŞTİRME GENEL MÜDÜRÜ



İÇİNDEKİLER

BÖLÜM 1: PROGRAMLAMAYA GİRİŞ	1
1.1. Bilgisayar Nasıl Çalışır?.....	1
1.2. Program, Programlamanın Tanımı ve İlgili Terimler	2
1.3. İkili Sayı Sistemi ve Makine Dili	5
1.4. Bilgisayar Programları Nasıl Çalışır?.....	7
BÖLÜM 2: ALGORİTMALAR VE AKIŞ ŞEMALARI.....	9
2.1. Bilgi İşlemsel Düşünmenin Temelleri.....	9
2.2. Problem Çözme	18
2.3. Algoritmik Düşünme	21
2.4. Temel Algoritma Öğeleri	21
2.5. Akış Şemaları Oluşturma	24
2.5.1. Sıralı Akış Yapısı.....	26
2.5.2. Koşullu Akış Yapısı	27
2.5.3. Tekrarlı Akış Yapısı	28
2.6. Uygulama Programları.....	29
2.6.1. Flow Chart Visual Programming	29
2.6.2. Diagrams	31
2.7. Algoritma ve Akış Diyagramı Etkinlikleri	33
BÖLÜM 3: BİLGİSAYARSIZ KODLAMA	57
3.1. Programlamaya Giriş Etkinlikleri	59
BÖLÜM 4: BLOK TEMELLİ PROGRAMLAMAYA GİRİŞ	103
4.1. Blockly.....	103
4.2. Code.org.....	108
4.3. ScrtachJr.....	114

4.4. Scratch	120
4.4.1. Scratch Kurulumu	120
4.4.2. Web Arayüzünden Scratch Çalıştırma	122
4.4.3. Arayüz Tanıtımı	126
4.4.3.1. Scratch Proje Oluşturma.....	127
4.4.3.2. Proje Kayıt	129
4.4.3.3. Kukla Ekleme	130
4.4.3.4. Dekor Ekleme	131
4.4.3.5. Kostüm Ekleme	133
4.4.3.6. Ses Ekleme	133
4.4.4. Kodlama Blokları	134
4.4.4.1. Hareket	134
4.4.4.2. Görünüm.....	138
4.4.4.3. Ses	142
4.4.4.4. Olaylar	144
4.4.4.5. Kontrol	146
4.4.4.6. Algılama.....	148
4.4.4.7. Operatörler	151
4.4.5. Doğrusal Mantık Yapısı ile Problem Çözme	154
4.4.6. Karar Yapıları ile Problem Çözme	163
4.4.7. Döngü Yapıları ile Problem Çözme	167
4.4.8. Değişkenler	182
4.4.9. Diziler (Listeler).....	187
4.4.10. Olaylar.....	193
4.4.11. Projelere Eklenti Ekleme.....	213
4.4.12. Scratchte Fonksiyon Kullanımı	231

BÖLÜM 5: KODLAMA PLATFORMLARI VE EĞİTSEL ROBOTİK KİTLER.....239

5.1. Arduino	239
5.2. Raspberry Pi.....	240
5.3. CeBot.....	241
5.4. LEGO	241
5.5. Mblock	242
5.6. Micro:bit.....	244
5.7. Vex.....	245
5.8. Tinkercad	245

KAYNAKÇA248

PROGRAM ÇIKTILARI

Bu mesleki gelişim programını tamamlayan öğretmenler:

- Programlamanın temel ilkelerini açıklar.
- Programlamanın eğitsel açıdan önemini açıklar.
- Öğrencinin bilgi işlemsel düşünme, problem çözme ve algoritmik düşünme becerilerini geliştirecek eğitim materyali tasarlar.
- Algoritma ve akış şeması oluşturmanın temel alındığı etkinlik planlar.
- İkili sayı sistemlerini ve ASCII dönüşümünü kullanarak programlama etkinliği planlar.
- Bilgisayar kullanmadan kodlama yapımının temel alındığı etkinlik planlar.
- Blok temelli programlamanın temel alındığı etkinlik planlar.
- Kodlama yapılan platformları ve eğitsel robotik kiti kullanarak eğitim materyali tasarlar.

BÖLÜM 1

PROGRAMLAMAYA GİRİŞ

Bu bölümde program ve programlamanın tanımı ile ilgili terimler, ikili sayı sistemi ve makine dili konularına yer verilmiştir.

1.1. Bilgisayar Nasıl Çalışır?

Bilgisayar, donanım ve yazılım olmak üzere iki ana bileşenden oluşur. Donanım, bilgisayarda bulunan her türlü fiziksel parçaya verilen isimdir. Bilgisayarda donanım dışında bulunan her türlü program, belge, dosya ve uygulamalara ise yazılım denilir. Bilgisayardaki donanımlar, yazılım aracılığıyla yönetilir. İnsanın bedeni, donanım olarak kabul edilirse insan zekâsı da bu donanımı işlevsel hâle getiren yazılım olarak tanımlanabilir. Bilgisayarın işlemleri gerçekleştirebilmesi, yazılımın donanımı yönetmesi ile gerçekleşir. Bu yönetme işi farklı programlar aracılığıyla yapılır. Yazılımlar; programlardan, yardımcı programlardan, belgelemelerden (yardım dosyaları) ve veri kaynak dosyalarından oluşur.

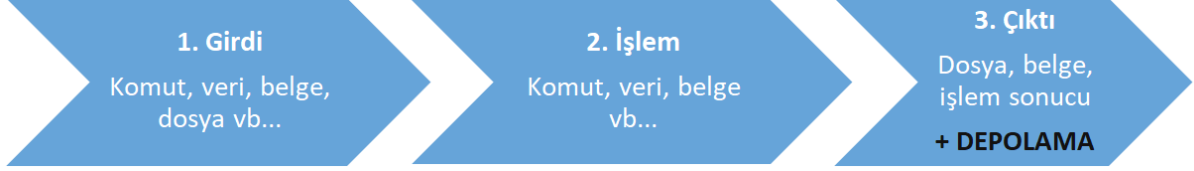
Bilgisayar yazılımları genel olarak iki ana gruba ayrılır:

- Sistem yazılımları;** bilgisayarı yöneten, denetleyen, kontrol eden yazılımlardır. Örnek: Linux, Pardus, Windows.
- Uygulama yazılımları;** belirli bir alana ve kullanıma yönelik olarak kullanıcılar için geliştirilmiş yazılımlardır. Örnek: Word, Excel, Virüs programı.

Bilgisayarın çalışma sistemi; girdi, işlem ve çıktı (Şekil 1.1) olarak özetlenebilir.

Girdi, kullanıcının bilgisayara komut, belge ve veri girdiği aşamadır. Girdiler, kullanıcıdan başka bir elektronik cihazdan, bir algılayıcıdan, bir programdan da yapılabilir.

BÖLÜM 1



Şekil 1.1. Bilgisayarın çalışma mantığı

İşlem aşaması, girdiler üzerinde istenen çıktıların alınması için işlemlerin yapılmasıdır. Bu aşamada bilgisayara komutlar verilerek (klavyede bir tuşa basmak, fareyi tıklamak ve bir komut yazmak gibi) istenilen işlemlere yönlendirilir. Bir dosya kaydetmek, bir yazı yazmak, basit bir toplama işlemi yapmak bu işlemlere örnek gösterilebilir.

Çıktı aşaması ise istenilen işlemler yapıldıktan sonra elde edilen sonuçlardır. Veri, belge, dosya gibi çıktıların verildiği aşamadır. Bu aşamada depolanan çıktılar başka işlemler için girdi olarak tekrar kullanılabilir. Örneğin; 2 sayının toplandığı bir işlemde girdi aşamasında kullanıcı tarafından iki sayı girilir, işlem aşamasında toplama (sayı1 +sayı2) işlemi yapılır ve çıktı aşamasında da sonuç ekrana yazdırılır.

1.2. Program, Programlamanın Tanımı ve İlgili Terimler

En temel anlamıyla **program**, bilgisayara bir işlemi yaptırmak için yazılan komutlar dizisidir (TDK, 2019). Başka bir ifade ile program; elektronik araçların, birbirleriyle haberleşebilmesini ve uyumunu sağlayan komut, talimatlar dizisidir. Her program, belirli bir amaca yönelik olarak bir problemi çözmek için geliştirilir. Örneğin bir web sayfasının tasarımı için ayrı, veri tabanı için ayrı bir program kullanılabilir.

Elektronik bir cihaz, programda yer alan talimat dizisine uygun olarak belirli bir görevi yerine getirir. Programlar “kod” diye tabir edilen kelimelerin belirli kurallar ile yazılmasından oluşur. Programların yazılması için özel olarak tasarlanmış, sınırlı talimatları, kuralları ve kodları içeren yapılar ise **programlama dilleri** denilir. Her programlama dilinin kendine özgü kuralları vardır ve komut dizimleri başka bir ifadeyle sentaksı (söz dizimi, **syntax**) çoğunlukla farklıdır. Farklı programlama dilleri kullanarak farklı amaçlar için farklı programlar geliştirilebilir. Bir oyun tasarımı için kullanılan programlama dili, üç boyutlu tasarım için kullanılan programlama dilinden farklı olabilir. Her programlama dilinin kullanım amacına göre avantaj ve dezavantajları vardır. **Programlama** ise belirli bir amaca ulaşmak için oluşturulan algoritmayı herhangi bir programlama dilinde uygun talimatlar dizisi hâline getirmektir. Bilgisayarlar, insandan farklı olarak programda verilen komutları, mantıksız olsa bile birebir uygular. Bu yüzden programların iyi yazılmış (talimatların açık) olması önemlidir. Hangi programlama dilinde olursa olsun program yazarlar (**programcı**), sorunun çözümüne ilişkin

adımları belirler (**algoritma**) ve programlama diline uygun olarak talimat dizisi oluştururlar. Programlar, yazıldıktan sonra oluşturulan talimat dizisinin istenildiği gibi çalışıp çalışmadığını anlamak, hataları bulmak ve ayıklamak için test edilir. Bir uzay mekiği fırlatılmasında, bir nükleer santralin çalışmasında veya bir tren rayındaki sinyalizasyonda çeşitli programlar çalışır. Çok küçük bir hata bile ciddi sorunlara neden olabilir. Bilgisayar programlarındaki hatalar, 1940'ların başındaki elektronik hesaplama makinesinin elektrik rölesinden çıkarılan bir güveden dolayı **böcek (bug)** olarak adlandırılır. Programda bulunan hataların temizlenmesine **hata ayıklama işlemi (debug)** denilir. Bunun için program veya yazılımların test aşamaları, tüm hataları bulmak amacıyla defalarca tekrarlanabilir. Hatta zaman içinde ortaya çıkan sorunlar giderilerek veya yeni özellikler eklenerek, programlarda sürekli güncellemeler yapılarak yeni sürümleri (**versiyon**) geliştirilebilmektedir. Hataların gözden kaçmaması için programın dikkatle test edilmesi gerekir. Ancak bazı programların tam olarak doğru çalıştığından emin olmak için test etmek mümkün olmayabilir.

Aşağıda Şekil 1.2'de Python'da yazılmış bir kod görülmektedir.

```
print("*****")
print("*****")
print("*****Merhaba öğretmenim*****")
print("*****")
print("*****")
```

Şekil 1.2. Python isimli programlama dilinde yazılmış kod satırları

Şekil 1.2'deki program çalıştırıldığında ekrana Şekil 1.3'teki görüntü gelir.

```
*****
*****
*****Merhaba öğretmenim*****
*****
*****
```

Şekil 1.3. Program çalıştırıldığında ekrana gelen görüntü

BÖLÜM 1

Gerçek dünyada olduğu gibi program yazmak için de farklı diller bulunmaktadır. Bir programlama dilinin tam olarak hangi karakterlerin nerede ve nasıl kullanılması gerektiğiyle ilgili ayrıntılı kurallarına **söz dizimi (syntax)** adı verilir. Aynı işlemi yapan bir program yazmak için farklı programlama dillerinde farklı söz dizimleri kullanmak gerekebilir. Söz diziminde her sembol önemlidir. Bir programdaki küçük bir hata, bazen bir sembol, bir karakter bile programın çalışmasını durdurabilir veya hatalı bir şekilde çalışmasına neden olabilir. Şekil 1.4'te farklı programlama dillerine ait örnek söz dizimleri sunulmuştur.

C Programlama Dili
<pre>#include <stdio.h> int main() { printf("Merhaba Dünya"); return 0; }</pre>
Java
<pre>public class MerhabaDunya{ public static void main(String []args){ System.out.println("Merhaba Dünya"); } }</pre>
Python
<pre>print ("Merhaba Dünya")</pre>

Şekil 1.4. Farklı programlama dillerine ait örnek söz dizimleri

Programlama dilleri, gerçek dünyadaki dillere çok benzemektedir. Örneğin, bazı programlama dilleri daha popülerdir, ayrıca söz dizimleri birbirine çok benzeyen ve ortak bir kökten gelen diller de bulunmaktadır.

1.3. İkili Sayı Sistemi ve Makine Dili

Bilgisayar için harf, rakam veya renkler bir anlam ifade etmez. Bilgisayar, fiziksel olarak sadece 0 ve 1'leri algılayabilir ve sadece bunlar üzerinden işlem yapar. Fiziksel olarak algılamada "0" elektrik akımının var olmadığı, "1" ise elektrik akımının var olduğu anlamına gelir. Bilgisayar için tek geçerli gösterim biçimi olan 0 ve 1'lerin olduğu sayı sistemine **ikili sayı sistemi** denir. Günlük hayatta **onlu sayı sistemi** kullanılır. Bilgisayar, 0 ve 1'ler üzerinden aritmetik ve mantıksal işlemleri çok hızlı bir şekilde gerçekleştirerek görevleri yerine getirir. Bilgisayar, tüm bu işlemlerde **ikili sayı sistemini** kullanarak bu iki sayının fiziksel olarak gösterimiyle donanımını kontrol edebilir.

Her 0 veya 1'e **bit** ("binary digit" in kısaltmasıdır.) denir. Bir bit, genellikle bilgisayarın ana belleğinde açık veya kapalı bir transistör veya şarj/deşarj olan bir kondansatör ile temsil edilir. Transistör, iletim hâlinde ise 1, iletim hâlinde değilse 0'ı ifade eder.



Şekil 1.5. Transistörde ve kondansatörde 0 ve 1 olarak gösterilmesi (Kaynak: <https://classic.csunplugged.org/>)

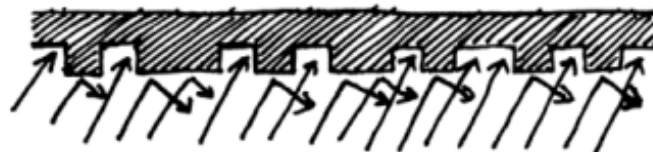
Verinin bir telefon hattı veya radyo frekansı üzerinden iletilmesi gerektiğinde, 0 ve 1'ler için alçak ve yüksek frekanslar kullanılır.

Manyetik disklerde (sabit diskler ve disketler) ve bantlarda, bitler manyetik alandan etkilenen özel demir tozları üzerinde tutulur. Kuzey-güney veya güney-kuzey şeklinde kaplanmış bir yüzeydeki manyetik alanın yönüne göre bilgi 1 veya 0'dır.



Şekil 1.6. Manyetik disklerde bilginin saklanması (Kaynak: <https://classic.csunplugged.org/>)

Ses CD'leri, CD-ROM'lar ve DVD'ler, bitleri optik olarak depolar, yüzeyin bir parçaya karşılık gelen kısmı ışığı yansıtır veya yansıtmaz. 0 ve 1 verisi, buna göre okunmuş olur.



Şekil 1.7. CD veya DVD'lerde verinin kodlanması (Kaynak: <https://classic.csunplugged.org/>)

BÖLÜM 1

Bilgisayarların yalnızca iki farklı değer kullanmasının nedeni, bunu yapabilen aygıtlar oluşturmanın çok daha kolay olmasıdır. **Onlu sayı sisteminde** bir kodlama ile tüm basamakların temsil edilebilmesi için 10 yansıma seviyesine sahip **CD**'ler yapılması gerekirdi. Ancak böyle bir **CD**'nin çalışmasını sağlamak için çok pahalı ve hassas cihazlar gerekirdi. Bilgisayarlarda sıfırlar ve birler donanımsal olarak sadece düşük ve yüksek voltajlara veya kuzey / güney mıknatıslanma gibi durumlara karşılık geldiği için 0 ve 1'leri kullanmak "Işık yansıdı, ışık yansımadı." gibi ifadeleri kullanmaktan daha pratiktir. Bu yüzden bilgisayardaki tüm veri, **bit** kullanılarak temsil edilir. Belgeler, resimler, şarkılar, videolar, sayılar, kullanılan programlar ve uygulamalar sadece çok sayıda **bit** ile ifade edilmektedir. Bilgisayarın hızı, bir defada işleyebileceği bit sayısına bağlıdır. Örneğin, 32 bitlik bir bilgisayar, bir defada 32 bitlik bilgi işlerken 16 bitlik bir bilgisayar, 32 bitlik bir işlemi daha küçük parçalara bölerek 2 defada işlemektedir. Bu da bilgisayarın hızını yavaşlatır. Tek başına bir bitin fazla bir anlamı yoktur, bu nedenle genellikle 0 ile 255 arasındaki sayıları temsil edebilen sekizlikler hâlinde gruplandırılır. Sekiz bitlik bir gruba **bayt** denir. Bir bayt; bir pikselin rengini, bir harfi, bir rakamı veya özel bir karakter gibi anlamlı bir bit grubunu ifade eder. Kullandığımız alfabe, sayılar ve semboller kullanıcının anlaması için **onlu sayı sisteminde** bir sayı olarak gösterilirken bilgisayarın anlaması için de bu sayı, **ikili sayı sisteminde** 0 ve 1'ler olarak gösterilir. Bilgisayarın tüm bu karakterleri temsil edebilmesi için kaç bit gerekir? Günümüzde çoğu bilgisayar, karakter başına bit kullanmaya dayanan **ASCII** (Bilgi Değişimi İçin Amerikan Standart Kodu) adlı bir temsili kullanmaktadır. Bu standart, bilgisayarların aynı dili konuşmasını sağlamaktadır. ASCII'de 33 tane basılmayan kontrol karakteri ve 95 tane basılan karakter bulunur. Bu karakterler ikili sayı sisteminde bir sayı ile temsil edilerek makine diline çevrilmektedir.

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Şekil 1.8. ASCII Kod Tablosu (Kaynak: <https://tr.wikipedia.org/wiki/ASCII>)

1.4. Bilgisayar Programları Nasıl Çalışır?

Bir program çalıştırıldığında bilgisayarın programı anında çalıştırdığı ve işlemi yaparak hemen çıktı verdiği düşünülebilir. Ancak gerçekte bilgisayar programları, başka bir program aracılığıyla çalıştırılır. Hatta bazı durumlarda programlar, bilgisayar tarafından daha kolay çalıştırılabilecekleri bir formatta depolanır.

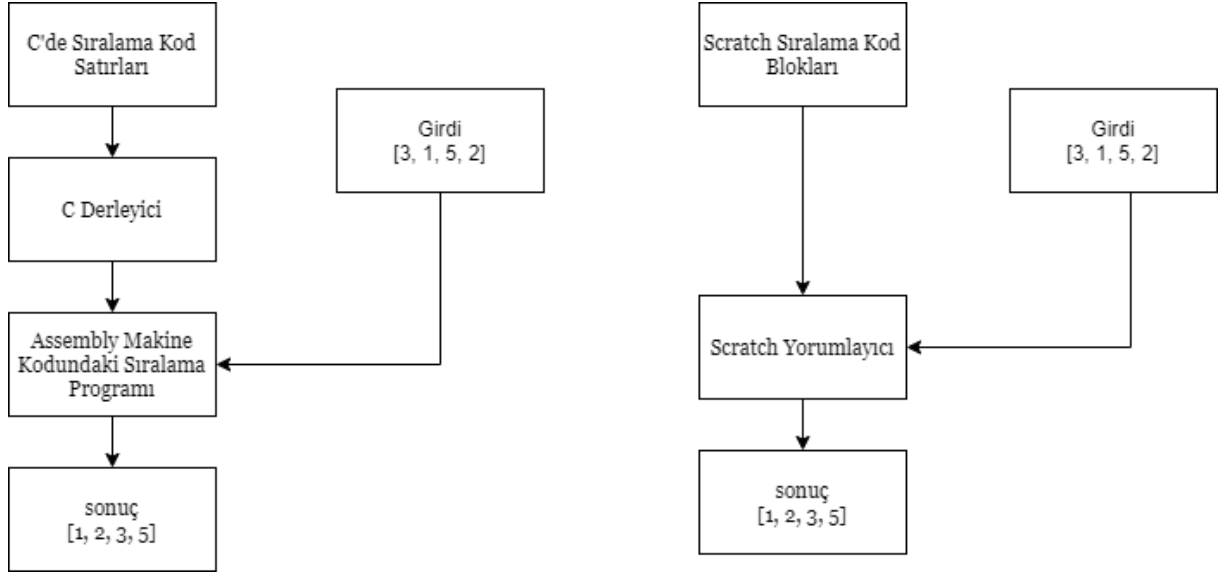
Neden programlama dillerine ihtiyaç duyulmaktadır? Bilgisayarlara neden konuşarak veya konuşulduğu gibi yazılarak talimatlar verilememektedir? Bilgisayara “3 ile 5’i çarp.” komutu verilirse bu, bilgisayar için anlamlı bir ifade veya talimat değildir. Aslında bunun için yapay zekâ yöntemlerinden yararlanmak gerekir. İnsan dilini anlamak, bilgisayar için karmaşık ve zor bir durumdur. Klasik programlamada bilgisayar, programlar aracılığıyla aldığı açık talimatları izleyebilir ve yorumlamadan yerine getirir. Bu nedenle, bilgisayarın anladığı talimatlar konusunda kısıtlanmış ve net olan dillere (**programlama dilleri**) ihtiyaç vardır. Ancak bir bilgisayar, doğrudan bu dillerde verilen kodları da çalıştıramaz. En düşük düzeyde, bir bilgisayarın kodları çalıştırmak için fiziksel donanım kullanması gerekmektedir. Toplama, çıkarma, çarpma ve bölme gibi aritmetik, daha büyük veya eşit gibi basit karşılaştırmalar, transistörler içeren fiziksel bilgisayar çipleri yoluyla elektrik ile ikili olarak temsil edilen sayılarla yapılmaktadır. Elektriksel çıktı, aynı zamanda ikili olarak temsil edilen bir sayıdır. İki sayının toplanması gibi basit bir aritmetik işlem yapmak için hızlı ve ucuz bir devre oluşturmak o kadar da zor değildir. Ancak insanlar bilgisayarlara biraz daha karmaşık işler (“aylık giderleri hesaplama” veya “asal çarpanları bulma” gibi) yaptırdığı için bu ihtiyaçlara yönelik devreler oluşturmak daha zordur. Bu nedenle yüksek seviyeli kodları doğrudan anlayabilen bilgisayarlar oluşturmak hem maliyetli hem de güçtür. Bunun yerine Python (veya Java, Basic, JavaScript, C ve benzeri) gibi diller kullanılarak programlar yazılabilmekte ve çok sınırlı bir dizi kodu takip edebilen bilgisayarlarda çalıştırılabilmektedir. Programcıların standart dillerde yazdığı kodlar, elektronik devrenin doğrudan gerçekleştirebileceği basit talimatlara (**makine dili**) derleyici ve yorumlayıcı adı verilen programlarla dönüştürülür.

Derleyici (Compiler): Bir programın kodlarını daha sonra çalıştırılabilecek makine koduna çeviren programdır.

Yorumlayıcı (Interpreter): Programlama dilinde yazılan komutları sırayla kontrol edip komutların yürütülmesini sağlayan bir yazılımdır. Programdaki kodları alan ve hemen yapan bir programdır. Her iki yaklaşımın da avantajları vardır. Gerçekte, çoğu modern dil derleme ve yorumlama karışımını kullanır. Örneğin, Java programı, makine koduna Java’dan daha yakın olan ByteCode adlı bir **ara dille** derlenir. ByteCode, daha sonra bir yorumlayıcı tarafından yürütülür. Program yaygın kullanım için dağıtılacaksa genellikle makine kodunda olması istenir çünkü daha hızlı çalışacaktır. Bu durumda kullanıcının kendi programlama dili için bir yorumlayıcıya da sahip olması gerekmez. Kullanıcı, kullanmak için makine kodunu indirdiğinde orijinal üst düzey programdaki kodların içeriğini göremez

BÖLÜM 1

(Örnek: C, Ada, Pascal ve Delphi). Yorumlanan programlar, daha hızlı bir şekilde test edilebildiği ve kodların içinde gerçekleşen adımların takibi daha kolay olduğu için avantajlı olur. Bazen derleme sürecinden geçmek zorunda kalmadan kodlar tek tek yazılarak anında sonuçlarına bakılabilir. Bu nedenle yorumlanan programlama dilleri, giriş dilleri için yaygın olarak kullanılır (Örnek: Scratch ve Java Script) Aynı zamanda hızlı bir şekilde yazılabildiği ve test edilebildiği için basit görevleri yerine getiren komut dosyaları gibi programlar için de kullanılır. Java ve Python gibi derleme ve yorumlama işlemini birlikte kullanan programlama dilleri de bulunmaktadır. Verilen sayıları sıralayan bir program yazılıp çalıştırıldığında bir yorumlayıcı ve derleyicide farklı işlemler (Şekil 1.9) gerçekleşmektedir.



Şekil 1.9. Derleyici ve yorumlayıcının çalışma mantığı CSFG, 2020'den uyarlanmıştır.

(Kaynak: <https://csfieldguide.org.nz/>)

Derleyici, sıralamayı yapacak bir makine kodunu üretmektedir. Girdide verilen sayıların sıralaması, bu ikinci program tarafından yapılmaktadır. Yorumlayıcı, programdaki kodları hemen işleyerek girdideki sayıları sıralar. Derleyici, dağıtılabilecek bir makine kodu programı oluşturur. Ancak bunun için fazladan bir işlem yapar. Girdi, oluşturulan makine kodu programı tarafından çalıştırılarak sıralama işlemi yapılır. Derleme işlemi, programın ilk çalışmasında bir kere yapılır ve daha sonra derlenen kod kullanılır. Yorumlama işlemi ise program her çalıştığında yapılmaktadır.

Sonuç olarak bilgisayar programı hangi dilde yazılırsa yazılsın kullanılmadan önce veya kullanılırken makine diline çevrilmesi/yorumlanması gerekir.

BÖLÜM

2

ALGORİTMALAR VE AKIŞ ŞEMALARI

Birçok programlama dili ve farklı bilgisayar türü mevcut olsa da temel nokta, ele alınan bir problemin çözümüdür. Bilgi işlemsel düşünme, bilgisayarlar için problem çözme sürecini tasarlamak, bilgisayarla düşünerek problemlere yaklaşmak ve çözmek için kullanılan fikir ve kavramların gelişimine destek olur. Bilgi işlemsel düşünme süreci, ilk kod satırını yazmadan önce başlayan fonksiyonel becerileri ifade eder. Bilgi işlemsel düşünme, salt bir kodlama yeteneğini değil; birden fazla soyutlama düzeyinde düşünme yolunu tanımlar. Bilgi işlemsel düşünme, içine kodlama becerilerini de alan daha geniş bir çatıyı ifade etmektedir. Bilgi işlemsel düşünme becerileri, sorunu çözmek için adımların tanımlandığı (algoritma) ve mümkünse çözümü simüle etmek, test etmek ve hata ayıklamak üzere modellerin oluşturulduğu süreçtir. Bilgi işlemsel düşünme becerileri kullanılarak oluşturulan algoritmalar, bir problemi çözmek için problemin bileşenlerini temsil edecek veri yapılarından ve çözümü üretmek için gerekli adımlardan oluşur. Algoritma olmadan kod olmaz. Kodlama, bir algoritmanın bir programlama dilinde ifade edilmesidir. Kodlama sayesinde algoritmalar bilgisayar tarafından yürütülebilir. Programlama dilleri, algoritma sürecinin ve veri türlerinin bilgisayarda oluşturulmasını ve çalıştırılmasını sağlar. Kodlamaya giriş yapılmadan önce bu konuların ayrıntılı olarak ele alınması yararlı olacaktır.

2.1. Bilgi İşlemsel Düşünmenin Temelleri

Bilgi işlemsel düşünme; bilgisayar biliminin kavramlarından yararlanarak problem çözme, sistem tasarlama ve insan davranışlarını anlama olarak tanımlanır. (Wing, 2006; Gülbahar ve Kalelioğlu, 2018).

Bilgisayar Bilimi Öğretmenleri Derneği (CSTA - Computer Science Teachers Association) ve Uluslararası Eğitim Teknolojileri Topluluğu'na (ISTE - International Society for Technology in Education) göre bilgi işlemsel düşünme aşağıdaki özellikleri kapsayan bir problem çözme sürecidir:

1. Problemleri bilgisayar veya başka araçlar yardımı ile çözebilmek için formülleştirme,

BÖLÜM 2

2. Mantıklı bir şekilde verileri düzenleme ve çözümleme,
3. Model ve simülasyonlar aracılığı ile verileri sunma,
4. Algoritmik düşünme çerçevesinde çözümleri otomatikleştirme,
5. Kaynakları etkin ve etkili bir şekilde kullanarak en uygun çözüm/leri tanımlama, çözümleme ve uygulama,
6. Bulunan çözümü farklı problemlere transfer etme ve genelleştirerek transfer etme.
(ISTE & CSTA, 2011).

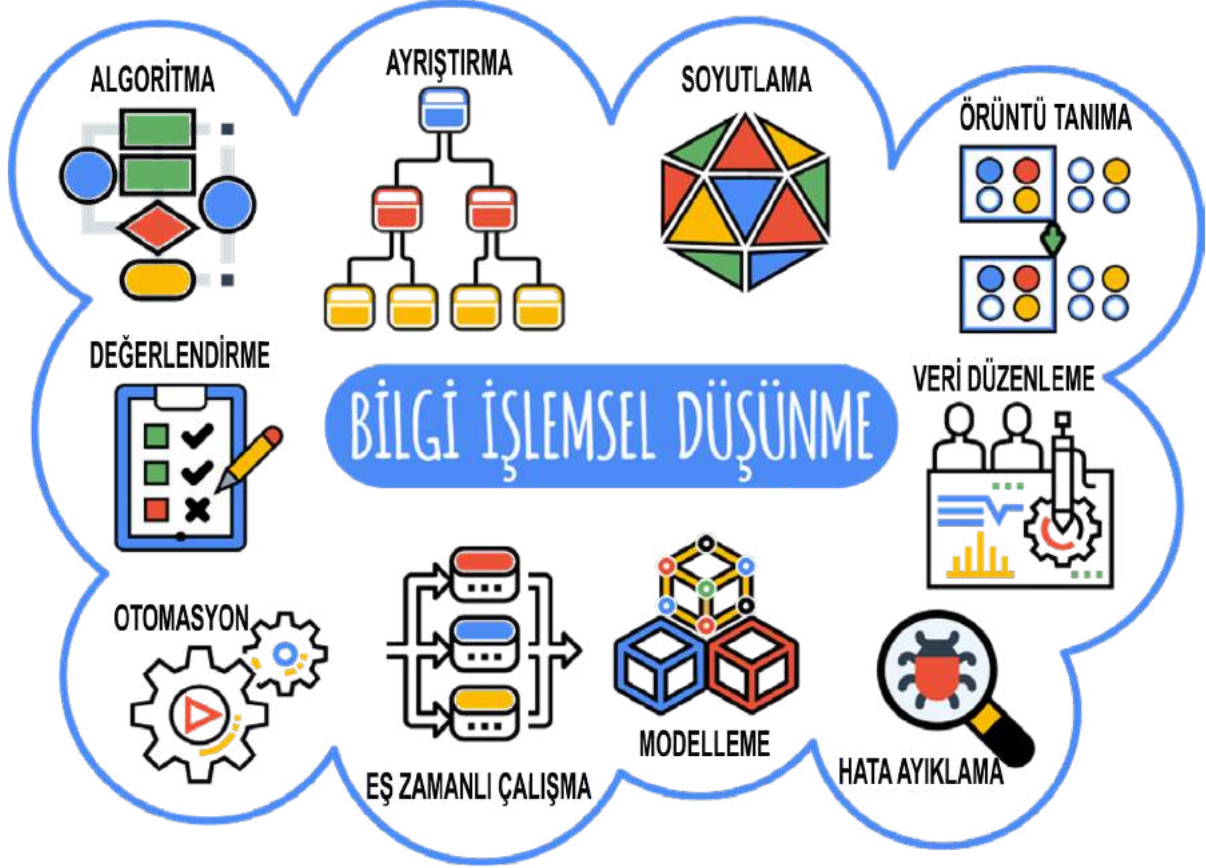
Bu bilgiler göz önünde bulundurulduğunda bilgi işlemsel düşünmenin birçok alt eylem ve kavramı da içinde bulundurduğu görülmektedir. Bilgi işlemsel düşünme sürecinde öğrencilerin tecrübe kazandığı süreçler genel anlamda; veri düzenleme, soyutlama, ayrıştırma, örüntü tanıma, eş zamanlı çalışma, algoritma tasarımı, modelleme ve otomasyon olarak sıralanır.

Veri düzenleme; veri toplama, veri çözümleme ve veri gösterimi gibi işlemleri içinde barındıran, veri ile ilgili tüm işlemlerin yapılmasıdır.

Veri toplama; nitel veya nicel yöntemler ile veri kayıt etme yani bilgi elde etme sürecidir. Günlük sıcaklık değerlerinin kaydedilmesi, öğrencilere ait cinsiyet, boy, kilo gibi bilgilerin anket kullanılarak toplanması ve çevre kirliliğine yönelik nitel ve nicel verilerin ortaya konulması veri toplama sürecine örnek olarak gösterilebilir.

Veri çözümleme; verileri anlamlı olarak ifade etmek için detaylı incelemek, toplanan veriyi anlamaya çalışmak, örüntü bularak sonuçlar çıkarmaya çalışmaktır. Günlük sıcaklık değerlerinin sabah, öğle ve akşam saatlerindeki değişimine yönelik genelleme yapmak, erkek ve kız öğrencilerin boy ve kilo ortalamalarını hesaplamak, deniz ve hava kirliliğinin sebeplerini ve sayısal değerlerini ortaya koymak veri çözümleme sürecine örnek olarak gösterilebilir.

Veri gösterim süreci ise yapılan analiz sonuçlarının görselleştirilmesi aşamasıdır. Günlük sıcaklık değerlerinin saatler içerisindeki değişim grafiğinin hazırlanması, kız ve erkek öğrencilerin boy ve kilo bilgilerinin sütun grafiklere aktarılması, hava ve deniz kirliliğine yönelik bilgilerin paylaşıldığı grafik ve raporların hazırlanması veri gösterim sürecine örnek olarak gösterilebilir.



Şekil 2.1. Bilgi işlemsel düşünme süreci

(Kaynak: Bilgi İşlemsel Düşünme Becerisinin Disiplinlerarası Yaklaşım ile Öğretimi. ÖYGGM, 2020.)

Ayrıştırma; verileri, işlemleri daha küçük parçalara bölerek problemi çözmek veya amaca ulaşmaktır. Karmaşık bir problemi alt basit problemlere bölmek sonuca ulaşmada kolay ve hızlı çözümler sağlar. Örnek olarak bir kitap fuarında hangi kitabı satın alacağınıza yönelik tercih süreci incelendiğinde aşağıdaki soruları kullanarak alt parçalara ve işlere ayırmak tercihe daha kolay ulaşmayı sağlar:

- Fuardaki reyonlarda hangi tür kitaplar var?
- Siz, hangi tür kitapları seviyorsunuz?
- Sevdiğiniz kitap türlerinin bulunduğu reyon numaraları nelerdir?

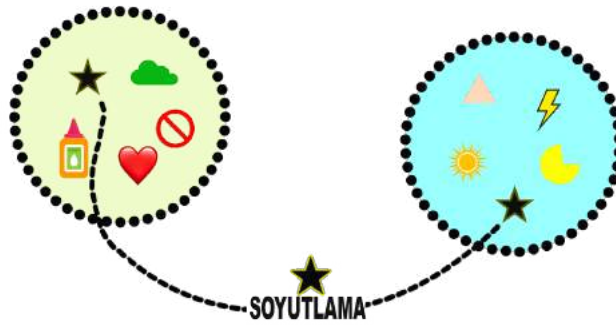
BÖLÜM 2

- Bu reyonlar, fuar yerleşimi içinde nerededir?
- Kaç kitap alacak bütçeniz var?

Bu tercih sürecinde de olduğu gibi karşılaşılan probleme yönelik tüm detaylar ayrı ayrı değerlendirildiğinde çözüme daha hızlı ve daha kolay ulaşılır. Aşağıda, problem durumları ayrıştırma sürecine yönelik örnekler bulunmaktadır.

- Suçlunun bulunabilmesi için suç işlenen yerlerin her bir bölümünün ve olay yerindeki her bir kişinin tek tek incelenmesi (BBC, 2017)
- Şiirin özelliklerine göre kafiyelerine, vezinlerine ve vurgu tonlamalarına ayrılması (Google Education, 2017)
- Tatil planlaması yapılırken otelin ayarlanması, yolculuk için bilet alınması, gereken kıyafetler için valiz hazırlanması (Lego, 2018)
- Bir evin bir anda temizlenmesi fikrine karşın oda oda günlere dağıtılarak temizlenmesi (MEB, 2018)

Soyutlama; bilgisayar biliminin temel kavramlarından biridir ve problemin çözümü için gerekli bilgiye odaklanmaktır. Yani, yapılan işlere odaklanmak için ihtiyaç duyulmayan özelliklerin görmezden gelinmesini (filtrelemesini) ifade eder. Böylece çözülmeye çalışılan konunun bir temsili (fikri) oluşturulmuş olur. Gösterilen bir haber metninin ya da videonun ana fikrinin bulunması, maddenin hâl değişim deneyine yönelik sonuçlarının özetlenmesi soyutlama için örneklerdendir. Şekil 2.2’de, nesnelere arasındaki ortak özellikleri tespit ederek gereksiz ayrıntıların gizlendiği soyutlama sürecine yönelik bir örnek verilmektedir.



Şekil 2.2. Bilgi işlemsel düşünmede soyutlama becerisi
(Kaynak: Bilgi işlemsel düşünmede soyutlama becerisi, Sert, 2017.)

Örüntü tanıma; veri yığnında bulunan benzerlikleri, farklılıkları veya kuralı ortaya koymaktır. Benzerlikler veya modeller, karmaşık problemlerin çözümünü kolaylaştırdığı gibi daha etkili ve verimli çözümler oluşturulmasını sağlar. Bir voleybol maçındaki örüntüleri ortaya koymak için aşağıdaki sorulara yanıt vermek yeterli olacaktır:

- Oyunda hangi taktikler kullanılmıştır?
- Bu maçta işe yarayan taktikler hangisi olmuştur?
- Hangi taktikler işe yaramamıştır?
- En etkili/etkisiz oynayan oyuncular kimler olmuştur?

Trafik lambalarının belirli bir işleyişi sürekli devam ettirmesi de örüntülere yönelik etkili bir örnektir (Lego 2018). Parmak izi, yüz tanıma gibi biyometrik sistemlerde de örüntünün tanımlanması ve eşleştirilmesi işlemleri günlük yaşamdaki diğer örneklerindendir (Üzümcü 2019).

Eş zamanlı çalışma (paralel işleme), aynı amaç doğrultusunda farklı işlerin aynı anda tamamlanmasına imkân sağlayan süreçtir. Örneğin okul tanıtım videosu yapım projesinde bir grup öğrenci, senaryoyu kurgular; bir grup, fon müziğini belirler ve diğer bir grup ise gerekli görsel materyalleri temin edebilir. Son aşamada eş zamanlı çalışmaların ürünleri birleştirilerek proje tamamlanır.

Algoritma tasarımı, herhangi bir problemi çözebilmek için gerekli adımları sıralamak ve tasarlamaktır. Algoritma, bir işlemin yapılması için gerekli olan yönergeler listesi veya kurallar dizisidir. Günlük yaşantımız içerisinde gerçekleştirdiğimiz birçok eylem veya etkinlik, algoritmaya örnek olarak verilebilir. Bir yemek tarifinin uygulanması, online alışveriş yapılması, doktorda muayene olunması, evden iş yerine gidilmesi veya satın alınan demonte bir ürünün birleştirilmesi gibi birçok eylem veya etkinlik algoritmalara örnektir. Bu örneklerde de görüleceği gibi yapılacak işlemler, komut olarak tanımlanan belirli bir sırada planlanarak algoritmalar oluşturulmaktadır. Bilgisayar programlarının oluşturulmasında başlangıç aşaması olan algoritmalar, bir akış şeması çizilerek veya sözde kod yazılarak oluşturulur.

Şekil 2.3'te bir öğrencinin sabah uyandığında gerçekleştireceği görevlere yönelik örnek bir algoritma görülmektedir (MEB, 2018).

BÖLÜM 2



Şekil 2.3. Bir öğrencinin sabah uyanığında gerçekleştireceği görevlere yönelik örnek bir algoritma
(Kaynak: Sabah kalkıp okula gidinceye kadar yapılacak işlerin algoritması. MEB, 2018)

Hata ayıklama, geliştirilen algoritmanın sonucu olan çözümün doğruluğundan ve verimli bir şekilde çalıştığından emin olmaktır. Bu işlem, test ile hatalar tespit edilerek yapılır. Test etme sürecinde çözüm algoritması gözden geçirilir, yönergeler uygulanır ve sonucun istediğimiz gibi olup olmadığı belirlenir. Eğer sorunlu bir kısım varsa ilgili komut veya yönerge değiştirilir ve tekrar test edilir. İstenilen çözüm algoritması ortaya çıkıncaya kadar bu süreç devam ettirilir. Bir yemek tarifinin uygulaması sonunda

istenilen lezzeti elde edemeyen kişinin, tüm süreci geriye dönük kontrol etmesi bu duruma örnek olarak verilebilir. Aşçı kullandığı malzemelerin miktarlarında değişiklikler yaparak tekrar yemek tarifini uygulayacaktır. Tam olarak istediği ve beklediği lezzeti elde edene kadar bu süreç döngüsel olarak devam edecektir. Günlük hayatta, yetişkinlerin sıklıkla karşılaşılabileceği buna benzer örnekleri çoğaltmak mümkündür. Ancak çocukların, bu soruları kendilerine sorarak cevaplar buldukları en etkili süreçler üretim süreçleridir. Çocuk, basit bir arduino devre tasarımının çoğu aşamasında, nerede hata yaptığını ya da neyi değiştirirse devresinin çalışacağını sürekli sorgular. Bu nedenle, öğrenme ortamlarında hata ayıklama becerilerinin gelişimini en fazla destekleyen çalışmalar, ürün geliştirilen uygulamalı öğretim süreçleridir.

Otomasyon, tekrarlayan işlemleri yapabilmek için bilgisayar veya makine kullanımı olarak açıklanabilir (Wing, 2008). Genel olarak robot programlama, oyun tasarlama veya diğer programlama etkinlikleri gibi tekrarlayan işlemlerin bilgisayar kullanılarak yapılması örnek olarak gösterilebilir. Anlık olarak birden fazla kişiye mail gönderilmesi, hastanede veya bankada bekleyenlere sıra numarası verilmesi ve trafikte hız sınırı ihlallerinin otomatik olarak belirlenmesi gibi günlük hayattan birçok örnek verilebilir.

Modelleme, gerçek yaşam süreçlerinin benzerinin veya bir modelinin geliştirilmesi işlemi olarak tanımlanabilir. Bilgisayarlar aracılığı ile nesnelerin üç boyutlu modelleri hazırlanabileceği gibi nesnelerin durum ve eylemlerinin tanımlanabileceği simülasyonlar da hazırlanabilir. Matematiksel işlemlerin, insan organlarının modellenmesi veya deney simülasyonları ve animasyonları örnek olarak verilebilir.

Değerlendirme, tüm tasarım süreçlerinin son adımında yer almaktadır. Bir probleme ilişkin ortaya çıkan çözümün hedeflerle uyumluluğunun ve verimliliğinin incelenmesidir. Bu amaçla, çözüme ilişkin olarak aşağıdaki sorular sorulur:

- İstenen kriterleri karşılıyor mu?
- Eksik veya hatalı bir noktası var mı?
- Geliştirilebilir mi?

Bilgi işlemsel düşünmenin eğitim süreçleriyle bütünleştirilmesi, öğrencilere günlük yaşam problemlerini farklı açılardan analiz etme ve problemlere çözüm üretme, yenilikleri keşfetme gibi önemli katkılar sağlayacaktır (Gülbahar ve Kalelioğlu, 2018). Bilgi işlemsel düşünmenin öğretiminde aşağıda şekillendirilen yaklaşım ve teknik etkili bir şekilde kullanılabilir. (Weinberg, 2013, Gülbahar ve Kalelioğlu, 2018).

BÖLÜM 2



Şekil 2.4. Bilgi işlemsel düşünme becerisi kazandırmak için yaklaşımlar

(Kaynak: Bilgi İşlemsel Düşünme Becerisi Kazandırmak için Yaklaşımlar. Gülbahar ve Kalelioğlu, 2018.)

Öğrenciler, bilgi işlemsel düşünme çerçevesinde kodlamayı kullanarak oyunlar, simülasyonlar, dijital hikâyeler gibi uygulamalar geliştirebilirler. Bu uygulamalarda problemlere, disiplinlerarası yaklaşımla teorik bilgiyi (matematik, fizik, kimya, biyoloji, yabancı dil vb.) bütünleştirerek çözümler üretebilirler.

Bilgi işlemsel düşünme becerisi; iş birlikçi öğrenmeyle birlikte eleştirel düşünme, algoritmik düşünme, yaratıcı düşünme, problem çözme ve iletişim becerilerini de kapsar (CSTA & ISTE 2011). Bu çok yönlü eğitim süreçlerinin değerlendirilmesinde ise Şekil 2.5'te verilen örnek araçlar kullanılabilir.

BÖLÜM 2



*Şekil 2.5. Bilişim eğitimi sürecinde kullanılacak bazı ölçme araçları
(Kaynak: Bilişim eğitimi sürecinde kullanılacak bazı ölçme araçları. ÖYGM, 2020.)*

Şekil 2.5'te verilen değerlendirme araçları, bilgi işlemsel düşünme öğretim süreçlerinde öğretmenler tarafından ekili bir şekilde kullanılabilir.

Günümüz bilgi toplumunda eğitim gören dijital çağın öğrencilerinin, sınıf içi öğrenmenin ötesine geçerek yaşam boyu öğrenen, problemlere yaratıcı ve yenilikçi çözümler üretebilen, içinde bulunduğu duruma uyum sağlayabilen, edindiği bilgileri farklı durum ve problemlere transfer edebilen, öğrenmeyi öğrenmiş donanımlı bireyler olması gerekmektedir (Gülbahar, Kert ve Kalelioğlu, 2019). Bu bağlamda bilgi işlemsel düşünme, yenilikçi bir yaklaşımı ele alır.

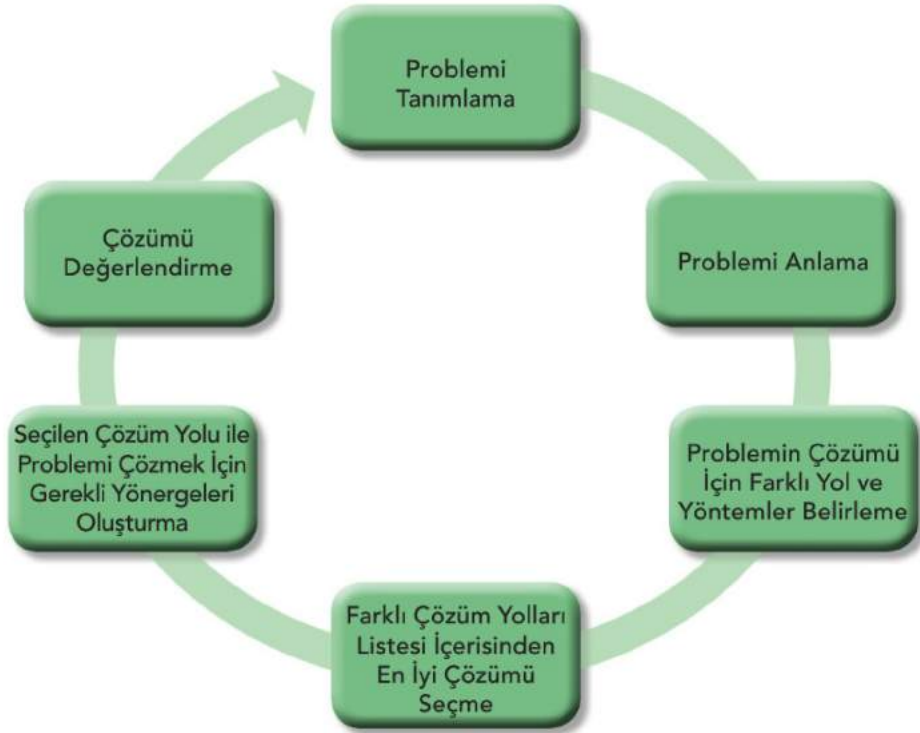
BÖLÜM 2

2.2. Problem Çözme

Bireyler, yaşantıları boyunca farklı problemlerle karşılaşır. Bu problemler bazen çözümü kolay bazen de bireyi zorlayacak karmaşık bir yapıda olabilir. Problemler, farklılık gösterse de ortak noktaları şu şekilde özetlenebilir:

- Problemler rahatsızlık yaratabilir.
- Bir belirsizlik içerir ve çözüm yolları açıkça görülmeyebilir
- Bu belirsizliklerin giderilmesi için zihinsel ya da fiziksel bir dizi işleme ihtiyaç duyulur.

Bu noktalardan hareketle, çocuklara problem çözme becerisi kazandırmak, gelecekte karşılaşılabilecekleri hayat zorluklarına karşı onları güçlendirecektir. Bu becerinin kazandırılması için izlenecek problem çözme süreci aşağıda verilen Şekil 2.6'daki gibidir:



Şekil 2.6. Problem çözme becerisi kazandırmak için izlenecek problem çözme süreci (Kaynak: MEB, 2019).

Problemi Tanımlama: Problemi çözmeye başlamadan önce problemin açık, anlaşılır ve doğru bir şekilde tanımlanması gerekir. Problemin ne olduğu bilinmezse o problem çözülemez.

Problemi Anlama: Çözüme geçmeden önce problemin çok iyi anlaşıldığından emin olunması gerekir. Problemin neler içerdiği ve kapsamı doğru anlaşılmalıdır. Ayrıca çözümü uygulayacak insan ya da sistemin hangi bilgilere sahip olduğu çok iyi bilinmelidir. Bilinmeyen herhangi bir kavram ya da yönerge problem çözüm sürecinde kullanılmamalıdır. Yani *“Problemi anlamak, problemi yarı yarıya çözmek demektir.”*

Problemin Çözümü İçin Farklı Yol ve Yöntemler Belirleme: Problemin çözümü için tüm olasılıkları içeren farklı yol ve yöntemler belirlenmelidir. Bunun için konu hakkında farklı kişilerin görüşleri alınabilir. Problem çözmek için tek bir yol yoktur.

Farklı Çözüm Yolları Listesi İçerisinden En İyi Çözümü Seçme: Çözüm önerilerinin olumlu ve olumsuz yönleri ortaya konulmalıdır. Bu nedenle değerlendirme yapılmasını sağlayacak ölçütler oluşturulmalıdır. Bu ölçütler, her bir çözüm yolunun değerlendirilmesinde çözüme rehberlik etmelidir. Problem çözmek için tek bir yol yoktur: En iyi yol vardır.

Seçilen Çözüm Yolu ile Problemi Çözmek İçin Gerekli Yönergeleri Oluşturma: Bu aşamada adım adım numaralandırılmış yönergeler oluşturulması gerekir. Yönergelerin ikinci adımda belirtilen bilgileri kapsamasına dikkat edilmelidir.

Çözümü Değerlendirme: Çözümü test etmek ya da değerlendirmek, sonucun doğruluğunu kontrol etmek anlamına gelir. Sonucun doğru olması ve problemi olan bireyin beklentilerini karşılama düzeyi önemlidir. Sonuç, yanlış çıkmış ya da bireyin beklentilerini karşılamamış ise problem çözme sürecine baştan başlanması gerekir.

BÖLÜM 2

Örnek Soru

Şekil 2.7’de beraberindeki nesneleri nehrin karşısına taşıması gereken bir çiftçi ile ilgili bir problem verilmiştir. Bu çiftçinin bir tilkiyi, bir kazı ve bir mısır çuvalını nehrin karşısına geçirmesi gerekmektedir. Çiftçinin bu işlemi gerçekleştirmek için küçük bir teknesi vardır ancak bu teknede çiftçi ile birlikte en fazla bir nesneye daha yer vardır. Ne yazık ki tilki ve kaz açtır. Bu yüzden tilki kaz ile yalnız kalmaz çünkü tilki, kazı yiyebilir. Aynı şekilde kaz ve mısır çuvalı yalnız bırakılamaz çünkü kaz, mısırı yiyebilir. Bu koşullarda çiftçi, nehrin karşısına tilki, kaz ve mısırı sorunsuz bir şekilde nasıl ve hangi sırayla geçirebilir?



Şekil 2.7. Çiftçinin bir tilkiyi, bir kazı ve bir mısır çuvalını nehrin karşısına geçirmesi görseli
(Kaynak: Tilki, Kaz ve Mısır Çuvalı. MEB, 2019.)

2.3. Algoritmik Düşünme

Algoritmik düşünme, problem çözümünde faydalanılan sıralamalarla ve kurallarla düşünme yoludur. Yeni bir algoritma üretmeyi veya var olan bir algoritmayı anlamayı sağlayan şu bilişsel becerileri kapsamaktadır (Futschek, 2006):

- Verilen problemi analiz etme becerisi
- Problemi açık olarak ifade edebilme becerisi
- Verilen problem için gerekli olan temel işlemleri belirleme becerisi
- Temel işlemleri kullanarak doğru bir algoritma oluşturma becerisi
- Problemin bütün olası durumlarını (genel veya özel) ele alma becerisi
- Bir algoritmanın verimliliğini artırma becerisi

Bu becerilerle bütünleştirilmiş öğretim süreçlerinde temel algoritma öğeleri kullanılır.

2.4. Temel Algoritma Öğeleri

Algoritma, bilgisayara hangi işlemi hangi sırada yapması gerektiğini söyleyen yönergeler bütünüdür. Programcı, oluşturulan algoritmadan hareketle grafiksel gösterimleri yani akış şemalarını oluşturabilir.

Algoritmaların temel öğeleri; tanımlayıcı, değişken, sabit, atama ve döngü gibi terimlerdir (Vatansever, 2009).

Tanımlayıcı; programı yazan geliştirici tarafından oluşturulan ve programdaki sabitleri, değişkenleri, kayıt alanlarını, özel bilgi tiplerini vb. adlandırmak için kullanılan kelimelerdir. Tanımlayıcılar oluşturulurken aşağıdaki kurallara uyulması gerekmektedir.

- Alfabedeki harflerden a-z ve A-Z arası kullanılabilir.
- Rakamlar kullanılabilir.
- Simgelerden alt çizgi (_) kullanılabilir.
- İsimlendirme, harf veya alt çizgi ile başlayabilir.
- İsimlendirmede, kullanılan programlama diline ait bir komut veya rezerve kelimeler kullanılmamalıdır.

BÖLÜM 2

Değişken; program çalıştırıldığında, farklı değerler alabilen veya aktarılabilen bilgi alanlarıdır. Değişkenler, programcının isteğine bağlı olarak isimlendirilir. Bunu yaparken tanımlayıcı kurallarına uyulması gerekir.

Sabit; programda değeri değişmeyen tanımlamalardır. Tanımlama kurallarına uygun olarak oluşturulan sabitlere, değer atanırken metinler tırnak içinde, sayılar ise doğrudan yazılır.

Atama; bilgi alanına veri yazma ve bir ifadenin sonucunu başka bir değişkene atamak için kullanılan operatördür.

Değişken = İfade

Yukarıdaki satırda ‘Değişken’ herhangi bir sabitin/değişkenin ismini, ‘İfade’ ise matematiksel veya metinsel değerleri temsil etmektedir. “=” sembolü atama operatörüdür ve sağdaki değeri soldaki değişkene atar.

Döngü; programlarda bazı işlemler belirli bir sayıda ve ardışık olarak gerçekleştirilmektedir. Döngüler, programlardaki tekrarlanan işlem bloklarını, belirtilen sayıda gerçekleştiren işlemdir.

Algoritma kullananlar (MEB, 2019):

- Problemleri daha hızlı ve sistematik olarak çözebilir.
- Problem çözme sürecini takip ederek nerede hata yaptığını görebilir.
- Tüm olasılıkları gözden geçirebilir.
- Hatalı işlem yapma olasılığını azaltabilir.
- Olası hataları düzeltebilir.
- Çözüme ulaşmak için farklı yolları deneyebilir.
- Problemin çözümü için harcanan süreyi kısaltabilir.

Algoritma oluşturmak için aşağıdaki örnek problem durumları kullanılabilir:

1. Hava çok sıcak ve iyice bunaldınız, serinlemek için yapacaklarınızı adım adım yazınız.
2. Taşınabilir belleği bilgisayarınıza taktınız fakat çalışmadığını fark ettiniz. Çalışır hâle getirmek için neler yapacağınızı adım adım yazınız.
3. Bir hafta sonra teslim edilmesi gereken bir rapor hazırlamanız gerekiyor ve bu rapor için ihtiyaç duyduğunuz 2 kitap elinizde yok. Kitapları en kısa sürede temin etmek için yapacaklarınızı adım adım yazınız.
4. Şekil 2.8’de verilen haritayı kullanarak farklı adres tariflerini yazınız.

BÖLÜM 2



Şekil 2.8. Adres tarifleri için kullanılacak harita

BÖLÜM 2

2.5. Akış Şemaları Oluşturma

Akış şemaları algoritmaların görsel gösterimi yani temsildir. Akış şemaları, program geliştirmeye başlamadan önceki son adımdır. Akış şemaları ile hatalar rahatlıkla görülüp düzeltilebilir. Akış şemalarını oluşturmak için kullanılan simgeler ve anlamları Tablo 2.1’de verilmiştir.

Tablo 2.1. Akış şemasında kullanılan simgeler ve görevleri

SİMGE	İŞLEV
	Başla / Bitir: Algoritmanın başladığı ve bittiği konumu gösterir.
	Giriş / Okuma: Klavye aracılığıyla veri girişi ya da okuma yapılacağını gösterir.
	Atama / İşlem: Aritmetik işlem yapımında ve değişkenlere değer atamada kullanılır.
	Denetim (Karar): Algoritmada karar verme işleminde kullanılır.
	Döngü: Birden çok tekrar gerektiren durumlarda kullanılır.
	Çıkış: Hangi bilginin çıktısının alınacağını gösterir.
	Akış Yönü: Şekiller arasındaki akış yönünü gösterir.
	Bağlaç: Bu sembol, akış şemalarında akışın bir noktadan diğerine zıplamasını göstermek için ve ana programdan ayrı bir alanda tanımlanan alt süreçlerden atlamak için de kullanılır.
	Önceden Tanımlı İşlem / Fonksiyon: Bir alt program algoritmasını gösterir.

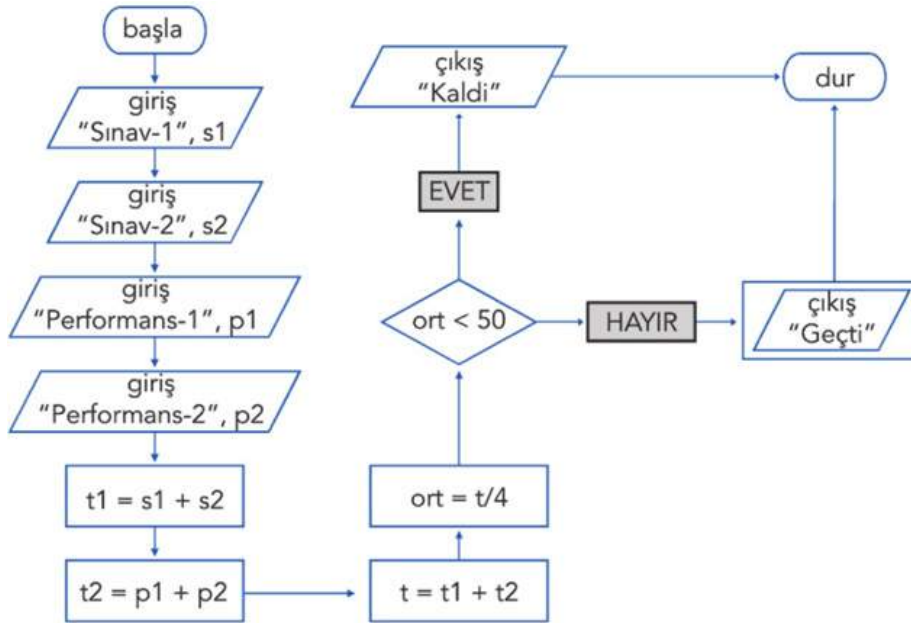
Kaynak: Akış Şemaları Sembolleri. MEB, 2019.

Akış şeması, bir problem çözümünün başlangıcından bitişine kadar olan süreci gösterir. Akış şemasında kullanılan her bir sembol, algoritmadaki bir işlemi ifade eder. Genellikle işlemler tek yönlü olmasına rağmen karar kutularından iki farklı ok çıkar. Bir karar simgesinden çıkan ok, bazı işlemlerin tekrarlanmasını sağlayabilir; böylece bir "döngü" oluşur.

Akış şemalarını oluştururken dikkat edilmesi gereken bazı noktalar şunlardır:

1. Yönergeler, simgeler içine yazılmalıdır.
2. Hatırlatıcı bilgiler, simgenin yanına yazılabilir. Böylece akış şeması, ek açıklamalı bir şemaya dönüşür.
3. Bir akış şeması, her zaman sayfanın başından başlar ve sonuna doğru gider. Eğer bir sayfaya sığmazsa bir ya da daha fazla bağlantı simgesi kullanılarak diğer sayfaya geçilebilir.
4. Akış şemasını çizmek için uygun yazılımlar kullanılırsa daha standart bir görünüm elde edilir (Flowchart vb.).
5. Simgeler, içeriğindeki yazının rahatça okunabileceği kadar büyük olmalıdır.

Örnek: Şekil 2.9'da iki yazılı ve iki performans puanı almış bir öğrencinin puan ortalamasını hesaplayarak dersten geçip geçmediğini belirleyen akış şeması görülmektedir.

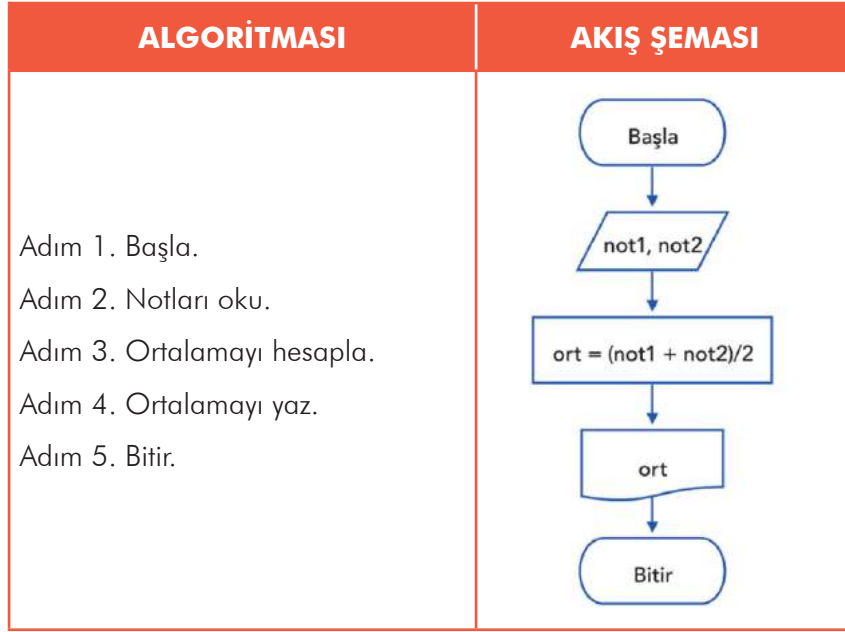


Şekil 2.9. İki yazılı ve iki performans puanı almış bir öğrencinin puan ortalamasını hesaplayarak dersten geçip geçmediğini belirleyen akış şeması

BÖLÜM 2

2.5.1. Sıralı Akış Yapısı

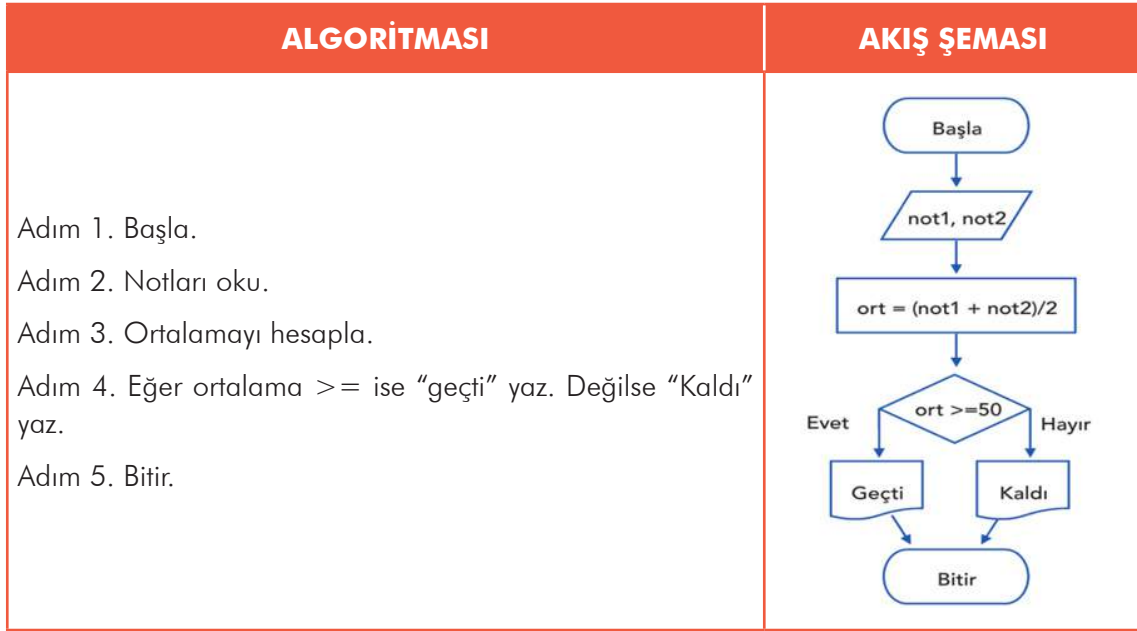
Sıralı akış yapısı, doğrusal yönde çalışan akış şemalarıdır. Bilgisayar, birbiri ardına algoritmanın başından sonuna kadar sırası ile işlenmesi gereken komutları verir. Şekil 2.10'da klavyeden girilen iki sınav puanının aritmetik ortalamasını hesaplayan algoritma ve akış şeması görülmektedir.



Şekil 2.10. İki sınav puanının aritmetik ortalamasını hesaplayan algoritma ve akış şeması

2.5.2. Koşullu Akış Yapısı

Koşullu akış şeması, iki olasılıktan birini seçmek ve devam etmek için kullanılan akış şemalarıdır. Şekil 2.11’de, klavyeden girilen iki sınav puanının ortalamasını bularak öğrencinin dersten geçip geçmediğini kontrol eden algoritma ve akış şeması görülmektedir.

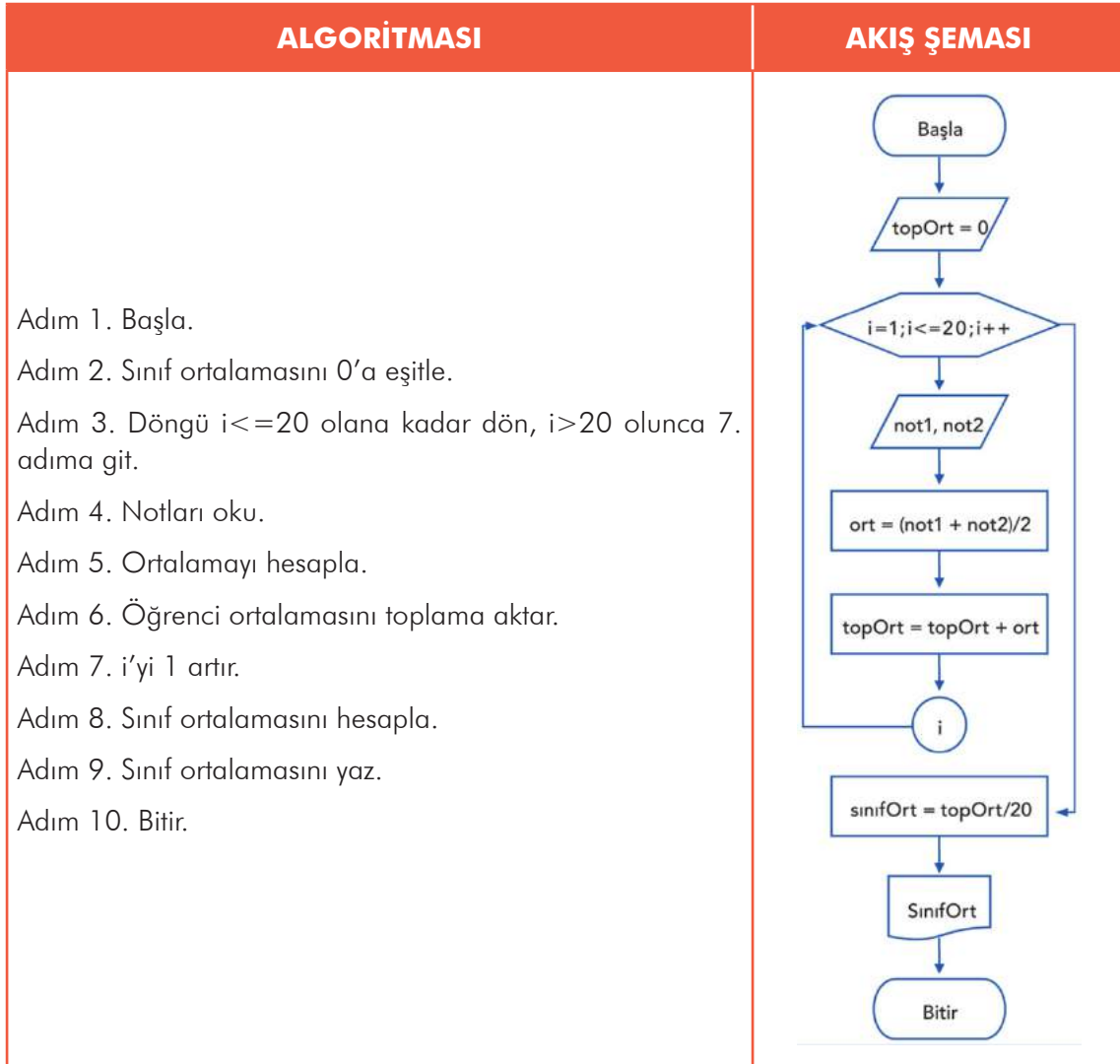


Şekil 2.11. Ders geçmeyi kontrol eden algoritma ve akış şeması

BÖLÜM 2

2.5.3. Tekrarlı Akış Yapısı

Tekrarlı akış yapıları yani döngüsel yapı, bir dizi işlemi tekrarlamak için kullanılan akış şemalarıdır. Şekil 2.12’de, 20 öğrencilik bir sınıfın bir dersten aldığı iki not üzerine sınıf ortalamasını hesaplayan algoritma ve akış şeması görülmektedir.



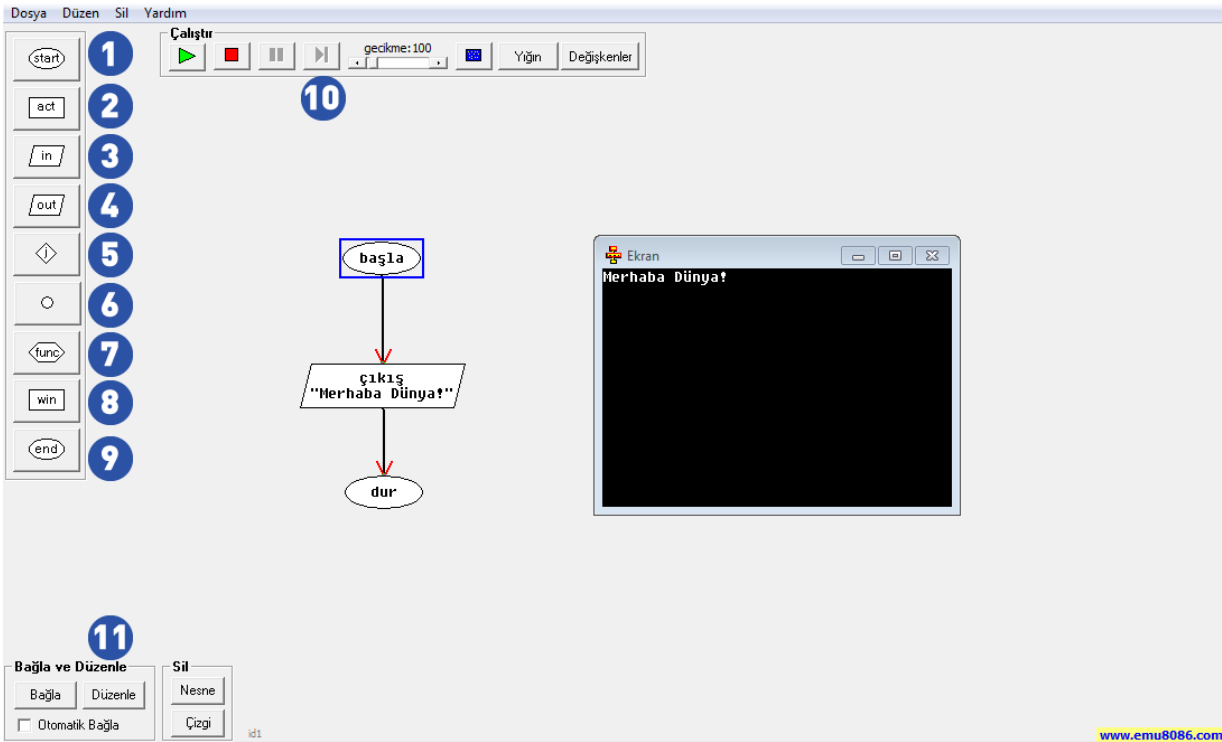
Şekil 2.12. Sınıf ortalamasını hesaplayan algoritma ve akış şeması

2.6. Uygulama Programları

Akış diyagramlarının öğretiminde kullanılabilecek birçok bilgisayar yazılımı vardır. Bu uygulamalar, ağırlıklı olarak 'sürükle bırak' tekniği ile akış diyagramlarının oluşturulmasını sağlar. Oluşturulan akış şemaları, bütünleşik olarak bu yazılımlar üzerinden çalıştırılarak sonuçları gösterir.

2.6.1. Flow Chart Visual Programming

Akış şemalarını hazırlamak için ülkemizde yaygın olarak kullanılan uygulamalardan biri de **Flow Chart Visual Programming**'dir. Programda ara yüzünden görsel olarak akış şeması kolaylıkla oluşturulabilir ve ilgili algoritmanın çalışıp çalışmadığı görülebilir. Açık kaynak bir yazılım olan bu programa internetten, arama motorları üzerinden kolaylıkla ulaşılabilir. Sade bir ara yüzü bulunan programın dosya menüsü altında öğrenimi kolaylaştıracak birçok örnek uygulama da mevcuttur.



Şekil 2.13. Flow Chart Visual Programming arayüzü

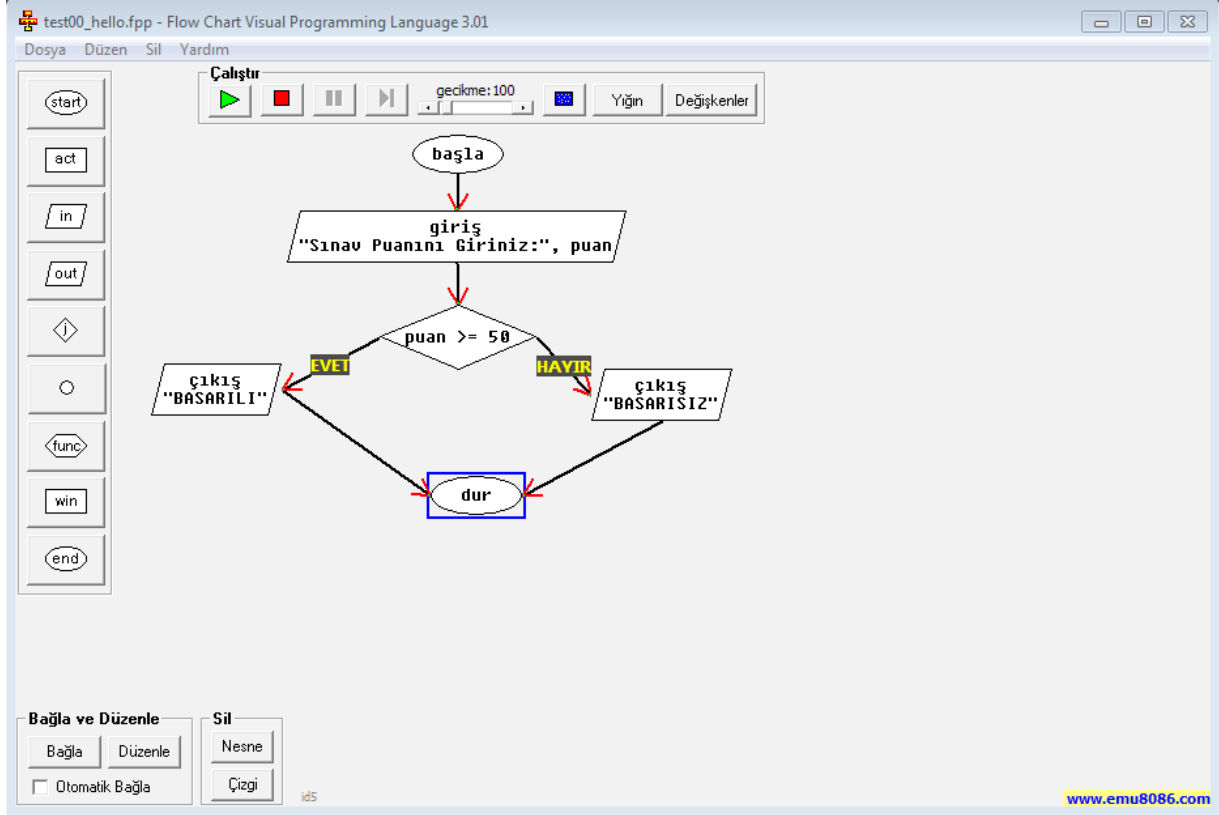
BÖLÜM 2

Dosya menüsü altından “Yeni” komutu ile proje oluşturulduktan sonra çalışma alanının solundaki bileşenler kullanılarak akış şemaları oluşturulur. Şekil 2.13’te numaralar ile gösterilen bu bileşenlerin kullanım amaçları şu şekildedir:

- 1. Başla (Start):** Bu bileşen, akış şemasının başlangıcında yer alır. Bütün akış şemaları bununla başlar.
- 2. İşlem (Action):** Bu bileşen, işlem yapmak için kullanılır. Değişken tanımlamaları, aritmetik ve metin işlemleri bu işlem kutusu içinde yapılır.
- 3. Giriş (Input):** Bu bileşen, kullanıcıdan veri almak için kullanılır. Program, bu adıma geldiğinde veri girişi yapılması için bekler.
- 4. Çıkış (Output):** Bu bileşen, sonuçların yazdırılması ve ekranda görüntülenmesi için kullanılır.
- 5. Şart (Junction):** Bu bileşen, “eğer” sorgu yapıları için kullanılır.
- 6. Bağlantı (Connection):** Bu bileşen, diğer bileşenleri birbirine bağlamak için kullanılır.
- 7. Fonksiyon (Function):** Bu bileşen, fonksiyonların akış şemasına dâhil edilmesinde kullanılır. Fonksiyonlar; kendilerine tanımlanmış belirli bir görevi yapan küçük algoritma parçalarıdır. Tek başlarına bir algoritma gibi çalışmazlar ama bir algoritma içerisinde görevlerini yerine getirirler.
- 8. Pencere (Window):** Bu bileşen, görsel program arayüzü için kullanılır.
- 9. Dur (End):** Bu bileşen, algoritmayı sonlandırmak için kullanılır.
- 10. Çalıştır:** Bu bölüm, akış şeması tasarımı tamamlandıktan sonra algoritmanın çalışıp çalışmadığını denetlemek için kullanılır. Algoritmayı, akış şeması üzerinden adım adım ve istenilen hızda çalıştırmak için kullanılır.
- 11. Bağla ve Düzenle:** Bu bölümdeki bileşenler, akış şeması üzerindeki nesnelere birbirine bağlamak, düzenlemek ve silmek için kullanılır. Bileşenler, çizgiler ile birbirine bağlanır. Ayrıca bileşenler arası çizgileri ve bileşenleri silmek için de bu bölüm kullanılır. Silinecek bileşen, fare ile seçildikten sonra “Sil” alanı altındaki “Nesne” butonuna tıklanarak silinebilir. Çizgi silmek için hangi iki bileşen arasındaki çizgi silinecekse o bileşenler seçilir ve “Çizgi” butonuna tıklanır.

Örnek Uygulama

Kullanıcının girdiği sınav puanı 50'den düşükse "BAŞARISIZ", 50'ye eşit ve 50'nin üstünde ise "BAŞARILI" mesajını ekrana yazdıran programın akış şeması aşağıdaki şekilde tasarlanabilir.



Şekil 2.14. Flow Chart Visual Programming ile yapılmış örnek bir uygulama

2.6.2. Diagrams

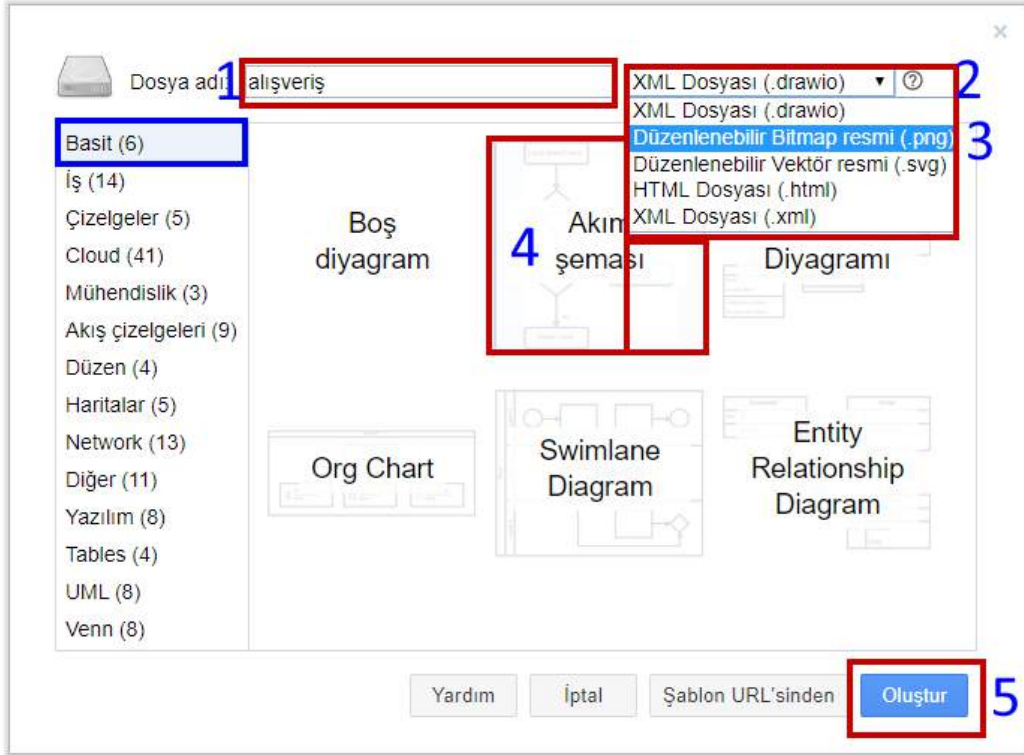
Akış şemalarını hazırlamak için kullanılan uygulamalardan biri de *Diagrams* uygulamasıdır. <https://app.diagrams.net/> adresine giriş yapıldığında *Device* ile yapılan çalışmaların bilgisayara mı veya Google Drive'a mı kaydedileceği seçilir. *Language* ile uygulama dili Türkçe olarak seçilebilir.

BÖLÜM 2

NOT

Ayrıca uygulamayı bilgisayara indirmek ve yüklemek için <https://github.com/jgraph/drawio-desktop/releases/tag/v13.2.4> adresinden işletim sistemine uygun sürümü indirilebilir. Masaüstü sürümü de benzer şekilde çalıştırılabilir.

Yeni bir proje açtığınızda Şekil 2.15'teki *Dosya Adı* yazan diyagrama isim verilir (1). Yapılan projenin kaydedilme uzantısı Şekil 2.15'teki 2 ve 3 ile ifade edilen yerlerden seçilebilir. Uygulama içinde birçok hazır şablon mevcuttur. Bunlar, yapılan projeye göre gruplandırılmıştır. Bunlardan projenize uygun olanı seçerek diyagramınızı tasarlayabilirsiniz.

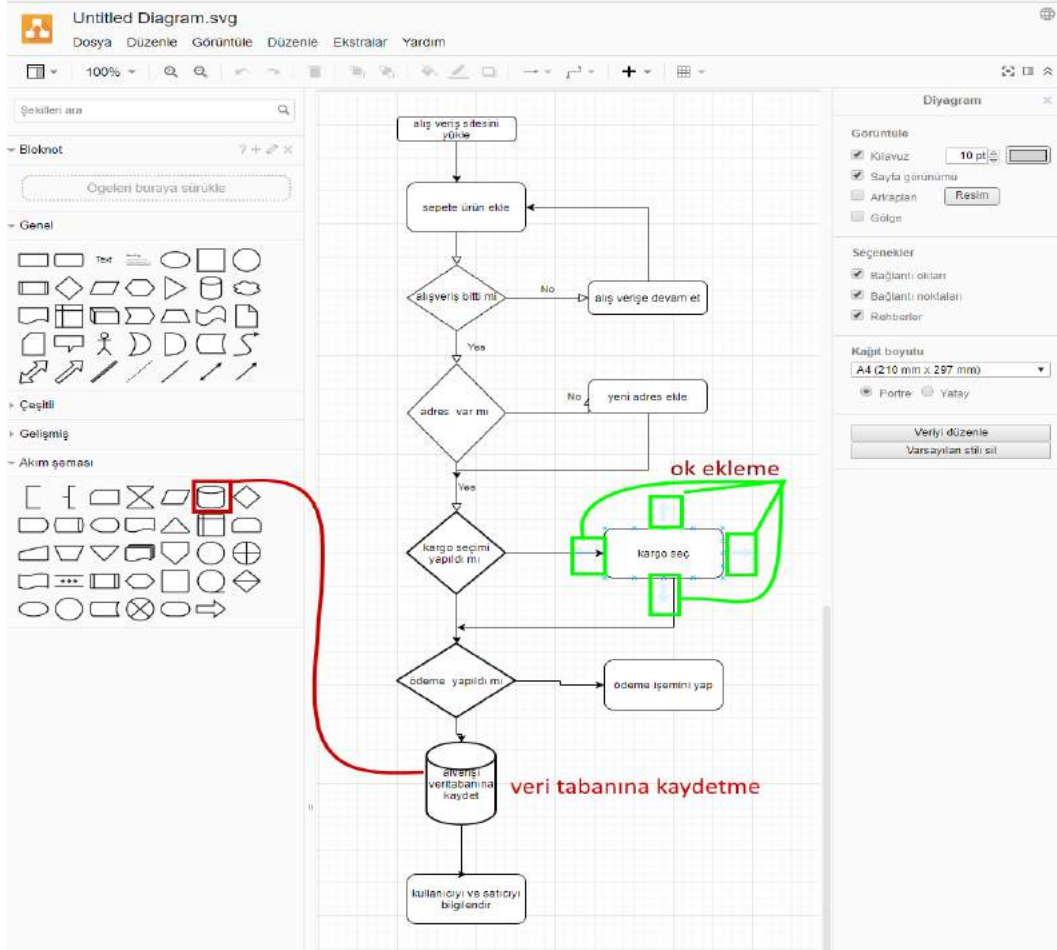


Şekil 2.15. Diagrams proje oluşturma ekranı

Diyagram oluşturma ekranında metinlere tıklayarak isimler değiştirilebilir, akış şeması sekmesinden karar yapısı ekleyebilirsiniz. Veri tabanı eklemek için akım şemasından ilgili görseli seçerek ekleyebilirsiniz. Nesne eklendikten sonra üzerine tıkladığında çıkan oklarla diğer nesnelere bağlantılarını yapılabiliyorsunuz.

Örnek Uygulama

Bir alışveriş sitesinin sipariş süreci, temel hatları ile akış şeması olarak düzenlenmiştir.



Şekil 2.16. Örnek Uygulama için akış şeması

2.7. Algoritma ve Akış Diyagramı Etkinlikleri

Günlük işler, sıradan gibi gelen işlerin belirli bir sırada yapılmasıdır. Her yapılan işin içerisinde adım adım takip edilen bir sıra vardır. Sıralar karıştırıldığında geri dönülerek uyulması gereken sıraya göre işlemler tekrar edilir.

BÖLÜM 2

1. ETKİNLİK: EŞİT ORANLI ÖRÜNTÜLER

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Algoritma oluşturmayı anlar.

Sıralı algoritmaları uygular.

Algoritmaların matematik ile ilişkisini anlar.

Algoritmaların şart yapılarını kullanır.

Akış şemalarını uygular.

YÖNTEM/TEKNİKLER

Gösterip yaptırma

ÖĞRETİM MATERYALLERİ

Bilgisayar ve yazı tahtası

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- 1 3 5 ? örüntüsünde sıradaki sayı ne olmalıdır?
- Bu sayıyı nasıl buldunuz, ayrıntılarıyla anlatabilir misiniz?
- Bu sayılara bir harf verilse nasıl tarif edersiniz?
- $a=1$ $b=3$ $c=5$

Adımlar	1	2	3	4
İşlemler	$k = a - b$	$m = c - b$	$K = M$ diye	$D = Z + M$

– Yukarıdaki tabloda bu işlemin nasıl çözüldüğü harfler ile gösterilmiştir.

Problem cümlesi: Kullanıcının girdiği üç sayının arasında eşit sayıda örüntü var mı, diye bakıp var ise 4. sayıyı bulan algoritmayı yazarak akış şemasını çiziniz.

SÜREÇ

Problemin algoritması:

Adım 1. Başla.

Adım 2. "1. sayıyı giriniz.", a

Adım 3. "2. sayıyı giriniz.", b

Adım 4. "3. sayıyı giriniz.", c

Adım 5. $k = a - b$

Adım 6. $m = c - b$

Adım 7. $k = m$ ise adım 8'e git, değilse 9'a git.

Adım 8. $d = c + k$ adım 10. adıma git.

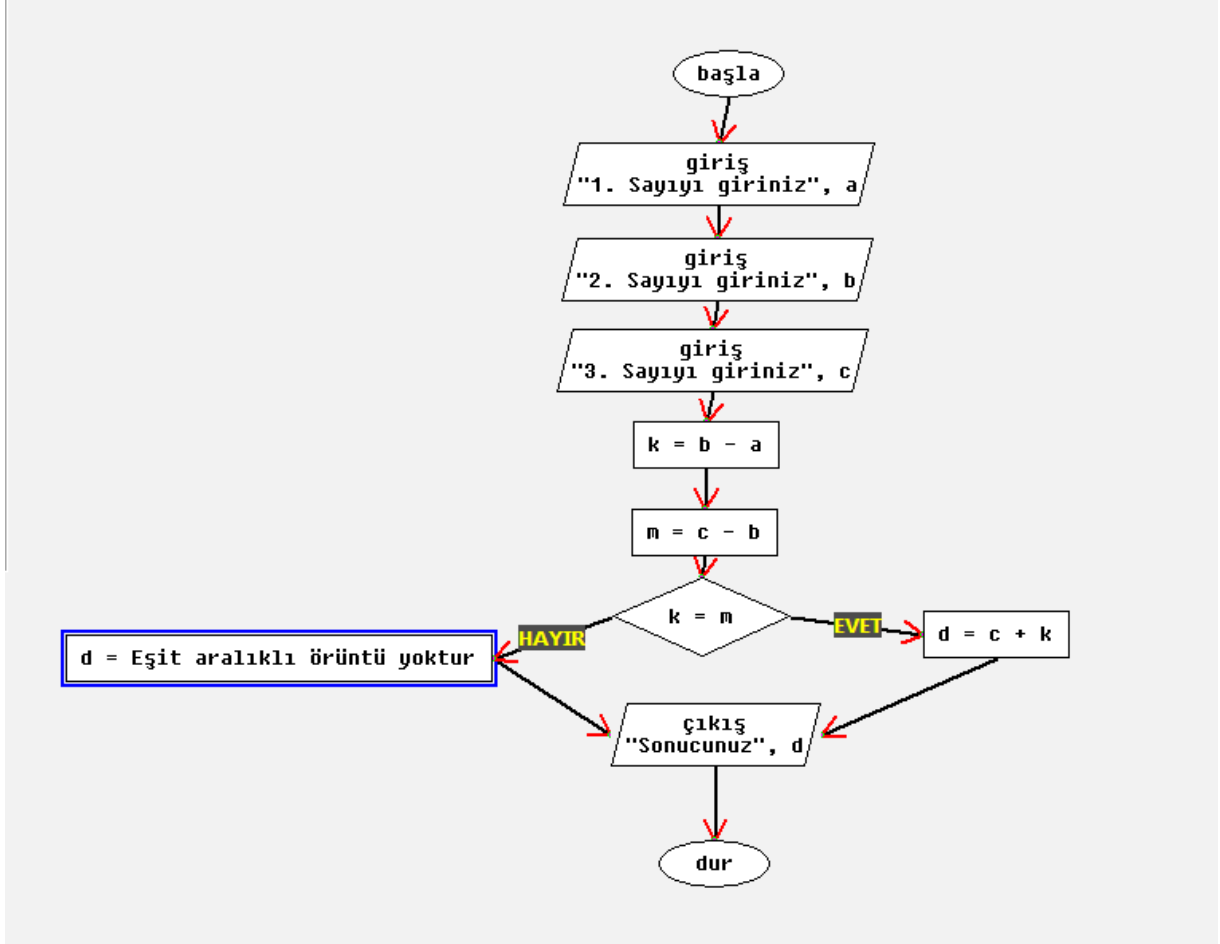
Adım 9. $d =$ "Eşit aralıklı örüntü yoktur." 11. adıma git.

Adım 10. "Sonuç", d

Adım 11. Bitir.

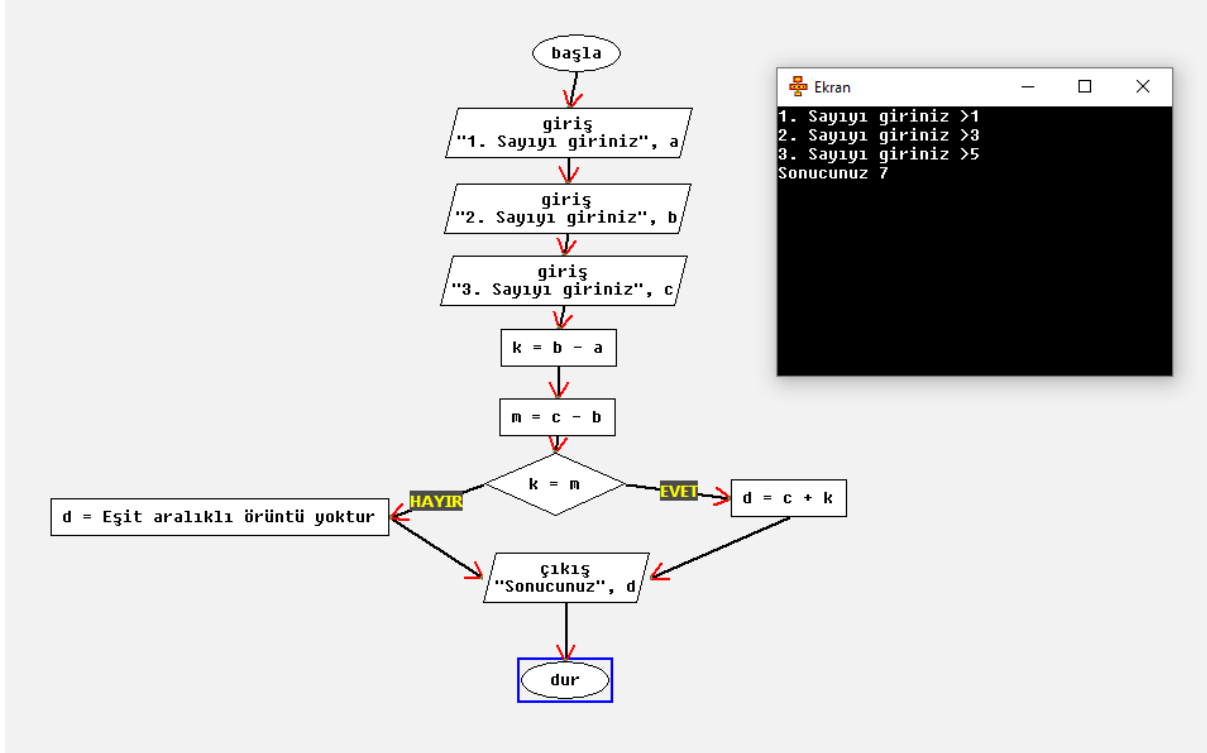
- Bu uygulamanın akış şemasını Flowchart programı veya el ile çizebilirsiniz.

BÖLÜM 2



Şekil 2.17. Örüntü akış şeması

Şekil 2.17’de Eşit aralıklı örüntüyü çözebilen programın akış şeması çizilmiştir.

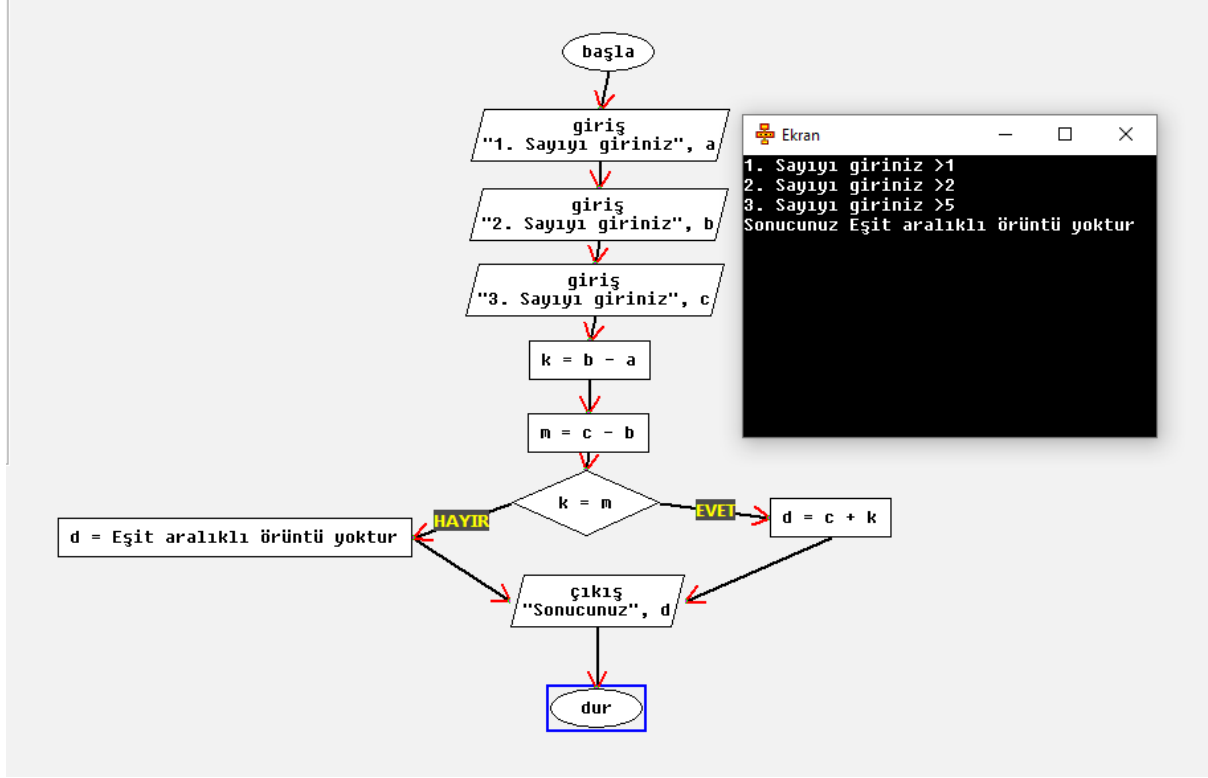


Şekil 2.18. Matematiksel örüntüler algoritmasını çalıştırma

Şekil 2.18'de görüldüğü gibi program eşit aralık matematiksel örüntüleri tanımlayarak bir sonraki sayının ne olacağını veriyor.

Programı çalıştırmak için 1, 2, 5 gibi bir sayı girerek yeni sonuçları görebilirsiniz.

BÖLÜM 2



Şekil 2.19. Matematiksel Örüntüler Çalıştırma 2

Şekil 2.19’da görüldüğü gibi 1, 2 ve 5 arasında eşit oranlı matematiksel örüntüler olmadığını algoritması ile fark ederek “Eşit aralıklı örüntü yoktur.” diye yazıyor.

ÖLÇME VE DEĞERLENDİRME

- Algoritma nedir?
- Sıralama algoritmalar günlük yaşamımızda kullanılır mı?

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Algoritma oluşturma			
Algoritmada sıralı işlemleri bilme			
Günlük yaşamda algoritmaları örnek verme			
Matematikte örüntü kavramını anlama			

BÖLÜM 2

2. ETKİNLİK: HANGİ LAMBA YANAR

BİLİŞSEL DÜZEY

Analiz

KAZANIM(LAR)

Bir problemin çözümü için algoritma geliştirir.

Bir problemin çözümü için akış şeması geliştirir.

Elektrik devrelerini anahtarların kullanımını öğrenir.

YÖNTEM/TEKNİKLER

Yaparak yaşayarak öğrenme

ÖĞRETİM MATERYALLERİ

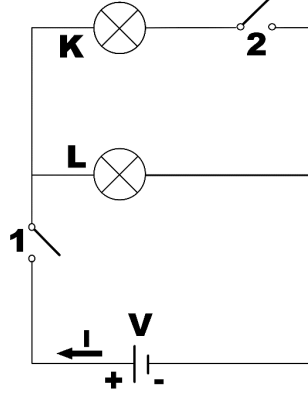
Lamba devresi çıktısı

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

“Lambaların ışık verebilmesi için neye ihtiyaç vardır?” sorusu sorulur. Sonrasında aşağıda verilen lamba devresinin çıktıları öğrencilere dağıtılarak 1 ve 2 numaralı anahtarların kapalı ve açık olma durumlarına göre K ve L lambalarının yanıp yanmayacağı konusunda öğrencilerin düşüncelerini sorunuz.

- “Anahtarların açık (0) ve kapalı (1) olma durumuna göre K ve L lambalarının yanıp yanmadığını ortaya koyan algoritma ve akış şeması nasıl hazırlanır?” problem cümlesine öğrencilerle birlikte cevap arayınız.



Şekil 2.20. Basit elektrik devresi

- 1 numaralı anahtar kapalı ve 2 numaralı anahtar açık ise hangi lamba/lar yanar? (Cevap: L lambası)
 - 1 numaralı ve 2 numaralı anahtarlar kapalı ise hangi lamba/lar yanar? (Cevap: K ve L lambaları)
 - 1 numaralı anahtar açık ve 2 numaralı anahtar kapalı ise hangi lamba/lar yanar? (Cevap: Lambalar yanmaz.)
 - 1 numaralı anahtar açık ve 2 numaralı anahtar açık ise hangi lamba/lar yanar? (Cevap: Lambalar yanmaz.)
- Anahtarların açık veya kapalı olma durumuna göre hangi lambaların yandığının tespitinde izlenen zihinsel süreçler katılımcılarla tartışılır.
 - Lambanın yanması için bulunduğu elektrik hattı üzerindeki anahtarın kapalı olması gerekmektedir. Elektrik kaynağının (V) + ucundan çıkan elektrik akımı (I) anahtar/ların kapatılması sayesinde lamba üzerinden geçerek kaynağın (V) – kutbuna ulaşıyorsa ilgili lamba/lar ışık vermektedir. Buradan hareketle problem çözme süreci şu şekildedir:
 - 1 numaralı buton kapalı mı?
 - Evet ise 2 numaralı buton kapalı mı?
 - Evet ise K ve L lambaları yanar.
 - Hayır ise L lambası yanar.
 - Hayır ise hiçbir lamba yanmaz. 2 numaralı butonun durumuna bakmaya gerek yoktur.
 - Nasıl bir çözüm süreci izlediğimizi belirledikten sonra problemin algoritmasının yazılmasına geçilir.

BÖLÜM 2

SÜREÇ

Problemin algoritması:

Adım 1. Başla.

Adım 2. "Anahtar1'in değeri 1 (Kapalı) mı 0 (Açık) mı?", anahtar1

Adım 3. "Anahtar2'nin değeri 1 mi 0 mı?", anahtar2

Adım 4. anahtar1 == 1 ise adım 5'e git değilse adım 8'e git.

Adım 5. Anahtar2 == 1 ise adım 6'ya git değilse adım 7'ye git.

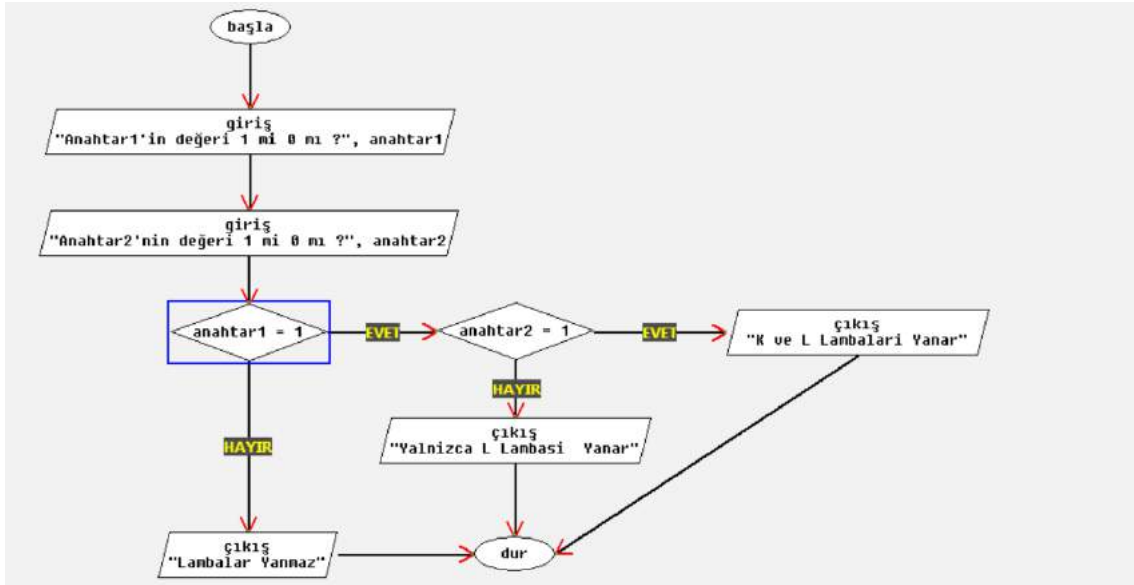
Adım 6. "Sonuç", "K ve L lambaları yanar" adım 9'a git.

Adım 7. "Sonuç", "L lambası yanar" adım 9'a git.

Adım 8. "Sonuç", "Lambalar yanmaz" adım 9'a git.

Adım 9. Bitir.

- Bu uygulamanın akış şemasını Flowchart programını kullanarak veya elle kâğıda çizebilirsiniz.



Şekil 2.21. Algoritmanın çizilmesi

- Başta verilen elektrik devresindeki anahtarların açık veya kapalı olma durumlarına bağlı olarak lambaların yanıp yanmama durumlarına yönelik algoritma ve akış şeması tanımlanmıştır.

ÖLÇME VE DEĞERLENDİRME

Günlük yaşamımızda gerçekleştirdiğimiz algoritmalardan örnek verebilir misiniz?

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Algoritma oluşturma			
Koşullu algoritmaları bilme			
Günlük yaşamda algoritmaların kullanıldığını anlama			
Anahtarların görevini bilme			
Basit elektrik devrelerini kontrol edebilme			

BÖLÜM 2

3. ETKİNLİK: RENKLİ ŞEKİLLER

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Algoritma oluşturmayı anlar.

Koşullu algoritmaları anlar.

Günlük yaşamdan sıralama algoritmalarına örnek verir.

Nesnelerin ikinci özelliklerini fark eder.

YÖNTEM/TEKNİKLER

Oyunlaştırma

ÖĞRETİM MATERYALLERİ

Renkli şekiller çıktısı

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- Sıralı algoritmaları öğrendik, peki başka algoritma türü var mıdır? Varsa sizce nasıl olabilir?
- Aileniz veya öğretmeniniz sizlere “Uslu durursanız oyuncak alırım.” benzeri cümleler söyledi mi?

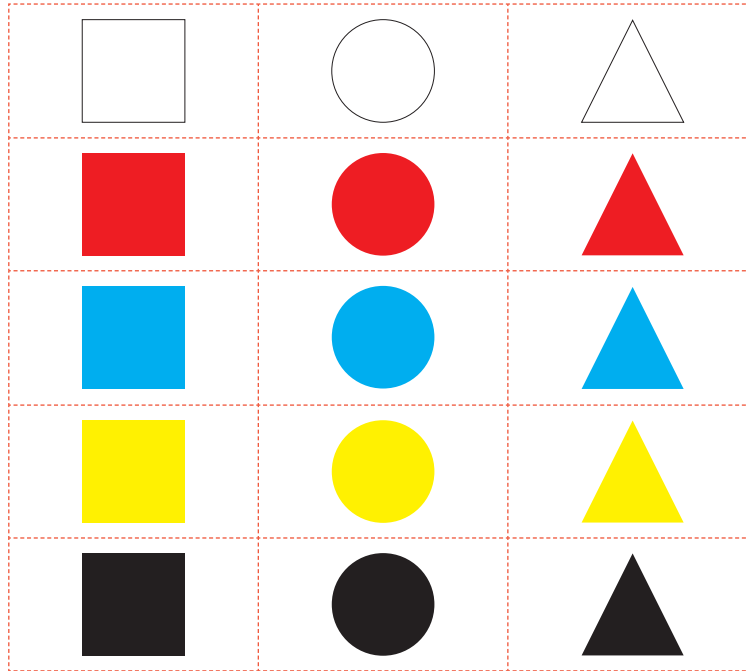
Öğrencilerinizden gelen cevapları dinleyerek sıralı algoritmaların tanımını tekrar hatırlatınız. Koşullu yapıları, günlük yaşam üzerinden örneklerle öğreneceğinizi söyleyiniz.

SÜREÇ

Öğrencileri üçerli gruplara ayırınız. Ayırdığınız bu gruplara, Şekil 2.22'deki platformun ve Şekil 2.23'te verilen geometrik şekillerin çıktılarını alarak dağıtınız. Dağıttığınız bu renkli şekilleri, yönergeler vererek öğrencilerin platform üzerine yerleştirmelerini sağlayınız.

KARE	ÜÇGEN	YUVARLAK

Şekil 2.22. Renkli şekil platformu (Satır sayısı isteğe bağlı olarak çoğaltılabilir.)



Şekil 2.23. Renkli geometrik şekiller

BÖLÜM 2

Elinizdeki şekilleri eğer kare ise sola, üçgen ise ortaya, yuvarlak ise sağa yerleştiriniz. (Sağ, sol ve orta yerine “ilk sırada, ikinci sırada ve üçüncü sırada” ifadelerini kullanabilirsiniz.). Renkli şekil çerçevesini kullanabilirsiniz.

Eğer kare ise
platformda sola bırak.

Eğer üçgen ise
platformda ortaya bırak.

Eğer yuvarlak ise
platformda sağa bırak.

Gruplardan, “mavi renkteki şekilleri sağa, değil ise sola” koymaları istenir.

Eğer mavi ise
platformda sola bırak.

Değilse
platformda sağa bırak.

Bütün şekilleri karıştırınız. Öğrencilerden birinin gözünü kapattırınız. Bu öğrenciye rastgele bir şekil seçtiriniz. Gözü açık olan öğrencilerden birinin gözü kapalı olan öğrenciyi kırmızı üçgeni buluncaya kadar yönlendirmesini sağlayınız.

Döngüye kırmızı üçgeni buluncaya kadar devam et.
Yeniden seç.

Öğrencilere siyah yuvarlak şekli seçip o şeklin hangi koşullu durumda seçileceği sorulur.

Eğer siyah ise ve yuvarlak ise koşullu

Öğrenci gruplarındaki bir kişiye bir şekil seçtirip diğer arkadaşlarının platformda hangi koşulla nereye yerleştirileceğini söylemelerini isteyiniz. Bu işlemi, dönüşümlü olarak gruptaki öğrencilerin hepsinin yapmasını sağlayınız.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI - 1

Öğrencilerinizin seçilen herhangi bir geometrik şekil için koşullu algoritma yazmalarını isteyiniz. Örnek olarak sarı ise sağdaki kareye, değilse soldaki kareye ayıran algoritma verilmiştir.

Adım 1: Başla.

Adım 2: Bir şekil al.

Adım 3: Sarı ise sola bırak, değil ise sağa bırak.

Adım 4: Bitir.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI - 2

Öğrencilere, Şekil 2.23'teki renkli geometrik şekilleri dağıtarak özelliklerine göre öğrencilerin Şekil 2.24'teki platforma yerleştirmelerini isteyiniz.

Siyah kare	Mavi üçgen	Beyaz yuvarlak	Siyah üçgen
Üçgen olmayan kare olmayan mavi olmayan kırmızı olmayan bir şekil	Siyah olmayan bir şekil	Mavi veya sarı bir şekil	Sarı veya mavi olan kare olmayan bir şekil

Şekil 2.24. Özelliklerine göre yeni platform

BÖLÜM 2

ÖLÇME VE DEĞERLENDİRME

Yaşantınızda karşılaşılabileceğiniz koşullu yapılara örnekler veriniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Algoritma oluşturma			
Koşullu algoritmaları bilme			
Temel geometrik şekilleri tanıma			
Renkleri tanımlayabilme			
Döngüleri bilme			

4. ETKİNLİK: FONKSİYON İLE İKİ SAYIYI TOPLAMA

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Algoritma oluşturmayı anlar.

Koşullu algoritmaları anlar.

Akış diyagramların Matematik ile ilişkisini anlar.

Akış diyagramlarında fonksiyon yapısını kullanır.

Fonksiyon oluşturma, çağırma işlemlerini yapar.

YÖNTEM/TEKNİKLER

Gösterip yaptırma

ÖĞRETİM MATERYALLERİ

Bilgisayar ve yazı tahtası

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

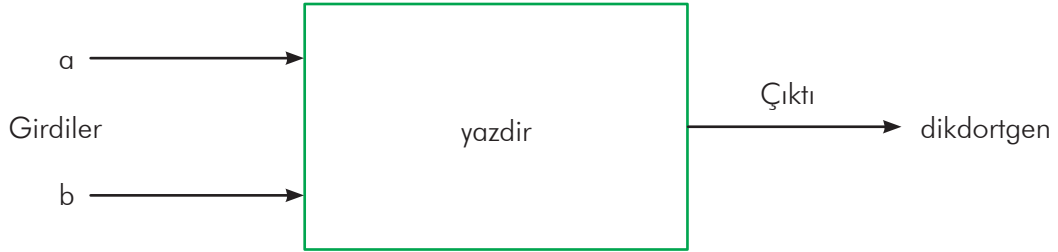
- Devamlı tekrarlamak zorunda kaldığınız işlerden sıkılıyor musunuz?
- Bu işlerin tekrarının olmaması için neler yapıyorsunuz?
- Örnek plastik malzemeden bir ürüne ait devamlı aynı çizim yapıp çıktı alınmaz. Sadece o ürünün kalıbı çıkartılarak kalıp üzerinden birçok çıktı alınabilir.
- Bilgisayar programlarının yazımında Türkçe karakterler kullanılmadığını anlatınız.

BÖLÜM 2

SÜREÇ

Dikdörtgenin kısa ve uzun kenarı girilerek fonksiyon yardımıyla alanın bulunması için aşağıdaki yönergeleri izleyiniz.

- ✓ Fonksiyon tipi : integer
- ✓ Fonksiyon adı : yazdir
- ✓ Parametreler : a ve b
- ✓ Geri dönüş değeri : dikdortgen



Şekil 2.25. Bir fonksiyonun örnek çalışma prensibi

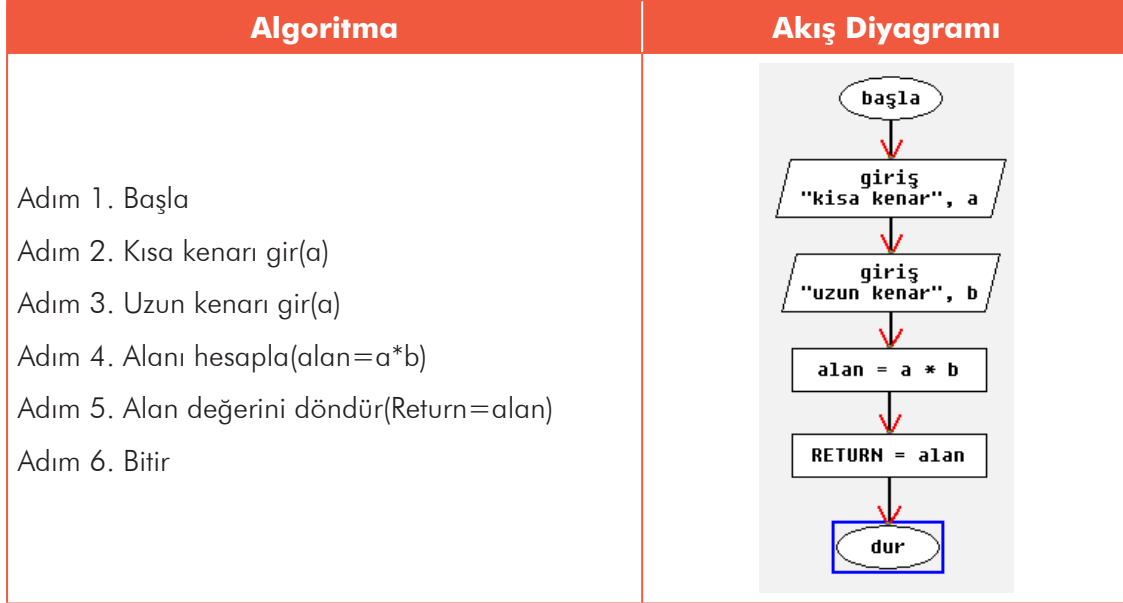
- Etkinlik için Flow Chart programında iki farklı dosya oluşturulacaktır. Bunlar “ana.fpp” ve “yazdir.fpp” dosyalarıdır.

ana.fpp dosyası için;

Algoritma	Akış Diyagramı
<p>Adım 1. Başla</p> <p>Adım 2. yazdir fonksiyonunu a ve b parametreleri ile çağır ve dikdortgen değişkenine ata.</p> <p>Adım 3. dikdortgen değişkenini ekrana yaz</p> <p>Adım 4. Bitir</p>	

Şekil 2.26. ana.fpp dosyası için algoritma ve akış diyagramı

yazdir.fpp dosyası için;

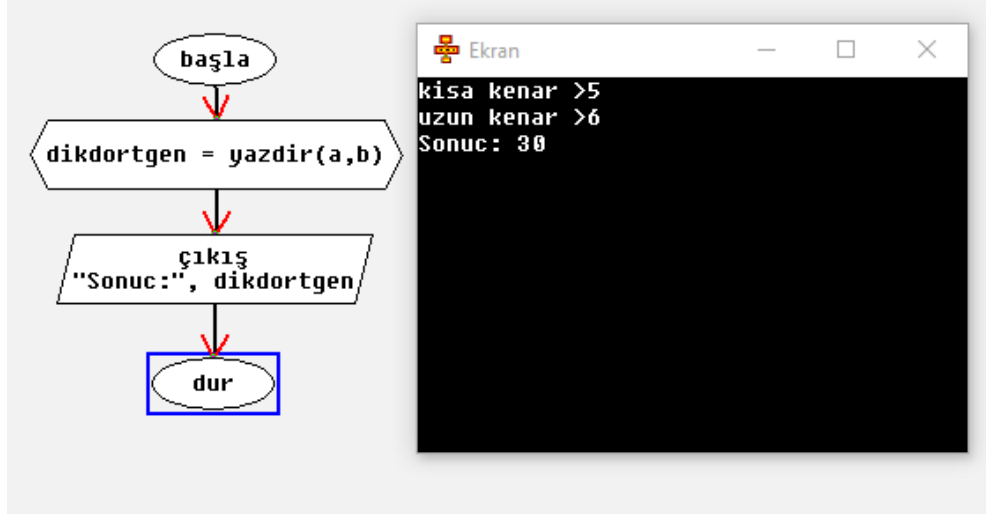


Şekil 2.27. yazdir.fpp dosyası akış diyagramı

ana.fpp dosyası çalıştırıldığında yazdir fonksiyonuna gider. Yani yazdir.fpp dosyasını çalıştırır. yazdir.fpp dosyasında kullanıcıdan dikdörtgenin alanın hesaplaması için kısa ve uzun kenarın girilmesi ister. Kenarlar girildikten sonra alanı hesaplayarak tekrar alan değişkenini ana.fpp dosyasına döndürür. Bu döndürülen değer dikdortgen değişkenine atanır. Devamında dikdortgen değişkeni ekrana yazdırılarak program bitirilir.

BÖLÜM 2

Ekran Çıktısı



Şekil 2.28. Dikdörtgenin alanının fonksiyon ile hesaplanması

ÖLÇME VE DEĞERLENDİRME

Bir karenin alanını hesaplayınız.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Algoritma oluşturma			
Algoritmalarda kullanıcıya değer girdirme			
Dikdörtgen veya karenin alanını hesaplamayı bilme			
Bir problemin farklı çözümlerini ortaya koyabilme			

5. ETKİNLİK: ARKADAŞIMIN YAŞINI BİLİYORUM

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Problemin çözümü için bir algoritma geliştirir.

Karar yapısını ve işlevlerini açıklar.

Karar yapıları içeren algoritmalar geliştirir.

Bir algoritmayı test eder ve değerlendirir.

YÖNTEM/TEKNİKLER

Grupla çalışma

ÖĞRETİM MATERYALLERİ

Makas, kalem, A4 kâğıt

GİRİŞ

Derse aşağıdaki sorular ile giriş yapılır:

- Yakın tarihte doğum günü olan kim var?
- Sizce arkadaşınız kaç yaşına giriyordur?

soruların cevaplarını öğrencilerinizin tahmin etmelerini sağlayınız.

Öğrencileriniz yaşıt oldukları için doğru tahminde bulunacaklardır. Onlara tahminlerinin doğru olduğunu belirtiniz. Hiç tanımadıkları birinin yaşını nasıl tahmin edebilecekleri hakkında öğrencilerinizin fikir yürütmelerini sağlayınız.

Bilgisayarların böyle tahminler yapmadıklarını, bunun için sıralı işlemler kullandıklarını anlatınız.

BÖLÜM 2

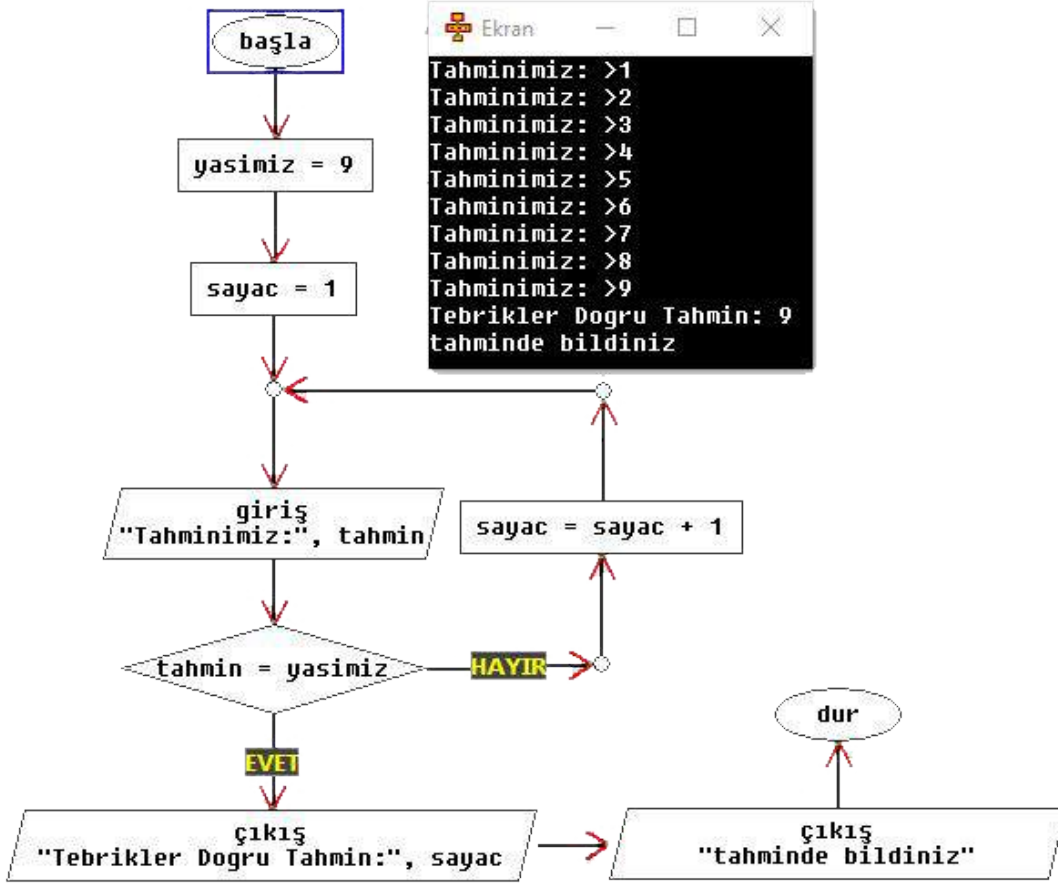
SÜREÇ

- Sınıftaki öğrenciler ile ikiyeşerli gruplar oluşturunuz.
- Her öğrenciden 1’den 10’a kadar küçük kâğıt parçalarına yaşlarını yazmalarını isteyiniz.
- Gruptaki bir öğrencinin, diğer öğrencinin elindeki yaşı ifade eden kâğıttaki rakamı 1’den başlayarak bulmaya çalışmasını isteyiniz. Bunun için Tablo 2.2’deki yaş bulma sorularının onlara yardımcı olacağını, doğru tahminde bulunup 2. öğrenci “evet” cevabını verene kadar devam edilmesi gerektiğini anlatınız.
- Bu tür tahminlerde küçük rakamlarda program yazarken Tablo 2.2’nin yardımcı olacağını ama çok büyük rakamların tahmininde bu kadar çok satır yazmanın yorucu ve zaman kaybı olduğunu belirterek Şekil 2.29’daki akış şemasını adım adım anlatınız.

Tablo 2.2. Yaş bulma soruları

1-Öğrenci	2- Öğrenci	Sonuç	Tahmin Sayısı
1 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	1
2 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	2
3 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	3
4 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	4
5 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	5
6 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	6
7 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	7
8 yaşında mısın?	HAYIR	SORMAYA DEVAM ET.	8
9 yaşında mısın?	EVET	BİLDİNİZ TEBRİKLER!	9

- Gruplardaki öğrencilerin yerlerini değiştirerek etkinliği devam ettiriniz.



Şekil 2.29. Yaş bulma akış diyagramı

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

1. Etkinliği, öğrencilerin ayın hangi gününde doğduklarını bulmaları için 1'den 30'a kadar sorarak da yaptırabilirsiniz.
2. Bir ismin içindeki harfi bulma etkinliği olarak da düzenleyebilirsiniz.

BÖLÜM 2

ÖLÇME VE DEĞERLENDİRME

Öğrencilere “Bilgisayarın çalışması esnasında istediğimiz bilgilere nasıl bir yöntem ile çözüme gittiğini kavrayabildiniz mi?” diye sorarak gelen yanıtları değerlendiriniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Arkadaşının yaşını bulmak için bir algoritma geliştirme			
Karar yapısını ve işlevlerini açıklama (Günlük hayattan örnek olabilir.)			
Örnek problemde karar yapısını kullanan bir algoritma geliştirme (Tahmin, arkadaşının yaşından farklıysa koşul cümlesinin eklenmesi)			
Geliştirdiği algoritmayı test etme ve değerlendirme (Arkadaşına sorular sorarak yaşını bulabiliyor mu? Daha kısa çözümler olabilir mi?)			

BÖLÜM

3

BİLGİSAYARSIZ KODLAMA

Bilgisayar; insanoğlunun hayatını kolaylaştıran, bilgileri saklayan, istenildiğinde bilgileri hızlı bir şekilde sunabilen elektronik bir cihazdır. İnsanoğlunun hayatını kolaylaştırmak, problemler karşısında çözüm üretmekle başlar. İnsanın varoluşuyla birlikte sorunların çözümüne yönelik arayışları da daima olmuştur. Milattan önceki dönemlerden beri insanoğlu, hayatını kolaylaştıracak yenilik ve gelişmeler için çaba göstermiş ve bunun için birçok girişimde bulunmuştur. Kesici ve kazıyıcı aletlerle kayalara semboller çizmiş, elindeki bilgileri sonraki zamanlara aktarmışlardır. Bu bilgileri, sonraki nesiller kullanmış ve kendi aralarındaki iletişimi bu sayede geliştirmişlerdir. Ateş ile oluşan dumanı işaretlere dönüştürerek kendi aralarında haberleşme yöntemlerini çeşitlendirmişlerdir.

Günümüze kadar yapılan birçok yenilik, bilgi ve beceri sonraki nesillere aktarılmaya çalışılmıştır. İnsanoğlunun nüfusu arttıkça hayatlarını kolaylaştıracak ulaşım araçları yapmışlar, sahip oldukları malları başka ihtiyaçları doğrultusunda değiştirmek için parayı kullanmaya başlamışlar ve paraya bir anlam yükleyerek değiştirme aracı hâline dönüştürmüşlerdir. Zamanla değişen ihtiyaçlar ve sorunlar sebebiyle birçok sembol sistemi insanoğlunun hayatına yerleşmiştir. Kısaca insanoğlu, hayatını kolaylaştıracak birçok sembolü belirli kurallar çerçevesi içinde ve bu sembolleri geliştirerek problemlerini çözmeye çalışmıştır.

Hayatta karşılaşılan problemler ya da ihtiyaçlara çözüm bulma becerilerinin artırılmasına yönelik öğrenme ve gelişme ihtiyacı hayatımızı daha da kolaylaştıracaktır. Daha önce öğrenilen bilgi ve beceriler, birçok yeniliğe ve gelişmeye ışık tutacaktır.

Bireyin yaşam döngüsü içinde ihtiyaçları her zaman var olacak ve birey, bu ihtiyaçlar karşısında çözümler üretmek isteyecektir. Düşünme becerilerini geliştirmek için öğrenme sürecinde olan bireyin geçmişten gelen bilgi birikimleri yanında problem karşısında çözüm odaklı düşünme becerilerini de geliştirmesi gerekir. Öğrenme sürecindeki bireyin oyun ile kavramları keşfetmesi ve süreç içinde hedeflenen amaçlara ulaşması beklenir.

BÖLÜM 3

Bilgisayarsız kodlama eğitimi, bilgisayar bilimini oyunlaştırma etkinlikleri ile çevremizde bulunan materyalleri (kartlar, ip, bulamaca vb) kullanarak birey üzerinde merak, ilgi ve motivasyonun artırılması sonucunda bilgisayar bilimini öğretmeye çalışır.

Bilgisayarsız kodlamada yapılacak etkinliklerde bireyin etkinlik içinde kullanacağı materyaller, ucuz ve kolay temin edilebilmeli, diğer bireyler ile farklı ortamlarda rahat bir şekilde tasarlanabilir olmalıdır. Böylelikle bireyin merak ve ilgisi daha çok artarak bilgisayar kavramlarını öğrenirken oyun içinde çözüm odaklı düşünme becerisi gelişecektir. Burada dikkat edilmesi gereken husus, bilgisayarlı kodlama etkinliklerinin öğretmenler tarafından iyi tasarlanmış olmasıdır. Bilgisayarsız kodlama etkinlikleri sayesinde belirli bir aşamaya geldikten sonra bireyin blok tabanlı kodlamaya geçişi daha kolay olacaktır.

3.1. Programlamaya Giriş Etkinlikleri

1. ETKİNLİK: SAYMAYI ÖĞRENİYORUM-1

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Onlu sayı sistemini ve ikili sayı sistemini ve özelliklerini tanımlar.

Onlu sayı sistemindeki sayıyı ikilik sayı sistemine çevirir.

İkili sayı sistemindeki sayıyı onlu sayı sistemine çevirir.

İkili sayı sisteminde sayma ve sıralama yapar.

YÖNTEM/TEKNİKLER

Oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

İkili sayı sistemi kartları

Noktalar yerine gülen yüz gibi alternatif oyun kartları da kullanılabilir.

GİRİŞ

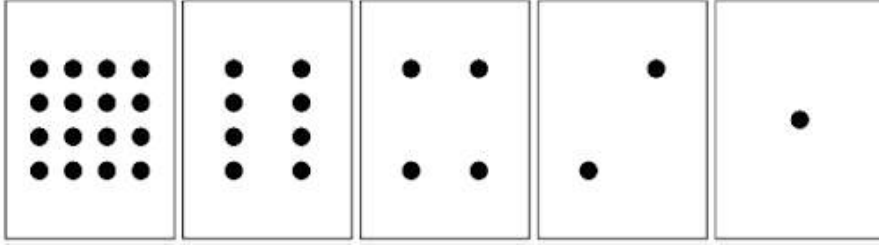
Aşağıdaki ifadelerle derse giriş yapılır:

- Saymayı biliyorsunuz değil mi? Şimdi farklı bir şekilde saymayı deneyeceğiz. Bilgisayarların yalnızca sıfır ve bir kullandığını biliyor muydunuz?
- Bilgisayarda gördüğünüz veya duyduğunuz her şey -kelimeler, resimler, sayılar, filmler ve hatta sesler- sadece bu iki sayı kullanılarak saklanır.
- Sınıfta bir oyun oynayacağız. Bu oyunda, bilgisayarla aynı yöntemi kullanarak arkadaşlarınıza gizli bir mesaj göndermeniz istenmektedir.

BÖLÜM 3

SÜREÇ

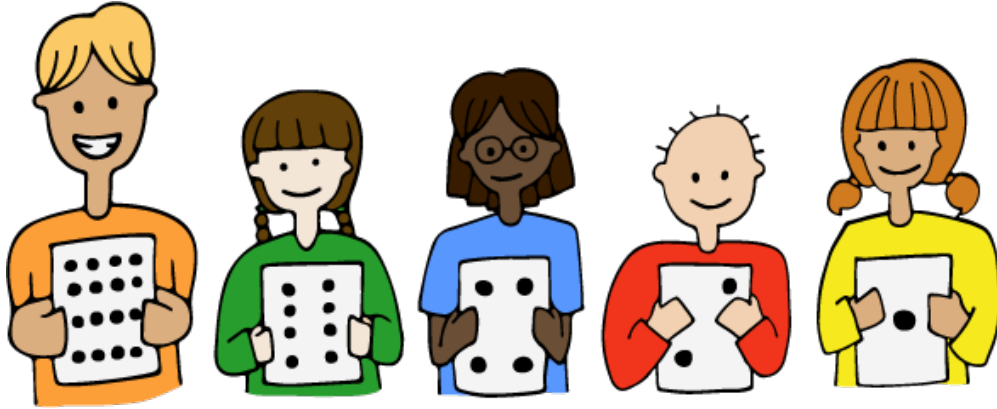
- Bu etkinlik için aşağıda gösterildiği gibi, bir yüzlerinde noktalar olan ve diğer yüzlerinde hiçbir şey olmayan Şekil 3.1'deki kartlara ihtiyacınız olacaktır.



Şekil 3.1. İkili sayı sistemi oyun kartları

Kaynak: <https://csunplugged.org/en/topics/binary-numbers/unit-plan/how-binary-digits-work/>

- Kartları, Şekil 3.2'deki gibi sınıfa göstermeleri için beş öğrenci seçiniz. Kartları dağıtırken (sağdan sola), sınıftaki diğer öğrencilerin bir sonraki kartta kaç nokta olduğunu tahmin edip edemeyeceklerine dikkat ediniz.

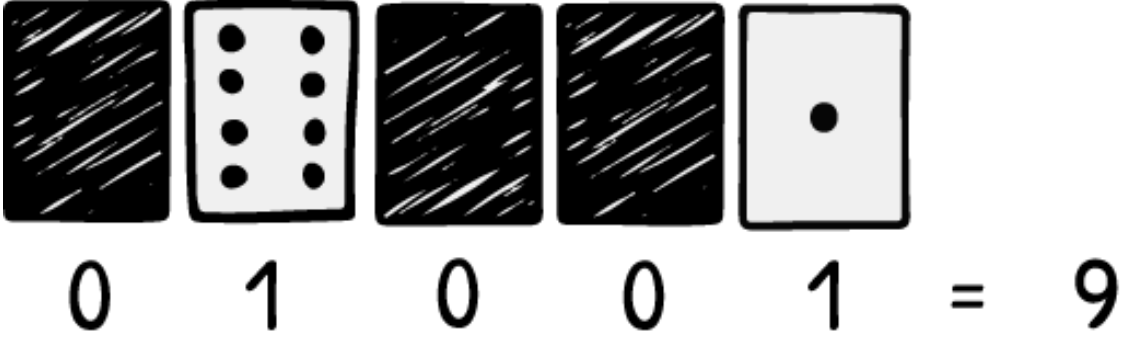


Şekil 3.2. Kartları tutan öğrenciler

Kaynak: <https://csunplugged.org/en/topics/binary-numbers/unit-plan/how-binary-digits-work-junior/>

■ Öğrencilerinize aşağıdaki soruları sorunuz:

1. Bir kart ile kartın sağındaki nokta sayısı nasıl bir artışla değişiyor? (Her kartın sağındaki karttan iki kat daha fazla nokta vardır.)
 2. Turuncu öğrencinin soluna 6. kart konulacak olsa bu kartta kaç nokta olurdu? (32)
 3. 7. kart konulacak olsa bu kartta kaç nokta olur? (64)
- Bu kartlar, bazılarının noktaları görünecek şekilde çevrilip üzerindeki noktaları toplayarak sayılar yapmak için kullanılabilir. Örnek olarak öğrencilerden 6 (4 noktalı ve 2 noktalı kart), sonra 15 (8, 4, 2 ve 1 noktalı kartlar), sonra 21 noktalı kartları (16, 4 ve 1) göstermelerini isteyiniz.

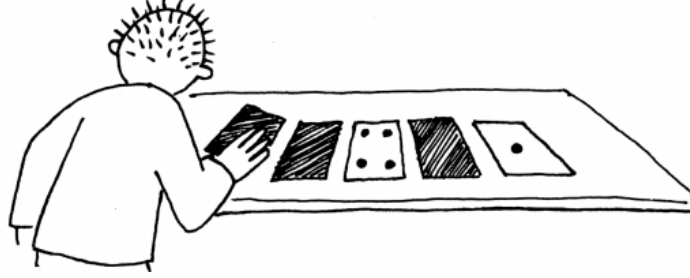


Şekil 3.3. 9 sayısının ikili sayı sisteminde gösterimi (01001)

Kaynak: <https://csunplugged.org/en/topics/binary-numbers/unit-plan/how-binary-digits-work/>

- Öğrencilerden 01001 yapmalarını isteyiniz. Bunun onluk sayı sistemindeki karşılığı olan sayıyı bulmalarını isteyiniz. (9)
- Öğrencilerden, 17 sayısının ikilik sayı sistemindeki karşılığının kaç olduğunu bulmalarını isteyiniz. (10001)
- “Bu kartlarla oluşturulabilecek olan en küçük sayı kaçtır? (1 yanıtını verebilirler ancak doğru yanıt sıfırdır.). Şimdi sıfırdan başlayarak saymayı deneyiniz.” diyerek sınıfın geri kalanının, kartların nasıl yerleştiğini gösteren bir örüntü görüp göremediklerine dikkat ediniz (Her kart, sağındaki kartın iki katını verir.). Bunu yaparken sınıfınızı gruplara ayırabilirsiniz. İkili bir sayı kartı gösterilmediğinde 0 ile, gösterildiğinde ise 1 ile temsil edilir. Bu, ikili sayı sistemidir. Kartların üzerindeki noktalar 16, 8, 4, 2 ve 1 olacak şekilde kartları soldan sağa doğru sıralayınız. Şimdi kartların arka yüzünü (noktalar görülmeyecek) çevirip aynı hizada yerleştiriniz.

BÖLÜM 3



Şekil 3.4. İstenen Sayıların Bulunması

Kaynak: <https://csunplugged.org/en/topics/binary-numbers/unit-plan/how-binary-digits-work/>

- Öğrencilerinize bu kartlardaki noktaları kullanarak toplamda 3, 12 ve 19 noktayı nasıl elde edeceklerini sorunuz? Bu sayıları elde etmek için kartlardan uygun olanların noktalı yüzlerini çeviriniz ve toplamda (5 kartta) görünen nokta sayısının istenen sayıya eşit olup olmadığını sayarak kontrol ediniz.
- Öğrencilere “Bu sayılardan herhangi birini elde etmek için birden fazla seçenek var mı?” sorusunu yöneltiniz.

Örnek: “3 nokta için hangi kartları çevirdiniz? 3 nokta elde etmek için başka olasılıklar var mı?” dedikten sonra diğer sayılar için de bu alıştırmaları yapınız.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

Kartları kullanarak öğrencilere sırasıyla 1, 2, 3, 4 sayılarını yaptırmayı deneyiniz. Öğrencilere “Herhangi bir sayıyı bir arttırmak için kartları nasıl çevirmek gerekiyor?” diye sorunuz. Önceki kartın noktatasız yüzeyini çevirirken bir sonraki kartın noktalı yüzeyini çevirmek gerektiğini belirtiniz.

<https://youtu.be/Wy6-FXtLMV8> adresinden sınıf içinde gerçekleştirilmiş bu tarz bir uygulamayı inceleyebilirsiniz.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

Sorular:

Bu kartları kullanarak oluşturabileceğiniz

1. En büyük sayı kaçtır?
2. En küçük sayı kaçtır?
3. En küçük ve en büyük sayılar arasında yazamayacağınız bir sayı var mı?

Yanıtlar

1. Tüm kartlardaki noktaların toplamı kadardır. $16+8+4+2+1=31$
2. Tüm kartların kapalı olduğu (noktasız yüzlerin görüldüğü) durumda en küçük sayıyı elde edebilirsiniz. Yani yanıt 0'dır.
3. 0-31 arasındaki tüm sayıları, kartlar kullanılarak elde edebilirsiniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Onlu sayı sistemini ve ikili sayı sistemini ve özelliklerini tanımlama			
Onlu sayı sistemindeki sayıyı ikili sayı sistemine çevirme			
İkili sayı sistemindeki sayıyı onlu sayı sistemine çevirme			
İkili sayı sisteminde sayma (00001, 00010, 00011)			
İkili sayı sistemindeki sayıları sıralama (Örnek 10000, 10001, 11001 sayılarını büyükten küçüğe sıralayınız.)			

Etkinlikte kullanılan oyun kartlarını indirmek için <https://csunplugged.org/en/resources/> adresini kullanabilirsiniz.

2. ETKİNLİK: SAYMAYI ÖĞRENİYORUM-2

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Sembolleri ikili sayı sisteminde gösterir.

Verilen sayıyı ikili sayı sisteminde göstermek için değişik sembollere çevirir.

İkinin kuvvetlerini kullanarak (1, 2, 4, 16, ...) sayma işlemi yapar.

YÖNTEM/TEKNİKLER

Oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

Öğrenci etkinlik kâğıdı

İkili sayı sistemi kartları

GİRİŞ

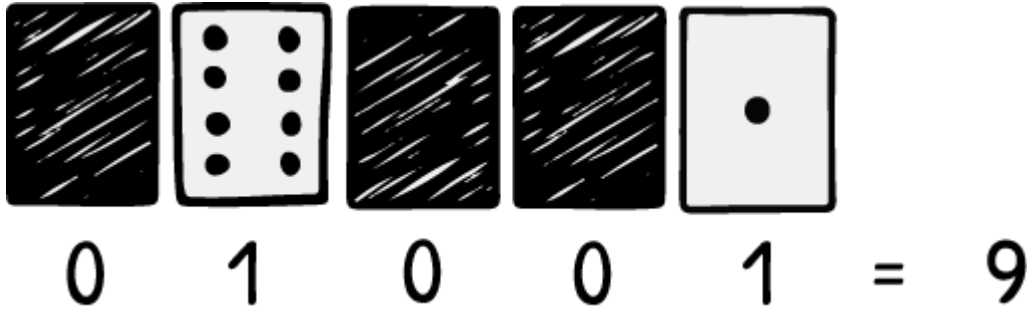
Aşağıdaki ifadelerle derse giriş yapılıır:

- Saymayı biliyorsunuz değil mi? Şimdi farklı bir şekilde saymayı deneyeceğiz! Bilgisayarların yalnızca "0" ve "1" kullandığını biliyor muydunuz?
- Bilgisayarda gördüğünüz veya duyduğunuz her şey -kelimeler, resimler, sayılar, filmler ve hatta sesler- sadece bu iki sayı kullanılarak simgelenir.
- Bu oyunda, bilgisayarla tamamen aynı yöntemi kullanarak arkadaşlarınıza gizli bir mesaj göndermeniz istenmektedir.

SÜREÇ

- İkili sistemde bir kartın açık olup olmadığını göstermek için sıfır ve bir kullanılır. 0, bir kartın gizlendiğini gösterir ve 1, noktaları görebileceğiniz anlamına gelir.

Örneğin:

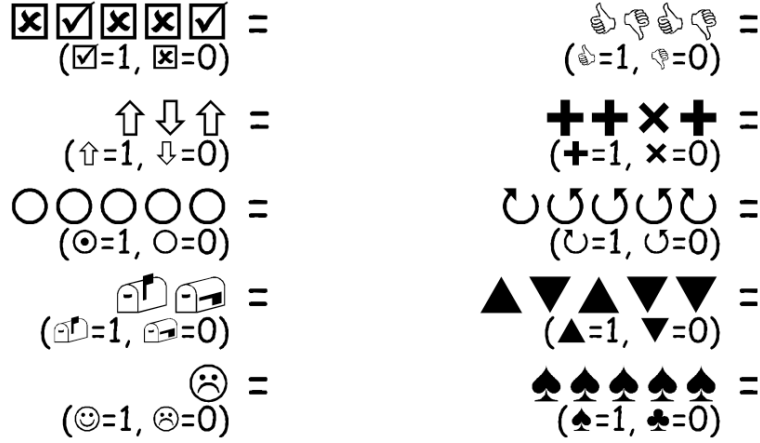


Şekil 3.5. Örnek: “9” sayısı İkili Sayı Sisteminde Gösterimi (01001)

Kaynak: <https://csunplugged.org/en/topics/binary-numbers/unit-plan/how-binary-digits-work/>

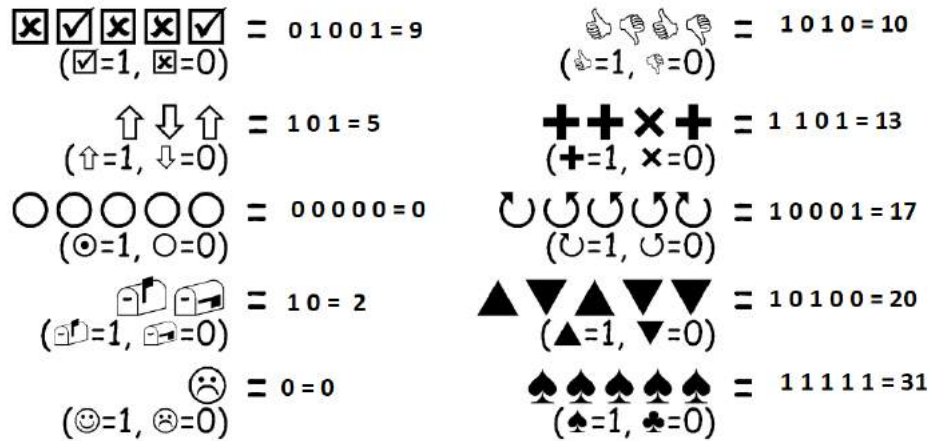
- Şekil 3.5’te verilen örnekteki gibi kartları kullanarak aşağıdaki soruları cevaplamalarını sağlayınız.
 - 10101’in kaç olduğunu bulabilir misiniz?
 - Peki “11111” kaçtır?
 - Ayın hangi günü doğduğunuzu ikilik sayı sisteminde gösterebilir misiniz?
 - Arkadaşlarınızın doğum günlerinin ikilik sayı sistemine göre ne olduğunu öğreniniz.
- Aşağıdaki örneklerde noktalar yerine farklı simgelerle kodlamalar verilmiştir. Öğrencilerinizden verilen kodlamalardaki ikilik sayı sistemindeki değerlerini bulmasını isteyiniz. Bulduğunuz bu ikili sayı sistemindeki değerlerin onlu sayı sistemindeki karşılığını şekildeki ikilik sayı kartlarını kullanarak hesaplayabilirsiniz.

BÖLÜM 3



Şekil 3.6. Farklı kod işaretleri (Kaynak: <https://classic.csunplugged.org/>)

- Öğrencilerin yanıtlarını kontrol ediniz. İlk değer, ikili sayı sistemindeki karşılığı, ikinci değer ise onlu sayı sistemindeki karşılığıdır.



Şekil 3.7. Farklı kod işaretleri sorularının cevapları

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

- 1, 2, 4, 8 ve 16 birim uzunluktaki çubuklardan oluşan bir set kullanarak 31 birime kadar herhangi bir uzunluğun nasıl yapılacağına ilişkin problemler gösterebilirsiniz.
- Öğrencilerinize bir eşit kollu terazi ve birkaç ağırlık (uzunluk ölçülerinde olduğu gibi) vererek değişik ağırlıkları nasıl tartılabileceklerine ilişkin problemler geliştirebilirsiniz.

ÖLÇME VE DEĞERLENDİRME

1. Öğrenciler, doğum günlerini ve aylarını sırasıyla kodlayarak arkadaşlarının bunları bilmelerini isteyebilir. Öğrenciler karşılıklı olarak bunu yapabilir ve doğruluğunu kontrol edebilir.
2. Kendi kodlama işaretlerini (değişik sembol veya simgeler) kullanarak 0 ve 1 simgelerini bu işaretlerle arkadaşlarına sormalarını ve yanıtlarını kontrol etmelerini isteyiniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Verilen sembolleri ikili sayı sisteminde (0, 1 olarak) gösterme			
İkili sayı sisteminde verilen sembolleri onlu sayı sistemine çevirme			
Onlu sayı sisteminde verilen sayıyı ikili sayı sisteminde sembollere çevirme			
İkinin kuvvetlerini kullanarak 31'e kadar sayıları hesaplama (Örnek: 1, 2, 3 için $2+1$ kullanılır.)			

BÖLÜM 3

3. ETKİNLİK: MAKİNE DİLİ

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Harflerin kodlara (İkili sayı sisteminde sayılara) karşılık geldiğini bilir.

İkili sayı sistemindeki sayıyı onlu sayı sistemine çevirir.

Karakter kodlarıyla harfleri eşleştirir.

Harfleri birleştirerek mesajı bulur.

Farklı mesajlar için yeni kodlar oluşturur.

YÖNTEM/TEKNİKLER

Oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

Öğrenci etkinlik kâğıdı

GİRİŞ

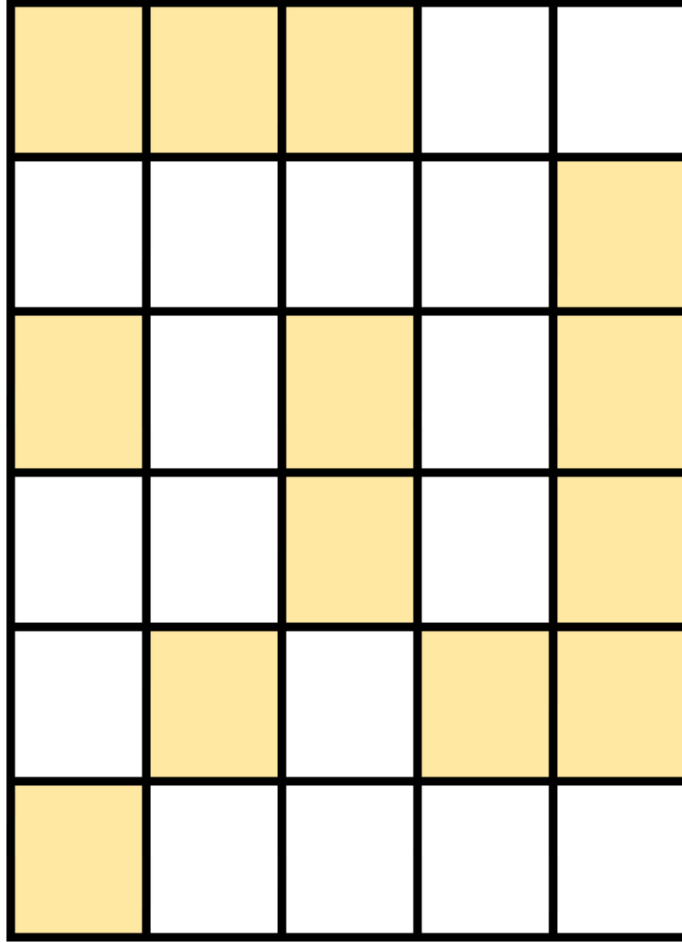
Derse girişte **ikili sayı sistemini** hatırlatıp ardından aşağıdaki hikâyeyi sınıfınızla paylaşınız.

“Emre, yılbaşı öncesi çıkış saatini kaçırarak bir mağazanın en üst katında mahsur kalır. Mağazada Emre’den başka kimse yoktur. Emre bu durumdan kurtularak evine gitmek ister, etrafına bakınır, seslenir ama ona yardım edecek kimseyi bulamaz. Caddenin karşısında, gece geç saatlere kadar çalışan yazılım şirketindeki programcıları görür. Birden Emre’nin aklına bir çözüm gelir. Emre etrafına bakınır ve tüm mağazadaki katların ışıklarını kontrol edebileceği şalterleri bulur. Bu şalterler ile katlardaki mağazaların ışıklarını açıp kapatmak mümkündür.”

Öğrencilerinize “Emre, bu ışıkları kullanarak yardım isteyebilir mi?” diye sorunuz.

SÜREÇ

- Öğrencilere “Emre, bu ışıkları karşıdaki programcılara mesaj göndermek için kullanabilir. Karşıdaki programcının anlayacağından emin olduğu basit bir ikili kod kullanan Emre’nin Şekil 3.8’de verilen kodunu çözebilecek misiniz?” dedikten sonra Tablo 3.1’de verilen harflerin kodlarını iki tarafın da bildiğini belirtiniz.



Şekil 3.8. Mağazanın dışarıdan görünüşü

- Öğrenciler, her satırdaki odalardan ışıklı olanlarının üzerine 1, diğerlerine 0 yazarak Emre’nin kodladığı ikili sayıyı bulabilirler.

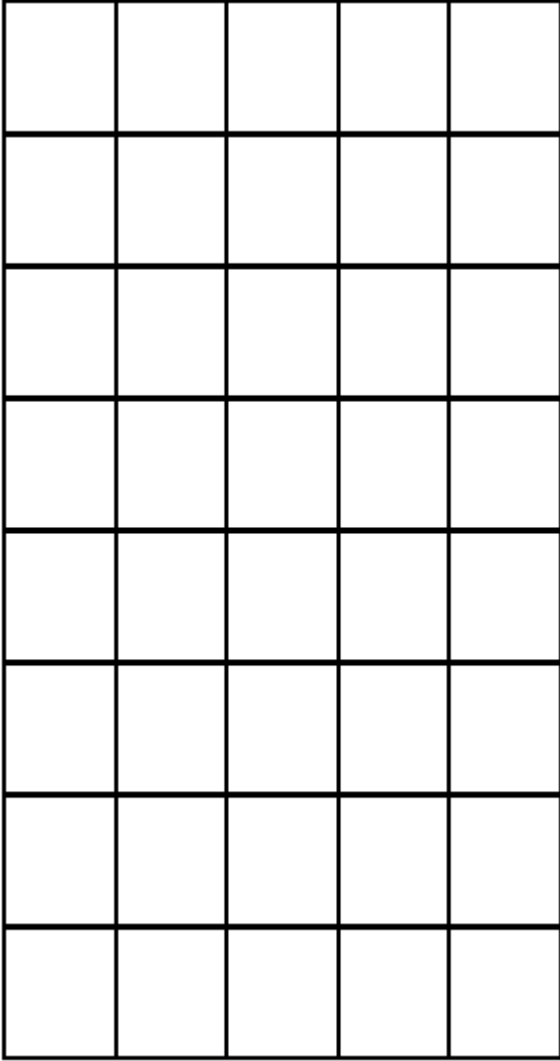
Tablo 3.1. Kodların harf karşılıkları

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k	l
16	17	18	19	20	21	22	23	24	25	26	27	28	29	
m	n	o	ö	p	r	s	ş	t	u	ü	v	y	z	

- Bu 1. satır, 1 1 1 0 0 yani onlu sayı sisteminde 28'dir. Tablo 3.1'de 28'in karşılığındaki harfi bu-
lularak satırın yanına yazar. Her satır için bu işlemler tekrarlandığında ve harfler üstten alta doğru
okunduğunda Emre'nin vermek istediği mesaj ortaya çıkar.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

Hikâyede kullanılan olay, mesaj ve ortamda değişiklikler yapılabilir. Şekil 3.9'daki kart tablosu üze-
rindeki karelerin içine Şekil 3.10'daki kartları yerleştirerek öğrencilerden yazılanları okumalarını ya
da kendi isimlerini yazmalarını isteyebilirsiniz.



Şekil 3.9. Boş kart tablosu



Şekil 3.10. Kareleri kapatma kartları

BÖLÜM 3

ÖLÇME VE DEĞERLENDİRME

Sorular

1. Emre'nin mesajını buldunuz mu?
2. Tablo 3.1'deki kodlama sistemini kullanarak arkadaşınıza en fazla 8 harfli kelimelerle mesajlar yollayabilir misiniz?

Yanıtlar

1. "Yardım"
2. Öğrenciler, arkadaşlarının kendi mesajlarını doğru alıp almadığını kontrol edebilir. Etkinlikte kullanılan oyun kartlarını indirmek için <https://csunplugged.org/en/resources/> adresini kullanabilirsiniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Harflerin kodlara (İkili sayı sisteminde sayılara) karşılık geldiğini bilme			
İkili sayı sistemindeki sayıyı onlu sayı sistemine çevirme			
Karakter kodlarıyla harfleri eşleştirme			
Harfleri birleştirerek mesajı bulma			
Farklı mesajlar için yeni kodlar oluşturma			

4. ETKİNLİK: BİLGİSAYAR, PROGRAMLARI NASIL ÇALIŞTIRIR?

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Seçilen şekil için uygun talimatlar kümesini oluşturur.

Çizim için uygun talimatları sırasıyla verir.

İstenilen şekli çizer/çizdirir.

YÖNTEM/TEKNİKLER

Oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

Şekilleri gösteren kartlar, kalem, kâğıt ve cetvel

GİRİŞ

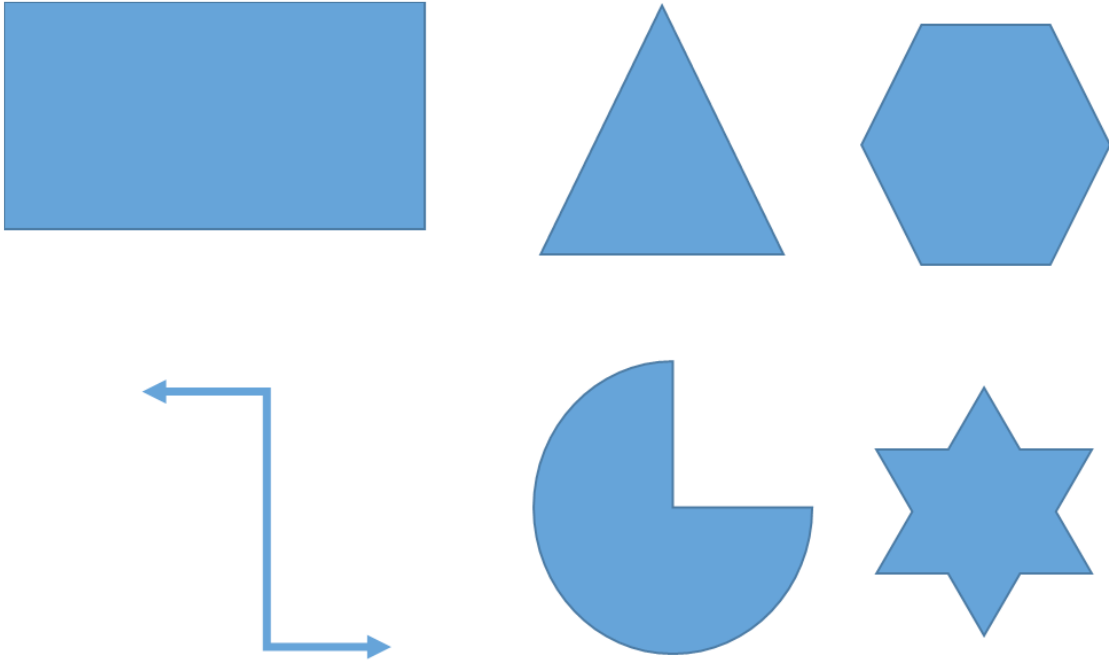
Bu etkinlik, programlamanın bazı özelliklerini vurgulamak amacıyla hazırlanmıştır. Konuyla ilgili aşağıdaki ifadeleri kullanarak kısa bir giriş yapınız.

“Bilgisayarlar, genellikle uyulması gereken talimatları ifade eden kısa sözcükler olan bir “dil” kullanılarak programlanır. Programlama ile ilgili en zorlayıcı konu, bilgisayarların her zaman talimatlara harfiyen uymasıdır. Siz, mantıksız bir komut verseniz bile bilgisayarlar, programlar bunu yerine getirmeye çalışır. Örneğin, kapalı bir kapıyı işaret edip “O kapıdan git.” dediğinizde öğrenciler, bilgisayar gibi davranırlarsa ne yaparlar? Sadece söylediğinizi yapmaya çalışmalı, kapalı kapıyı açmadan çıkmaya çalışmalıdır.” şeklinde açıkladıktan sonra anlattıklarınızı sınıfta uygulayabilirsiniz.

BÖLÜM 3

SÜREÇ

- Seçtiğiniz öğrenciye Şekil 3.11'deki şekillerden birini veriniz. Öğrenci, aşağıdaki şekillerden birini tüm öğrencilere komutlar vererek çizdirmeye çalışmalıdır. Öğrenciniz "Kalemi sayfanın başına getiriniz, 5 cm düz bir çizgi çiziniz." gibi komutlar kullanmalıdır. Diğer öğrenciler, anlamadıkları komutlarla ilgili sorular sorabilirler. Amaç, etkinliğin ne kadar hızlı ve doğru bir şekilde tamamlanabileceğini görmek için bir prova yapmaktır.
- Etkinliği, başka bir şekil seçerek tekrarlayınız ancak bu sefer öğrencilerin soru sormalarına izin vermeyiniz.
- Etkinliği bu kez de herhangi bir soruya izin vermeden diğer öğrencileri görmeyen bir öğrencinin talimat vermesiyle tekrarlayınız.
- Bu iletişim biçiminin, bilgisayar programcılarının program yazarken karşılaştığı iletişim şekline benzediğine dikkat ediniz. Kullanıcılar, bilgisayara bir dizi talimat verir ve işlemlerin nasıl yapıldığını göremez ancak işlem bittiğinde sonuçlarını monitörden veya yazıcıdan bir çıktı aldıklarında görebileceklerini bilmelerini sağlarınız.

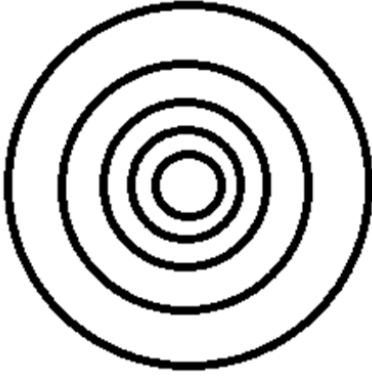


Şekil 3.11. Öğrencilerin çizeceği şekiller

- Şekillerin çizilmesinde tüm sınıfın anlayabileceği talimatlar dizisini oluşturmaya çalışınız. Bu talimatlar dizisi, tüm sınıfta aynı şeklin çizilmesini sağlamalıdır. Öğrencilere verilen talimatlar, tıpkı bilgisayarlara verilen talimatlar gibi açık ve net olmalıdır.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

Dart hedef tahtası çizme (iç içe daireler) etkinliğini Şekil 3.12'deki iç içe daireleri talimatlar ile çizdirerek uygulayabilirsiniz.



Şekil 3.12. Basit bir dart

"X metre ileri git.", "Sola dön." (90 derece) ve "Sağa dön." (90 derece) gibi yönergeleri kullanarak sınıf içindeki bir yere nasıl ulaşılacağına ilişkin talimatları yazdırınız. Öğrenciler, konuyu tam olarak anlayana kadar talimatları test etmeli ve geliştirmelidir.

Bir öğrencinin diğer öğrenciler tarafından sınıf içinde talimatlarla kapıya yönlendirilmesini sağlayınız. Öğrenci, sadece verilen talimatları yerine getirmelidir.

ÖLÇME VE DEĞERLENDİRME

Öğrencilerin, talimatları tam olarak anlayıp yerine getirdiklerini kontrol ediniz. Yukarıdaki etkinlikler için seçilen şekillerin öğrenciler tarafından nasıl çizildiğini kontrol ediniz.

Farklı sonuçlara ulaşan öğrencilerle talimatları tekrarlayarak gözden geçirebilirsiniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Çizim için uygun talimatları sırayla verme			
Verilen komutlara uygun hareketleri yapma			
İstenilen şekli çizme/çizdirme			

5. ETKİNLİK: İSMİMİ BİLEKLİĞİME KODLUYORUM

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Harflerin kodlara (ikili sayı sisteminde sayılara) karşılık geldiğini bilir.

ASCII kodlarının işlevini bilir.

Bir harfin ASCII kodunu ve ikili sayı sistemindeki karşılığını bulur.

Harfin ikili sayı sistemindeki karşılığını boncukları kullanarak görselleştirir.

Farklı harflerin ASCII kodunu boncukları kullanarak görselleştirir.

YÖNTEM/TEKNİKLER

Yaparak yaşayarak öğrenme

ÖĞRETİM MATERYALLERİ

Makas, kalem, şeffaf bant, ip, ASCII kod tablosu, renkli boncuklar

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- “İsimleriniz neyi ifade etmektedir?” Sınıfınızdaki öğrencilerinizden bir kaçının ismini söyleyerek bu isimlerin bizleri tanımlamak için kullanıldığını belirtiniz.
- “Sınıf adınız neyi ifade etmektedir?” Sınıf adlarının, içinde buldukları sınıfı tanımlayan kodlar olduğunu, aynı şekilde okul numaralarının da öğrencileri tanımladığını ifade ediniz.
- Bilgisayarlar da okul numaralarımız ve isimlerimiz gibi kendine göre kodlar kullanmaktadır ve harflerle rakamları kendisinin anlayacağı kodlara çevirir. Bu kod tablosuna da ASCII kod tablosu denilmektedir.
- Bu kod tablosunu kullanarak öğrencilerin isimlerinin baş harflerini ifade eden bileklikler yapınız.

SÜREÇ

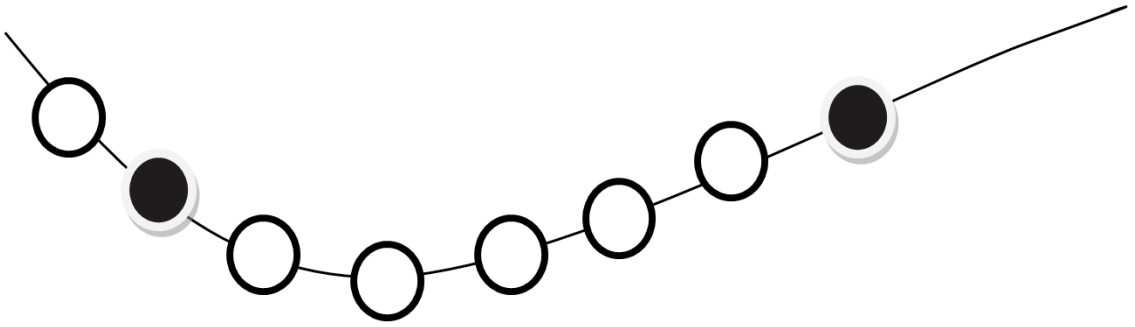
- Sınıftaki öğrencilerinizi ikiye bölün ve gruplara ayırınız.
- Her gruba bir adet ASCII tablosu, ip, sekiz adet siyah ve beyaz (Biri koyu ve diğeri açık olmak üzere farklı renkler de olabilir.) boncuk veriniz.
- Öğrencilerin kendi isimlerinin ve arkadaşının isminin baş harflerinin ne olduğunu söylemelerini isteyiniz. Bu harfleri ASCII tabloda bularak göstermelerini isteyiniz. (Not: Okuma yazma bilmeyen öğrencilerinizin isimlerinin baş harflerini tahtada gösterebilirsiniz. Bu harfe benzeyen resmi ASCII tabloda göstermelerini isteyebilirsiniz.)
- ASCII tablosunda buldukları harfin karşısındaki sayıları söylemelerini isteyiniz. Bu sayıların sadece 1 ve 0 rakamlarından oluştuğunu fark etmelerini sağlayınız. Bunun adının ikili (binary) sayı sistemi olduğunu ve bilgisayarların sadece bu rakamları tanıdığını belirtiniz.

Tablo 3.2. İkili (Binary) ASCII Kod Tablosu

İKİLİ (BİNARY) ASCII KOD TABLOSU					
HARFLER	ASCII	BİNARY	HARFLER	ASCII	BİNARY
A	65	01000001	N	78	01001110
B	66	01000010	O	79	01001111
C	67	01000011	P	80	01010000
D	68	01000100	Q	81	01010001
E	69	01000101	R	82	01010010
F	70	01000110	S	83	01010011
G	71	01000111	T	84	01010100
H	72	01001000	U	85	01010101
I	73	01001001	V	86	01010110
J	74	01001010	W	87	01010111
K	75	01001011	X	88	01011000
L	76	01001100	Y	89	01011001
M	77	01001101	Z	90	01011010

BÖLÜM 3

- Harflerin karşılığında bulunan ikili (binary) sayıların 1 için siyah (koyu renk), 0 için beyaz (açık renk) boncukları ifade ettiğini belirtiniz.
- Ellerindeki boncukları harfin karşısındaki rakamlara göre sıralı bir şekilde ipe dizmelerini isteyiniz.
- Gruptaki diğer öğrencinin ise bu oluşan boncuk diziliminin binary karşılığına bakarak hangi harfe karşılık geldiğini bulmasını ve ismini tahmin etmesini isteyiniz.
- Gruplardaki eşleşmeleri değiştirerek etkinliğin devam ettirilmesini sağlayabilirsiniz.
- Yeteri kadar pekiştirme yapıldıktan sonra öğrencilerin hazırladıkları bileklikleri takmalarına yardım ediniz.
- Öğrencilerinize ATATÜRK sözcüğündeki ilk harfi örnek olarak gösterebilirsiniz. A harfinin ASCII tablosunda karşılığını bularak "01000001" olduğunu gösteriniz. Buna göre boncuk diziliminde takip edecekleri sıranın beyaz, siyah, beyaz, beyaz, beyaz, beyaz, beyaz ve son olarak siyah olduğunu söyleyerek boncukları ipe dizebilirsiniz.



Şekil 3.13. A harfinin boncuk dizilimi

ÖLÇME VE DEĞERLENDİRME

Adınızın ilk harfini boncuklarla diziniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Harflerin kodlara (ikili sayı sisteminde sayılara) karşılık geldiğini bilme			
ASCII kodlarının işlevini bilme (Niçin kullanılır?)			
Bir harfin ASCII kodunu ve ikili sayı sistemindeki karşılığını bulma (ASCII Tablosunu kullanma)			
Bir harfin ASCII kodunu, boncukları kullanarak görselleştirme (Örnek A harfi)			
Farklı harflerin ASCII kodunu, boncukları kullanarak görselleştirme (ATATÜRK kelimesinde bulunan diğer harfler için aynı işlemleri tekrarlama)			

6. ETKİNLİK: BAHÇEDEN ELMA TOPLAMA

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

- Doğrusal mantık yapısını açıklar.
- Doğrusal mantık yapısını kullanan algoritmalar geliştirir.
- Karar yapısını ve işlevlerini açıklar.
- Doğrusal mantık yapısını kullanan algoritmalar geliştirir.
- Bir algoritmayı test eder ve değerlendirir.

YÖNTEM/TEKNİKLER

Oyun temelli öğrenme, grup çalışması

ÖĞRETİM MATERYALLERİ

Makas, kalem, A4 kâğıt

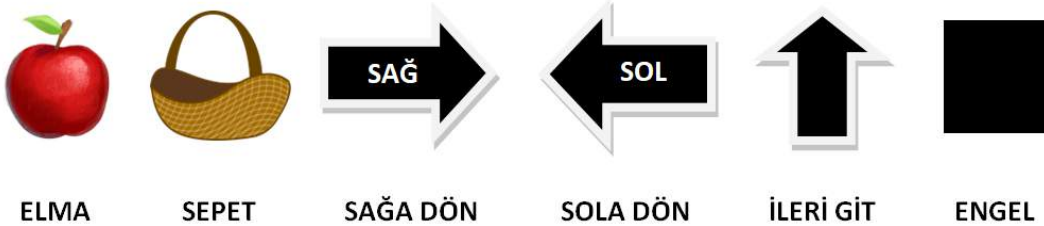
GİRİŞ

Aşağıdaki hikâye ile derse başlayınız.

“Ayşe, annesine hafta sonu dedesinin çiftliğine gitmek istediğini söyler ve hep birlikte çiftliğe giderler. Dedesinin çiftliğinde elma ağaçları vardır. Dedesi, Ayşe’ye elma ağaçlarından istediği kadar elma toplayabileceğini söyler. Ayşe, elma ağaçlarını evin balkonundan görür ama bir türlü elma ağaçlarına giden yolu bulamaz. Üzgün bir şekilde dedesinin yanına gider ve dedesine “Dedeciğim, elma ağaçlarına giden yolu bulamadım.” der. Dedesi, Ayşe ile birlikte elma ağaçlarına giderler. Ayşe ve dedesine elma ağaçlarına giden yolu bulmalarında yardımcı olmak ister misiniz?”

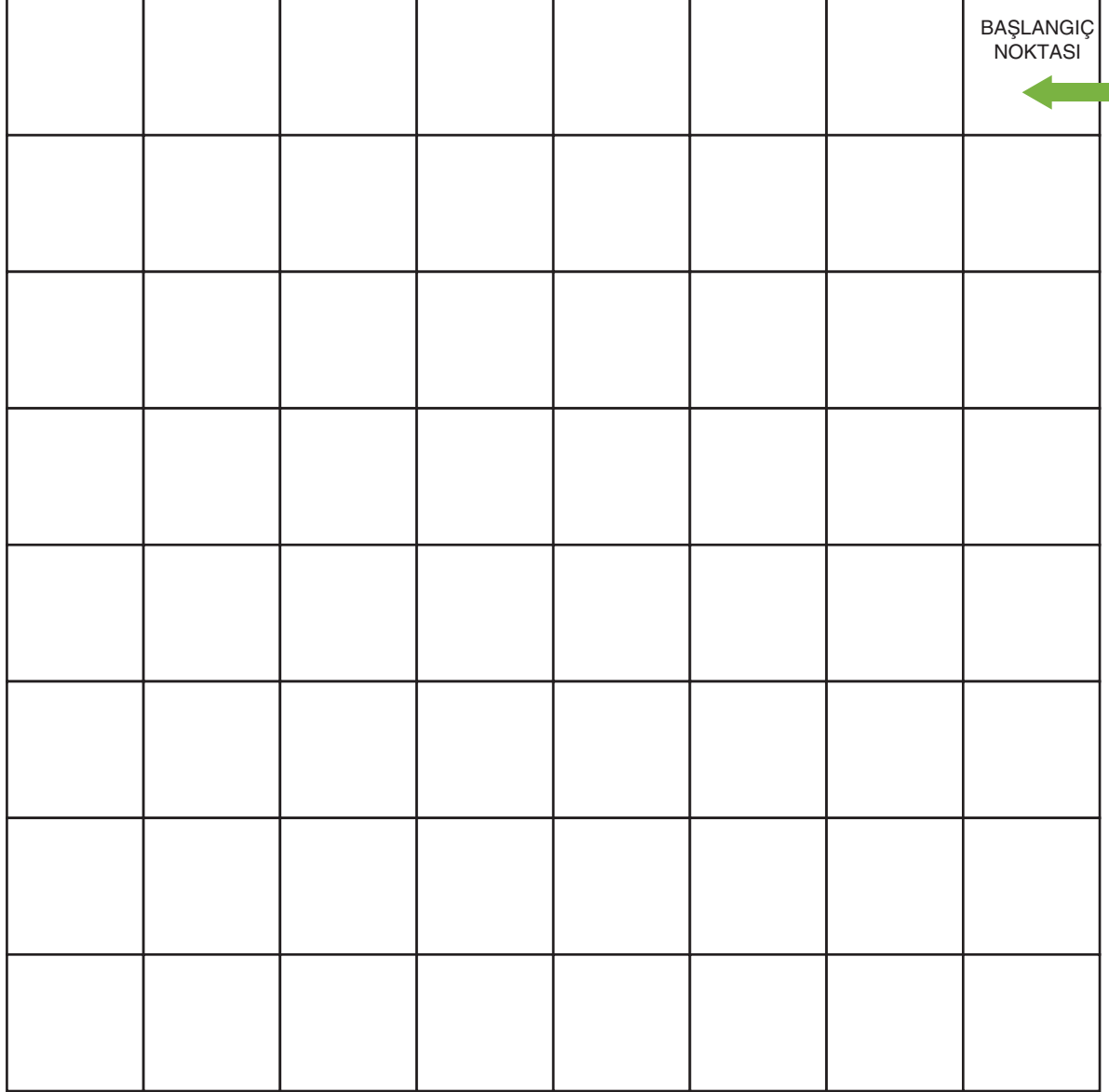
SÜREÇ

- Sınıftaki öğrencileri üçer kişilik gruplara ayırınız.
- Gruptaki öğrencilerinizi bu etkinlik için görevlendiriniz. Birinci öğrenci (hedef gösteren), oyuncacı karelerden birisine yerleştiresin. İkinci öğrenci (programcı), kod tablosunu doldursun. Üçüncü öğrenci ise (bilgisayar) arkadaşının doldurmuş olduğu kod tablosuna göre başlangıç noktasından itibaren kodları çalıştırarak hedefe (elmalar) gitmeye çalışsın.
- Şekil 3.14'teki yön bloklarını, Şekil 3.15'teki oyun platformunu (boş olarak) ve Tablo 3.3'teki kod tablosunun çıktısını alarak ya da çizerek her gruba veriniz. İstenilirse kod tablosundaki adımları arttırabilirsiniz.
- Tablonun üzerine herhangi bir nesne (elma resmi) koyabilirsiniz.
- Elmaların üzerine gelince elmayı toplamak için sepet komutunu veriniz.
- Elmaları, herhangi bir kareye yerleştiriniz. İsterseniz aralara engeller koyabilirsiniz. Öğrencilerin, başlangıç karesinden başlayarak yön işaretlerini kod tablosuna adım adım yerleştirmelerini veya çizmelerini sağlayınız.



Şekil 3.14. Oyunda kullanılacak görseller

BÖLÜM 3



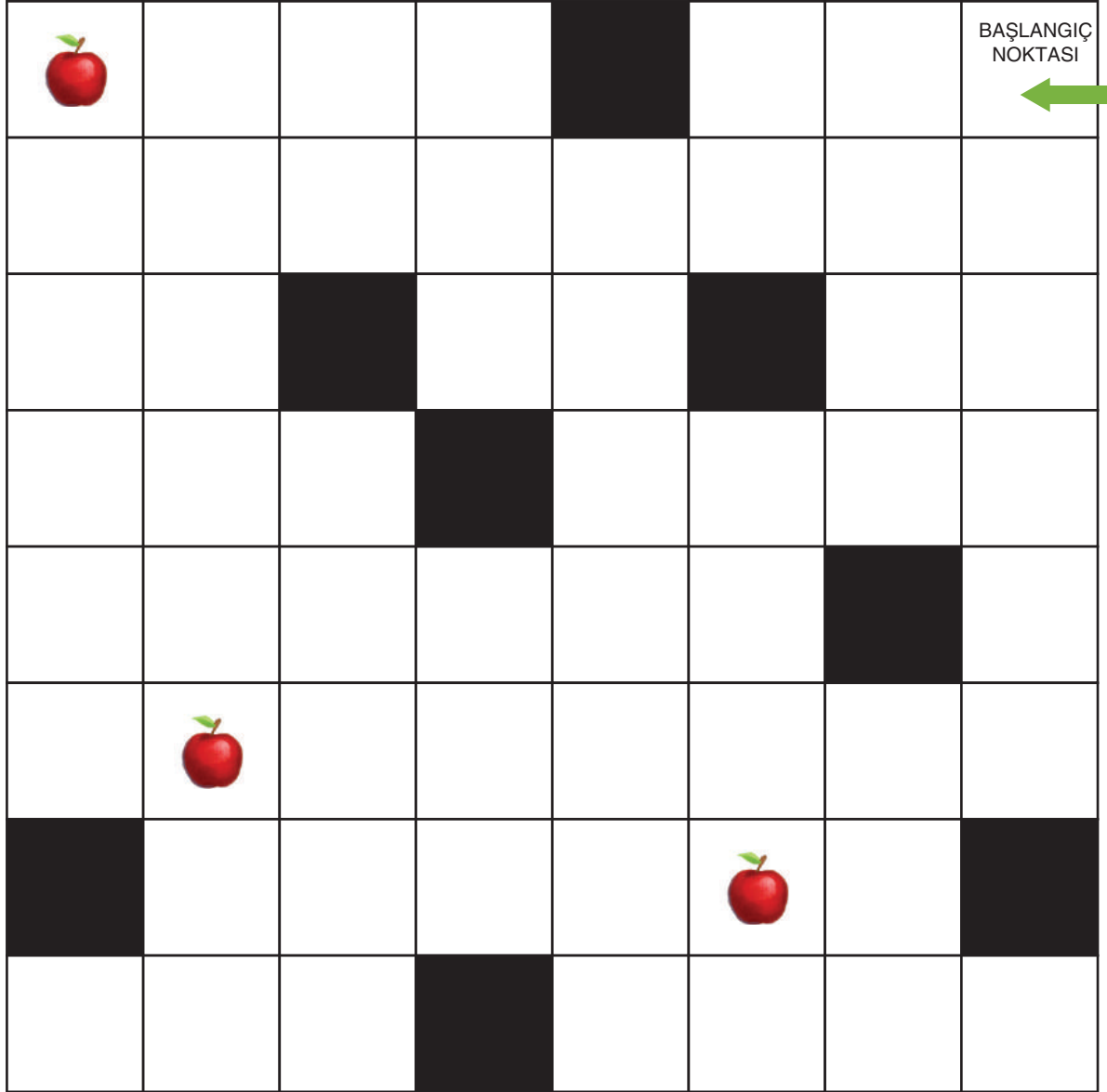
Şekil 3.15. Oyun platformu

Tablo 3.3. Göreve göre kodların yazılacağı Kod Tablosu (Satır sayısı isteğe göre artırılabilir.)

KOD TABLOSU	
ADIMLAR	KODLAR
Adım 1	
Adım 2	
Adım 3	
Adım 4	
Adım 5	
Adım 6	
Adım 7	
Adım 8	
Adım 9	
Adım 10	

BÖLÜM 3

Örnek Uygulama



Şekil 3.16. Örnek uygulama platformu

Tablo 3.4. Örnek uygulama Kod Tablosu (Başlangıç oku ile bahçeye giriş yapılacak.)

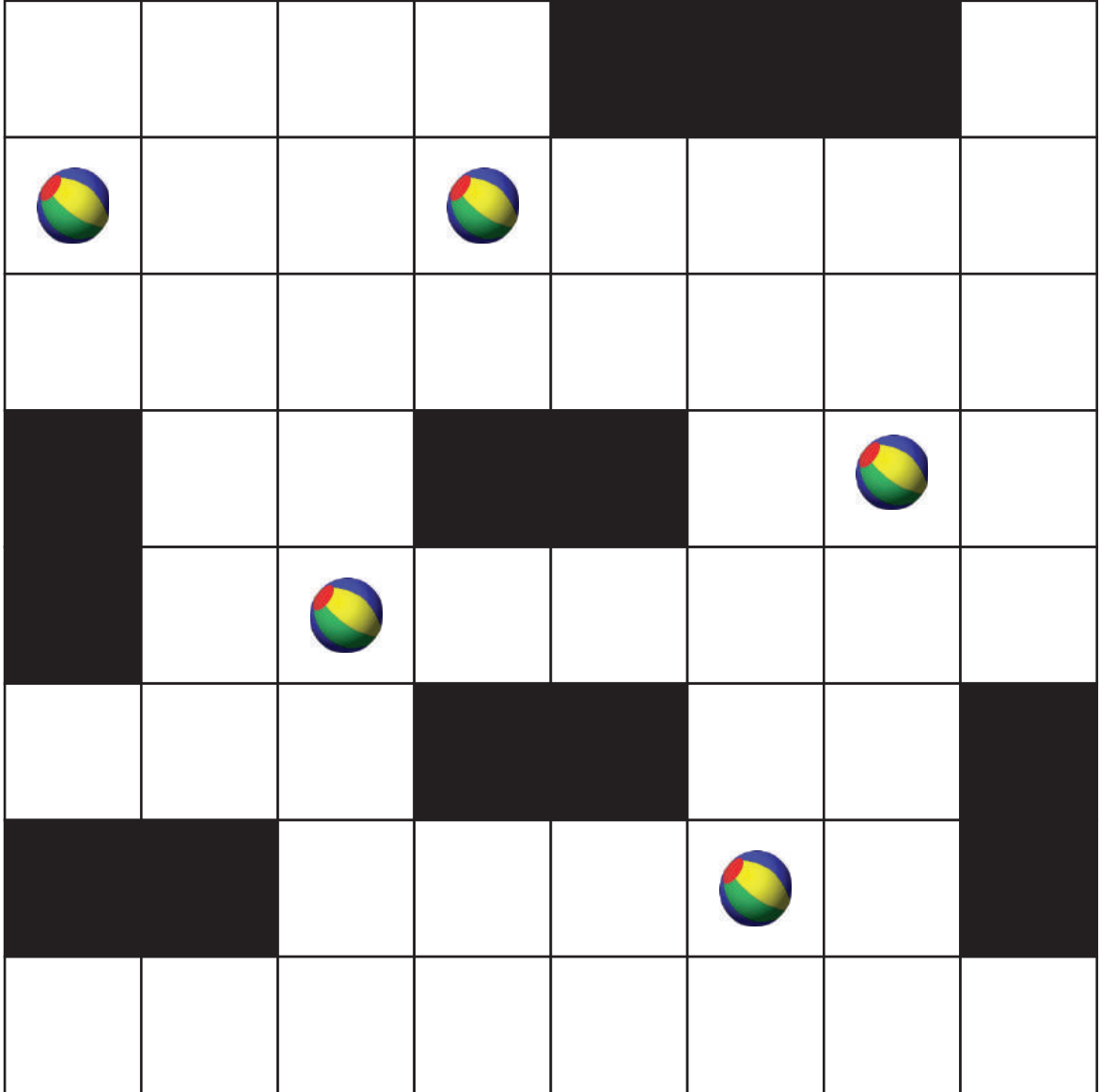
KOD TABLOSU	
ADIMLAR	KODLAR
Adım 1	3 adım ileri git.
Adım 2	Sola dön.
Adım 3	1 adım ileri git.
Adım 4	Sağa dön.
Adım 5	5 adım ileri git.
Adım 6	Sağa dön.
Adım 7	İleri git.
Adım 8	Elmayı sepete at.
Adım 9	2 sefa sola dön.
Adım 10	5 adım ileri git.
Adım 11	Sola dön.
Adım 12	İleri git.
Adım 13	Elmayı sepete at.
Adım 14	4 adım ileri git.
Adım 15	Sağa dön.
Adım 16	İleri git.
Adım 17	Elmayı sepete at.

Öğrencilerin yerlerini değiştirip elmaları farklı karelere koyarak yeniden kod oluşturmalarını isteyebilirsiniz.

BÖLÜM 3

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

Şekil 3.17'deki gibi platformu farklılaştırabilirsiniz.



Şekil 3.17. Alternatif platform

BÖLÜM 3

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Doğrusal mantık yapısını açıklama			
Doğrusal mantık yapısını kullanan algoritmalar geliştirme (Komut kümesindeki hareket oklarını doğru kullanmış mı?)			
Hedefe ulaşmak için doğrusal mantık yapısını kullanarak algoritma geliştirme			
Geliştirdiği algoritmayı test etme ve değerlendirme (Talimatlar takip edilerek oyuncağa ulaşıyor mu? Başka çözümler var mı? Kısa çözüm var mı?)			

7. ETKİNLİK: SESLER NEYİ İFADE EDİYOR?

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Doğrusal mantık yapısını açıklar.

Karar yapısını ve işlevlerini açıklar.

Amacına uygun olarak farklı yapılar arasında ilişki kurar.

Amacına uygun olarak elindeki verilerden çıkarımda bulunur.

Bir algoritmayı test eder ve değerlendirir.

YÖNTEM/TEKNİKLER

Soru cevap, drama

ÖĞRETİM MATERYALLERİ

Mors harfleri görselleri, ses notası görselleri

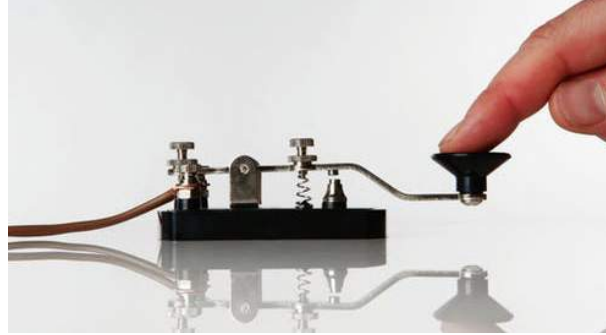
GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- Sınıfınıza “dit dit dit, dat dat dat, dit dit dit” şeklinde bir uyarın kullanarak öğrencilerin dikkatini üzerinize çekiniz.
- Öğrencilerinize seslerin neyi ifade ettiğini ve seslerden ne anladıklarını sorunuz.
- Aşağıdaki hikâyeyi öğrencilerinize anlatınız.

“Ali, tatil için köydeki dedesinin yanına gitmişti. Ali’nin geldiğine çok sevinen dedesi, onunla güzel vakitler geçirmek istiyordu. Dedesi, askerliğinde telsizci olduğunu ve o zamanlarda cep telefonları olmadığından haberleşmek için farklı bir sistem kullandıklarını anlatmış. Bu sistemde harfleri, sese çevirerek telefon hatları üzerinden mesaj gönderirlermiş. Ali, bu sistemi çok merak etmiş ve dedesinden bu telsiz kodları ve mors alfabesi kullanımını öğretmesini istemiş. Dedesi “Hadi, öyleyse öğrenmeye başlayalım.” demiş.

BÖLÜM 3



Şekil 3.20. Mors Cihazı

SÜREÇ

- Dedesi, Ali'ye öncelikli olarak harflerin nokta ve çizgilerle nasıl ifade edildiğini Tablo 3.5 üzerinden göstermiştir. (Tablo 3.5'te Türkçede olmayan özel karakterlerin de bulunduğunu belirtiniz.).

Tablo 3.5. Harfler ve Mors alfabesi karşılıkları

A	·-	J	·---	S	···	2	··---
B	---·	K	---·	T	-	3	··---
C	---·	L	·---	U	··-	4	··---
D	---·	M	--	V	··-	5	·····
E	·	N	··	W	·---	6	·····
F	··---	O	---	X	---·	7	---···
G	---·	P	··---	Y	---·	8	---···
H	·····	Q	---·	Z	---·	9	---···
I	··	R	··-	1	·-----	0	-----

- Dedesi, Ali'ye bu sesleri elde etmek için noktada kısa, çizgide uzun süreli bastıklarını söylemiştir.
- Ali ve dedesi, noktaları "DİT", uzun çizgileri "DAT" olarak seslendirmeye karar vermişlerdir (Tablo 3.6).

Tablo 3.6. Nokta ve çizgide çıkarılacak ses eşleşmesi

●	DİT
■	DAT

- Dedesi, Ali'ye "Hadi, senin ismini Mors harfleri ile hem yazılı hem de sözlü olarak nasıl ifade ederiz? Birlikte yapalım." demiş ve Tablo 3.7'yi oluşturmuştur.

Tablo 3.7. ALİ isminin Mors alfabesi ve ses ile ifade edilmesi örneği

MORS ALFABESİ	● ■	● ■ ● ●	● ●
OKUNUŞU (SESİ)	DİT DAT	DİT DAT DİT DİT	DİT DİT
KARŞILIĞI	A	L	İ

- Siz de öğrencilerinizin Tablo 3.8'de sırayla sesleri (şekil) verilen kodlarda ne söylenmek istendiğini bulmalarını isteyiniz.

Tablo 3.8. Öğrencilerinizle çalışacağınız Kod Tablosu

KOD TABLOSU	
ADIMLAR	KODLAR
Adım 1	● ■■■■
Adım 2	■■■■
Adım 3	● ■■■■
Adım 4	■■■■
Adım 5	● ● ■■■■
Adım 6	● ■■■■ ●
Adım 7	■■■■ ● ■■■■
Adım 8	● ■■■■ ■■■■ ■■■■ ■■■■
Adım 9	■■■■ ■■■■ ■■■■ ● ●
Adım 10	■■■■ ■■■■ ■■■■ ● ●
Adım 11	● ■■■■ ■■■■ ■■■■ ■■■■

- Sınıftaki öğrencileri ikiyeşerli gruplara ayırınız.
- Öğrencilerin kendi isimlerini Mors harfleri ile kâğıtlara yazmalarını isteyiniz.
- Yazdıkları Mors kodlarını sesli olarak (DİT ya da DAT) arkadaşına söylemelerini isteyiniz.
- Sesleri dinleyen diğer öğrencinin kodları yazmasını ve kontrol etmesini isteyiniz. Mors alfabesi kullanımında, gelen sesler önce nokta ve çizgi olarak kâğıda yazılır sonra harflere dönüştürülür. Bu pratik yöntemi öğrencilerinizle paylaşarak daha hızlı çözümlene yapmalarını sağlayabilirsiniz.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI-1

- Aynı etkinliği öğrencilerinize el feneri ile de yaptırabilirsiniz. Fenerin kısa yanıp sönmesinin, noktayı (ses olarak DİT); fenerin uzun süreli yanıp sönmesinin, uzun çubuğu (ses olarak DAT) ifade ettiğini öğrencilerinize söyleyiniz.

Tablo 3.9. Işık ile uygulanacak Kod Tablosu

KOD TABLOSU	
ADIMLAR	KODLAR
Adım 1	■
Adım 2	● ● ■
Adım 3	● ■ ●
Adım 4	■ ● ■
Adım 5	● ●
Adım 6	■ ● ■ ■
Adım 7	●

- Öğrencilerinizi üçerli gruplara bölerek Tablo 3.9'daki işaretleri anlamlarını sağlayınız. Doğru anladıklarını teyit ettikten sonra grupların uzak mesafeden birbirlerine mesajları iletmelerini sağlayınız.
- Etkinliği daha eğlenceli hâle getirmek için sizin belirleyeceğiniz çeşitli görevlerin bir grup, tarafından diğer gruba bir el feneri ile anlatılması ve karşıdaki grubun anlatılan görevi yerine getirmesi şeklinde bir sınıf içi yarışma düzenleyebilirsiniz.

BÖLÜM 3

ÖLÇME VE DEĞERLENDİRME

Sesler ile kodlama nasıl yapılır?

Sesli kodlama, günlük yaşamımızda kullanılır mı? Gruplar, seslerle yapılmış kodlamalara günlük yaşamımızdan örnek verebilir mi?

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Karar yapısını ve işlevlerini açıklama (Sesleri birbirinden ayırt etme)			
Amacına uygun olarak farklı yapılar arasında ilişki kurma (Çıkan ses ile nokta ve çizgiyi doğru şekilde eşleştirme)			
Amacına uygun olarak elindeki verilerden çıkarımda bulunma (Oluşan şekillerle harfler arasında ilişki kurma)			
Geliştirdiği algoritmayı test etme ve değerlendirme (Kendi isimlerini harflerle ifade etme)			
Geliştirdiği algoritmayı test etme ve değerlendirme (Kendi ismini seslerle ifade etme)			

Kaynak: <http://learnmorsecode.info/>[MA40]

8. ETKİNLİK: RESİMLERİ KODLUYORUM

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Bir problemin çözümü için algoritma oluşturur.

Bir problemi alt problemlere böler.

Bir algoritmayı oluşturmak için en uygun karar yapılarını seçer.

Problem ve alt problemlere göre sıralı algoritmaları uygular.

Bilgisayarda verilerin nasıl saklandığını bilir.

YÖNTEM/TEKNİKLER

Soru cevap

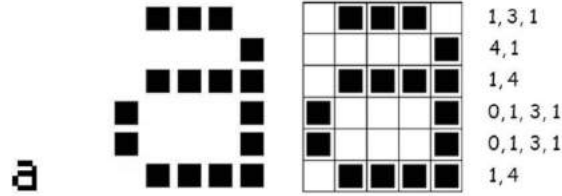
ÖĞRETİM MATERYALLERİ

Emoji görselleri, makas, kalem

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- Bilgisayarlar rakamlarla çalışıyor ama nasıl oluyor da resimleri tutabiliyor?
- Görseller bilgisayarda nasıl saklanıyor?
- Öğrencilerinizin aşağıdaki resme bakıp fikir yürütmelerini isteyiniz. Tüm görsellerin birer noktadan oluştuklarını, bu noktalar ne kadar küçük ve birbirine yakın olursa resimlerin kalitesinin de o kadar arttığını ve bunun renkli resimlerde de geçerli olduğunu belirtiniz. Oradaki noktaların da buldukları resmin o noktadaki rengini oluşturduklarını söyleyerek öğrencilerinize açıklama yapabilirsiniz.



Şekil 3.21. Bilgisayarda verilerin saklanması (MEB, 2017)

Her resim küçük noktalardan oluşmaktadır.

Şekil 3.22'ye bakıldığında 5x6 olan dikdörtgen ve içinde bazı kare kutucukları yandaki kodlamaya göre siyah renk ile boyanmıştır. Şekil 3.22'de 1. satırda 5 sütun vardır. 2, 3, 4. sütunlar siyaha boyanmıştır.

- Öğrencilerinize aşağıdaki karelerin boş olduğu çıktıyı göstererek “Aslında burada bir resim var, hatta sevebileceğiniz bir resim. Bu resmi bulabilir misiniz?” diye sorunuz.
- Öğrencilerinize “Resmin ne olduğunu söyleyemem ama size satır satır ipuçları verebilirim, bu ipuçları sonucunda resimde ne olduğunu görebileceksiniz.” şeklinde bilgilendiriniz.

SÜREÇ

- Öncelikli olarak öğrencilerinize Şekil 3.22'deki boyama platformunu dağıtınız.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	R	S	T	U	V	Y	Z	
1																							
2																							
3																							
4																							
5																							
6																							
7																							
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							

Şekil 2.22. Öğrencilere dağıtılacak boyama platformu

- Her adımda bir satırdaki beyaz ve siyah alanları sırası ile söyleyeceğinizi belirtiniz. Bu kısımda her adımı, sınıfın tümünün aynı zamanda boyamasına dikkat ediniz.

BÖLÜM 3

Tablo 3.10. İlk örnek boyama Kod Tablosu

Adım 1	9 BEYAZ, 3 SİYAH, 10 BEYAZ
Adım 2	8 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 9 BEYAZ
Adım 3	7 BEYAZ, 1 SİYAH, 5 BEYAZ, 1 SİYAH, 8 BEYAZ
Adım 4	7 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 1 BEYAZ, 1 SİYAH, 8 BEYAZ
Adım 5	7 BEYAZ, 1 SİYAH, 6 BEYAZ, 1 SİYAH, 7 BEYAZ
Adım 6	8 BEYAZ, 1 SİYAH, 3 BEYAZ, 2 SİYAH, 8 BEYAZ
Adım 7	3 BEYAZ, 1 SİYAH, 5 BEYAZ, 1 SİYAH, 1 BEYAZ, 1 SİYAH, 11 BEYAZ
Adım 8	2 BEYAZ, 1 SİYAH, 1 BEYAZ, 1 SİYAH, 2 BEYAZ, 2 SİYAH, 2 BEYAZ, 1 SİYAH, 10 BEYAZ
Adım 9	2 BEYAZ, 1 SİYAH, 2 BEYAZ, 2 SİYAH, 5 BEYAZ, 1 SİYAH, 9 BEYAZ
Adım10	2 BEYAZ, 1 SİYAH, 10 BEYAZ, 1 SİYAH, 8 BEYAZ
Adım11	2 BEYAZ, 1 SİYAH, 5 BEYAZ, 2 SİYAH, 4 BEYAZ, 1 SİYAH, 7 BEYAZ
Adım 12	3 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 6 BEYAZ, 1 SİYAH, 7 BEYAZ
Adım 13	3 BEYAZ, 1 SİYAH, 2 BEYAZ, 1 SİYAH, 7 BEYAZ, 1 SİYAH, 7 BEYAZ
Adım 14	4 BEYAZ, 1 SİYAH, 8 BEYAZ, 1 SİYAH, 8 BEYAZ
Adım 15	4 BEYAZ, 1 SİYAH, 8 BEYAZ, 1 SİYAH, 8 BEYAZ
Adım 16	5 BEYAZ, 2 SİYAH, 5 BEYAZ, 1 SİYAH, 9 BEYAZ
Adım 17	7 BEYAZ, 5 SİYAH, 10 BEYAZ
Adım 18	8 BEYAZ, 1 SİYAH, 1 BEYAZ, 1 SİYAH, 11 BEYAZ
Adım 19	22 BEYAZ

Birinci uygulama bittikten sonra Tablo 3.11'deki adımlara göre yeni bir boyama daha yaptırabilirsiniz.

Tablo 3.11. İkinci örnek boyama Kod Tablosu

Adım 1	22 BEYAZ
Adım 2	6 BEYAZ, 2 SİYAH, 5 BEYAZ, 2 SİYAH, 7 BEYAZ
Adım 3	5 BEYAZ, 1 SİYAH, 2 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 2 BEYAZ, 1 SİYAH, 6 BEYAZ
Adım 4	4 BEYAZ, 1 SİYAH, 4 BEYAZ, 1 SİYAH, 1 BEYAZ, 1 SİYAH, 4 BEYAZ, 1 SİYAH, 5 BEYAZ
Adım 5	3 BEYAZ, 1 SİYAH, 6 BEYAZ, 1 SİYAH, 6 BEYAZ, 1 SİYAH, 4 BEYAZ
Adım 6	3 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 1 SİYAH, 4 BEYAZ
Adım 7	3 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 1 SİYAH, 4 BEYAZ
Adım 8	3 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 1 SİYAH, 4 BEYAZ
Adım 9	3 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 1 SİYAH, 4 BEYAZ
Adım10	3 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 1 SİYAH, 4 BEYAZ
Adım11	4 BEYAZ, 1 SİYAH, 11 BEYAZ, 1 SİYAH, 5 BEYAZ
Adım 12	4 BEYAZ, 1 SİYAH, 11 BEYAZ, 1 SİYAH, 5 BEYAZ
Adım 13	5 BEYAZ, 1 SİYAH, 9 BEYAZ, 1 SİYAH, 6 BEYAZ,
Adım 14	6 BEYAZ, 1 SİYAH, 7 BEYAZ, 1 SİYAH, 7 BEYAZ
Adım 15	7 BEYAZ, 1 SİYAH, 5 BEYAZ, 1 SİYAH, 8 BEYAZ
Adım 16	8 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 9 BEYAZ
Adım 17	9 BEYAZ, 1 SİYAH, 1 BEYAZ, 1 SİYAH, 10 BEYAZ
Adım 18	10 BEYAZ, 1 SİYAH, 11 BEYAZ,
Adım 19	22 BEYAZ

BÖLÜM 3

ETKİNLİĞİN ALTERNATİF UYGULANIŞI-1

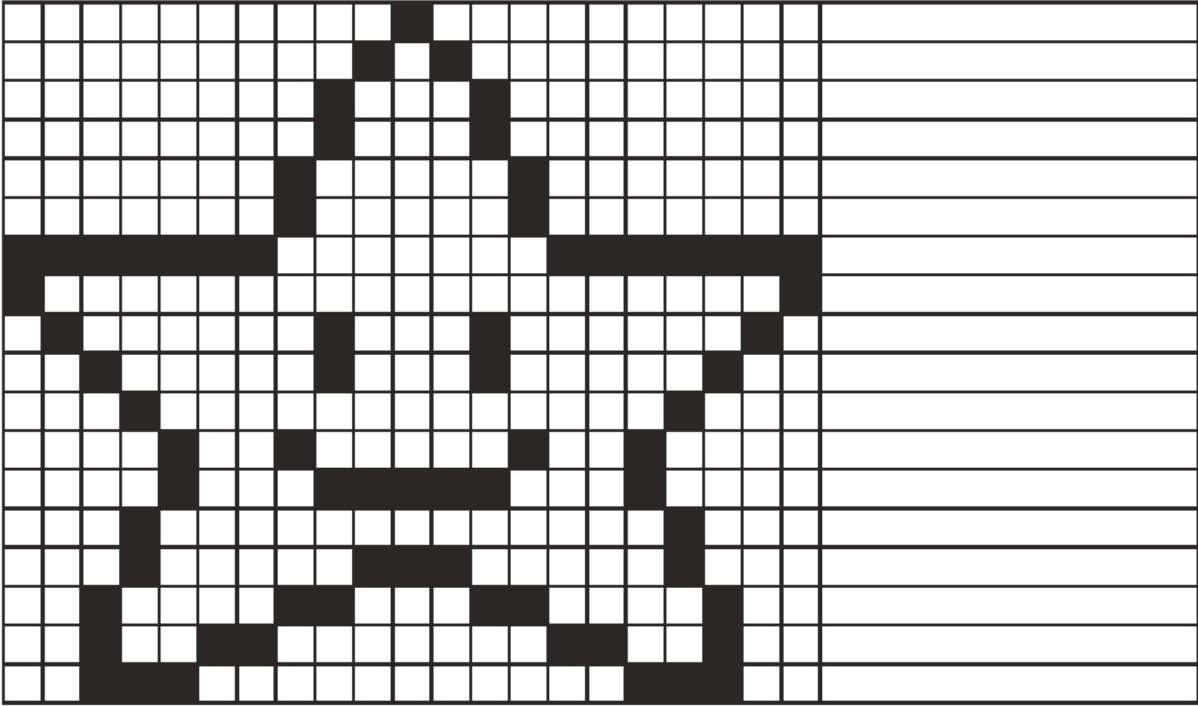
Öğrencilere el işi kâğıtlarından Şekil 3.22'deki küçük kareler boyutunda siyah ve beyaz (Farklı iki renk de tercih edilebilir.) parçalar kesmelerini söyleyiniz. Kesilen renkli parçaları, Tablo 3.12'de verdiğiniz adımlara göre sırasıyla karelere yapıştırılmalarını sağlayınız.

Tablo 3.12. Boyama için Kod Tablosu

Adım 1	10 BEYAZ, 3 SİYAH, 9 BEYAZ
Adım 2	8 BEYAZ, 2 SİYAH, 3 BEYAZ, 2 SİYAH, 7 BEYAZ
Adım 3	7 BEYAZ, 1 SİYAH, 7 BEYAZ, 1 SİYAH, 6 BEYAZ
Adım 4	6 BEYAZ, 1 SİYAH, 9 BEYAZ, 1 SİYAH, 5 BEYAZ
Adım 5	5 BEYAZ, 1 SİYAH, 11 BEYAZ, 1 SİYAH, 4 BEYAZ
Adım 6	4 BEYAZ, 1 SİYAH, 4 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 4 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım 7	4 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım 8	4 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım 9	4 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım10	4 BEYAZ, 1 SİYAH, 13 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım11	4 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 5 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım 12	4 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 3 BEYAZ, 1 SİYAH, 4 BEYAZ, 1 SİYAH, 3 BEYAZ
Adım 13	5 BEYAZ, 1 SİYAH, 4 BEYAZ, 3 SİYAH, 4 BEYAZ, 1 SİYAH, 4 BEYAZ
Adım 14	6 BEYAZ, 1 SİYAH, 9 BEYAZ, 1 SİYAH, 5 BEYAZ
Adım 15	7 BEYAZ, 1 SİYAH, 7 BEYAZ, 1 SİYAH, 6 BEYAZ
Adım 16	8 BEYAZ, 2 SİYAH, 3 BEYAZ, 2 SİYAH, 7 BEYAZ
Adım 17	10 BEYAZ, 3 SİYAH, 9 BEYAZ
Adım 18	22 BEYAZ
Adım 19	22 BEYAZ

ETKİNLİĞİN ALTERNATİF UYGULANIŞI-2

Görsellerin bilgisayarda nasıl kodlandığını anlamaları amacıyla Şekil 3.23'te verilen görselin kodlamasını yaparak rakamları bulmalarını sağlayınız. Her bir satırın rakamlarını, sırasıyla ilk rakam beyaz ifade edecek şekilde resmin yan tarafına yazmalarını isteyiniz.



Şekil 3.23. Saklanmış kodlar (MEB, 2017)

BÖLÜM 3

ÖLÇME VE DEĞERLENDİRME

Öğrencilerinizin bir emoji görseline göre adımları yazmalarını isteyiniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Problemlere göre algoritma oluşturma			
Bir problemi alt problemlere bölme			
Algoritmanın oluşturulması için en uygun kararın seçilmesi			
Nesnenin ortaya çıkması için sıralı görevleri yerine getirebilme			
Çeşitli şekilleri oluşturmaya yönelik örnekler verme			
Algoritma oluştururken tümevarım ve tümdengelim metotlarını uygulayabilme			
Bilgisayardaki kayıtlı verilerin nasıl tutulduğunu anlayabilme			

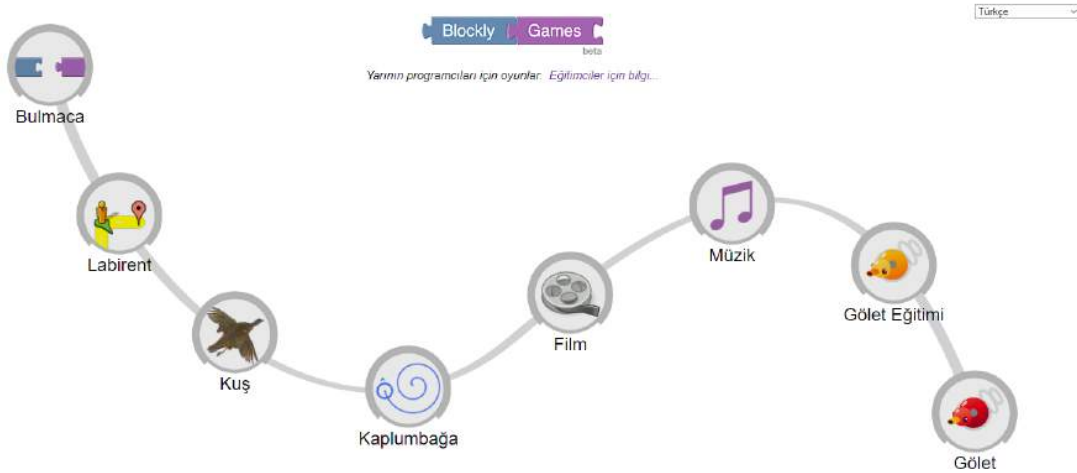
BÖLÜM 4

BLOK TEMELLİ PROGRAMLAMAYA GİRİŞ

Blok temelli programlama; çeşitli kod bloklarını birleştirerek uygulama, program, oyun vb.nin yapılmasıdır. Bloklar, ezberlenmeye gerek kalmadan sürükle-bırak mantığıyla çalışır. Arayüzleri küçük yaş gruplarına hitap edecek şekilde görsel unsurlar içerir ve daha çok ürüne yönelik uygulamalar geliştirilerek, öğrencilerin eğlenerek, oynayarak, uygulayarak ve son aşamada kendi özgün ürünlerini tasarlayarak öğrenmelerini sağlar.

4.1. Blockly

Blockly Google tarafından geliştirilen görsel tabanlı bir programlama aracıdır. Araç, daha önceden bilgisayar programlama deneyimi olmayan çocuklar için tasarlanmıştır. Uygulama 8 aşamadan oluşur (Şekil 4.1). Kodlama işlemi bloklarla ya da java script komutları ile yapılır.



Şekil 4.1. Blockly arayüzü

BÖLÜM 4

Bulmaca: Çeşitli resim ve blokların doğru şekilde birleştirilmesi amaçlanır.

Labirent: Döngüler ve koşullu ifadelerden oluşur. Verilen görevi tamamlamak için uygun algoritma oluşturulur. Bu bölüm, 10 aşamadan oluşur ve yapılan işlem her aşamada zorlaştırılır.

Kuş: Karakter, kendisine verilen görevleri yerine getirmeye çalışır. Koşullu durumlar ve döngü ön plana çıkar.

Kaplumbağa: Karaktere verilen komutlarla şekiller oluşturmak amaçlanır.

Film: Bu bölümde döngüler, çeşitli algoritmalarla matematiksel şekiller oluşturur. Şekillerin genişlik, yükseklik ve çapları ayarlanarak boyut kavramı öğretilir.

Müzik: Bu bölümde bloklarla müzik tonları oluşturulur. Farklı vuruş ve tonların ayarlanması sağlanır.

Gölet Eğitimi: Bu bölümde karaktere açı değeri ve mesafe kavramları ile kodlama yapılır.

Gölet: Öğrenilen bilgilerin tamamı kullanılarak bir oyun kurgulanır.

1. ETKİNLİK: BLOCKLY UYGULAMASI

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanıır.

Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.

Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.

Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar.

YÖNTEM/TEKNİKLER

Sunu, anlatım, gösterip yaptırma, oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, tablet

GİRİŞ

Aşağıdaki sorularla derse giriş yapılır:

- Daha önce puzzle yaptınız mı?
- Puzzle parçalarını birleştirmek için önce nerelerden başlarsınız?

SÜREÇ

- <https://blockly.games/> adresine giderek Şekil 4.3'te gösterilen sayfanın arayüzünü öğrencilerinize tanıtır.

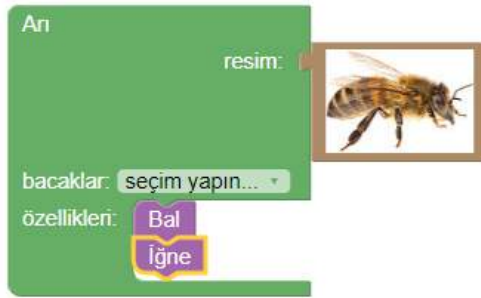
BÖLÜM 4



Şekil 4.2. Blockly arayüzü

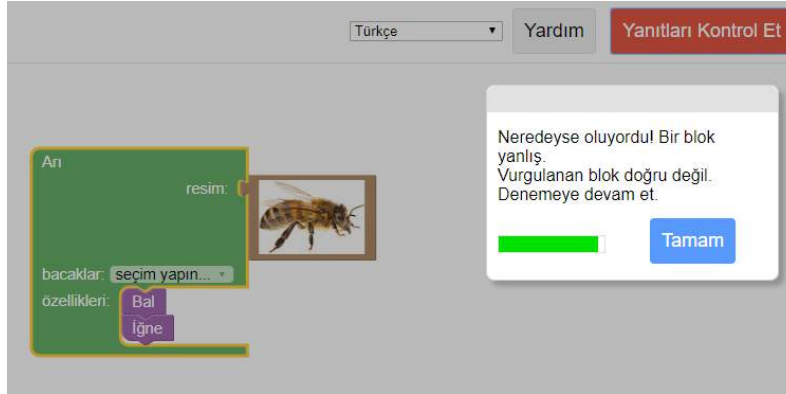
- İlk etkinlik olan “Bulmaca” etkinliğine girerek sayfada bulunan 4 adet uygulamadan birincisiyle ilgili aşağıdaki bilgileri öğrencilerinize aktarınız.

“Bu ekranda 4 hayvana ait yeşil blok, bu yeşil blok içinde de bacak sayıları seçeneği yer almaktadır. Bu hayvanların resimleri ve özellikleri ayrı bloklar âlinde görülmektedir. Hayvan resimlerinin yer aldığı bloklar, mouse yardımıyla sürükleyip bırakılarak yeşil blokta “resim” yazan yere eklenir. Hayvanlara ait özelliklerin yer aldığı mor bloklar da yeşil blok üzerinde “özellikleri” yazan boşluğa yine sürükleyip bırakarak yöntemi ile eklenir. Yeşil blokta adı verilen hayvanın bacak sayıları “bacaklar” kısmında seçilir.



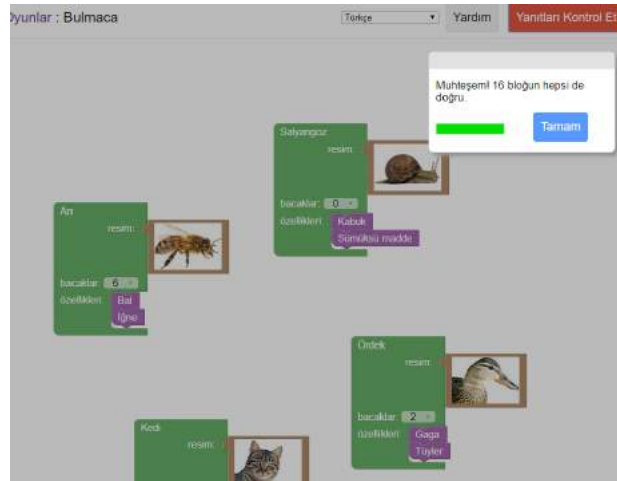
Şekil 4.3. Örnek bulmaca etkinliği

- İlk örneği öğrencilerinizle soru- cevap şeklinde yaparak gösteriniz. Sonra öğrencilerinizin kendi bilgisayarlarında bunu yapmalarını isteyiniz. Öğrencilerinizin sizin müdahaleniz olmadan kontrol etmelerini, hataları kendilerinin bulup düzeltmelerini sağlayınız.



Şekil 4.4. Yapılan etkinliğin doğruluğunun kontrol edilmesi

- “Burada ne gibi bir eksiklik ya da hata olmuş olabilir? Arıların bacakları var mıdır? Varsa kaç bacağı vardır?” gibi sorularla hata durumlarının çözümleri için öğrencilerinizi yönlendirebilirsiniz.
- Tüm hata durumları giderildikten sonra yanıtlar kontrol edilerek öğrencilerinizin etkinliği doğru bir şekilde bitirmelerini sağlayınız.



Şekil 4.5. Tüm eşleştirmeler doğru ise alınan cevap ekranı

BÖLÜM 4

ÖLÇME VE DEĞERLENDİRME

Tamamlanan etkinlikleri kontrol ediniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Kukla ile ilgili blokları eşleştirme			
Kuklaya ait özellikleri seçme			
Kuklanın bacak sayısını belirleme			

4.2. Code.org

Code.org, www.code.org adresinden girilen, temel kodlama becerisini müfredat tabanlı ve bilgisayar üzerinden gerçekleştirmenizi sağlayan bir platformdur. Code.org sitesine giriş yaptığınızda karşınıza dil seçeneği gelir.



Şekil 4.6. code.org dil seçeneği

Açılan ana sayfada “**Öğrenmeye başla**” linkine tıkladığınızda kursların olduğu sayfaya geçiş yapabilirsiniz.



Şekil 4.7. code.org giriş sayfası

Öğrencilerin seviyeleri ve yaşlarına göre “**Bilgisayar Bilimi Temelleri**” adı altında kurslar seçilerek etkinlikler yapılabilir.



Şekil 4.8. code.org sitesindeki kurslar

Her ne kadar kurslar, yaşlara göre ayrılmışsa da kodlamayla yeni tanışacak öğrenciler için **Kurs 1**'den başlamakta fayda vardır. Kurs 1'e giriş yapıldığında 18 etkinliğin olduğu bir liste ekrana gelecektir (Şekil 4.9).

BÖLÜM 4

Ders adı	İlerleme Durumu
1. Mutlu Haritalar	Bilgisayarsız Etkinlik 1
2. Oynat, Oynat	Bilgisayarsız Etkinlik 1
3. Yapboz: Nasıl sürükleyip bira...	1 2 3 4 5 6 7 8 9 10 11 12
4. Labirent: Sıra	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
5. Labirent: Hata Ayıklama	1 2 3 4 5 6 7 8 9 10 11 12
6. Gerçek-yaşam Algoritmaları: ...	Bilgisayarsız Etkinlik 1 2
7. Arı: Sıralama	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
8. Aktör: Sıra	1 2 3 4 5 6 7 8 9 10 11 12
9. Bir Temel Oluşturma	Bilgisayarsız Etkinlik 1
10. Sanatçı: Şekiller	1 2 3 4 5 6 7 8 9 10
11. Heceleme Yarışması	1 2 3 4 5 6 7 8 9 10 11 12

Şekil 4.9. Kurs 1'deki etkinlik listesi (Yeşiller gerçekleştirilen etkinlikler göstermektedir.)

Burada 4. etkinlik olan **“Labirent: Sıra”** etkinliği ile başlanarak çocukların verilen görevleri yapması amaçlanmaktadır.

Bilgisayarsız kodlama etkinliğine benzer olarak bloklar, çalışma alanına, çalıştığı zaman bloğunun altına sürüklenerek bloklar birleştirilir ve çalıştır düğmesine basılır. Bu bölümdeki 18 etkinliğin tamamlanması sağlanır. Kurs 1 bittikten sonra diğer kurslara geçiş yapılabilir. Sitede ayrıca öğrencilere yönelik farklı etkinlikler de mevcuttur.

2. ETKİNLİK: HAYVANCIĞI YAKALAYALIM

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanır.

Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.

Blok tabanlı programlama aracında sunulan bir programı, verilen ölçütlere göre geliştirerek düzenler.

Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar.

YÖNTEM/TEKNİKLER

Sunu, anlatım, gösterip yaptırma, oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar/tablet, internet bağlantısı

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- Angry Bird isimli oyunu oynadınız mı? Bu oyunda hangi karakterler var?
- Yapacağınız etkinlikte “Angry Bird” isimli çizgi film karakterinin hayvancığı yakalaması için nasıl hareket etmesi gerektiğini öğreteceğinizi söyleyiniz.

SÜREÇ

- Code.org sitesinde **Kurs 1**'de bulunan **4. Labirent: Sıra** etkinliğini açınız.

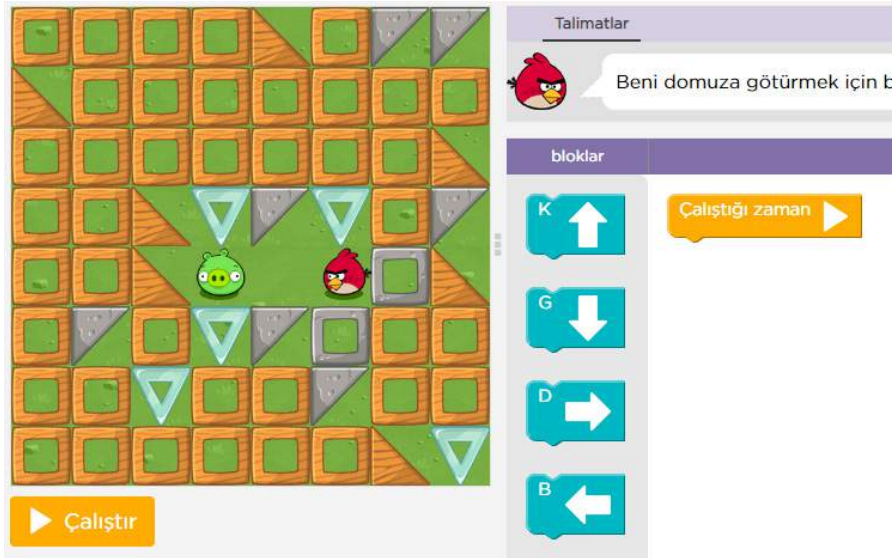
BÖLÜM 4

Ders adı	İlerleme Durumu
1. Mutlu Haritalar	Bilgisayarsız Etkinlik 1
2. Oynat, Oynat	Bilgisayarsız Etkinlik 1
3. Yapboz: Nasıl sürükleyip bira...	1 2 3 4 5 6 7 8 9 10 11 12
4. Labirent: Sıra	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
5. Labirent: Hata Ayıklama	1 2 3 4 5 6 7 8 9 10 11 12

Şekil 4.10. Kurs 1 etkinlik listesi

- Öğrenci seviyesine göre öğrencilerinizin pratik kazanması için ilk üç etkinliği yaptırınız. 4. etkinlik uygulamasında öğrencilerinize aşağıdaki bilgileri vererek ilk örneği siz yapınız.

“Hayvancığa gidilmesi için gereken yönü belirten oklar, mavi bloklar içinde yer almaktadır. Doğru yönü belirten mavi bloğu, çalışma alanında, “**çalıştığı zaman**” bloğunun altına sürükleyerek bloklar birleştirilir ve “**Çalıştır**” düğmesine basılarak işlemin doğru yapılıp yapılmadığı kontrol edilir.”



Şekil 4.11. 4. Etkinlikteki ilk uygulama

BÖLÜM 4

- İlk uygulamayı öğrencilerinizle beraber yaptıktan sonra bu bölümdeki 12 etkinliğin tamamlanmasını sağlayınız. Öğrencilerinize yön, sağa-sola dönme ve adım sayıları konusunda ara açıklamalar yaparak yardımcı olunuz.

ÖLÇME VE DEĞERLENDİRME

Sayfanın üst alanında bulunan etkinlik şemasından eksik olan ya da doğru tamamlanmamış etkinlik olup olmadığını kontrol edilir.



Şekil 4.12. Web sayfasındaki etkinlik takip listesi

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Doğru blokları birbirleriyle eşleştirme			
Yön kavramını öğrenme (K-G-D-B)			
Kuklanın bacak sayısını belirleme			

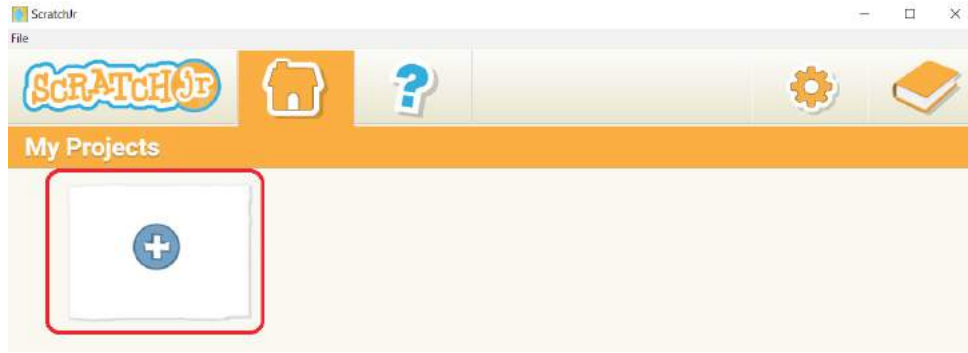
BÖLÜM 4

4.3. ScrtachJr

ScratchJr, dünyada pek çok öğrencinin kullandığı Scratch uygulamasının okul öncesi öğrenciler için geliştirilmiş bir platformudur. Uygulama, 5-7 yaş arası çocukların kendi interaktif hikâye, animasyon ve oyunlarını yapabildiği, aynı zamanda programlama öğrenmeye yardımcı olabilecek bir uygulamadır.

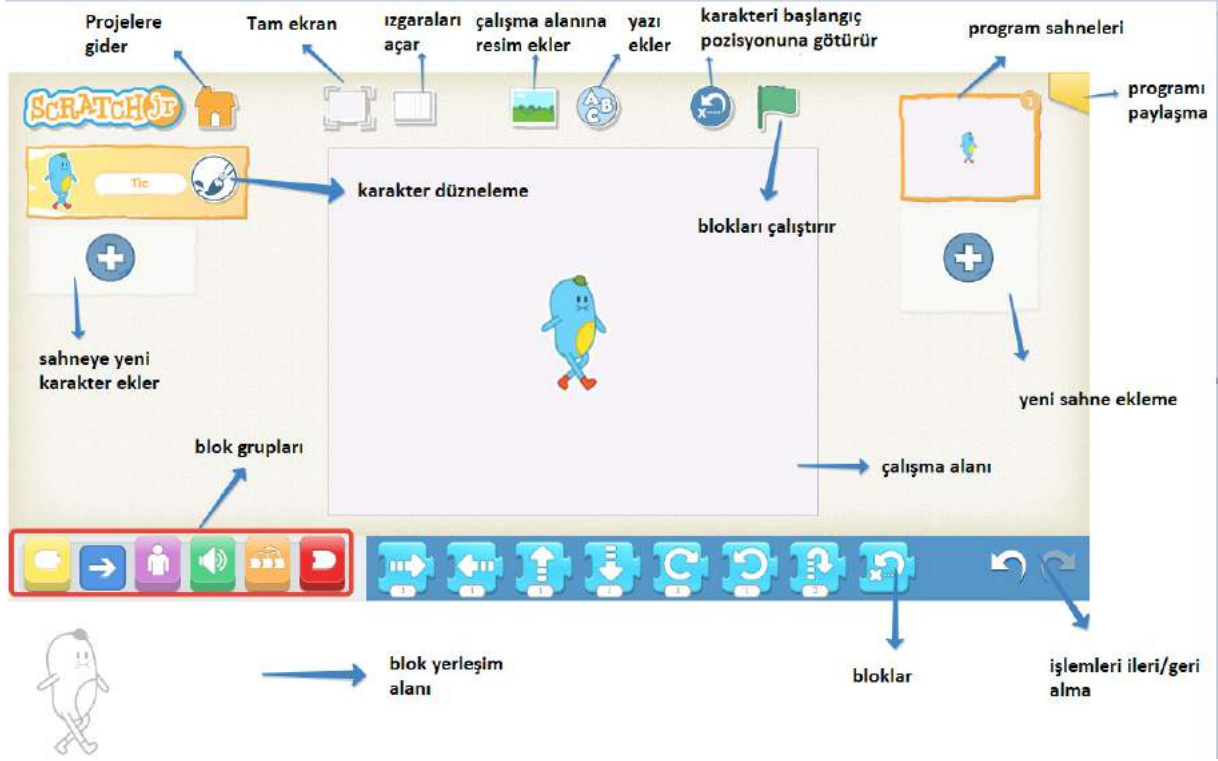
Bu uygulama; bilgisayara, tablete kurulabildiği gibi online olarak da kullanılabilir. Kurulum için <https://jfo8000.github.io/ScratchJr-Desktop/> bağlantısından indirilir. Tablet ve telefon üzerinden, uygulama marketinden ücretsiz olarak da yüklenebilir.

ScrtachJr uygulaması açılarak yeni bir proje oluşturulur.



Şekil 4.13. ScrtachJr uygulamasında yeni proje ekleme

Uygulamanın arayüzü Şekil 4.14'teki gibidir. Masaüstü sürüm ve tablet sürümündeki karakterler birbirinden farklı olabilir.



Şekil 4.14. Scratch Jr uygulama ekranı ve simgelerin görevleri

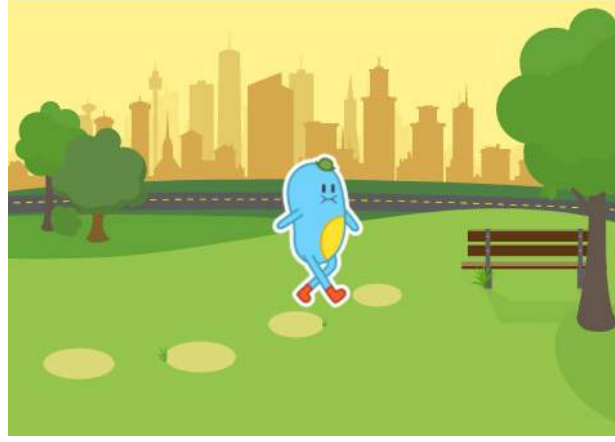
Uygulama açıldığında çalışma alanı boş gelir. Pencerenin üst tarafında bulunan "Çalışma alanına resim ekleme" butonu aracılığı ile arka plan resmi eklenebilir. Uygulama içinde Şekil 4.15'te gösterildiği gibi çöl, ova, orman, oda gibi pek çok arka plan bulunur.

BÖLÜM 4



Şekil 4.15. Arka plan ekleme ekranı

Eklenmek istenen arka plan resmi seçilip sağ üst tarafta bulunan onay işaretine basılarak arka plan resmi değiştirilebilir. Kullanıcı, ayrıca kendi arka plan resmini de oluşturabilir.



Şekil 4.16. Eklenen arka plan ile karakterin birlikte görüntüsü

ScratchJr içerisinde karakter olarak pek çok insan, hayvan, bitki ve eşya bulunmaktadır. Bunun haricinde ilgili karakterin üzerindeki palet işaretine tıklanarak da yeni karakter çizilebilir.

Uygulamada karakterlere hareket verebilmek için alt tarafta bulunan blok grupları kullanılır (Şekil 4.17). Bunun için karakter seçili durumda iken **olaylar** blok grubunda bulunan bloklardan faydalanılır.



Şekil 4.17. Karakterlere hareket verme blokları

Uygulama penceresinin üst tarafında bulunan yeşil bayrağa tıklanarak uygulama başlatılabilir. Alternatif olarak; karakterin üzerine tıklanınca, karakterler birbirine değince gibi farklı aksiyonlarla da uygulamanın ya da karakterin harekete geçmesi sağlanabilir. Aynı şekilde **hareket** blok grubu üzerinde de karakterlerin hareket edebilmesi için bloklar bulunur (Şekil 4.18).



Şekil 4.18. Hareket blok gurubu

Hareket blokları sayesinde karaktere ileri – geri, yukarı – aşağı, zıpla- başa dön gibi komutlar verilebilir. Bu blokların alt tarafında bulunan sayılar ise bloğun kaç defa çalışacağını gösterir.

3. ETKİNLİK: KENDİ ÇİZGİ FİLMİMİZİ YAPALIM

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Blok tabanlı programlama aracının arayüzünü ve özelliklerini tanıır.

Blok tabanlı programlama aracında sunulan bir programın işlevlerini açıklar.

Blok tabanlı programlama aracında sunulan bir programı verilen ölçütlere göre geliştirerek düzenler.

Blok tabanlı programlama aracında sunulan bir programın hatalarını ayıklar.

YÖNTEM/TEKNİKLER

Sunu, anlatım, gösterip yaptırma, oyun temelli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, tablet

GİRİŞ

Aşağıdaki sorularla derse giriş yapılır:

- En sevdiğiniz çizgi film karakteri hangisidir?
- Siz de kendi çizgi filminizi yapmak ister misiniz?

SÜREÇ

- ScratchJr arayüzünü öğrencilerinize tanıttikten sonra karakter seçili iken aşağıdaki blok dizisini oluşturunuz. Blokların altındaki rakamların nasıl değiştirildiğini anlatarak bu rakamların istenilen hareketin tekrar sayısı olduğunu belirtiniz.



Şekil 4.19. Yapılması istenen blok dizisi

- Ardından uygulamanın üst tarafında bulunan yeşil bayrağa tıklayarak uygulamayı çalıştırınız.



Şekil 4.20. Yaptığınız uygulamayı çalıştırma butonu

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

ScratchJR uygulaması ile sahneye bir arka plan resmi eklenerek aşağıdaki gibi karakterin hareket ettirilmesi sağlanır.



Şekil 4.21. Yapılması istenen blok dizisi

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmalarını kontrol ediniz. Öğrencilerinize ikinci bir karakter ekleterek farklı uygulamalar yapmalarına olanak sağlayınız.

Öğrencilerine kısa senaryolar yazdırarak bunları uygulamalarını isteyiniz.

BÖLÜM 4

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Sahnenin arka plan resmini değiştirme			
Tüm blokları doğru şekilde eşleştirme			
Doğru karaktere doğru kodları ekleme			

4.4. Scratch

Scratch; MIT tarafından geliştirilen, 8-16 yaş arası çocukların kodlamayı öğrenebilmesi için tasarlanmış, çok basit bir arayüze sahip bir programlama aracıdır.

İlk çıkış yılı 2003'tür. Şu an Scratch programının en son versiyonu olan Scratch 3'ü <https://scratch.mit.edu/download> adresinden indirilerek bilgisayara kurulabildiği gibi, <https://scratch.mit.edu/projects/editor/?tutorial=getStarted> adresinden de programı indirmeden tarayıcı üzerinden çalışmalar yapılabilir.

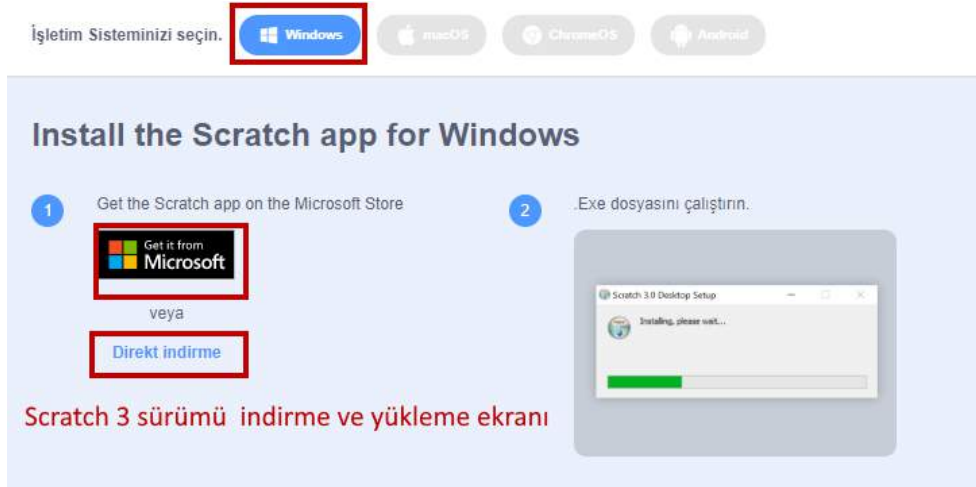
4.4.1. Scratch Kurulumu

Scratch masaüstü yazılımını bilgisayara yükleyerek çalışmak için <https://scratch.mit.edu/download> adresine gidilerek bilgisayarda yüklü işletim sistemine uygun Scratch sürümü seçilmelidir (Şekil 4.22). Bu etkinliğin yazıldığı gün itibarıyla en güncel Scratch masaüstü sürümü 3.9.0'dır. Scratch masaüstü sürüm 3.9'u bilgisayarınıza indirdikten sonra çalıştırıp varsayılan ayarlarıyla kurulum yapılmaktadır.



Şekil 4.22. İşletim sistemi seçme ekranı

İşletim sistemini seçtiğinizde sayfanın altında Şekil 4.23'teki gibi indirme ekranı gelir. Buradan "**Direkt İndir**" butonuna tıklanarak bilgisayarda belirlenen yere kurulum dosyası indirilmektedir.

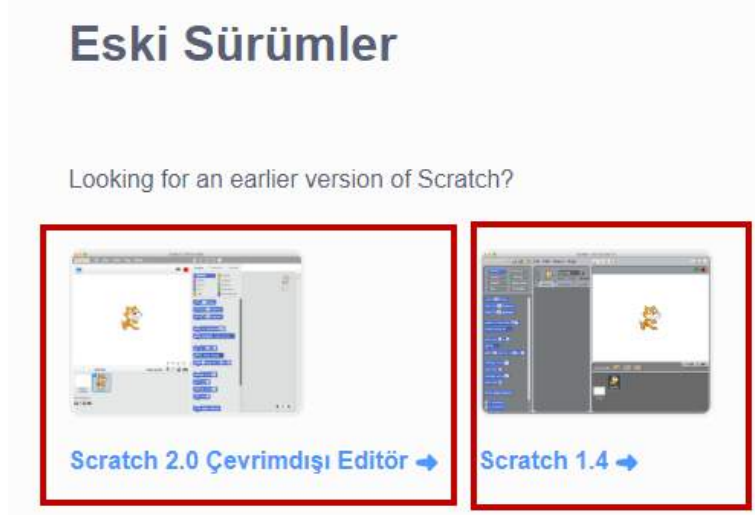


Şekil 4.23. Seçilen işletim sistemine göre Scratch sürümünü indirme ekranı

BÖLÜM 4

İndirilen .exe uzantılı dosya üzerine çift tıklanarak açılan pencerelerde sırasıyla çalıştır, kur ve bitir butonlarına basıldığında Scratch programı bilgisayara kurulur.

Scratch'in daha eski sürümleri olan 1.4 veya 2.0 sürümleri de Şekil 4.24'te görülen ekran üzerinden bilgisayara yüklenebilir.



Şekil 4.24. Scratch eski sürümleri

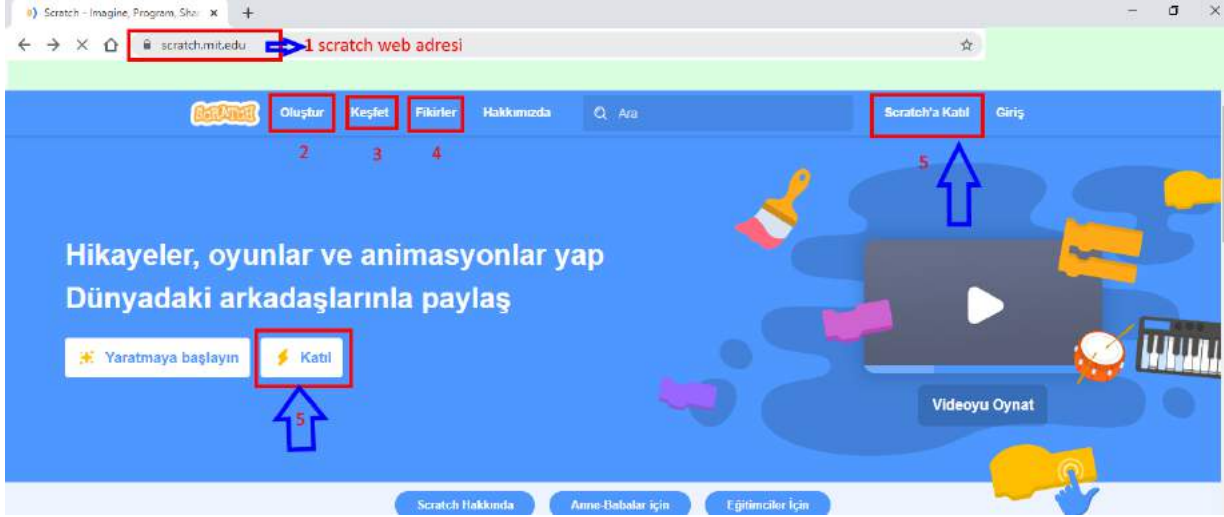
4.4.2. Web Arayüzünden Scratch Çalıştırma

<https://scratch.mit.edu/> web sitesine girildiğinde karşınıza Şekil 4.26'daki görüntü gelir.

Bu ekranda doğrudan proje oluşturmak için **Oluştur** butonuna (2), hazır projeleri görmek için **Keşfet** butonuna (3) ve öğretici etkinliklerini incelemek için **Fikirler** butonuna (4) tıklanır. **Katıl** (5) butonuna tıklayarak Scratch e üye olunur. Üye olmanın avantajı, yapılan projelere internet bağlantısının olduğu her yerden ulaşılarak farklı bilgisayarlar ile de çalışılabilir olmasıdır. Aşağıdaki adımlar sırayla uygulanarak Scratch hesabı oluşturulabilir.

İlk adımda bir kullanıcı adı ve bu kullanıcı adı ile kullanılacak parola oluşturulur. Oluşturulan parolanın güçlü bir parola olması, internet güvenliği için önemlidir. **Sonraki** butonuna tıklanarak ülke seçme adımına geçilir.

BÖLÜM 4



Şekil 4.25. Scratch web sayfası

The image shows the 'Scratch'a Katıl' registration form. It includes a title 'Scratch'a Katıl' and the subtitle 'Create projects, share ideas, make friends. It's free!'. The form has three main sections: 'Create a username' with a text input field labeled '1' and a red box 'Geççek adını kullanma'; 'Bir parola oluştur' with a text input field labeled '2' and a red box 'Unutmayacağıın Güçlü bir şifre oluştur'; and a section for '3 Tücc password again' with a checked checkbox 'Yeni şifre'. A red box '4 Sonraki' is at the bottom.

Şekil 4.26. Kullanıcı adı ve parola oluşturma ekranı

The image shows the 'Scratch'a Katıl' registration form with a country selection dropdown. The title is 'What country do you live in?' and the dropdown menu is open, showing 'Turkey' selected. A red box '4 Sonraki' is at the bottom.

Şekil 4.27. Katılım sağlanan ülke seçimi

Kullanıcı, doğum tarihinin ay ve yılını seçtikten sonra cinsiyetini seçerek sonraki adıma geçer. Burada kullanıcı, 18 yaşından küçük ise onay için veli e-posta adresi istenir.

BÖLÜM 4

When were you born?
Doğum tarihimizi seçiyoruz

Ay ▼ Yıl ▼

We will keep this information private. ⓘ

* yaşımızın 18 den küçük olması durumunda veli e posta adresi isteyeceğğini hatırlayalım

Sonraki

Şekil 4.28. Doğum tarihi giriş ekranı

Cinsiyetimizi soruyor
What's your gender?
Scratch welcomes people of all genders.

Kadın
 Erkek
 Non-binary
 Another gender:
 Prefer not to say * belirtmek istemeyenler için

We will keep this information private. ⓘ

Sonraki

Şekil 4.29. Cinsiyet seçme ekranı

Son adım olarak kullanıcı kendisine ait olan bir e-posta adresi girer. Bu e-posta adresine bir onay mesajı gönderildiği bilgisi ekranda görünür.

What's your email?
E posta adresimizi girmemiz gerekiyor

E-posta adresi

We will keep this information private. ⓘ

I'd like to receive emails from the Scratch Team about project ideas, events, and more.

By creating an account, you acknowledge the Gelişik İlerlemiz and you accept and agree to the Kullanım Şartları.

Artık hesap oluşturabiliriz
Create Your Account

Şekil 4.30. Mail adresi giriş ekranı

Welcome to Scratch,
Scratch kullanıcı adı yazacak

You're now logged in! You can start exploring and creating projects.

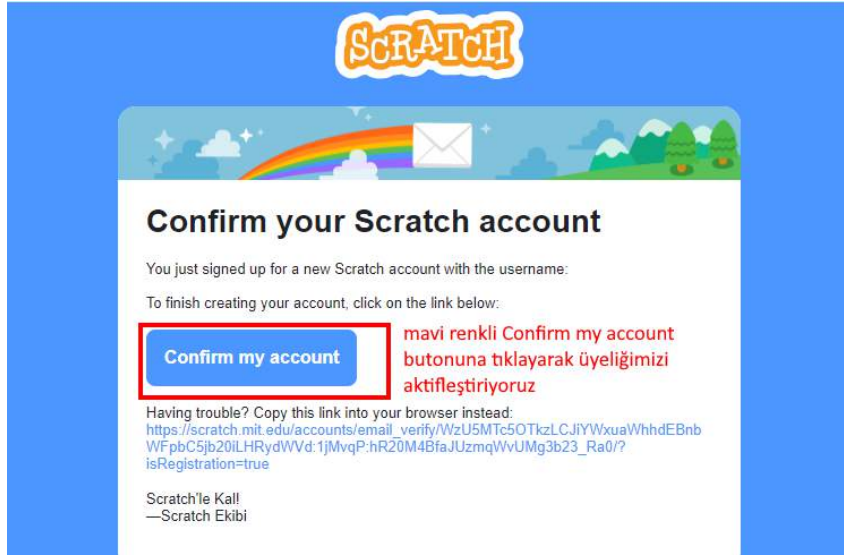
Want to share and comment? Click the link on the email we sent to

* eposta adresimize gelecek olan posta ile ücretsiz olan üyeliğimizi tamamlıyoruz

Başlayın →

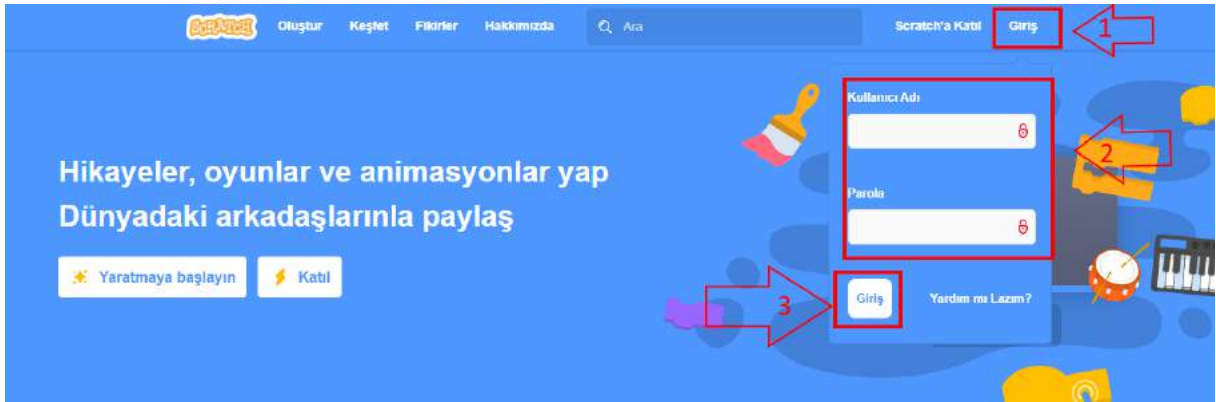
Şekil 4.31. Mail gönderildiği bilgi ekranı

Kayıt işlemleri tamamlandıktan sonra Şekil 4.30'da yazılan e-posta hesabına Scratch tarafından gönderilen e-posta açılıp Şekil 4.31'deki ekranda görülen **"Confirm my account"** butonuna tıklanarak üyelik işlemleri tamamlanır.



Şekil 4.32. Scratch hesabı aktivasyon e-postası

Üyelik işlemleri tamamladıktan sonra <https://scratch.mit.edu/> adresine gidilerek Şekil 4.33'te gösterildiği gibi sisteme giriş yapılır.



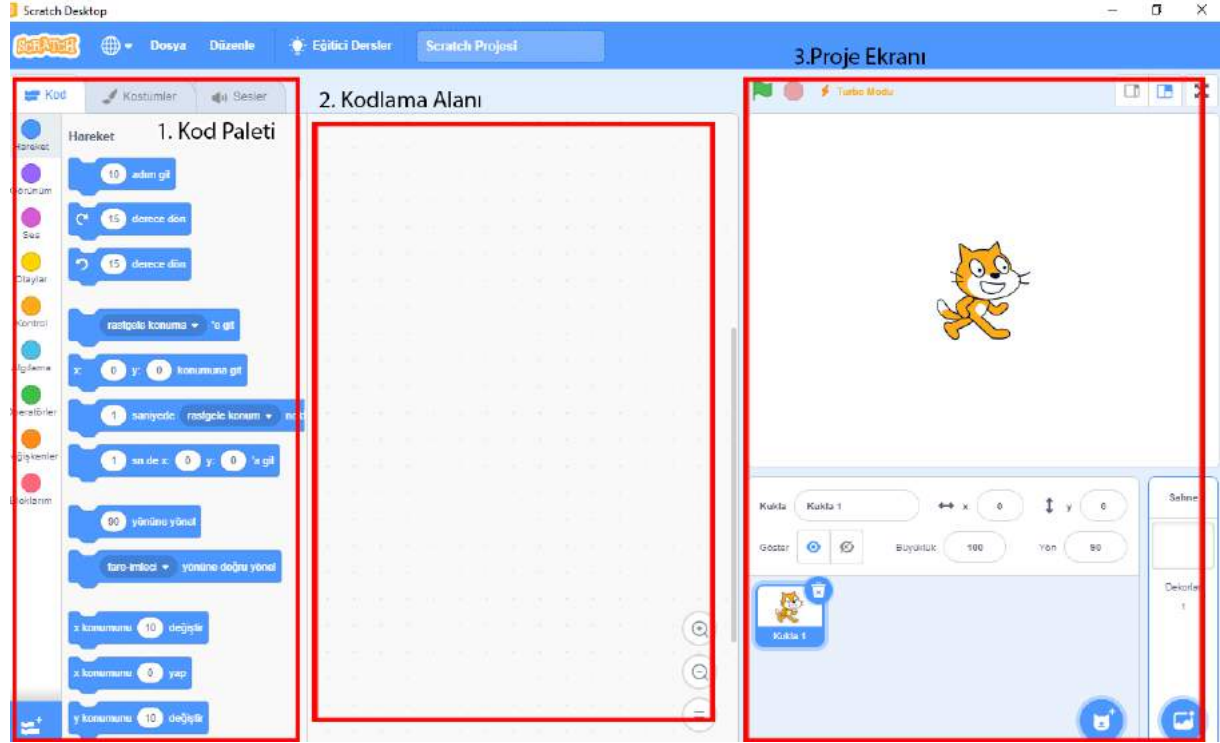
Şekil 4.33. Scratch 3 üyelik hesabı ile giriş yapma ekranı

BÖLÜM 4

4.4.3. Arayüz Tanıtımı

Scratch arayüzü; 1. Kod Paleti, 2. Kodlama Alanı, 3. Proje Ekranı olmak üzere Şekil 4.34'te de görüldüğü gibi üç alandan oluşmaktadır.

1. **Kodlama Paleti:** Kodlama paleti, kodlama görsellerinin alındığı alandır. Bu alandan görseller alınarak kodlama alanına "sürükle bırak" ile atılır.
2. **Kodlama Alanı:** "Sürükle bırak" ile atılan kodların çalıştırıldığı alandır.
3. **Proje Ekranı:** Sahnenin ve kuklaların bulunduğu alandır. Kodların çalışması bu alanda incelenebilir.



Şekil 4.34. Arayüz tanıtımı

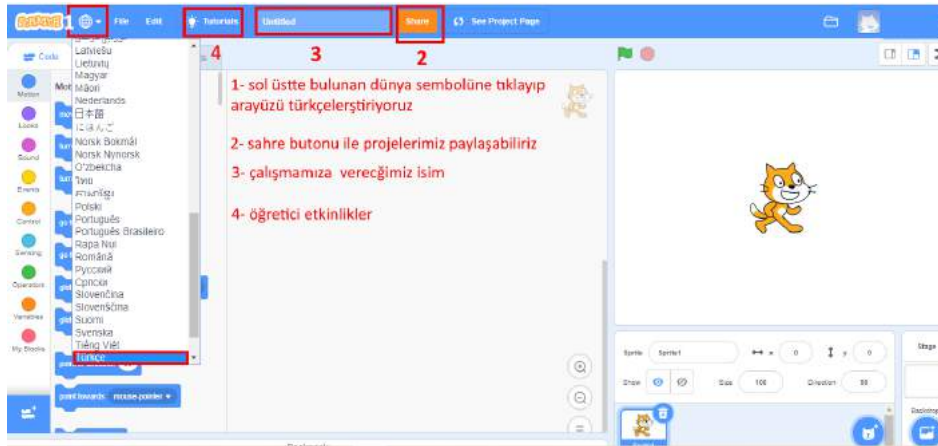
4.4.3.1. Scratch Proje Oluşturma

Scratch ana sayfasında, Şekil 4.35'te gösterildiği gibi **“Oluştur”** düğmesine tıklanarak yeni bir proje oluşturulabilir. Bu ekranda profil ayarları da değiştirilebilir.



Şekil 4.35. Yeni bir proje oluşturma ve hesap ayarları ekranı

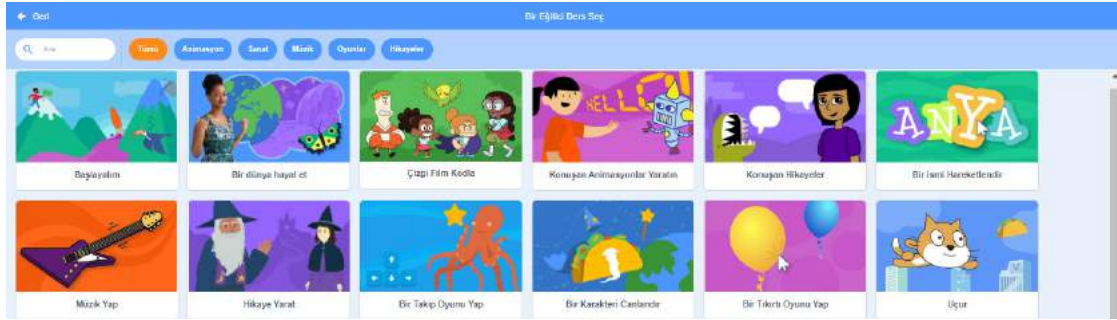
Oluştur tıklandıktan sonra Şekil 4.36'da görülen Scratch kodlama arayüzü ekrana gelir. Arayüzü, Türkçeleştirmek için sol üstte bulunan ve dünya sembolü ile gösterilen menüden Türkçe seçilerek dil tercihi yapılabilir.



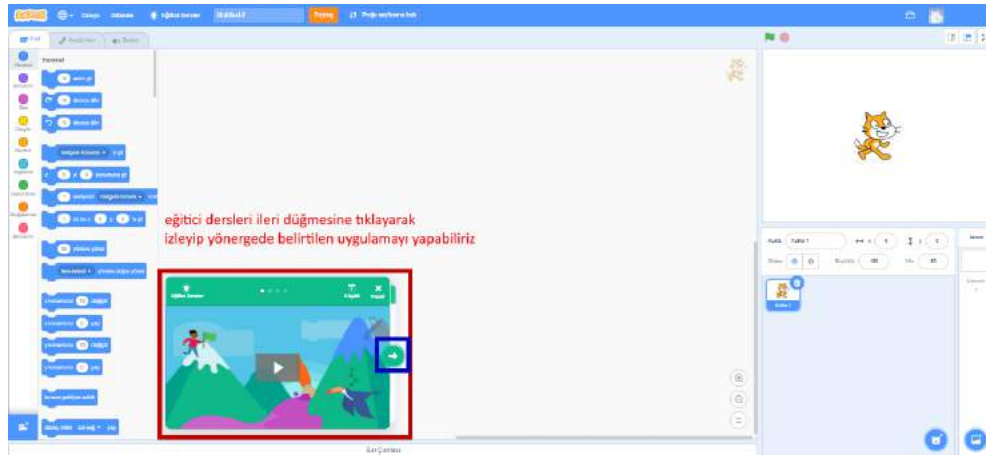
Şekil 4.36. Scratch web arayüzü

BÖLÜM 4

Untitled kısmına projenin adı yazılır. **Tutorials** (Eğitici dersler) menüsüne tıklayarak Şekil 4.37’de gösterilen eğitici derslerden herhangi birini seçip Şekil 4.38’deki gibi yönergeleri takip ederek ilk uygulamaya başlanabilir. **Share** butonu ile yapılan proje paylaşılabilir.

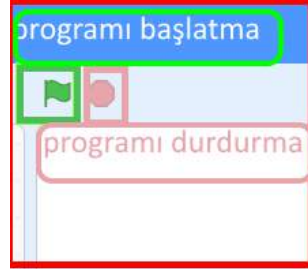


Şekil 4.37. Scratch eğitici dersler ekranı

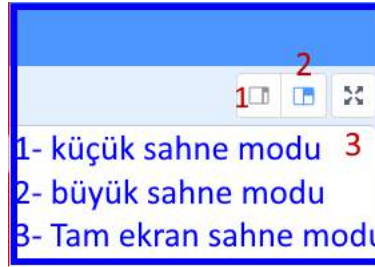


Şekil 4.38. Bir eğitici ders ekranı örneği

Arayüzün sağ üst kısmında Şekil 4.39’da gösterildiği gibi proje başlatma ve durdurma düğmeleri ile Şekil 4.40’ta arayüz görünüm türleri bulunmaktadır. Varsayılan olarak Şekil 4.40’ta (2) rakamıyla gösterilen büyük sahne modu açılmaktadır. Daha fazla kodlama alanına ihtiyaç duyulduğunda (1) rakamıyla gösterilen küçük sahne modu simgesine tıklanarak geçilebilmektedir. Projenin test edilmesinde tam ekran çalışma deneyimine ihtiyaç duyulur ise (3) rakamıyla gösterilen simgeye tıklanarak tam ekran moduna geçiş yapılabilir.



Şekil 4.39. Proje başlatma ve durdurma

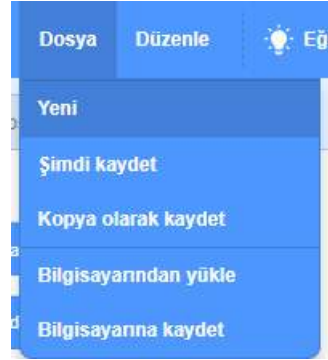


Şekil 4.40. Projede ekran görünürlüğü modları

4.4.3.2. Proje Kayıt

Ekranın en üst kısmındaki menüde **Dosya** ve **Düzenle** sekmeleri bulunur. Dosya menüsünde Şekil 4.41’de görüldüğü gibi dosya ile ilgili oluşturma ve kaydetme seçenekleri vardır. Kaydetme seçeneklerinde dikkat edilmesi gereken nokta, **web ortamı ve kişisel bilgisayar için ayrı kayıt yapılmasıdır**. Kişisel bilgisayarda bulunan bir projenin içeriği hem masaüstü sürümüne hem de web sürümüne aktarılabilir. Böylelikle önceden başlanmış bir projeye kalınan yerden ister web ister masaüstü ortamdan devam edilebilir. Scratch masa üstü sürümü kullanılıyor ise **öncelikle yazılımı açmak ardından dosya menüsü aracılığıyla projeyi içe aktarmak gerekir**. Scratch sürüm 3 dosyaları.sb3 dosya uzantısına sahiptir. Scratch sürüm 2 dosyaları ise .sb2 dosya uzantısına sahiptir.

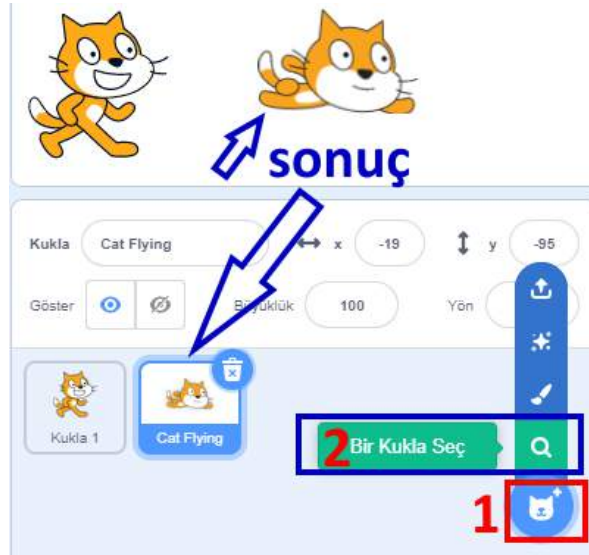
BÖLÜM 4



Şekil 4.41. Scratch dosya menüsü

4.4.3.3. Kukla Ekleme

Bir kukla eklemek için öncelikli olarak Şekil 4.42’de gösterildiği gibi “bir kukla seç” butonuna tıklanır. Kuklaların olduğu Şekil 4.43’deki ekran gelir. Kukla seç ekranında hayvanlar, insanlar, fantezi, dans... gibi birçok kategoride kukla mevcuttur. Bu kuklalardan biri seçildiğinde, seçilen kukla hem sahneye hem de kuklalar bölümüne eklenir.



Şekil 4.42. Kukla ve sahne bileşenleri



Şekil 4.43. Kukla seçme ekranı

Kukla eklemenin farklı yolları bulunmaktadır. Bunlar:

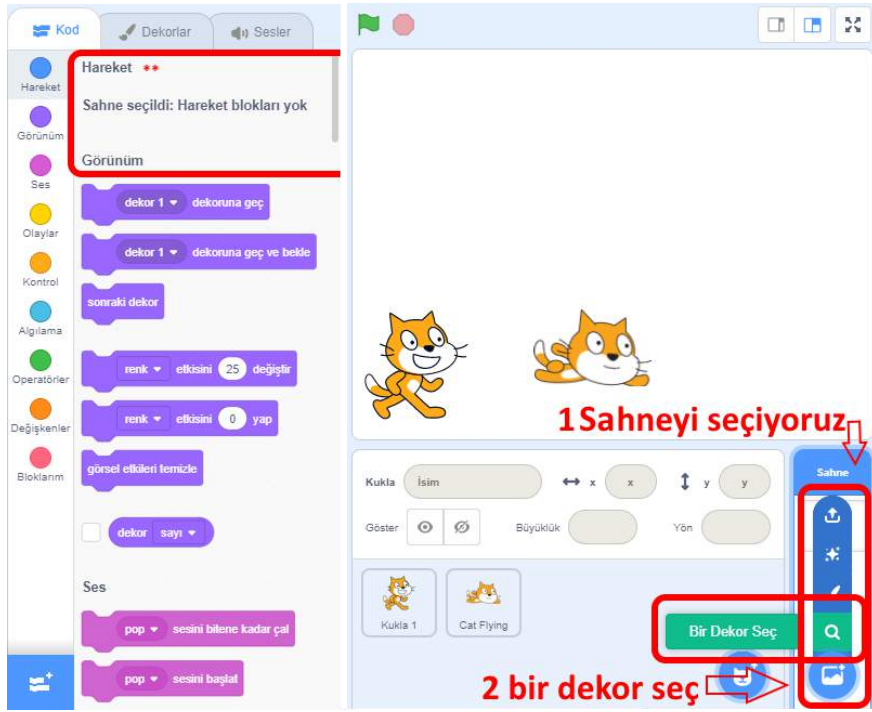
- Kütüphaneden seçmek
- Çizim ekranı ile kukla çizmek
- Bilgisayarınızdaki dosyadan veya internetten indirdiğiniz resimleri kukla olarak içe aktarmak
- Başka bir projeden kukla indirip çalışılan projeye kuklaya ait kodlar eklemek

4.4.3.4. Dekor Ekleme

Kukla eklemeye benzer yöntemler kullanılarak sahneye dekorlar eklemek de mümkündür. Sahneye hareket blokları dışında kodlar da yazılabilmektedir.

Sahneye bir dekor eklemek için Şekil 4.44'teki ekranda "**Sahne**"yi seçip ardından "**dekor ekle**" butonuna tıkladığında Şekil 4.45'te görülen dekor seçme ekranı gelmektedir. Dekor seçme ekranında fantezi, müzik, sporlar, bina dışı... gibi birçok kategoride dekor mevcuttur.

BÖLÜM 4



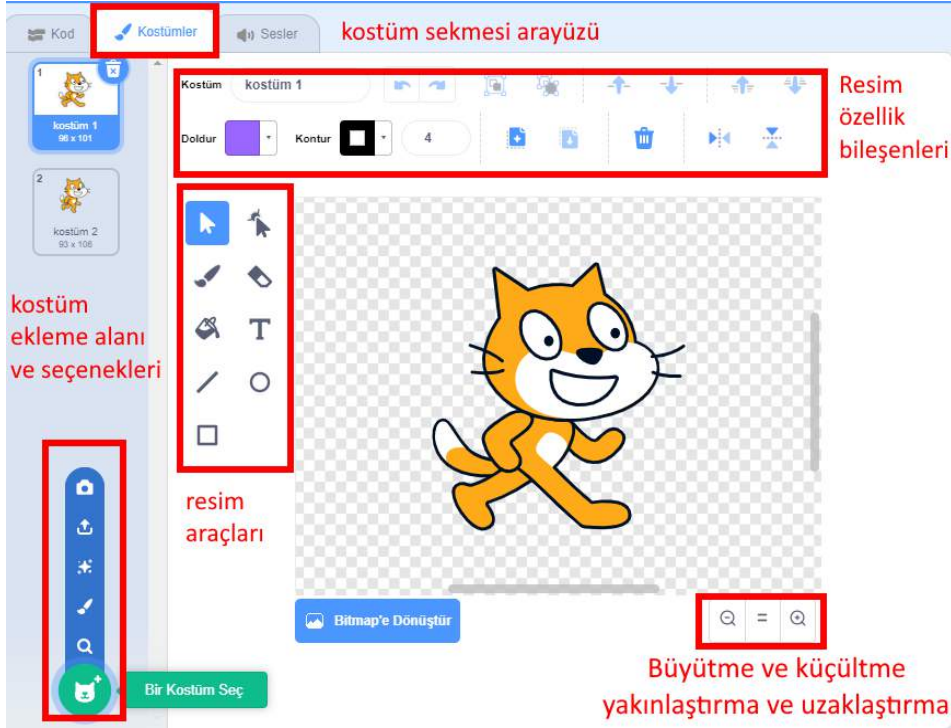
Şekil 4.44. Dekor ekleme aşamaları



Şekil 4.45. Dekor seçme ekranı

4.4.3.5. Kostüm Ekleme

Kostümler sekmesine tıklandığında seçilen kuklaya ait kostüm sekmesi Şekil 4.46'daki gibi açılır. Bu sekme ile kuklalar için kostümler eklenebilir, kostümlere görsel eklenir ve düzenlemeler yapılabilir.

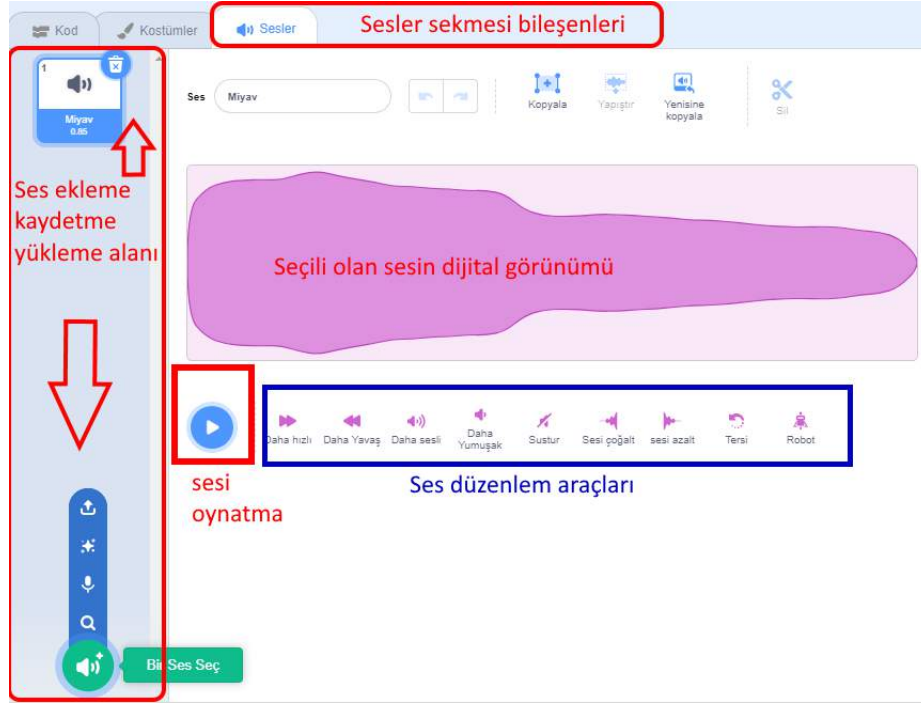


Şekil 4.46. Kostümler sekmesi alanı görünümü

4.4.3.6. Ses Ekleme

Sesler sekmesinde ise Şekil 4.47'de görülen ekranda seçilen kuklaya ait sesler ile ilgili işlemler yapılabilir. Ses ekleme, düzenleme ve ses kaydı burada yapılır.

BÖLÜM 4



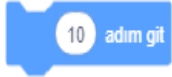

Şekil 4.47. Sesler sekmesi alanı görünümü

4.4.4. Kodlama Blokları



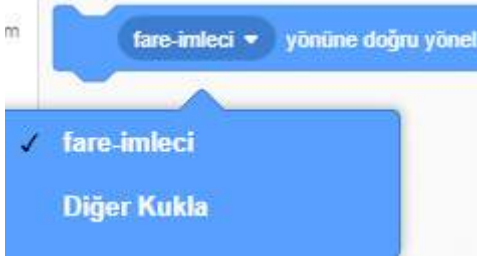


4.4.4.1. Hareket



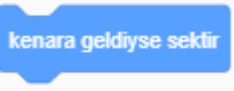


Sahnedeki kuklanın hareket etmesi, yön değiştirmesi gibi komutları içeren kategoridir. Sahnelerin herhangi bir hareketi olamayacağından sahne seçili olduğunda hareket kod bloklarında herhangi bir blok görünmeyecektir. Hareket kategorisinde bulunan bloklar ve açıklamaları Tablo 4.1’de gösterilmektedir.

Tablo 4.1. Hareket blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Kuklayı, belirtilen adım kadar hareket ettirir.
2		Kuklanın saat yönünde belirtilen açı değeri kadar (15 derece) dönmesini sağlar.
3		Kuklanın saat yönünün tersinde belirtilen açı değeri kadar (15 derece) dönmesini sağlar.
4		Kuklanın rastgele bir konuma, fare imlecine veya sahnedeki diğer kuklalardan herhangi birine gitmesini sağlar.
5		Kuklanın belirtilen x ve y koordinatlarına (x:39 y:44) gitmesini sağlar.
6		Kuklanın belirtilen sürede (1 saniye), rastgele bir konuma, fare imlecine veya sahnedeki kuklalardan herhangi birine gitmesini sağlar. Bu blok sayesinde kukla, sahne üzerinde süzülme eylemini gerçekleştirir.

BÖLÜM 4

7		Kuklanın belirtilen sürede (1 saniye), belirtilen koordinata (x:39 y:44) gitmesini sağlar.
8		Kuklanın belirtilen açı yönüne doğru dönmesi sağlar.
9		Kuklanın fare imleci veya sahnedeki diğer kuklalardan birine doğru dönmesini sağlar.
10		Sahnedeki kuklanın x eksenindeki konumunu belirtilen değer kadar (10 birim) değiştirmek için kullanılır. Kuklanın sağa gitmesi için x eksenindeki konum değeri 10 artırılırken kuklanın sola gitmesi için x eksenindeki konum değeri -10 değiştirilir.
11		Sahnedeki kuklanın x ekseninde belirli bir noktaya (39) gitmesi sağlanır.

<p>12</p>		<p>Sahnedeki kuklanın y eksenindeki yerini, belirtilen değer (10 birim) kadar değiştirmek için kullanılır. Kuklanın sahnenin üst kısmına gitmesi için y eksenindeki değeri 10 arttırılırken sahnenin alt kısmına doğru hareket etmesi için y eksenindeki değeri -10 değiştirilir.</p>
<p>13</p>		<p>Sahnedeki kuklanın y ekseninde belirli bir noktaya (44) gitmesi sağlanır.</p>
<p>14</p>		<p>Hareket hâlindeki kuklanın, sahnenin kenarlarına geldiği anda yönünün değişmesini sağlar.</p>
<p>15</p>		<p>Kuklanın sağa-sola dönme, etrafında dönebilme ve hiç dönememe ayarlarını yapar.</p>
<p>16</p>		<p>Kuklanın x pozisyonu bilgisini verir. Bu seçenek işaretlendiğinde x değeri ekranda gösterilir.</p> <p>Kuklanın y pozisyonu bilgisini verir. Bu seçenek işaretlendiğinde y değerini ekranda gösterilir.</p> <p>Kuklanın yön bilgisini verir. Bu seçenek işaretlendiğinde yön bilgisi ekranda gösterilir.</p>





BÖLÜM 4

4.4.4.2. Görünüm

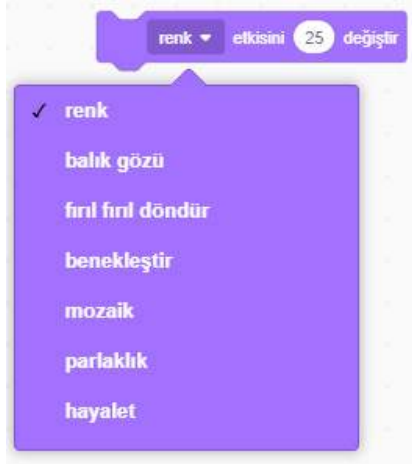

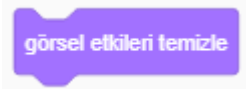

Görünüm kategorisi, kuklaların ve sahnenin görünümüyle ilgili kod bloklarını barındırır. Görünüm kategorisindeki kod blokları Tablo 4.2’de gösterilmiştir.


Tablo 4.2. Görünüm kategorisi kod blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Girilen ifadenin girilen süre (2 saniye) kadar kuklanın konuşma balonunda görün-tülenmesidir.
2		Girilen ifadeyi kuklanın sürekli olarak söy-lemesini sağlar.
3		Kuklanın girilen süre kadar, girilen ifadeyi düşünüyor görünmesini sağlar.
4		Kuklanın sürekli olarak girilen ifadeyi düşü-nüyor görünmesini sağlar.
5		Kuklanın kostümlerinin seçilmesini sağlar.
6		Kuklanın kostümleri arasında sırasıyla geçiş yapmak amacıyla kullanılır.

7		<p>Sahne dekorlarından seçilen bir dekora veya rastgele bir dekora geçiş yapılmasını sağlar.</p>
8		<p>Sahne dekorları arasında sırasıyla dekor değiştirme işlemi yapılır.</p>
9		<p>Kuklanın boyutunu, belirtilen değerde değiştirir. Küçültme işlemi için – ifadesi kullanılır.</p>
10		<p>Kuklanın boyutunu % olarak değiştirmeyi sağlar. Küçültme işlemi için – ifadesi kullanılır.</p>

BÖLÜM 4

11		Kuklaya çeşitli efektleri belirtilen değerde eklemeyi ve azaltmayı sağlayan komuttur.
12		Belirtilen efektin sabit bir değerde kuklaya uygulanmasını sağlar.
13		Sahnedeki tüm görsel etkilerin temizlenmesini sağlar.
14		Sahnedeki gizlenmiş bir kuklanın görüntülenmesini sağlar.

15		Sahne de görüntülenen bir kuklanın gizlenmesini sağlar.
16		Sahne üzerinde kuklaların katmanlar şeklinde yerleştirilmesi sağlanır. Ön katmanda bulunan kukla, arka katmanda bulunan kuklanın üstündeymiş gibi görünür.
17		Sahnedeki kuklanın katmanlar arasında ileri veya geri hareket etmesini sağlar.
18		Seçili olan kuklanın aktif kostümünün tercihe göre ismini veya sırasını görüntüler. Sahne de aktif olan dekorun tercihe göre ismini veya sırasını görüntüler. Sahnedeki kuklanın büyüklüğünü görüntüler.



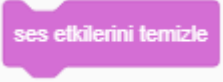


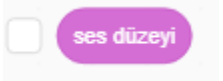
BÖLÜM 4

4.4.4.3. Ses

Projelerde kullanılacak önemli özelliklerden biri de ses bloklarıdır. Hem oyun projelerinde hem animasyonlar hazırlarken hem de çeşitli uygulamalarda ses ile kullanıcıyı uyarmak projenin ilgi çekici olmasını sağlar. Tablo 4.3'te ses kategorisinin kod blokları gösterilmektedir.

Tablo 4.3. Ses kategorisinin kod blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Seçilen ses dosyasının bitene kadar çalınmasını sağlar. Ayrıca bilgisayarın mikrofonu ile kendi ses dosyanızı da oluşturabilirsiniz.
2		Mevcut kütüphanelerden seçilen herhangi bir sesin veya kendi kaydedeceğiniz bir ses dosyasının oynatılmasını sağlar.
3		Uygulamada çalınan tüm sesleri durdurur.


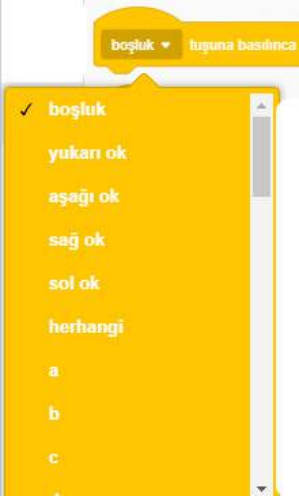

<p>4</p>		<p>Ses perdesi (pitch) veya sağa sola kaydır (left/right pan) efekt etkisini değiştirmek için kullanılır. Pitch efekti sesin kayması anlamında kullanılan bir efekttir. Sesin sağa sola yönlmesi için kullanılan bir efekttir.</p>
<p>5</p>		<p>Ses perdesi (pitch) veya sağa sola kaydır (left/right pan) efekt etkisini sabit bir değere ayarlamak için kullanılır.</p>
<p>6</p>		<p>Ses dosyası üzerindeki etkilerin temizlenmesini sağlar.</p>
<p>7</p>		<p>Uygulamanın ürettiği ses şiddetini değiştirmek için kullanılır.</p>
<p>8</p>		<p>Uygulamanın ürettiği ses şiddetini % olarak değiştirmek için kullanılır.</p>
<p>9</p>		<p>Sahne üzerinde mevcut ses düzeyini görüntülemek veya bir kontrol ifadesinde kullanmak için kullanılan ses düzeyi değişkenidir.</p>

BÖLÜM 4

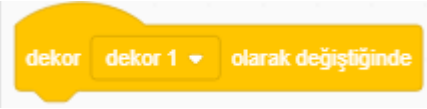
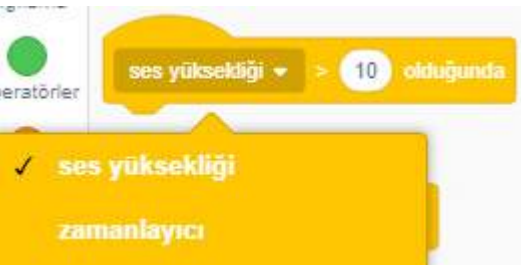
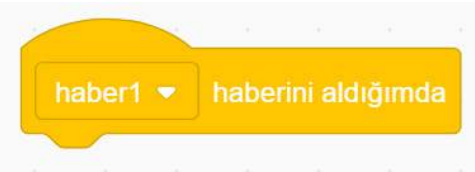


4.4.4.4. Olaylar

Olaylar blokları, bir uygulamanın başlayabilmesi için çeşitli tetikleyiciler sunar. Tetikleyicilerin en sık kullanılanı sahne üzerinde bulunan **“yeşil bayrak”** simgesidir. Uygulamayı başlatmak için yeşil bayrağa, durdurmak için ise kırmızı butona basılmalıdır. Aşağıdaki tabloda Olaylar kategorisi kod bloklarının açıklamaları gösterilmektedir.

Tablo 4.4. Olaylar kategorisinin kod blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Scratch uygulamalarının başlayabilmesi için bir tetikleyiciye ihtiyaç vardır. “Yeşil bayrak” butonuna basıldığında bu kod bloğu altına eklenen tüm bloklar sırası ile çalıştırılır. Birden fazla “yeşil bayrağa tıklandığında” blok kullanılarak aynı anda birden fazla iş yapılması sağlanabilir.
2		Klavyeden herhangi bir butona basıldığında çalıştırılması istenen bloklar, bu bloğun altına eklenmelidir.
3		Sahnedeki herhangi bir kuklaya tıklandığında çalıştırılması istenilen bloklar, bu bloğun altına eklenir.

BÖLÜM 4

4		Sahne dekoru belirtilen bir dekor olarak değiştiğinde çalıştırılması istenen bloklar, bu bloğun altına eklenir. Oyunlarda seviye değişikliğinde kullanılabilen bir bloktur.
5		Mikrofondan algılanan ses şiddeti veya zamanlayıcı belirli bir değer altında iken çalıştırılması istenen kodlar bu bloğun altına eklenir.
6		“Haber sal” komutu ile birlikte çalışan bir komuttur. Uygulamadaki tüm kuklaların ve sahnelerin belirli bir haber geldiğinde tepki vermesi gerekiyorsa bu kod bloğu kullanılır.
7		Uygulamada yeni bir haber üretmek için yeni haber sekmesine tıklanır. Bu “ yeni haber ” oluşturulduğunda kod içinde kullanılması gerekir. Kullanılmaması durumunda oluşturulan haber silinir.
8		Bir haber ortama duyurulduğunda bu haberi algılayan kuklaların görevleri bitinceye kadar kodların tamamlanmasını bekler.


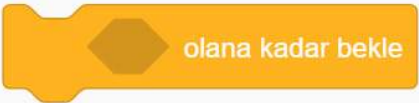


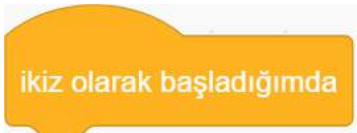

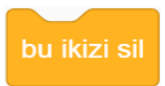
BÖLÜM 4

4.4.4.5. Kontrol

Karar durumlarının uygulama üzerinde çalışmasını belirli şartlara göre şekillendirmek amacıyla kullanılır.

Tablo 4.5. Kontrol kategorisinin kod blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Belirtilen süre kadar kodların çalışması bekletilir.
2		Belirtilen sayı kadar, bloğun içerisine yazılan kodlar tekrar edilir. Döngü olarak isimlendirilen bu kod yapısı, tekrar eden görevler için daha az kod yazmayı ve kodun yönetilmesini kolaylaştırır.
3		Uygulamada sürekli olarak çalıştırılmasını, tekrar edilmesini istediğiniz kodlar, bu kod bloğunun içine yazılır. Bu blok içine yazılmayan bloklar bir defa çalışır.
4		Uygulamada kodların bir koşula bağlı olarak çalıştırılması istendiğinde "eğer bloğu" kullanılır. Eğer bloğun içindeki kutucuğa eklenecek olan bir mantıksal ifadeye göre eğer bloğu içindeki kodlar çalıştırılır. Eklenecek olan mantıksal ifadenin sonucu "doğru" değer üretiyorsa yani koşul sağlanıyorsa eğer şart bloğu içindeki kodlar çalıştırılacaktır.






5	 <p>eğer <input checked="" type="checkbox"/> ise</p> <p>değilse</p>	Şarta bağlı komutların çalıştırılmasında mantıksal şart ifadesinin sonucu "doğru" değer üretiyorsa eğer bloğu içindeki kodlar çalışırken "yanlış" değer ürettiğinde ise "değilse bloğu" içindeki kodlar çalıştırılacaktır.
6	 <p><input checked="" type="checkbox"/> olana kadar bekle</p>	Uygulamada bir koşul sağlanana kadar uygulama kodlarının çalışması bekletilir.
7	 <p><input checked="" type="checkbox"/> olana kadar tekrarla</p>	Uygulamada belirli bir koşul durumu sağlandığı sürece belirli kodların çalıştırılmasını sağlar.
8	 <p>durdur <input checked="" type="checkbox"/> tümü ▼</p> <p><input checked="" type="checkbox"/> tümü</p> <p><input type="checkbox"/> bu dizi</p> <p><input type="checkbox"/> Kukladaki diğer diziler</p>	Uygulama kodlarının tamamını, sadece belirtilen diziyi veya kuklaya ait diğer dizileri durdurmak için kullanılır.
9	 <p>ikiz olarak başladığımda</p>	Bir kuklanın ikizi, kodlar yardımı ile üretilebilmektedir. Kuklanın ikizi üretildiğinde çalıştırılmasını istediğiniz kodlar, bu bloğun altına eklenmelidir.
10	 <p>kendim ▼ 'in ikizini yarat</p> <p><input checked="" type="checkbox"/> kendim</p> <p><input type="checkbox"/> Apple</p>	Sahnedeki bir kuklanın veya seçili olan kuklanın bir ikizini yaratmak için kullanılır. Bir oyunda aynı karakterden 2 tane kullanılacağı durumda bu karakterin ikizi, bu kod bloğu ile oluşturulabilir.
11	 <p>bu ikizi sil</p>	Sahne oluşturulan ikiz kuklanın silinmesini sağlar.







BÖLÜM 4

4.4.4.6. Algılama


Algılama kategorisi; sahnedeki kuklaların birbiri ile, kullanıcı ile ve sahnenin kendisi ile olan etkileşimlerini algılamayı sağlayan kod bloklarını içerir. Bloklarda ifade edilen soru cümleleri, mantıksal şart ifadeleridir. Bir durumun meydana gelip gelmediğini sorgular. Eğer ki sorguda bahsedilen durum meydana geldi ise **“Doğru”** sonucunu üretir, bahsedilen durum meydana gelmedi ise **“Yanlış”** sonucu üretir. Bu sonuçlara göre uygulamanın akışı yönlendirilir. Tablo 4.6’da algılama kategorisi kod bloklarının açıklamaları verilmiştir.

Tablo 4.6. Algılama kategorisinin kod blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Sahnedeki kuklanın fare imlecine, sahnenin kenarına veya sahnedeki bir dięer kuklaya deęip deęmedięi algılanır.
2		Sahnedeki kuklanın belirtilen renge dokunup dokunmadıęı algılanır.
3		Sahne de belirtilen bir rengin, dięer bir renge deęip deęmedięi algılanır.
4		Sahnedeki kuklanın, fare imlecine veya sahnedeki dięer bir kuklaya olan mesafe deęerini verir.
5		Kullanıcıya bir soru sordurup vereceęi cevapla cevap deęiřkeninde iřlem yapılmasını saęlayan kod bloęudur.

6		<p>Sahnedeki kuklanın kullanıcı ile etkileşime girmesi amacıyla bilgisayarın klavyesinden de faydalanılır. Klavyeden basılan herhangi bir tuşu algılayan kod bloğudur.</p>
7		<p>Verilen kod blokları sırası ile fareye basılı olup olmadığını, imlecin x eksenindeki yerini ve imlecin y eksenindeki yerini belirtir.</p>
8		<p>Sahnedeki kuklanın uygulama çalıştırıldığında fare imleci ile sürüklenebilmesini veya sürüklenmemesini sağlayan kod bloğudur.</p>
9		<p>Uygulamada mevcut ses yüksekliği değerini bildirir.</p>
10		<p>Zamanlayıcı, uygulama çalıştırıldığında çalışmaya başlayan ve istediğiniz anda sıfırlayabildiğiniz saniyeleri sayan bir sayaçtır. Bu kod bloğunu kullandığınız yerde size sayaç değeri bildirilecektir.</p>
11		<p>Zamanlayıcıyı sıfırlamak için kullanılan kod bloğudur.</p>








BÖLÜM 4

12		Sahnedeki kuklalar veya dekor hakkında bazı bilgileri, bu kod bloğu sayesinde elde edebilmekteyiz.
13		Mevcut tarih ve zaman bilgisini elde edebilmek için kullanabileceğiniz kod bloğudur.
14		2000 yılından beri geçen gün sayısını verir.
15		Scratch online üzerinde çalışırken kullanıcı adı bilgisini veren kod bloğudur. Offline modda çalışırken boş bir değer döndürür.

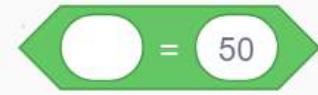

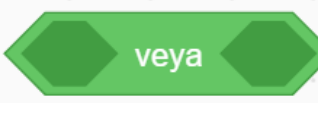



4.4.4.7. Operatörler




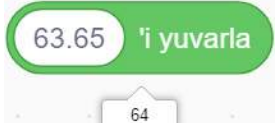
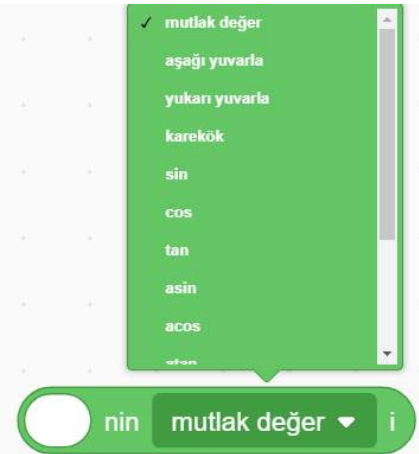
Mantıksal işlemler, matematiksel işlemler ve metinler üzerinde yapılacak işlemler için kod bloklarını içerir. Tablo 4.7’de ilgili kod blokları gösterilmektedir.

Tablo 4.7. Operatörler kategorisinin kod blokları ve görevleri

Sıra	Kod Bloğu	Açıklama
1		Girilen iki sayının toplam değerini sonuç olarak bildirir. Metinsel ifade girişine izin verilmez.
2		Girilen iki sayının farkını sonuç olarak bildirir.
3		Girilen iki sayının çarpımını sonuç olarak bildirir.
4		Girilen iki sayıdan soldaki sayının, sağdakine bölümünü sonuç olarak bildirir. Elde edilen sonuç ondalıklı ise olduğu gibi bırakır, yuvarlama işlemi yapılmaz. 10/4 işleminde sonuç olarak 2.5 değerini döndürür.
5		Verilen iki sayı arasında rastgele bir sayı üretmeyi sağlar. Girilen sayılar tam sayı ise üretilen sayı da tam sayı olacaktır. Girilen değerlerinden 1 tanesi virgüllü sayı ise üretilen değer de virgüllü bir sayı olacaktır.
6		Girilen iki sayıdan soldaki sayının sağdaki sayıdan büyük olup olmasını sorgular.
7		Girilen iki sayıdan soldaki sayının sağdaki sayıdan küçük olup olmasını sorgular.

BÖLÜM 4

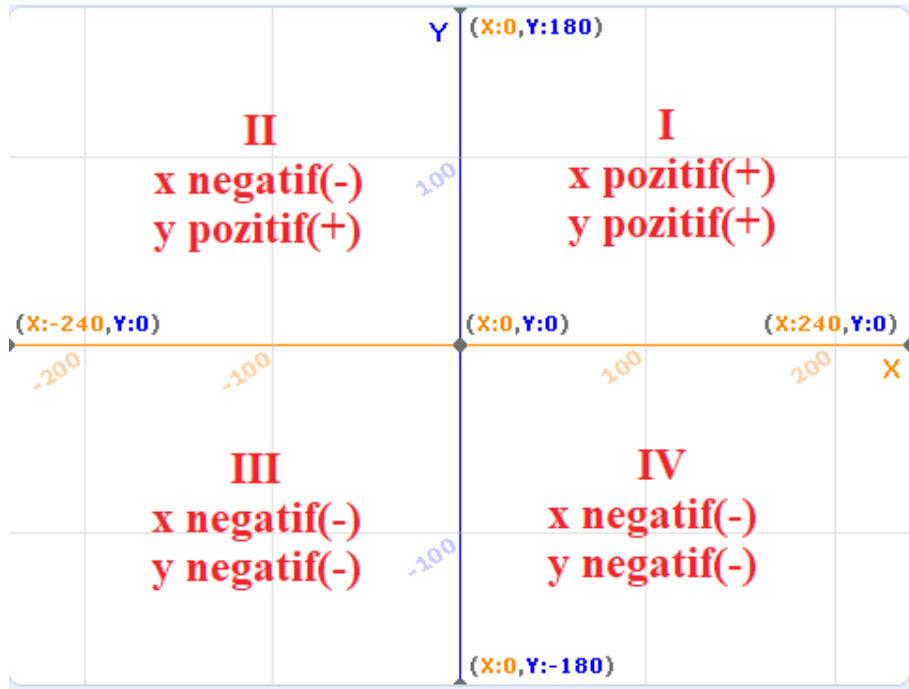
8		Girilen iki sayının birbirine eşit olup olmadığı sorgulanır.
9		Mantıksal iki ifadenin “doğru” sonuç üretmesini sorgular. Her iki mantıksal ifade de “doğru” sonuç üretiyorsa “ve” işleminin sonucu da “Doğru” olacaktır. Verilen iki mantıksal ifadeden herhangi biri “yanlış” sonuç üretmesi durumunda “ve” işleminin sonucu “yanlış” olacaktır.
10		Mantıksal iki ifadeden herhangi birinin “doğru” sonuç üretmesini sorgular. Her iki mantıksal ifadeden biri “doğru” sonuç üretiyorsa “veya” işleminin sonucu da “Doğru” olacaktır. Verilen iki mantıksal ifadenin de “yanlış” sonuç üretmesi durumunda “veya” işleminin sonucu “yanlış” olacaktır.
11		Mantıksal “değil” işlemi kendisine verilen mantıksal ifadenin üretmiş olduğu değerin tersini üretir. Parametre olarak verilen ifade “Doğru” sonuç üretiyorsa, “değil” işlemi sonucu “Yanlış” olarak değer üretecektir.
12		İki metinsel ifadeyi birleştirmek için kullanılır. Verilen örnekte sonuç olarak “elma-muz” değeri bildirilecektir.
13		İlk parametrede verilen metinsel ifadenin istenilen karakteri elde edilir. Örnekte verilen elma ifadesinin 1. harfi olarak “e” karakteri döndürülür.

14		<p>Elma ifadesinin karakter uzunluğu sonuç olarak döndürülür. Örnekte verilen ifade sonuç olarak 4 değerini döndürecektir.</p>
15		<p>Verilen iki parametre ile soldaki metinsel ifade içinde, sağdaki metinsel ifade aranır. Eğer ki sağdaki metinsel ifade, soldaki metinsel ifade içinde bulunursa sonuç "doğru" olarak döndürülür. Bulunamazsa yanlış olarak değer bildirilir.</p>
16		<p>Verilen iki sayısal ifadede soldaki sayının sağdaki sayıya göre modu alınır. Başka bir deyişle soldaki sayının sağdaki sayıya bölümünden kalan sayı sonuç olarak döndürülür. 10 mod 2 işleminin sonucu 0 olarak döndürülür.</p>
17		<p>Kendisine parametre olarak verilen sayıyı kendisine en yakın onluğa yuvarlar. Örnekte görüleceği üzere 63.65 sayısı 64'e yuvarlanmıştır.</p>
18		<p>Kendisine parametre olarak verilen sayısal ifadenin çeşitli matematiksel fonksiyonlar ile işlem yapılmasını sağlayarak sonucu bildirir.</p>

BÖLÜM 4

4.4.5. Doğrusal Mantık Yapısı ile Problem Çözme

Doğrusal mantık yapısı ile problem çözme, bilgisayara birbiri ardına algoritmanın başından sonuna kadar sırası ile işlemesi gereken kodların alt alta konulmasıdır. Yönergeler alt alta konularak işlemler yapılır. Doğrusal mantık yapısı ile problem çözmenin Scratch üzerinden anlatılabilmesi için sahne düzeni bilinmelidir.



Şekil 4.48. Koordinat düzlemi

Scratch'te sahne, yatay ekseninde (x) $240+240=480$ pixel, düşey ekseninde (y) ise $180+180=360$ pixel bir dikdörtgen çalışma alanı sunar. Yapılacak uygulamalar, bu alan içinde sınırlı olacak şekilde çalışırlar. X ve Y koordinat sistemi 4 alana ayrılır. Kuklalar, bu alanlarda hareket ederken x ve y koordinatlarının pozitif ve negatif işaretleri Şekil 4.48'deki gibi olur.

4. ETKİNLİK: FARE OYUNU

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

- Blok tabanlı kodlamada “olaylar” kullanımını bilir.
- Blok tabanlı kodlamada olayları kullanır.
- Blok tabanlı kodlamada nesnelerin hareketini gerçekleştirir.
- Projeye kukla eklemeyi bilir.
- Projeye dekor eklemeyi bilir.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

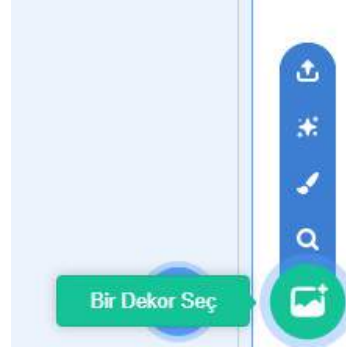
GİRİŞ

Bu etkinlikte Scratch masaüstü veya web arayüzü kullanarak kodlama ve oyunlarda hareket ve olaylar kullanımı ile ilgili örnek yapılacaktır.

SÜREÇ

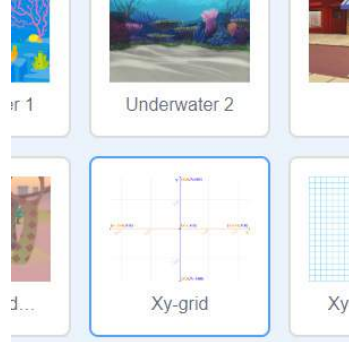
- Scratch masaüstünü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde kedi karakterinin standart geldiğini gösteriniz. Sahnenin arka planı ilk açıldığında boş olarak geldiği için ekranın sağ alt tarafında bulunan simgeye tıklayarak **Bir Dekor Seç** seçeneği ile dekor resmi seçebilirsiniz.

BÖLÜM 4



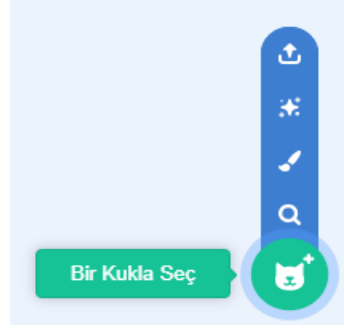
Şekil 4.49. Sahneye yeni dekor ekleme

- Açılan listede en altta bulunan **Xy-grid** dekorunu seçerek sahneye ekleyiniz.



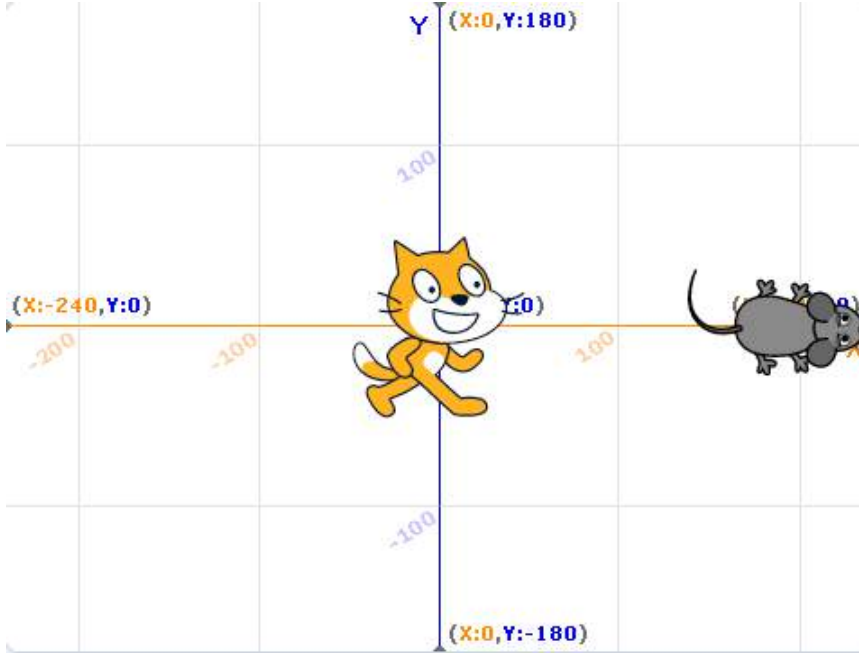
Şekil 4.50. Xy-grid isimli dekorun sahneye eklenmesi

- Bu işlemten sonra sahneye dekor resmi eklenecektir. Uygulama üzerine ayrıca yeni karakterler ekleyebilirsiniz. Yine ekranın sağ alt tarafında bulunan kukla simgesi tıklanarak **Bir Kukla Seç** seçeneğini seçmelisiniz.



Şekil 4.51. Sahneye yeni bir kukla ekleme

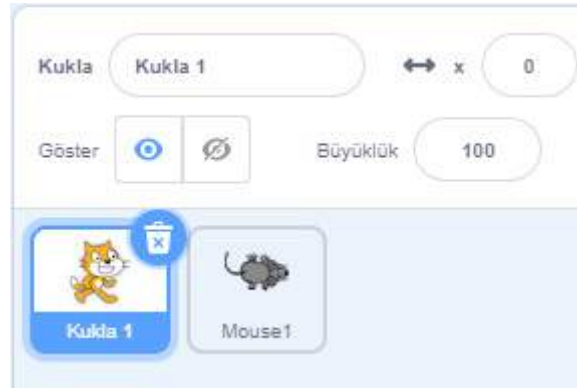
- Bu işlemden sonra ekrana gelen kukla listesinden **Mouse 1** isimli kuklayı seçiniz. Bu işlemden sonra kukla, sahne üzerine konumlanacaktır.



Şekil 4.52. Sahne üzerinde kuklalar.

- Kedi kuklasına bir eylem yaptırabilmek için Şekil 4.53'teki gibi **Kedi** kuklasını seçiniz.

BÖLÜM 4



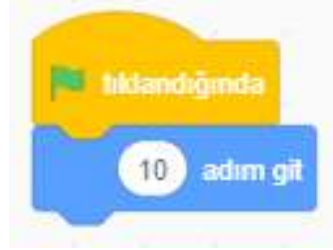
Şekil 4.53. Kuklanın seçilmesi

- Karaktere bir eylem yaptırabilmek için kukla seçili iken blokların "Olaylar" sekmesinde bulunan **Yeşil Bayrağa Tıklandığında** bloğu ile başlayınız. Blokları, kodlama alanına sürükleyerek kodlama işlemi yapınız.



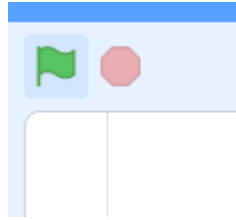
Şekil 4.54. Blokların kodlama alanına sürüklenmesi

- Hareket blokları altında bulunan **10 adım git** bloğu sayesinde kuklaya hareket verilebilirsiniz. Eklenecek blokların tamamını, **Yeşil Bayrak Tıklandığında** bloğunun altına eklemelisiniz.



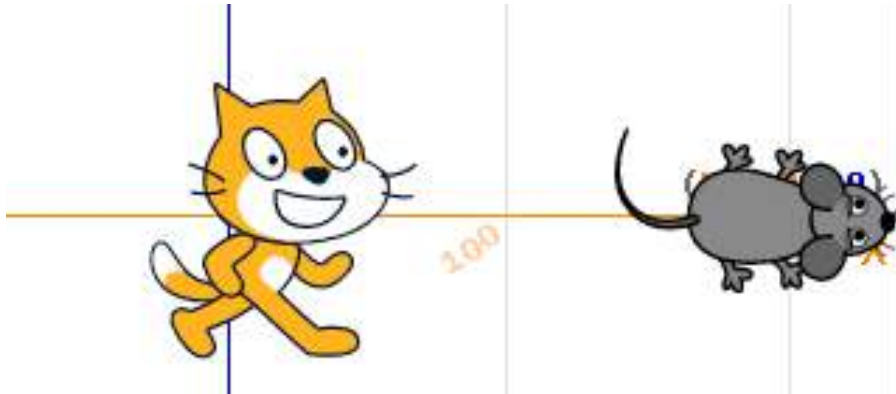
Şekil 4.55. Bloklarının birleştirilmesi

- Blokların çalışabilmesi için sahne üzerinde bulunan **Yeşil Bayrak** işaretini tıklanmanız gerekecektir.



Şekil 4.56. Yeşil Bayrak

- Yeşil Bayrağa tıklayarak uygulamayı başlatabilirsiniz. Bu işlem yapıldıktan sonra kukla, sahne üzerinde bir miktar hareket etme eylemi gerçekleştirecektir.

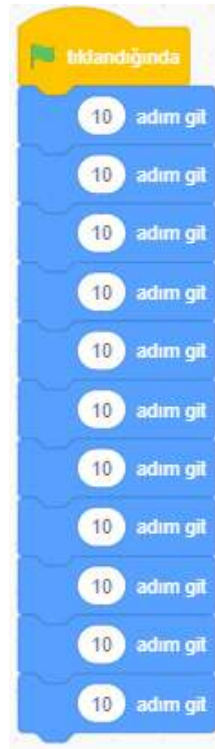


Şekil 4.57. Kuklanın sahne üzerinde hareketi

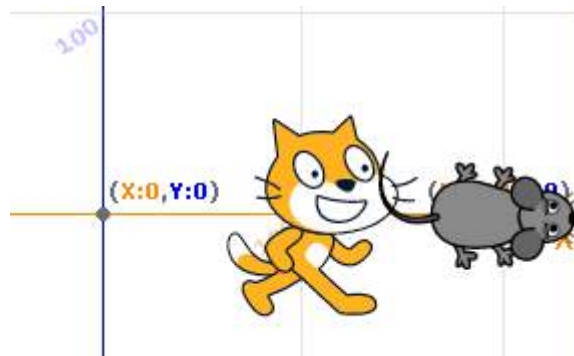
- Kedi kuklasının, fare kuklasına ulaşabilmesi için **10 Adım Git** bloğundan birkaç tane daha eklenerek ulaşmasını sağlayabilirsiniz. Şekil 4.57'de iki kukla arasında 100 birim olduğu için

BÖLÜM 4

10 Adım Git bloğundan 10 tane daha eklemeniz gerekecektir. Şekil 4.58'deki gibi kod bloklarını alt alta sıralayarak **Doğrusal Programlama** yapmış oldunuz.



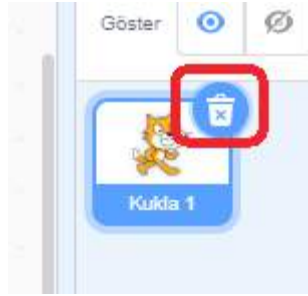
Şekil 4.58. Kukla hareketi için yeni blokların eklenmesi



Şekil 4.59. Uygulama çıktısı

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

Scratch masaüstü veya scratch web arayüzünü açınız. Boş gelen sayfada ekranın sağ alt tarafında bulunan simgeye tıklayarak **Bir Dekor Seç** seçeneği ile koordinat sistemi Xy-grid resmi ekleyiniz. Kuklalar sekmesinde kedi karakterinin üzerindeki X(sil) butonuna basılarak karakter silinir.



Şekil 4.60. Sahnedeki kuklanın silinmesi

Sahneye fare kuklasını ekleyiniz. Kuklanın konumunu x: -180 y:-130 olacak şekilde ayarlayınız. Bu işlemi yaparken kukla özelliklerinden x-y konumuna değerleri giriniz.



Şekil 4.61. Kuklanın koordinatlarının ayarlanması

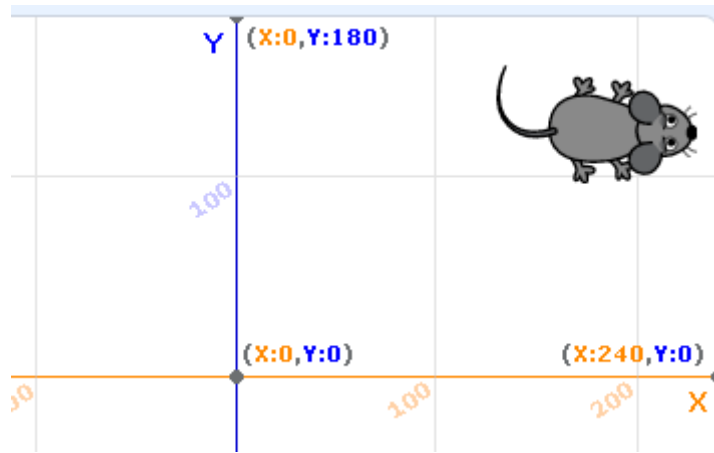
Kod bloklarını aşağıdaki gibi oluşturunuz.

BÖLÜM 4



Şekil 4.62. Kuklanın belirli sürede belirli bir koordinata hareketi

Yeşil Bayrağa basıp uygulamayı çalıştırdığınızda kukla, belirtilen sürede belirlenen koordinata hareket edecektir.



Şekil 4.63. Kuklanın süzülme hareketi

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmaları kontrol ediniz. Öğrencilerinize ikinci bir karakter ekleterek farklı uygulamalar yapmalarına olanak sağlayınız.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Blok tabanlı kodlamada tıkladığında olayını yapma			
Blok tabanlı kodlamada kuklaları 10 adım git ile hareket ettirme			
Dekor ekleme			
Kukla ekleme			
Kuklayı silme			
Kuklayı istenilen koordinatlara gönderebilme			

4.4.6. Karar Yapıları ile Problem Çözme

Karar yapıları ile problem çözme, günlük hayatın temelini oluşturur. Örnek olarak insanların karnı acıktığında yemek yer, acıkmadığında ise yemezler ya da hava yağmurlu ise şemsiye kullanılır, yağmurlu değilse şemsiye kullanılmaz. Aynı şekilde **Scratch** programında da bir olayın farklı sonuçları için karar yapıları olarak “eğer-ise” blokları kullanılır.

BÖLÜM 4

5. ETKİNLİK: EHLİYET ALABİLİR MİYİM?

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Görünümden kuklaya düşünce balonu oluşturur.

Operatörleri karşılaştırma ifadelerini bilir.

Koşullu yapıları bilir.

Günlük hayatta koşullu yapıların kullanıldığını fark eder.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

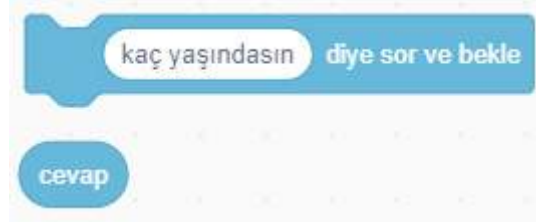
Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Bu etkinlikte Scratch masaüstü veya web arayüzü kullanarak kodlama, oyunlarda hareket ve olaylar kullanımı, birlikte koşullu kullanımla ilgili örnek yapılacaktır.

SÜREÇ

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde hazır gelen kedi karakterine, kullanıcıya soru sormak ve kullanıcıdan veri almak için algılama bloklarından “... **diye sor ve bekle**” bloğunu tanımlayınız.



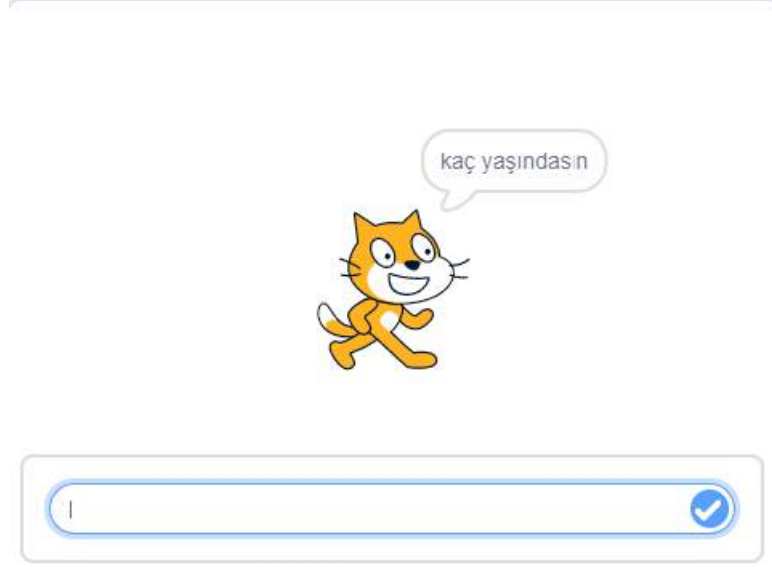
Şekil 4.64. Kullanıcıdan veri alma bloğu

- Bu blok ile kullanıcıya bir soru sorulacak ve kullanıcının bir cevap girmesi sağlanacaktır. Kullanıcıdan alınan veri ile işlem yapabilmek için **Operatör** bloklarını kullanarak **eğer** bloğu içinde karşılaştırma işlemi yapılması için Şekil 4.65'teki gibi blokları oluşturunuz.



Şekil 4.65. Uygulama blokları

- Uygulama çalıştığında kuklanın soru soracağını ve kullanıcının cevabına göre ehliyet alıp alamayacağı cevabını vereceğini öğrencilerinize aktarınız.



Şekil 4.66. Kuklanın kullanıcıdan veri girişi yapmasını beklemesi

- Girilen sayıya göre kuklanın, **eğer değer 18'den küçükse ehliyet alamazsın**, büyükse **ehliyet alabilirsin** mesajını verdiği 3-4 farklı değer girerek gösteriniz.



Şekil 4.67. Uygulamanın çıktısı

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmaları kontrol ediniz. Öğrencilerinize başka bir karakter ekleterek farklı uygulamalar yapmalarına olanak sağlayınız.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Blok tabanlı kodlama ile soru sorma ve cevap verme			
Koşullu yapıları kullanma			
Görünümden yazı yazma			
Operatörlerden karşılaştırma yapılarını kullanma			

4.4.7. Döngü Yapıları ile Problem Çözme

Döngüler “*herhangi bir olayın birden fazla tekrarlanması*” olarak tanımlanır. Döngüler; resim, matematik gibi birçok alanda da kullanılır. Programlamada döngüler, bir kodun birden fazla tekrarlanmasının kodlamasını kolaylaştırılması için kullanılır.



Şekil 4.68. Sürekli tekrarla ve .. defa tekrarla blokları

Sürekli tekrarla bloğu, bir işlemin kaç defa çalıştırılacağı bilinmeyen durumlarda uygulamayı sonlandırılana kadar tekrar etmesi için kullanılır. .. **defa tekrarla** bloğu ise bir işlemin kaç defa yapılacağı belli olduğu durumlarda kullanılır. Örnek olarak bir top 26 defa sektirilecekse .. **defa tekrarla** bloğunu kullanılmalı, eğer top sürekli sekmesi ve program kullanıcı tarafından sonlandırılana kadar devam etmesi isteniyorsa **sürekli tekrarla** bloğunu kullanılmalıdır.

6. ETKİNLİK: TOP SEKTİRME

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Döngüleri fark eder.

Koordinat düzenini fark eder.

Döngüleri kullanır.

YÖNTEM/TEKNİKLER

Gösterip yaptıрма, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

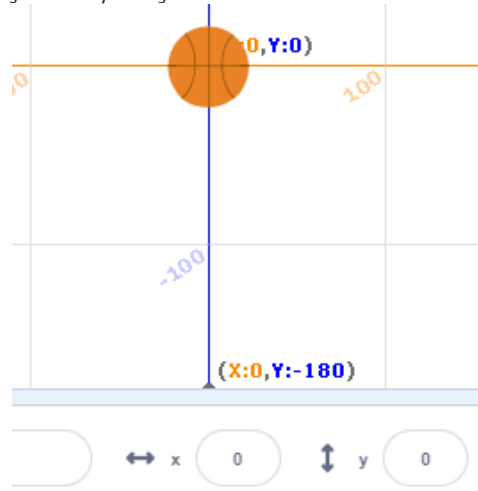
GİRİŞ

Konuya aşağıdaki soru ve ifadelerle giriş yapınız:

- Günlük hayatta süreklilik arz eden pek çok olay bulunmaktadır. Bunlar nelerdir?
- Öğrencilerinizle birlikte günlük yaşamda tekrarlayarak yaptıkları işleri tartışınız.
- Acıktığımızda yemek yeriz, güneş her sabah yeniden doğar ya da bir yerden bir yere hareket etmek için yürüme eylemi yaparız. Bu iş için sürekli komut vermek yerine tekrar eden işlemler için sürekli tekrarlama işlemi yapmak daha iyi bir yöntemdir.
- Bu etkinlikte Scratch masaüstü veya web arayüzü kullanarak kodlama ve oyunlarda hareket ve olaylar kullanımı ile birlikte döngü kullanımına yönelik örnek yapılacaktır.

SÜREÇ

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri silebilirsiniz. Kukla olarak **Top** kuklasını sahneye ekleyiniz. Sahnenin arka planı ilk açıldığında boş olarak geldiği için dekor olarak **Xy-grid** koordinat düzlemini arka plan olarak seçiniz. Top kuklasının konumunu sahne üzerinde **x: 0 y: 0** olacak şekilde yerleştiriniz.



Şekil 4.69. Top kuklasının sahne üzerine konumlandırılması

- Top kuklasının sekme hareketi yapabilmesi için Top kuklasına eklenecek blokları Şekil 4.70'teki gibi oluşturunuz.



Şekil 4.70. Top kuklasının sekme hareketi

BÖLÜM 4

- Uygulama çalıştırıldığında top kuklası üç defa sekme hareketi yaptıktan sonra sabit kalacaktır. Üçten fazla topu zıplatmak için top kuklasına eklenen blokları alt alta eklemeniz gerekecektir. Bloklar alt alta eklendiğinde hem çok fazla blok eklemek gerekeceğini hem de uygulamanın kontrolünün zorlaşacağını öğrencilerinize aktarınız. Bu işlemler için **kontrol** bloğu içinde tekrarlama bloklarını kullanmalısınız.



Şekil 4.71. Sürekli tekrarla ve .. defa tekrarla blokları

- Uygulamada topun 30 defa sekmesi için **..defa tekrarla** bloğunu kullanarak Şekil 4.72'deki kod bloklarını Top kuklasına ekleyiniz.



Şekil 4.72. Top kuklasının belirli sayıda sekmesi için gerekli kod blokları

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmalar kontrol edilir.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Belli sayıda tekrarlamaı kullanma			
İstenilen koordinata kuklayı gönderme			
Görünümden yazı yazma			
Operatörlerden karşılaştırma yapılarını çalışmalarında kullanma			

7. ETKİNLİK: TOP SEKTİRME-2

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Döngüleri fark eder.

Koordinat düzenini fark eder.

Döngüleri kullanır.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Bu etkinlikte Scratch masaüstü veya web arayüzü kullanılarak kodlama ve oyunlarda hareket ve olaylar kullanımı ile birlikte döngü kullanımı ile ilgili örnek yapılacaktır. Bu örnekte **Sürekli tekrarla** bloğunu kullanarak bir işlemin sürekli çalışması gerektiğinde ya da kaç defa çalıştırılması gerektiğinin bilinmediği durumlarda nasıl kullanıldığı görülecektir.

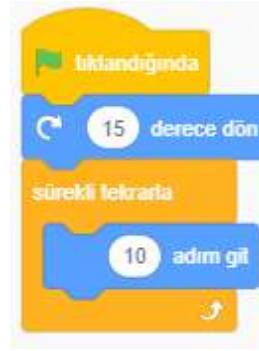
SÜREÇ

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri silebilirsiniz. Kukla olarak **Be-achball** kuklasını sahneye ekleyiniz.



Şekil 4.73. Beachball kuklası

- Uygulamada **Beachball** kuklasına Şekil 4.74'teki kod bloklarını ekleyiniz. Bu kuklanın hareketine belirli bir açıyla başlaması için .. **derece dön** bloğunu kullanabilirsiniz.



Şekil 4.74. Kuklanın sahne üzerinde hareketi

- Uygulama başlatıldığında, kukla sahnenin köşelerine geldiğinde köşelere yapışıp kalacak ve hareketine devam edemeyecektir. Sahnenin köşelerine çarptığında kuklanın yönünün değişmesi için **kenara geldiyse sek** bloğunu Şekil 4.75'teki gibi kullanabilirsiniz.



Şekil 4.75. Kenara gelen kuklanın geri dönerek hareketine devam etmesi

BÖLÜM 4

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmalarını kontrol ediniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Belli sayıda tekrarlamaı kullanma			
İstenilen koordinata kuklayı gönderme			
Görünümden yazı yazma			
Operatörlerden karşılaştırma yapılarını çalışmalarında kullanma			

8. ETKİNLİK: GÜVERCİN UÇURMA

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Döngüleri kavrar.

Koşullu yapıları bloklar ile kullanacağını fark eder.

Koşullu yapıları bilir.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Bazı kuklaların birden fazla kostümü bulunabilir. Bu kostümler sayesinde bazı kuklalara yürüme efekti verilebilirken bazıları ile dans edilebilir ya da çeşitli aktiviteler yaptırılabilir. Kuklanın kostümlerine erişebilmek için kukla seçili iken kod bloklarının üstündeki Kostümler sekmesine tıklanıldığında o kuklaya ait kostümler görüntülenir.

Bu etkinlikte Scratch masaüstü veya web arayüzü kullanılarak kodlama ve oyunlarda hareket ve olaylar kullanımı ile birlikte döngü kullanımı ile ilgili örnek verilirken kuklaya ait kostümlerle uygulamanın görselliği artırılabilecektir.

BÖLÜM 4

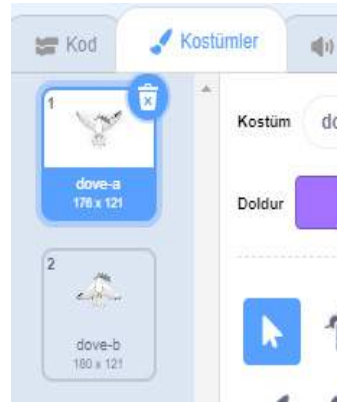
SÜREÇ

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde görünen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri silebilirsiniz. Kukla olarak **Dove** kuklasını sahneye ekleyiniz.



Şekil 4.76. Dove kuklası

- Dove kuklası seçili iken **Kostümler** sekmesine tıklarsanız Dove kuklasına ait kostümleri görebilirsiniz. Dove kuklasının kostümlerini incelediğinizde **Dove-a** ve **Dove-b** isiminde iki adet kostümü olduğunu göreceksiniz.



Şekil 4.77. Dove kuklasının kostümleri

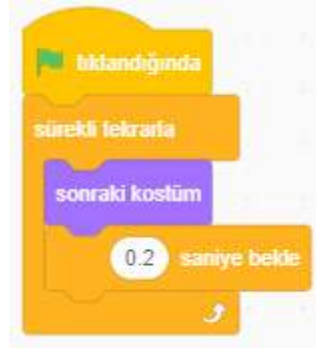


Şekil 4.78. Dove-a



Şekil 4.79. Dove-b

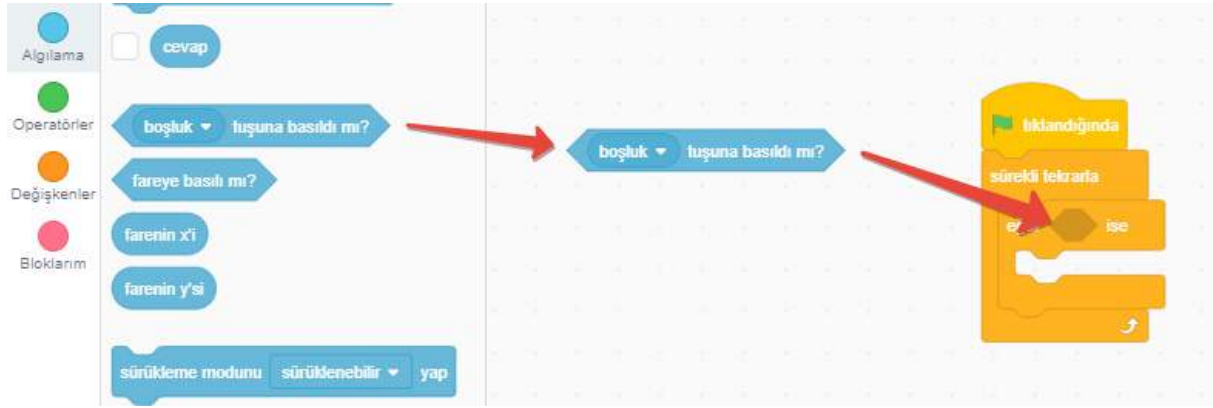
- Kuklaya ait kostümleri incelendiğinizde **Dove-a** ve **Dove-b** sayesinde kuklanın kanat çırpma hareketi yapabileceğini göstermektedir. Bu kuklaya ait kostümleri sırayla oynattığınızda kanat çırpma eylemini oluşturabilirsiniz. Dove kuklasına kanat çırpma hareketini yaptırmak için Şekil 4.80'deki kod bloklarını tanımlayınız. Bloklar arasında **0.2** saniye bekleme eklenerek kanat çırpma eyleminin belirli aralıklarla yapılmasını sağlayabilirsiniz. Bu süreyi artırdığınızda kostümler arası geçiş süresi uzayacak, süreyi kısalttığınızda ise kostümler arasındaki geçiş süresi kısalmaktadır. Böylelikle kuklaya hareketleri yavaş veya hızlı yaptırabilirsiniz.



Şekil 4.80. Dove kuklasının kanat çırpma eylemi için gerekli kod blokları

- Kuklanın klavyenin yön tuşları ile koordinat düzleminde hareketlerini sağlayabilmek için **Algılama** blokları içinde bulunan **..tuşuna basıldı mı?** bloğunu **Eğer .. ise** bloğunun içine ekleyiniz.

BÖLÜM 4



Şekil 4.81. Algılama bloğunun Eğer ise bloğunun içine eklenmesi



Şekil 4.82. Algılama bloğunun yukarı ok tuşu olarak ayarlanması

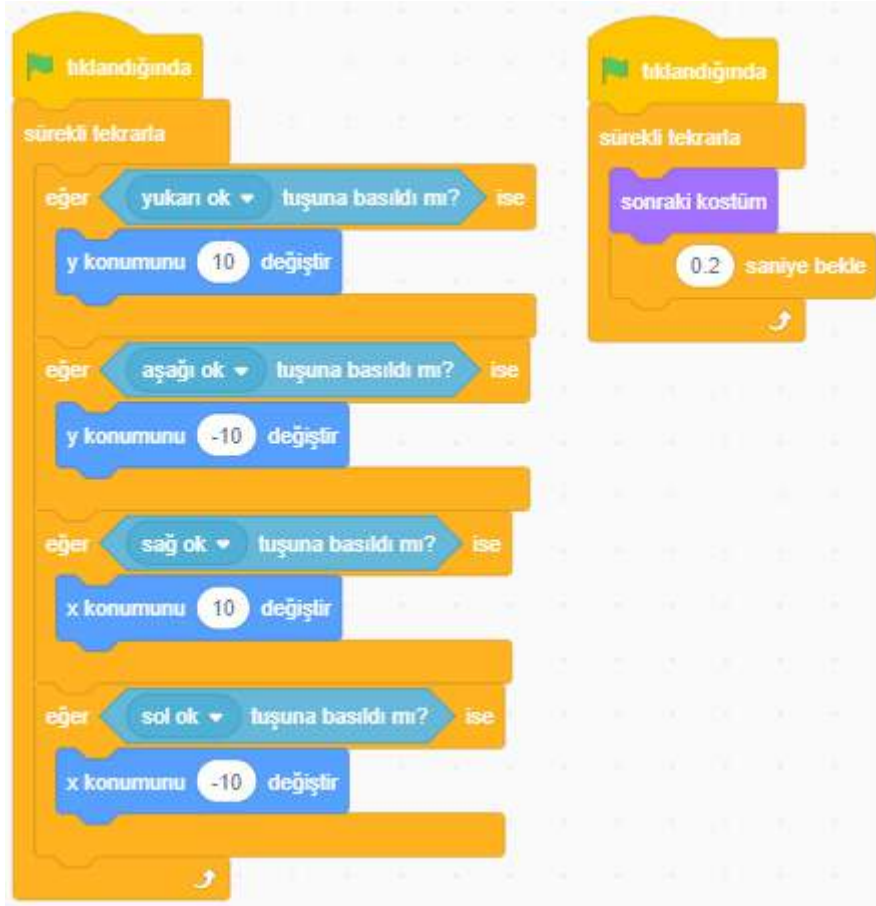
- Son aşamada tüm yön oklarını aktif hâle getirmek için bloklarını Şekil 4.83'teki gibi oluşturabilirsiniz.



Şekil 4.83. Yön tuşlarını tanımlamak için kullanılan kod blokları

- Uygulama tamamlandığında **Dove** kuklasının kod blokları, Şekil 4.84'teki gibi olacaktır.

BÖLÜM 4



Şekil 4.84. Dove kuklasının tamamlanmış kod blokları

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmaları kontrol ediniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Kuklanın kostümlerini değiştirme			
Algılama menüsünde “tuşa basıldı mı” komutunu kullanma			
Yön tuşları ile kuklayı hareket ettirme			
Operatörlerden karşılaştırma yapılarını bilme			
Döngüler ve şart yapılarını bir arada kullanabilme			

BÖLÜM 4

4.4.8. Değişkenler

Bilgilerin geçici olarak tutulduğu alanlardır. Programcı tarafından tanımlanan verilerin geçici olarak tutulduğu alanlardır.

$$1 + \square = 5$$

Şekil 4.85. Değişken tanımlaması için örnek

Örneğin, Şekil 4.85'te gösterilen formülde dikdörtgen değişkenlere benzetilebilir. İçinde gizli 4 rakamı bulundurmaktadır. Dikdörtgen, 4'ün yerini tutmaktadır. Değişkenler içindeki veriler de bu şekilde tutulur. Kullanıcı, sabit 1 sayısı ile klavyeden girilen bir değere göre sonuç elde ettiğinde Dikdörtgen klavyeden girilen değeri ifade edecektir. Sonuçta buna göre değişir. Değişkenlerle uygulama içerisinde zaman tutucu, sayı tutucu ve isim tutucu gibi işlevler de kullanılabilir.

9. ETKİNLİK: TANIŞMA OYUNU 1

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Değişkenleri kavrar.

Değişken oluşturur.

YÖNTEM/TEKNİKLER

Gösterip yaptırma

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- Tahtaya $1 + \square = 5$ yazılıp öğrencilere kare alanına hangi sayı geleceği sorulur. Örnekler çoğaltılabilir.
- Öğrencilerden gelen cevaplar kareye yazılır. Karenin buradaki işlevi sorulur. Karenin, 4 sayısının yerini tutan bir şekil olduğu belirtilir.
- Programlama da bazı sayıların, yazıların yerini tutan karelerin olduğu ve buna da değişkenler denildiği söylenir.

Öğrencilere değişkenler hakkında bilgi vererek etkinliğe başlayınız.

BÖLÜM 4

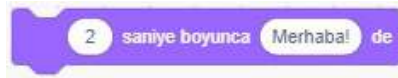
SÜREÇ

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde gelen kedi karakterini kullanınız. Dekor seçimini öğrencilerinize bırakabilirsiniz.
Kedi kuklasına **Olaylar** bölümünden **Yeşil Bayrağa Tıkladığında** bloğu eklenerek başlatma tetikleyicisi belirlenmiş olur.



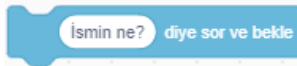
Şekil 4.86. Başlatmak için gerekli kod bloğu

- **Görünüm** bölümünden .. **saniye boyunca de** bloğu eklenerek boşluklara Şekil 4.87'deki gibi gerekli yazıları yazınız.



Şekil 4.87. Selamlama bloğu

- **Algılama** bölümünden ...**diye sor ve bekle** kod bloğu eklenerek soru kısmını doldurunuz. Şekil 4.88'deki "İsmin ne?" sorusu dibi kullanıcıdan bilgi alınmak istenen soruları sorabilirsiniz.



Şekil 4.88. Kullanıcıya soru sorma bloğu

- **Değişkenler** bölümüne giderek değişken oluşturmalsınız. (Örn: İsim)



Şekil 4.89. Değişken 0 yap

- Değişkeni oluşturduktan sonra **Değişken i .. yap** bloğunu ekleyerek oluşturduğunuz değişkeni Şekil 4.89'daki gibi **Değişken** boşluğuna yerleştiriniz.



Şekil 4.90. Cevap kod bloğu

- **Algılama** bölümünden **cevap** bloğu 0 içine atılır.
- Öğrencilere bu araçlarla isim ve soyismi soran ve bunları aklında tutan Şekil 4.91'deki kod bloğunu kedi kuklasına eklemelerini sağlayınız.

BÖLÜM 4



Şekil 4.91. Örnek uygulama kod bloğu

- Öğrencilerinize değişkenlerin bilgileri hafızada tuttuğunu ve unutmadığını ama bunu sadece program bitinceye kadar yaptığını ve program yeniden başlatıldığında bu bilgilerin silindiğini söyleyiniz.

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmaları kontrol ediniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Değişkenleri tanımlayabilme			
Değişken oluşturma			
Değişkenleri doğru kullanma			

4.4.9. Diziler (Listeler)

Birden fazla bilginin bir arada tutulmasını sağlayan yapılar **diziler** olarak adlandırılır. Bazı programlarda liste olarak da adlandırılır. Daha kolay anlaşılması için listeler, bir alışveriş listesine benzetilebilir. Birbiri ile ilgili birçok özelliğin bir arada tutulabildiği, geçici hafızada yer tutan bilgilerdir. Bir alışveriş listesinde marul, domates, patates gibi değişkenlerin adetleri ve özellikleri yer alır.

10. ETKİNLİK: TANIŞMA OYUNU 2

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Değişkenleri kavrar.

Değişken oluşturur.

Liste oluşturur.

Listeye öge ekler.

YÖNTEM/TEKNİKLER

Gösterip yaptırma

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Aşağıdaki ifadelerle derse giriş yapılır:

- Bir sınıftaki öğrencilerin isim, sınıf numarası gibi bilgilerini toplayan bir program yapmak istersek hangi adımları takip etmemiz gerekir?

Öğrencilerden gelen cevapları dinledikten sonra öğrencilere değişkenler ve diziler hakkında bilgi vererek etkinliğe başlayınız. Bir değişkenin birden çok bilgiyi bir arada tutabileceğini söyleyiniz. Değişkenler ile yapılan çalışmalara göre diziler ile yapılanların daha az kodla birçok benzer verinin toplanacağını ve listelerin birçok değişkenin tutulduğu alan olduğunu hatırlatınız.

SÜREÇ

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde gelen kedi karakterini kullanınız. Dekor seçimini öğrencilerinize bırakabilirsiniz.
- Kedi kuklasına öncelikli olarak başlatma işareti olarak **Yeşil Bayrağa tıkladığında** kod bloğu eklenir.
- Liste elemanlarını yazmak için **sayaç** isminde değişken oluşturunuz. Ve başlangıç değerini 0 (sıfır) olarak atayınız. Bu, listeleri yazdırmak için oluşturulan bir değişkendir. Algılama bloklarından “... **diye sor bekle**” kod bloğu ekleyiniz. Değişkenler bölümünden **yeni bir liste oluştura** tıklanır.

Bir Liste Oluştur

Şekil 4.92.

- Liste adını, kişisel olarak değiştiriniz.

Şekil 4.93.

- Sonra Değişkenler bölümünden “...i değişken adı’ ün pozisyonuna ekle” liste elemanı eklenir. Bu kısımda şey **Kişisel** listesinin **1 numaralı dizisine şey** tanımlanacağını gösterdiğinden bahsedebilirsiniz.
- **Şey** kısmına **Algılama** kod bloklarından **verilen cevap** eklenir.



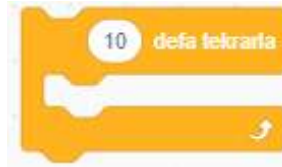
Şekil 4.94.

BÖLÜM 4



Şekil 4.95.

- Şekil 4.95'te liste elemanlarının her bir dizisine aşağıdaki sorulan soruların cevapları eklenmektedir.
 - Kişisel 1. elemanına isim atanmaktadır.
 - Kişisel 2. elemanına soyisim atanmaktadır.
- Bundan sonra değişkenlerin sırasıyla birden fazla bilginin yazdırılması için yapılacak kod blokları görülmektedir.
- Kontrol kod bloklarından 10 defa tekrar kod bloğu eklenir.



Şekil 4.96.

- Değişkenler bölmesinden sayaç değişkenini 1 değiştir kodu atılır.



Şekil 4.97.

- Değişkenler bölmesinden "... nin ... ögesi" kod bloğu atılır.

kisisel ▾ nin 1 öğesi

Şekil 4.98.

- **Kisisel**'in **sayac** bloğu **2 Saniye boyunca...** de bloğunun içine aktarılır.



Şekil 4.99.

- Etkinlikte kullanılan kod bloklarının sıralı hâli Şekil 4.100'de gösterilmektedir.



Şekil 4.100. Etkinlikte kullanılan kod blokları

BÖLÜM 4

ÖLÇME VE DEĞERLENDİRME

Yapılan çalışmalarını kontrol ediniz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Değişkenleri tanımlayabilme			
Değişken oluşturma			
Liste oluşturma			
Listeye öge ekleme			

4.4.10. Olaylar

Programlamada **olaylar**, bir işaret verildiğinde belirli durumların başlatılması olarak anlaşılabilir. Örneğin, zil çaldığında dışarı çık. Zil çalma, bir olaydır ve dışarı çıkma zil çaldığında gerçekleştirilen durumdur. Programlama da belirli durumlar için belirli eylemler gerçekleştirilebilir.

11. ETKİNLİK: KLAVYE TUŞLARIYLA OYUNLAR

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Blok tabanlı kodlama ile kodlama yapar.

Olaylar mantığını bilir.

Koordinat düzlemini bilir.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Öğrencilere Scratch kod menülerinden bahsedilir. Benzer kod bloklarının ortak menülerinde bulunduğu ifade edilir. Öğrencilere bir programda, örneğin bir bilgisayar oyununda bir karakteri hareket ettirmek için neler yaptıkları sorulur. Örnek yanıtlar: Klavyede yön ok tuşlarına basmak, farklı hareketler yapmak için a, s, d, w ve x gibi tuşlara basmak, fare ile bir noktaya veya bir hedefe tıklamak. Öğrencilerden gelen yanıtlara göre programı başlatmak, bir tuşa basmak, bir kuklaya tıklamak gibi eylemler gerçekleştiğinde kodların çalışabilmesi için Scratch'de hangi kod menüsünün kullanılması gerektiği sorulur.

BÖLÜM 4

Örneğin öğrencilere “Klavyeden boşluk tuşuna basılınca kuklamızın hareket etmesini isterseniz hangi blokları kullanırsınız?” diye sorulur. Scratch **olaylar** menüsünde yer alan bloklar gösterilir.

SÜREÇ

Bu etkinlikte öğrencilerinize koordinat düzleminde belirli bir hızda X ve Y ekseninde hareket eden iki topu merkezde üst üste getirme amaçlı bir oyun yaptırınız.

Oyunun kuralları: Ekranı kırmızı ve yeşil renkte iki top yerleştiriniz. Kırmızı basketbol topu X ekseninde (sağ-sol) hareket etsin. Yeşil basketbol topu ise Y ekseninde (yukarı-aşağı) hareket etsin. Oyunda klavyeden “a” tuşuna basıldığında kırmızı top dursun. Yeşil topu durdurmak için ise “s” tuşuna basınız.

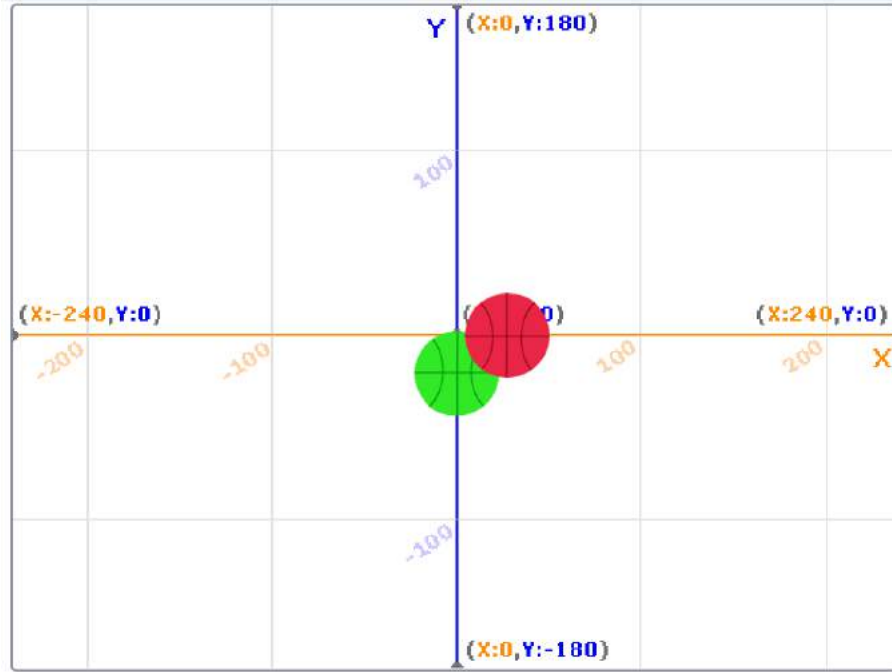
Şekil 4.102’de örnek oyuna ait ekran görüntüsü yer almaktadır.

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri siliniz. Kukla olarak **Basketball** kuklasını sahneye ekleyiniz. İki tane topa ihtiyacınız olacağı için işlemi tekrar ediniz ve sahnede iki adet **Basketball** kuklasından olmasını sağlayınız. Oyunda **Basketball1** ve **Basketball2** kuklaları yer alacaktır.



Şekil 4.101. Kukla kopyalama

- Kukla eklendikten sonra kuklalar bölümünde kuklaların simgelerini görebilirsiniz. Aynı kuklayı eklemek için bir başka yöntem de bu kuklayı kopyalamaktır. Şekil 4.101’deki gibi kuklanın üzerine sağ klik yaptığınızda açılan menüden **kopyasını çıkart** seçerseniz kuklayı kopyalamış olursunuz. Bir kuklaya kodlar eklendikten sonra kopyalarsanız kodları da kopyalamış olursunuz.
- Sahnenin arka planı ilk açıldığında boş olarak geldiği için dekor olarak **Xy-grid** koordinat düzlemini arkaplanı seçiniz.



Şekil 4.102. Örnek oyunun görüntüsü

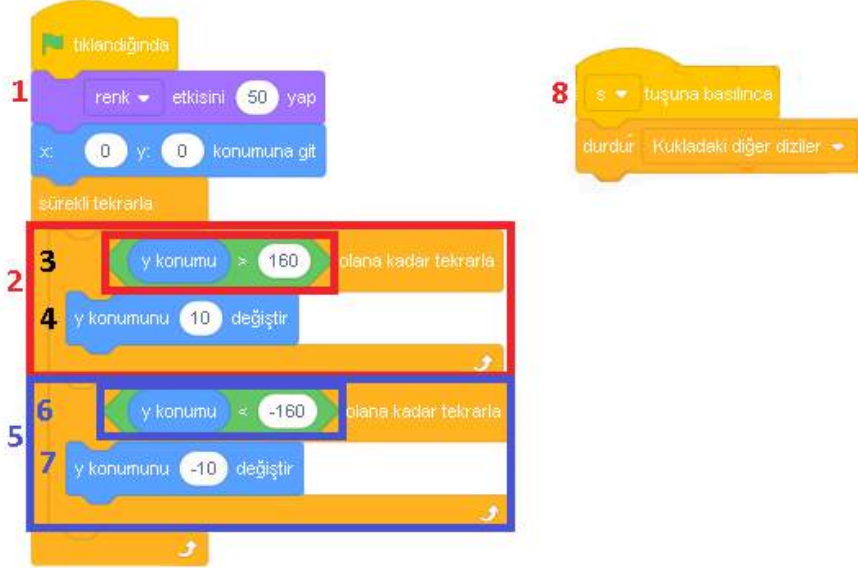
- **Basketball1** kuklası için Şekil 4.103'teki kod bloklarını ekleyiniz.

```
1 tiklandığında
2 renk etkisini 180 yap
3 x: 0 y: 0 konumuna git
4 sürekli tekrarla
5 kenara geldiyse sektir
6 10 adım git
7 a tuşuna basılınca
8 durdur Kukladaki diğer diziler
```

Şekil 4.103. Basketball1 kuklası için kod blokları

BÖLÜM 4

- **Olaylar** menüsü altında yer alan **yeşil bayrak tıklandığında** bloğunu (1) ekleyiniz. Program çalıştırıldığında gerçekleşecek kod bloklarını da bu bloğun altına ekleyiniz.
- Kuklaları ayırt etmek için **Basketball1** kuklası **renk etkisini 180 yap** bloğunu kullanarak kuklanın kırmızı olmasını sağlayabilirsiniz. Kırmızılık **yeşil bayrak** tıklandıktan sonra gerçekleştiğini görebilirsiniz.
- **Basketball1** kuklasını **x: 0 y: 0 konumuna git** bloğuyla (3) dekorun merkezine ortalamış olursunuz. Bu blok **Basketball1** kuklasının x eksenini ortalayarak hareket etmesini sağlayacaktır.
- Şekil 4.103'teki 2 ve 3. bloklar sadece **yeşil bayrak tıklandığında** bir kez çalışmaktadır.
- **Sürekli tekrarlar** bloğu (4) kullanılarak **Basketball1** kuklasına oyun sürecince çalışacak kodları ekleyebilirsiniz. **Hareket** menüsü altında yer alan **kenara geldiye sektir** bloğu kullanılarak sürekli olarak **Basketball1** kuklasının kenara geldiye sektirilmesi ile bu kuklanın X ekseninde (sol-sağ) kenarlara geldiğinde sekerek harekete devam etmesini sağlamış olursunuz. Kuklanın hareket hızını (adım sayısını) belirlemek için **10 adım git** bloğunu kullanınız.
- Hareket eden topu durdurmak için olaylar menüsünde yer alan **... tuşuna basılınca** bloğunu kullanınız. Bu bloğun altına yerleştireceğiniz olay, bu blokta belirlediğiniz klavye tuşuna basıldığında gerçekleşecektir. Bu blokta açılır menüden **"a"** harfini seçerek klavyeden **"a"** harfini aktif hâle getirirsiniz. **a tuşuna basılınca** kontrol menüsü altında yer alan **durdur tümü** bloğunu ekleyerek **durdur kukladaki diğer diziler** şeklinde seçerseniz Şekil 4.103'te görülen basketball1 kuklasına ait tüm blokların çalışmasını durdurmuş olursunuz.
- Programı, yeşil bayrağa tıklayarak çalıştırdığınızda basketball1 kuklasının önce kırmızı olup merkeze geldikten sonra sola sağa doğru x ekseninde hareket ettiğini görebilirsiniz. Program çalışırken klavyeden **"a"** harfine basarsanız kırmızı basketbol topunun hareketini durdurursunuz.
- **Basketball2** kuklası için Şekil 4.104'teki kod bloklarını kullanabilirsiniz.



Şekil 4.104. Basketball2 kuklası blok dizileri

- **Basketball2** kuklasının kodları **Basketball1** kuklasının kodlarından farklılıklar göstermektedir. İki topunda aynı anda başlaması için diğer kuklada da seçtiğiniz **yeşil bayrak tıkladığında** bloğunu kullanınız. Bu bloğun altına **renk etkisi 50 yap** bloğu ile **Basketball2** kuklasının rengini **yeşil** yapabilirsiniz. Bu kuklayı da ilk kukla gibi **x: 0 y: 0 konumuna git** bloğunu kullanarak merkeze getiriniz.
- **Sürekli tekrarlar** içinde **Basketball1** kuklasından farklı olarak **olana kadar tekrarlar** blokları yer almaktadır. Kuklayı Y ekseninde hareket ettirmek için **y konumunu ... değiştir** (4 ve 7) bloğunu kullanınız. Ancak bu blok **kenara geldiye sektir** bloğuyla birlikte **Basketball1** kuklasındaki gibi kullanılırsa oyun çalıştırıldığında yeşil basketbol topu kenara geldiğinde sekerek aşağı doğru yönelmemekte sürekli yukarı gitmeye çalışmaktadır. **Basketball2** kuklasının Y ekseninde (yukarı aşağı) sürekli hareket etmesini sağlamak için Y ekseninde en üst noktaya ve en alt noktaya geldiği durumları (2 ve 5) kontrol ederek bu noktalara gelene kadar (3 ve 6) sürekli olarak yükselmesini veya alçalması (4 ve 7) sağlamanız gerekir. X eksen -240 ile 240 arasında değerler alırken Y eksen -180 ile 180 arasında değerler almaktadır. Şekil 4.104'te 2. blokta **hareket** menüsünde yer alan **y konumu** (kuklanın Y ekseninde bulunduğu nokta) 160'tan büyük olana kadar (3) sürekli olarak **y konumunu 10 değiştir** (4) bloğunu kullanarak yukarı doğru hareket etmesini sağlarsınız. Şekil 4.104'te 5. blokta **hareket** menüsünde yer alan **y konumu** (kuklanın Y ekseninde bulunduğu nokta) -160'tan küçük olana kadar (6) sürekli olarak **y konumunu -10**

BÖLÜM 4

değiştir (7) bloğuyla aşağı doğru hareket etmesini sağlarsınız. Program çalışırken yeşil topun durması için klavyeden “s” tuşunu tanımlayabilirsiniz.

- Oyunu denemek ve kodlarına ulaşmak için <https://scratch.mit.edu/projects/402719536> adresini ziyaret edebilirsiniz.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

1. Oyun için farklı kuklalar kullanılabilir.
2. Dekor değiştirilebilir.
3. Öğrencilerin **olaylar** menüsünde yer alan farklı olay bloklarını kullanmaları sağlanabilir. Oyunun farklı tuşlara basılarak başlaması ve oynanması için kodlar değiştirilebilir. **Yeşil bayrak tıklanıldığında** bloğu yerine, **olaylar** menüsündeki diğer bloklar kullanılabilir. Oyunun **boşluk tuşuna basılınca** başlaması için kodlar değiştirilebilir.
4. Oyunun hızını değiştirmek için **10 adım git, y konumunu 10 değiştir, y konumunu -10 değiştir** bloklarındaki 10 sayısı yerine farklı sayılar kullanılabilir.

Etkinliğin alternatif uygulanışına ait kodlara <https://scratch.mit.edu/projects/402734872> adresinden erişebilirsiniz.

ÖLÇME VE DEĞERLENDİRME

Öğrencilerinizin yaptığı oyunların istenilen şekilde çalışıp çalışmadığını kontrol edebilirsiniz. Öğrencilere kullandıkları kod bloklarının oyundaki işlevlerini sorabilirsiniz.

Sorular

1. Oyunun farklı tuşlarla oynanması için hangi kod blokları değiştirilmelidir?
2. **Yeşil bayrak tıklanıldığında** bloğu yerine, olaylar menüsünde başka hangi bloklar kullanılabilir?
3. Oyunda topların hızını değiştirmek için hangi kod blokları değiştirilmelidir?

BÖLÜM 4

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Görünümden renk bloğunu kullanma			
Koordinat düzlemini kullanma			
Döngüleri uygulama içinde kullanma			
Kuklayı hareket ettirme			
Bir tuşa basıldığında kod bloğunu kullanma			

12. ETKİNLİK: BALON TIKLAMA OYUNU

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Olayları bilir.

Değişkenleri kullanır.

Sürekli tekrarlarla yapılarının içinde koşullu durumları kullanır.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

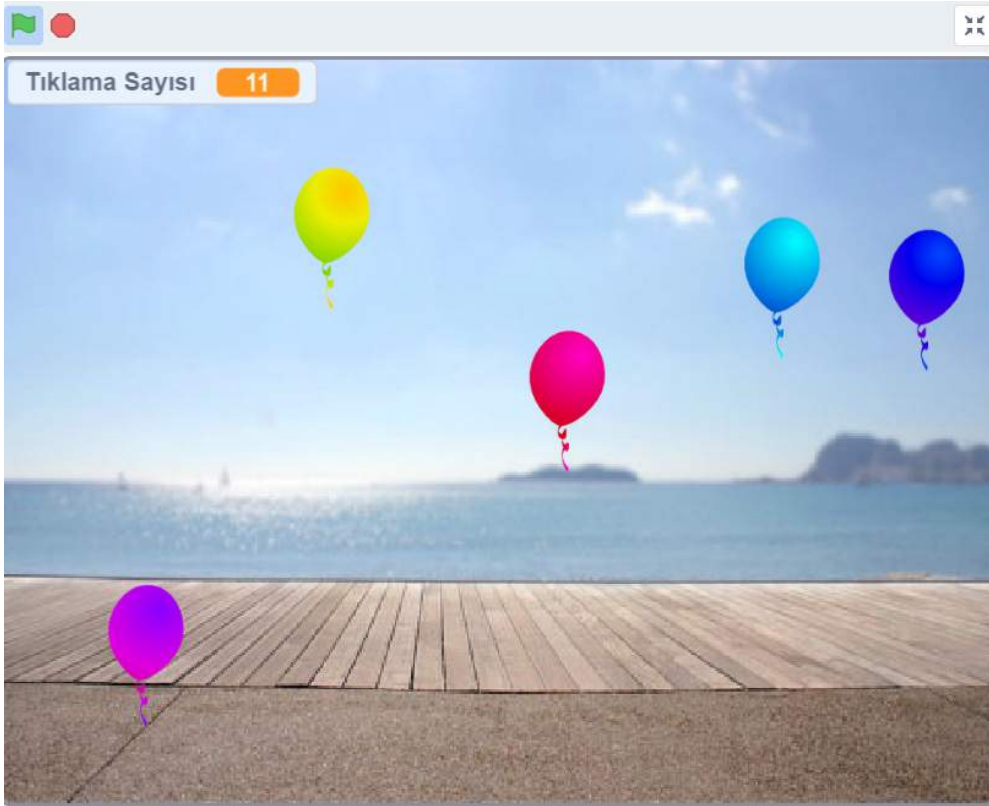
GİRİŞ

Öğrencilere bilgisayarda fare kullanarak oynadıkları oyunlar sorulur. Bu oyunlarda farenin tuşlarına bastıklarında, oyunda bir karaktere veya bir hedefe tıkladıklarında neler olduğu sorulur ve bu tür oyunlar yapılırken netür kod blokları kullanılmış olabileceği sorulur. Öğrencilerden gelen yanıtlar üzerine konuşulur. Öğrencilere Scratch projeler sayfasından <https://scratch.mit.edu/projects/152342065/> ve <https://scratch.mit.edu/projects/82087798/> gibi örnek projeler gösterilir. Bu projelerde belirtilen hedef tıkladıkça oyuncunun puanı artmaktadır. Örnekteki oyunlarda Scratch'deki **olaylar** menüsü altında hangi blokların kullanıldığını öğrencilerin bulması için sorular sorarak onları yönlendiriniz.

SÜREÇ

Bu projede öğrencilerinize uçan balonları tıklayarak patlatacakları bir oyun yaptırınız.

Oyunun kuralları: Bu oyunda uçan balonlar tıklanmakta, balonlar tıklandıkça renkleri değişmekte ve ses çıkmaktadır. Aynı zamanda balonlara tıklandıkça ekrandaki tıklama sayısı da artmaktadır. Örnek oyuna ait ekran görüntüsü Şekil 4.105'te gösterilmiştir.



Şekil 4.105. Örnek balon tıklama oyunu

- Scratch masaüstü veya scratch web arayüzünü açınız. Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri silebilirsiniz. Kukla olarak **Balloon** kuklasını sahneye ekleyiniz.
- Sahnenin arka planı ilk açıldığında boş geldiği için dekor olarak **Boardwalk** dekorunu seçiniz.
- Bu aşamada **Balloon 1** kuklasına Şekil 4.106'daki kod bloklarını kullanabilirsiniz.

BÖLÜM 4



Şekil 4.106. Balloon1 kuklası kod blok dizileri

- **Olaylar** menüsü altında yer alan **yeşil bayrak tıkladığında** bloğunu (1) ekleyerek oyunun yeşil bayrağa tıkladığında başlamasını sağlayabilirsiniz. Program çalıştırıldığında gerçekleşecek kod bloklarını bu bloğun altına ekleyiniz.
- **Değişkenler** menüsü altında ... **değişkenini göster** (2) bloğu balonlara tıklanma sayısının tutulması için oluşturulmuş ve **"Tıklama Sayısı"** adı verilerek oyun skorunu tutan değişkenin programda gösterilmesini sağlayabilirsiniz. Tıklama Sayısı'nı sıfırlamak için **Tıklama Sayısı' i 0 yap** (3) bloğunu kullanınız. Bu blok, oyun her başladığında yani yeşil bayrak tıkladığında **Tıklama Sayısını** sıfırlayacaktır.
- **Sürekli tekrarla** bloğu (4) kullanılarak **Balloon1** kuklasına oyun sürecince çalışacak kodları ekleyebilirsiniz. **Hareket** menüsü altında yer alan **y konumunu 1 değiştir** (5) bloğu kullanılarak **Balloon1** kuklasının sürekli olarak yukarı doğru hareket etmesini sağlıyorsunuz.
- Kontrol menüsü içinde yer alan **Eğer ... ise** bloğu içine **Algılama** menüsü altında **kenara değişiyor mu?** şart bloğunu eklerseniz bir şart yapısı oluşturmuş olursunuz. Bu şart yapısı, balon kenara değiştiğinde (uçarken sahnenin en üst noktasına değiştiğinde) kod blokları çalıştırmak için kullanılır. Şart bloğu içinde yer alan **y konumunu -150 yap** bloğu kullanılarak **Balloon1** kuklası yükselerek üstte kenara değiştiğinde tekrar alta (Y:-150) gönderilmesini sağlayacaktır.
- Şekil 4.106'da yer alan sağdaki dizi **olaylar** menüsü altında **bu kukla tıkladığında** (9) blok ile balon tıkladığında (**Balloon1** kuklası) kod bloklarının çalışmasını sağlamış olursunuz.

- Balona her tıkladığında Tıklama Sayısını 1 artırmak için **Değişkenler** menüsü altında yer alan **Tıklama Sayısı'ı 1 kadar değiştir** (10) bloğunu kullanabilirsiniz.
- Balona tıkladığında tıklamanın gerçekleştiğini bildiren bir efekt eklemek için **Ses** menüsü altında yer alan ... **sesini başlat** bloğu (11) nu kullanabilirsiniz. Bu ses blokta "Pop" sesini seçerseniz balona her tıklatıldığında çıkacak sesi seçmiş olursunuz.
- Her tıklamada balonun rengini değiştirmek için **Görünüm** menüsü altında yer alan **renk etkisini ... değiştir** (12) bloğunu kullanabilirsiniz. Örnekte renk etkisi 25 olarak belirlenmiştir.
- Balona tıklandıktan sonra balonun aşağı inmesi yani konumunun y:-150 konumuna gelmesi için **y konumunu -150 yap** (13) bloğunu kullanabilirsiniz.
- Oyunu çalıştırıp balona tıkladığında tıklama sayısının 1 arttığını, her tıklamada "Pop" sesi geldiğini, balonun renginin değiştiğini ve balonun aşağı indiğini görebilirsiniz. Ama sahnede tek balon vardır.
- Balon sayılarını artırmak için Şekil 4.107'deki gibi kuklanın üzerinde sağ tıklama ile **kopyasını çıkart** seçeneğini kullanabilirsiniz. Bu işlemi, istediğiniz balon sayısı kadar tekrarlayabilirsiniz. Kukla içindeki kodlar ile kopyalanacağından tüm balonlar aynı şekilde hareket edecektir. Balon çoğaltıldıktan sonra sahne üzerinde X ekseninde farklı konumlara getirilerek balonların üst üste gelmesini engelleyebilirsiniz. Bu işlemi kuklaların fare ile sürükleyip bırakarak yapılabileceğiniz gibi kuklaya tıkladıktan sonra sahnenin altında yer alan X: konumuna değer vererek de yapabilirsiniz.



Şekil 4.107. Balloon1 kuklasını çoğaltmak

- Oyunu denemek ve kodlarına ulaşmak için <https://scratch.mit.edu/projects/402789246> adresini ziyaret edebilirsiniz.

BÖLÜM 4

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

1. Oyun için farklı kuklalar kullanılabilir, dekor değiştirilebilir.
2. Öğrencilerin **olaylar** menüsünde yer alan farklı olay bloklarını kullanmaları sağlanabilir. Oyunun farklı tuşlara basılarak başlaması ve oynanması için kodlar değiştirilebilir. **Yeşil bayrak tıklan-
dığında** bloğu yerine, **olaylar** menüsündeki diğer bloklar kullanılabilir. Oyunun **boşluk tuşuna
basılınca** başlaması için kodlar değiştirilebilir.
3. Balonların hızını değiştirmek için **y konumunu 1 değiştir** bloğundaki 1 sayısı değiştirilebilir.
4. **Kopyasını çıkart** seçeneği kullanılarak balonlar çoğaltılabilir. Ancak kuklanın kodları değiştiril-
diğinde eski kukla kopyalarının silinip yeni kodların olduğu kuklanın **kopyasını çıkart** seçeneği
kullanılarak çoğaltılması gerektiği unutulmamalıdır.

Etkinliğin alternatif uygulamasına ait kodlara <https://scratch.mit.edu/projects/402808621> ad-
resinden erişebilirsiniz.

ÖLÇME VE DEĞERLENDİRME

Öğrencilerinizin yaptığı oyunların istenilen şekilde çalışıp çalışmadığını kontrol edebilirsiniz. Öğren-
cilere kullandıkları kod blokların oyundaki işlevlerini sorabilirsiniz.

Sorular

1. **Bu kukla tıklan-
dığında** bloğu yerine ... **tuşuna basılınca** bloğu kullanılırsa oyun nasıl çalışır?
Oyun hileleri bu şekilde yapılıyor olabilir mi?
2. **Yeşil bayrak tıklan-
dığında** bloğu yerine olaylar menüsünde başka hangi bloklar kullanılabilir?
3. Oyunda balonların hızını değiştirmek için hangi kod blokları değiştirilmelidir?
4. Bu oyunun daha eğlenceli hâle gelmesi için başka neler yapılabilir?
Süre sınırı koyma, 3 balon sahnedan çıktığında oyunun bitmesi gibi.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Olaylar kod bloğunu kullanma			
Değişkenleri amacına uygun kullanma			
Döngüler ile birlikte koşullu yapıları kullanma			

13. ETKİNLİK: GEZİYORUM

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Farklı sahne arka planlarını kullanır.

Algılama komutlarını bilir.

Haberini sal ile fonksiyonları bilir.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

“Veli toplantısı olduğunda tüm veliler nasıl haberdar oluyor? Tüm öğrenciler, dersin başladığından nasıl haberdar oluyor?” gibi sorular sorarak öğrencilerinizin konuya giriş yapmalarını sağlayınız. Bu tür durumlarda bir haber vericinin olduğunu belirtiniz. Bu haber vericinin bir okul zili, bir hatırlatma e-postası veya bir arkadaş olabileceğinden bahsediniz.

Scratch’de animasyonlar veya programlar yapılırken kodlar bazı olaylara bağlı olarak çalışır. Öğrencilerinize “Bir kukladaki değişiklik diğer kuklayı etkiliyorsa bunun için hangi blokları kullanmak gerekir?” diye sorunuz.

Örnek: Bir oyunda oyun hakkı bittiğinde oyunun durması (tüm kuklaların durması, oyun bitti mesajının verilmesi) nasıl sağlanır? Scratch’de bu tür durumlarda **olaylar** menüsü altında yer alan ... **haber aldığımda**, ...**haberini sal** ve ... **haberini sal ve bekle** blokları kullanılmaktadır.

Öğrencilerinize **olaylar** menüsünde yer alan **ses yüksekliği büyük ... olduğunda** bloğunun nelerde kullanılabileceğini sorunuz.

BÖLÜM 4

SÜREÇ

Bu projede öğrencilerinize dekorların etkili öğrenimi üzerine bir animasyon yaptırılacaktır.

Animasyonun kuralları: Bu animasyonda sahnede yürüyen biri, sahnenin kenarına geldiğinde farklı bir dekorda baştan yürümeye devam etsin. Bir bulut kuklası ise dekor değiştiğinde haber versin. Animasyonumuzda yüksek bir ses olduğunda karakterimiz dursun. Örnek oyuna ait ekran görüntüsü Şekil 4.108’de gösterilmiştir.



Şekil 4.108. Örnek animasyon

- Scratch masaüstü veya scratch web arayüzünü açınız. Sahnenin arka planı ilk açıldığında boş olarak geldiği için dekor olarak **Castle** dekorunu seçiniz. Aynı işlemi diğer dekor içinde uygulayarak **Greek Theatre** dekorunu ekleyiniz.
- Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri siliniz. Kukla olarak **Avery Walking** kuklasını sahneye ekleyiniz. İkinci kukla olarak **Cloud** kuklasını ekleyiniz.

- **Avery Walking** kuklasına Şekil 4.109'daki kod bloklarını ekleyebilirsiniz.



Şekil 4.109. Avery Walking kuklası blok dizisi

- **Olaylar** menüsü altında yer alan **yeşil bayrak tıklandığında** bloğunu ekleyerek oyunun yeşil bayrağa tıklandığında başlamasını sağlayabilirsiniz. Program çalıştırıldığında gerçekleşecek kod bloklarını bu bloğun altına ekleyiniz.
- Animasyon başladığında dekorun **Castle4** olarak ayarlanmasını sağlamak için **Değişkenler** menüsü altında ... **dekoruna geç** (1) bloğunu ekleyiniz.
- **Sürekli tekrarla** bloğu kullanılarak **Avery Walking** kuklasına oyun sürecince çalışacak kodları ekleyebilirsiniz. **Avery Walking** kuklasının sürekli olarak hareket etmesini sağlamak için **Hareket** menüsü altında yer alan ... **adım git** (2) bloğunu kullanabilirsiniz.
- Kontrol menüsü içinde yer alan **Eğer ... ise** (3) bloğuna **Algılama** menüsü altında **kenara değişiyor mu?** şart bloğunu ekleyerek bir şart yapısı oluşturulmuş olursunuz. Bu şart yapısı **Avery Walking** kenara geldiğinde (yürüyerek sahnenin sonuna geldiğinde) kod bloklarını çalıştırmak için kullanılır. Şart bloğu içinde yer alan **olaylar** menüsü altındaki ... **haberini sal** bloğu kullanılarak oluşturulan **kenar** haberi salınmaktadır. Bu bir ilan, duyuru, okuldaki zil sesi gibidir. **Olaylar** menüsü altında ... **haberini sal** bloğunda **yeni haber'**e tıklanarak çıkan pencerede habere **"kenar"** adını veriniz.
- Her programda birden fazla haber olabileceği için haberlere farklı adlar verebilirsiniz. Kullanılırken de ilgili haber adıyla oluşan blok alınarak kullanılır. Örnek: haber1, kenar. Şekil 3'te **olaylar**

BÖLÜM 4

menüsüne tıklayarak haber oluşturma gösterilmiştir. Şekil 4.110'da gösterildiği gibi sırayla 1 ve 2 numaralı bölümlere tıklanarak yeni haber oluşturabilirsiniz.



Şekil 4.110. Yeni haber oluşturma

- **Yeni haber**'e tıkladıktan sonra Şekil 4.111'de açılan pencerede habere isim vererek ve **OK**'e basınız.



Şekil 4.111. Yeni haber adı verme

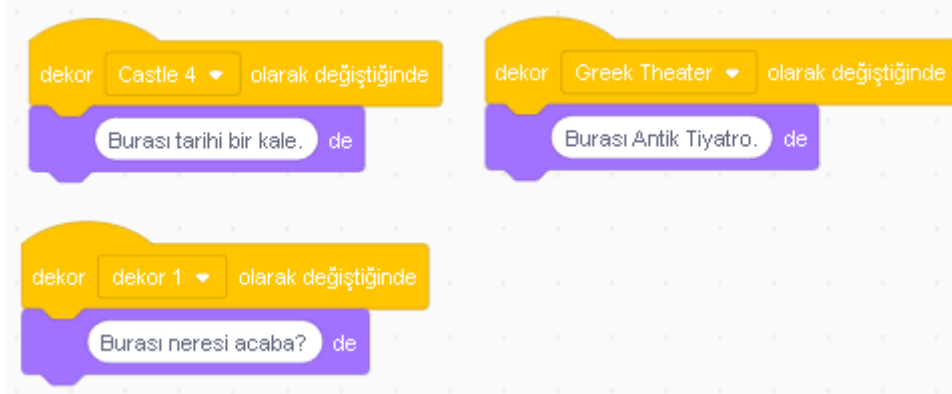
- Şekil 4.109’da soldaki dizideki kod blokları kullanılarak animasyon başlatıldığında **Castle4** dekoru gelir ve yürüyen insan kuklasının kenara değene kadar sürekli olarak hareket ettiğini görürsünüz. Kenara geldiğinde ise (3) **kenar** haberi salınır. Programdaki tüm bileşenlere **kenar** haberi iletilir.
- Şekil 4.109’da yer alan sağdaki dizide **olaylar** menüsü altında **yeşil bayrak tıklandığında** bloğu ile başlamaktadır. Bu bloğu iki kere kullanılmış olacaksınız. Soldaki dizide kuklanın ilerlemesi ve kenara geldiğinde haber salması için kullanılırken buradaki amacı, **Avery Walking** kuklasına yürüme efekti vermektir. **Avery Walking** kuklasının kostümlerine bakarsanız sonraki kostümlerde ayak ve kol hareketlerinin art arda kullanıldığında bir yürüme efekti oluşturduğunu görebilirsiniz. Bu işlemi animasyon sürecince sürekli yapmak için sağdaki dizide **sürekli tekrarlar** döngü bloğu içine **sonraki kostüm** (5) ve **0.2 saniye bekle** (6) bloklarını ekleyebilirsiniz. Bu kodları kullanarak kuklanın yürüme pozisyonları arasında (kostüm) geçiş yapmasını sağlamış olacaksınız. Kostüm geçişleri arasındaki 0.2 saniye farklı kostümleri görebilmek için bir bekleme süresidir. Bu blok olmasa kostüm geçişleri görünmez.
- **Avery Walking** kuklası içine eklenen **haberini alındığında** bloğu, belirlenen haber ulaştığında (**kenar** haberi salındığında) çalışacak kodların yazıldığı bloktur. Şekil 4.112’de kenar haberi alındığında **sonraki dekor**’a geçmek içindir. Yürüyen kuklanın sahnenin sol başına konumlanmasını sağlamak için **X konumunu -190 yap** bloğunu kullanmanız yeterlidir.

BÖLÜM 4



Şekil 4.112. Haber aldığımda kod bloğu

- Şekil 4.112'deki diziyi oluşturarak kuklanın kenara geldiği zaman kenar haberinin salınmasıyla sahneye sonraki dekorun ayarlanmasını ve kuklayı baştaki konumuna gelmesini sağlamış olursunuz. Kenara gelen kukla, dekorun değişmesi ve sahnenin soluna gelmesiyle farklı bir ortamda yürümeye devam ediyormuş görüntüsünü oluşturmuş olacaksınız.
- Dekor değiştiğinde kuklanın dekora ilişkin konuşmalarının görünmesi için **Avery Walking** kuklası içinde Şekil 4.113'te yer alan **görünüm** menüsü altındaki ... **de** bloğunu kullanmanız gerekir.



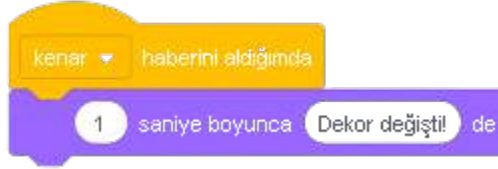
Şekil 4.113. Ddekor değiştiğinde olayı

- Programı çalıştırdığınızda ve dekor "Castle4" olduğunda kuklanın konuşma balonu içinde "**Burası tarihi bir kale**" yazdığını, dekor "Greek Theater" olduğunda "**Burası Antik Tiyatro**" yazdığını, dekor "dekor1" olduğunda ise konuşma balonu içinde ise "**Burası neresi acaba?**" yazdığını görebilirsiniz.
- Tüm animasyonu durdurmak için **Avery Walking** kuklası içinde 4.114'te gösterildiği gibi mikrofondan yüksek şiddette (80 den büyük) bir ses geldiğinde **durdur tümü** bloğunu kullanmanız yeterli olacaktır.



Şekil 4.114. Ses yüksekliği olduğunda bloğu

- Kukla kenara geldiğinde kenar haberi alınmasının başka bir kukla tarafından nasıl haber alındığını göstermek için **Cloud** kuklasının içine Şekil 4.115'te gösterilen kod bloklarını yerleştirmeniz gerekir.



Şekil 4.115. Kenar haberi alındığında

- Bu dizi aracılığıyla **kenar haberi alındığında** Cloud kuklasının konuşma balonunda 1 saniye boyunca "Dekor değişti!" yazmasını sağlarsınız.
- Oyunu denemek ve kodlarına ulaşmak için <https://scratch.mit.edu/projects/402789246> adresini ziyaret edebilirsiniz.

ETKİNLİĞİN ALTERNATİF UYGULANIŞI

1. Oyun için farklı kuklalar kullanılabilir, dekor değiştirilebilir.
2. Öğrencilerin **olaylar** menüsünde yer alan farklı olay bloklarını kullanmaları sağlanabilir. Oyunun farklı tuşlara basılarak başlaması ve oynanması için kodlar değiştirilebilir.
3. Kuklanın yürüme hızı değiştirilebilir. Kuklanın sonraki kostüm bloğunda 0.2 saniye artırılabilir veya azaltılabilir.
4. Ses yüksekliği yerine, zamanlayıcı kullanılabilir.

Etkinliğin alternatif uygulanışına ait kodlara <https://scratch.mit.edu/projects/402920227> adresinden erişebilirsiniz.

BÖLÜM 4

ÖLÇME VE DEĞERLENDİRME

Öğrencilerinizin yaptığı oyunların istenilen şekilde çalışıp çalışmadığını kontrol ediniz.. Öğrencilerin kullandıkları kod bloklarının oyundaki işlevlerini sorunuz.

Sorular

1. Animasyonda kuklanın yürüme hızını değiştirmek için hangi kod blokları değiştirilmelidir?
2. Animasyonda bulut kuklası diğer kuklanın kenara geçtiğini nasıl anlamaktadır?
3. Zamanlayıcı ve ses yüksekliği ... **olduğunda** blokları hangi oyunlar için kullanılabilir?
Ses ölçer yapmak, zaman karşı yarışlar vb.

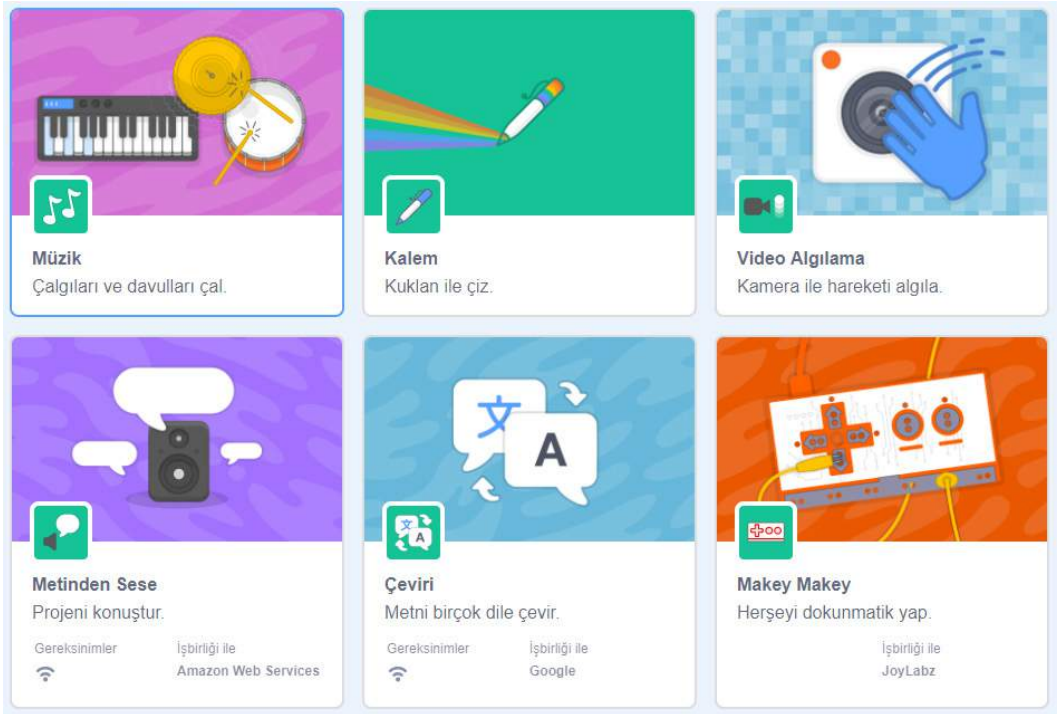
Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Farklı sahneler ile çalışma			
Algılama komutlarını kullanma			
Fonksiyonları bilme			
Haberini sal kodunu kullanma			

4.4.11. Projelere Eklenti Ekleme

Projelere çeşitli eklentiler eklenerek projenin zenginleşmesi sağlanabilir. Müzik çalma, video algılama, çizim yapma gibi fonksiyonlar eklentiler menüsünden projelere eklenebilir. Bunun için Kod Blokları menüsünün en altıda yer alan Şekil 4.116'da gösterilen simgeye tıkladığınızda Şekil 4.117'deki **Eklentiler** menüsü açılır. Buradan projede ihtiyaç duyulan eklenti seçilerek eklenebilir.



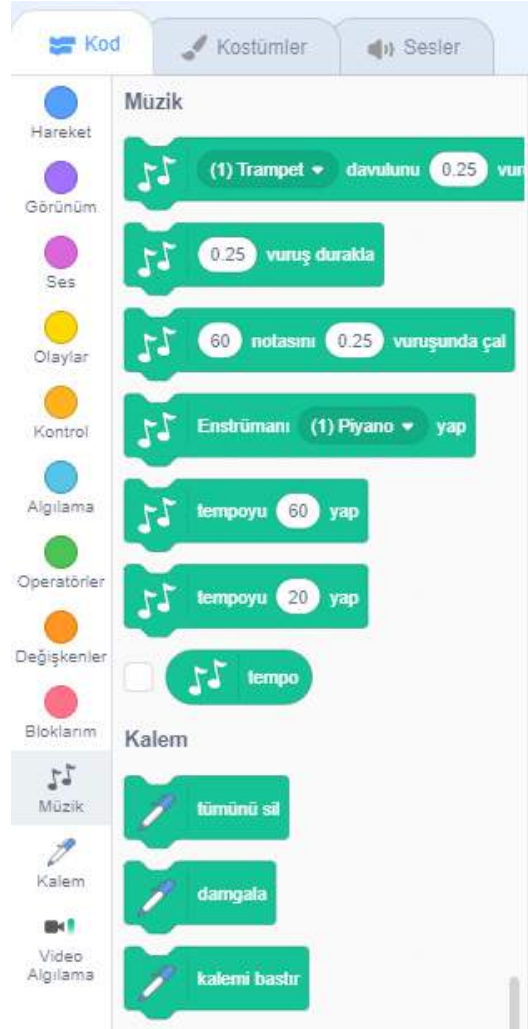
Şekil 4.116. Eklenti butonu



Şekil 4.117. Eklenti ekleme ekranı

Eklenmek istenen eklenti eklendiği zaman kod blokları listesine Şekil 4.118'deki seçilen eklentinin kod blokları gelir. Buradan eklenmek istenen nesneye kod blokları eklenebilir. Şekil 4.118'de eklenti olarak eklenmiş Müzik, Kalem, Video Algılama eklentilerinin kod blokları görülmektedir.

BÖLÜM 4



Şekil 4.118. Seçilen eklentilerin kod blokları

14. ETKİNLİK: GEOMETRİK SANAT KODLAMA

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Döngü kavramını kullanır.

İç içe döngü kavramını kullanır.

Eklenti ekler ve kullanır.

Tekrarlayan yapılar kullanarak desen oluşturur.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

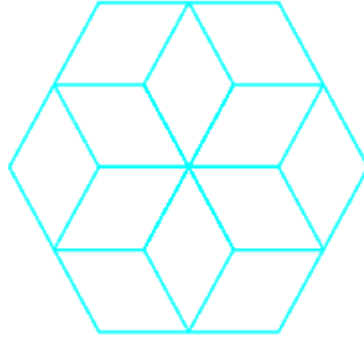
- Bilgisayarda kodlama yapılarak geometrik şekiller çizdirilebilir mi?
- Bilgisayarda kodlama yapılarak farklı desenler çizdirilebilir mi?
- Öğrencileriniz bu etkinlikte bilgisayara geometrik şekiller ve desenler çizdireceğinizi söyleyerek derse giriş yapınız.

SÜREÇ

Bu projede öğrencilerinize bir geometrik şekiller çizen animasyon yaptırınız.

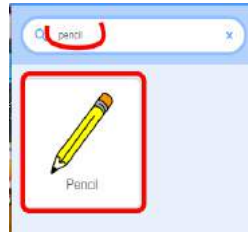
Animasyonun kuralları: Kalem kullanarak ekranda üçgen, dörtgen gibi çokgenler çizmekle birlikte renk, çokgen sayısı gibi bilgilere göre dinamik desen çizebilen uygulama geliştirmek. Örnek oyuna ait örnek ekran görüntüsü Şekil 4.119'da gösterilmiştir.

BÖLÜM 4



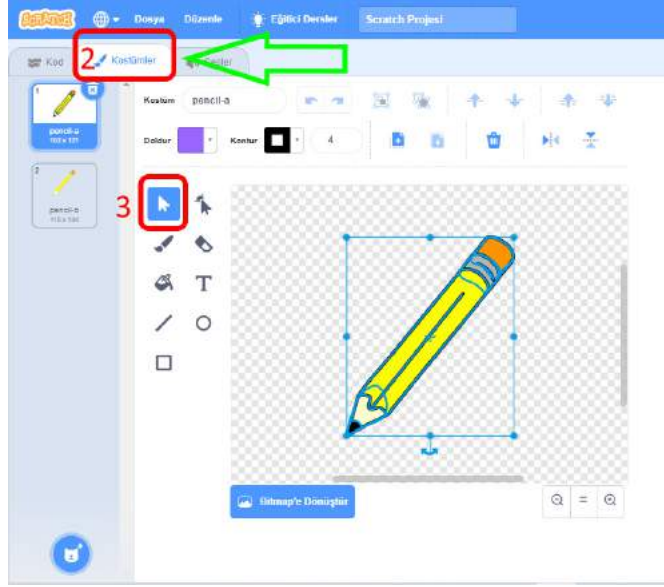
Şekil 4.119. Oyuna ait ekran görüntüsü

- Scratch masaüstü veya scratch web arayüzünü açınız. Sahnenin arka planı ilk açıldığında boş olarak geldiği için dekor olarak istediğiniz dekorunu seçebilirsiniz ya da boş bırakabilirsiniz.
- Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri siliniz. Kukla olarak **Pencil** kuklasını sahneye ekleyiniz.



Şekil 4.120. Pencil kuklası

- Kalem aracı kullanılarak resimler çizileceği için kuklanın merkez noktasını kalemin ortasından uç kısmına taşıyınız. Bu işlemi gerçekleştirmek için kalemi ekledikten sonra **Kostümler** sekmesinde **Pencil-a** kostümünü seçiniz (Şekil 4.121).



Şekil 4.121. Pencil kostüm görünümü

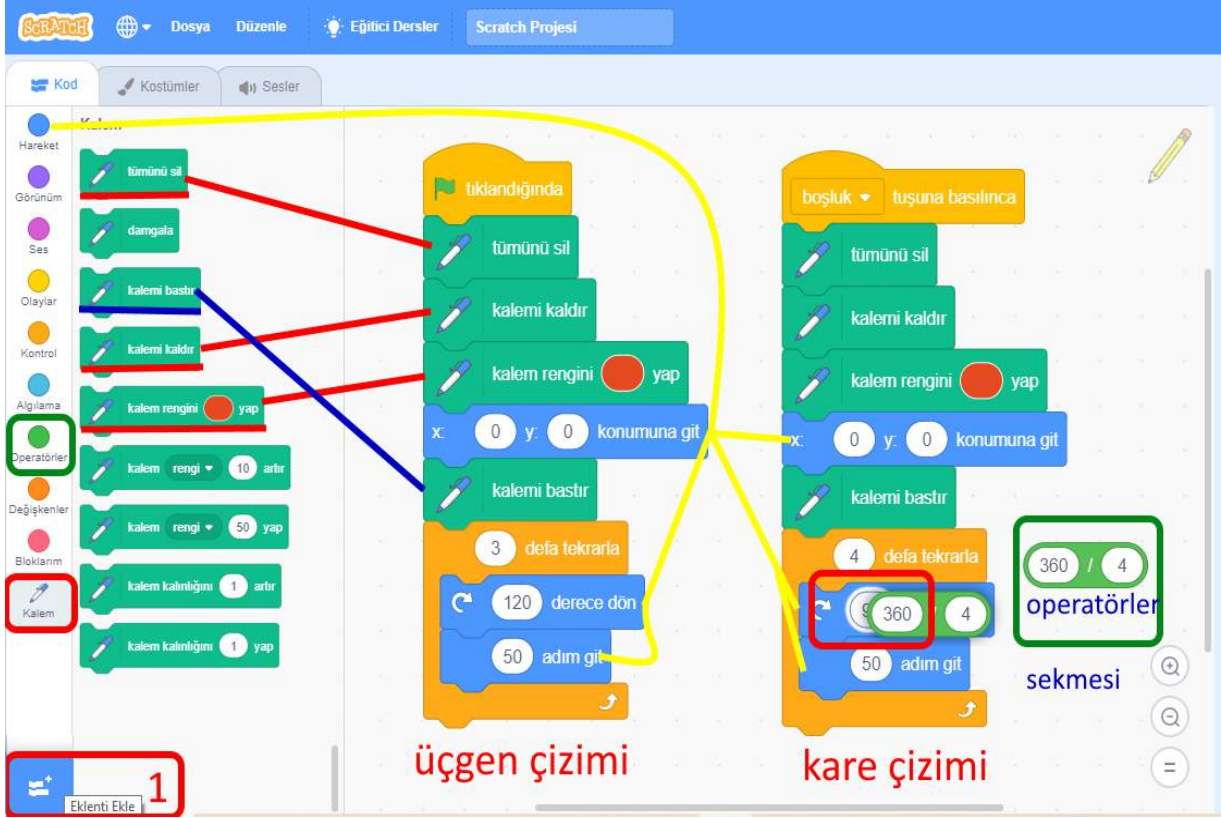
- Pencil-a nesnesi seçili iken orta alanda bulunan resim düzenleme bölümünde seçim (3) aracına tıklayıp kalemın tamamını seçiniz. Ardından sağ üst tarafa taşıyarak Şekil 4.122’de gösterildiği gibi kalemın ucunun tam ortada bulunan hedef sembolünün üzerine getiriniz.



Şekil 4.122. Pencil-a kostümlü kuklanın ucunu merkez yapma görünümü

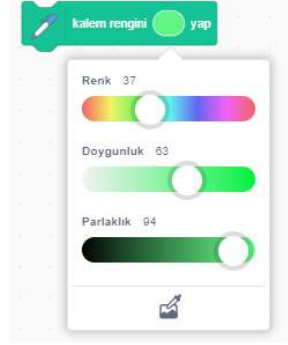
- Bu işlemden sonra çizim için kalem eklentisini eklemeniz gerekir. Bunun için arayüzün sol alt kenarında duran eklentiler düğmesine tıklattığınızda karşınıza gelen **Bir Eklenti Seç** ekranından **Kalem** eklentisini seçerek ekleyebilirsiniz.
- Kalem kuklası seçili iken üçgen ve kare çizmek için gerekli kod bloklarını Şekil 4.123’teki gibi ekleyebilirsiniz.

BÖLÜM 4



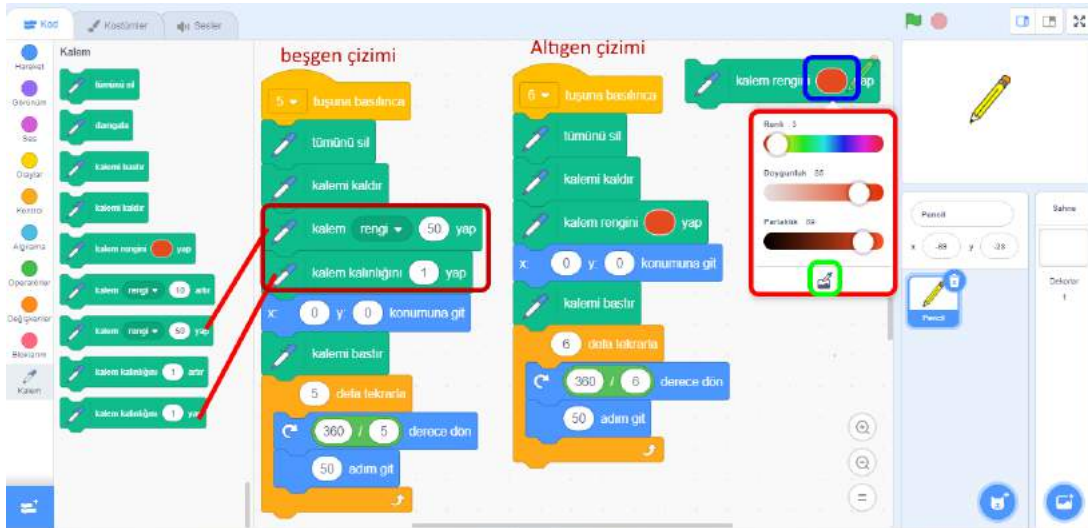
Şekil 4.123. Kalem kuklasına eklenecek kod blokları

- Şekil 4.123'teki **yeşil bayrak olayının** altında bulunan kodlarla kırmızı renkte üçgen ve boşluk tuşu olayın altında bulunan kod blokları ile kırmızı renkte kare çizimimizi sağlayan kod blokları bulunmaktadır. Hem üçgen çizimi hem de kare çizimi bloklarında **Tümünü Sil** kod bloğu daha önce yapılan çalışmaların sahneden temizlenmesi için kullanılır. Kodları tekrar tekrar çalıştırdığınızda ekranın her seferinde temizlendiğini görebilirsiniz. **Kalemi kaldır** kod bloğu da kalem daha önceki çalıştırmada basılı kaldı ise istenmeyen çizimlerin oluşmasının önüne geçmek için kullanılır. **Kalem rengini ... yap** kod bloğu çizim yapacağımız rengi ayarlamak için kullanılır. Renklerin sayısal karşılığı **kalem rengini ... yap** bloğunda renge tıkladıktan sonra altında açılan seçeneklerden Şekil 4.124'te görüldüğü gibi denemeler yaparak farklı renkler de elde edebilirsiniz.



Şekil 4.124. Kalem rengini belirleme işlemi

- **X:0 y:0 noktasına git** komutu ile kalemi merkez noktaya alırsınız. Böylece kodu her çalıştırdığınızda merkez noktadan çizim yapmaya başlamış olur. **Defa tekrarla** bloğu içinde yer alan kod blokları tekrar tekrar çalıştırmak için kullanılır. Bu kod bloklarında bölme işlemi kullanılarak kalemin çizim esnasında dönmesi için kullanılır. Burada **işlem operatörü** kullanmanın amacı beşgen, altıgen gibi farklı çokgen çizimleri de yapabilmemiz içindir.
- Beşgen ve altıgen çizimlerine ait kod blokları için Şekil 4.125'te görebilirsiniz. Şekil 4.125'te beşgen ve altıgen çiziminde kalem rengi ve kalem kalınlığı ile ilgili yeni bloklar eklenmiştir. Scratch'te kalem rengi 0-100 arasında değer alabilir. 100'den büyük değerler yazılsa bile Scratch mod 100'e göre işlem yapar.



Şekil 4.125. Beşgen ve altıgen çizime kodları

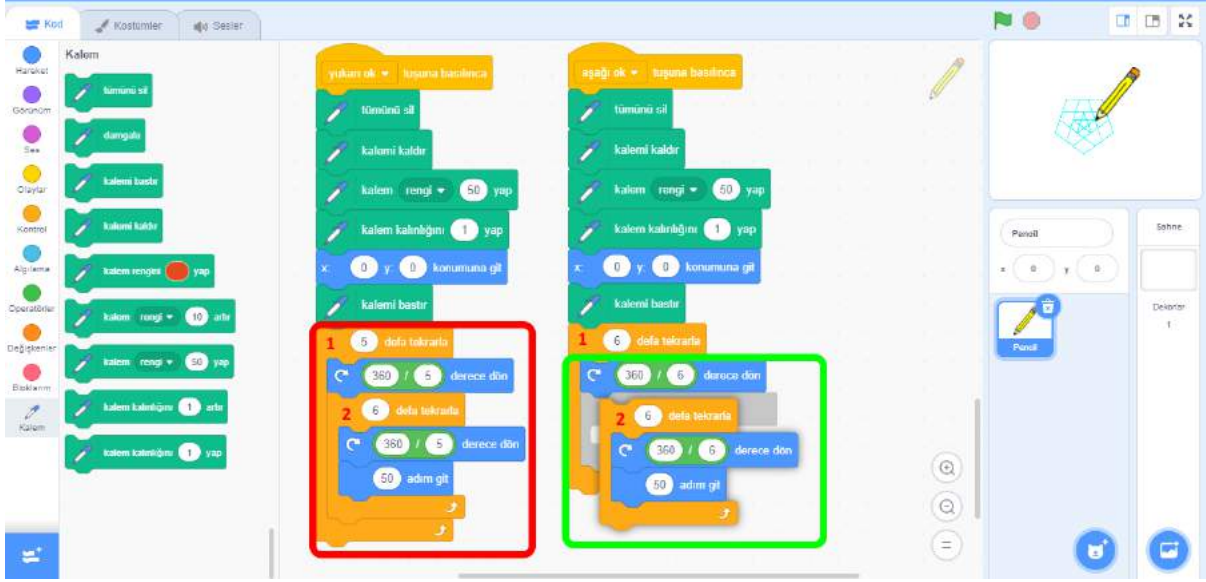
BÖLÜM 4

- Şekil 4.126'da gösterilen kod bloğunda kullanıcıya kaç kenarlı çokgen çizdirmek istediği sorusu sorularak kullanıcının verdiği yanıtı göre çokgen çizimi yapan kod blokları görülmektedir.



Şekil 4.126. Kullanıcıdan alınan kenar sayısına göre çizim yapma kodları

- Çizimin tekrar sayısının kullanıcının gireceği sayı ve her bir kenar açısının **360/kenar sayısı**'na göre program çalışacaktır. Bu etkinlikte kalem rengi, kalem kalınlığı, kenar sayısı gibi birden çok parametreyi daha dinamik hâle getirebilirsiniz.
- Komut blokları ile tek seferde daha çok sayıda çokgen çizdirerek farklı desenler elde etmek için gerekli olan kod blokları Şekil 4.127'de verilmiştir.



Şekil 4.127. Desen çizme kod blokları

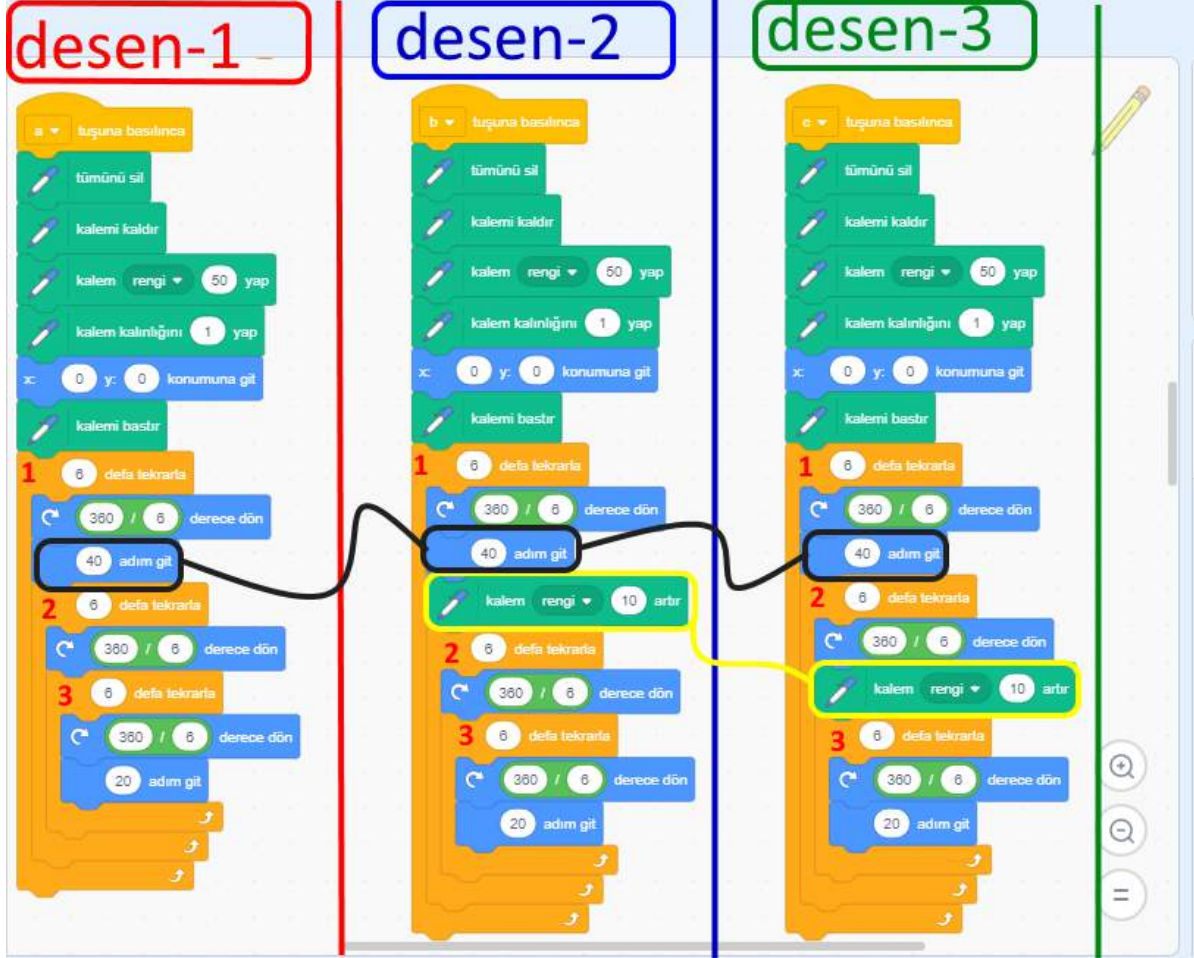
- Şekil 4.127’de **defa tekrarla** (döngü) bloklarının iç içe kullanılması ile Şekil 4.128’deki gibi desenler elde edebilirsiniz. Farklı sayılar, parametreler girerek çok farklı çıktılar ve renkleri değiştirerek rengârenk desenler elde edebilirsiniz.



Şekil 4.128. Örnek desen çıktıları

- 3 farklı desenin nasıl çizildiğini ile ilgili örnek kod blokları Şekil 4.129’da gösterilmiştir.

BÖLÜM 4



Şekil 4.129. Desen1-2-3 kod blokları

- Son olarak tekrar (döngü) bloğunun içinde bulunan **40 (kırk) adım git** bloğunun görevi farklı bölgelerde çizim yapmasıdır. Bu kod bloğu, iç içe tekrar kod bloklarının üst üste aynı noktada desen oluşmasını engellemek içindir.

Etkinliğin kodlarına v1 <https://scratch.mit.edu/projects/385730246/> adresinden ulaşabilirsiniz.

Etkinliğin kodlarına v2 <https://scratch.mit.edu/projects/385728853/> adresinden ulaşabilirsiniz.

ÖLÇME VE DEĞERLENDİRME

1. 10 kenarlı çokgen oluşturma uygulaması yaptırınız.
2. 8 kenarlı desen (iç içe 2 döngü kullanarak) oluşturma uygulaması yaptırınız.
3. 8 kenarlı desen (iç içe 3 döngü kullanarak) oluşturma uygulaması yaptırınız.
4. Yıldız çizdirme uygulaması yaptırınız. (İpucu: Açığı olarak 144'ü deneyiniz.)

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Döngü kullanma			
İç içe döngü kavramını kullanma			
Eklenti ekleme			
Tekrarlayan yapılar kullanarak desen oluşturma			
Kalem rengini ayarlama			

14. ETKİNLİK: MÜZİK ETKİNLİĞİ

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Döngü kavramını kullanır.

İç içe döngü kavramını kullanır.

Eklenti ekler ve kullanır.

Kostümleri kullanır.

YÖNTEM/TEKNİKLER

Gösterip yaptırma, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Öğrencilerinize aşağıdaki soruları yönelterek derse giriş yapınız.

- Bilgisayar kodlama yaparak sahne performansı sergilenebilir mi?
- Bilgisayar kodlama yaparak hareket animasyonu nasıl yapılır?

SÜREÇ

Bu projede öğrencilerinize bir müzik etkinliği yapabilen animasyon yaptırınız.

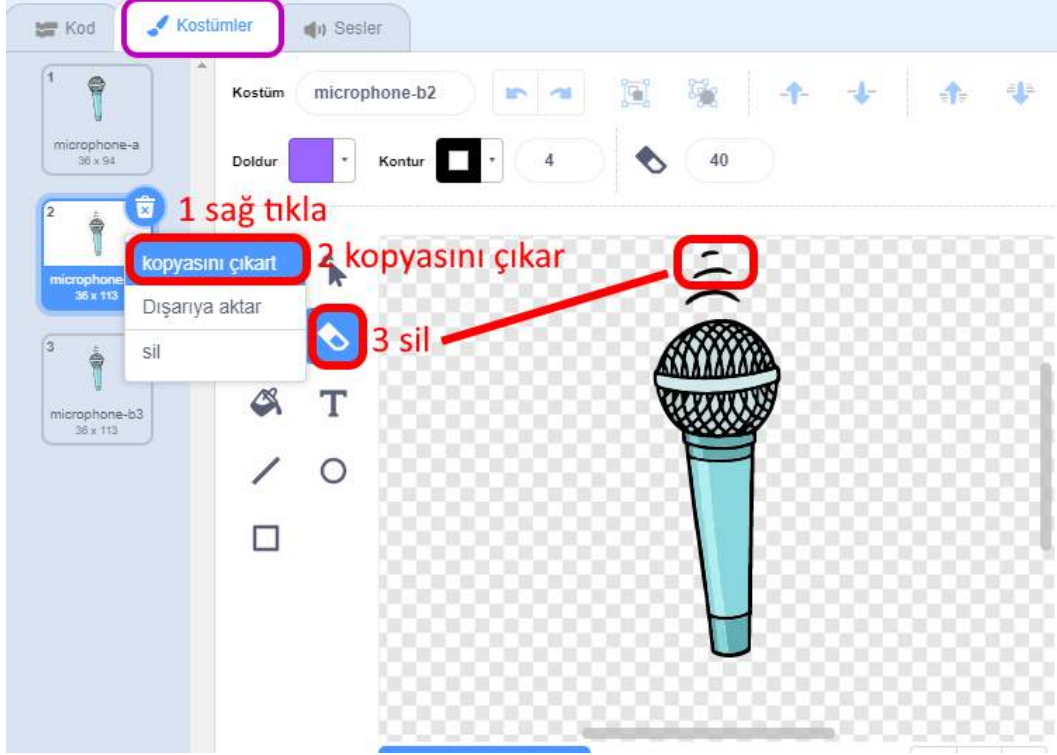
Animasyonun kuralları: Mikrofon dışındaki karakterlerin başlangıçta gizleneceği, heber geldikçe gözükken kuklaların olduğu sahne performansı sergileyiniz. Örnek oyuna ait örnek ekran görüntüsü Şekil 4.118’de gösterilmiştir.



Şekil 4.130. Oyuna ait ekran görüntüsü

- Scratch masaüstü veya scratch web arayüzünü açınız. Sahnenin arka planı ilk açıldığında boş olarak geldiği için dekor olarak müzik sekmesinden **Spotlight** dekorunu seçerek ekleyiniz.
- Uygulama açıldığında sahne üzerinde gelen kedi karakteri ile ilgili bir işlem yapılmayacağı için karakteri siliniz. Kukla olarak **Buttun 1**, dans kategorisinden **Cassy Dance** ve **Champ99**, müzik kategorisinden **Microphone** ve **Radio** kuklalarını sahneye ekleyiniz.
- Bu uygulamada mikrofon dışındaki karakterler başlangıçta gizlenecek, **haber1** haberi gelince kuklalar görünüp kodlarını çalıştıracaklardır.
- Öncelikle **Müzik** eklentisini eklentiler düğmesinden seçerek ekleyiniz.
- Kuklaları ekledikten sonra öncelikle **microphone** kuklasını seçip kostümler sekmesinden **microphone-b** kostümünü seçiniz. Şekil 4.131’de gösterildiği gibi **microphone b** kostüme gelip sağ tıklama ile gelen menüden **kopyasını çıkart** ile kostümün kopyasını oluşturunuz. Böylelikle **microphone-b2** kostümü oluşacaktır. Microphone-b2 kostümünü seçip **sil** butonu ile mikrofon görselinin üzerinde bulunan dalga efektlerinden ikisini siliniz. Bu işlemi tekrarlayarak **microphone-b3** kostümünü de oluşturunuz. **microphone-b3** kostümünde sadece mikrofonun üstündeki bir dalgayı siliniz.

BÖLÜM 4

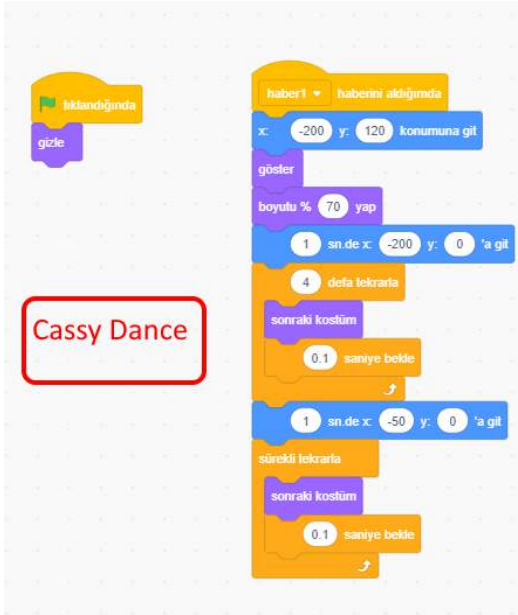


Şekil 4.131. Microphone kostüm ayarları görünümü

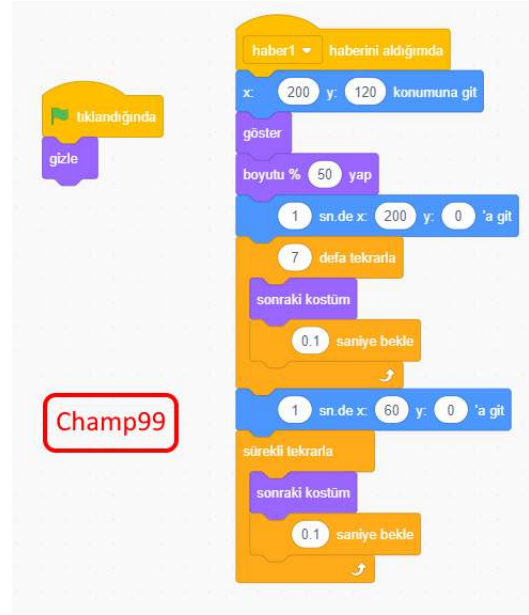
- Böylece Microphone kuklasına ait olan iki kostümü dörde çıkarmış olursunuz. Kodlama yaparken bu kostümleri sırayla göstererek ufak bir animasyon yapabileceksiniz.
- Microphone kuklasına eklenecek kod blokları Şekil 4.132’de gösterilmektedir.

BÖLÜM 4

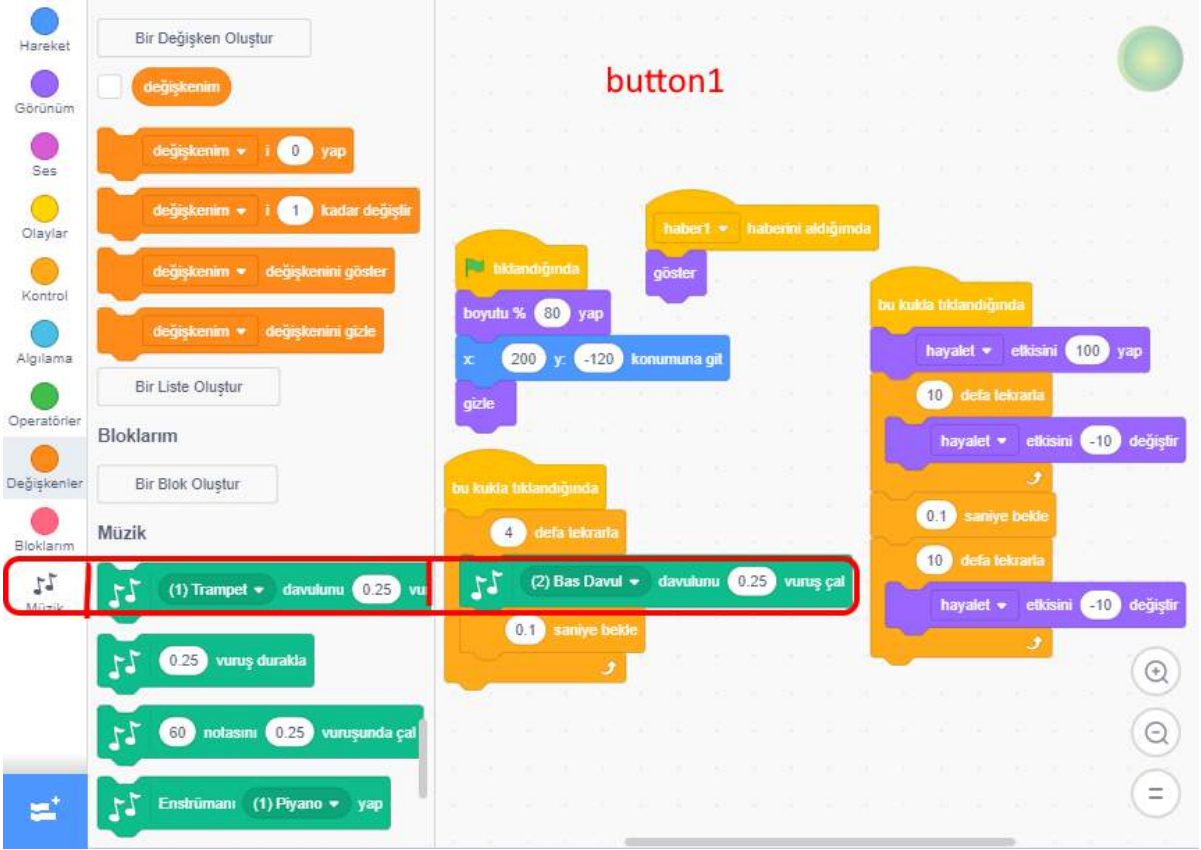
- **Radio** kuklasına Şekil 4.135'teki kod bloklarını ve **button1** kuklasına Şekil 4.136'daki kodlarını ekleyebilirsiniz. Bu kuklalar da **Haber1** haberi gelene kadar gizleneceklerdir. Bu kod blokları ile **Haber1** haberi gelince görünür hâle getirmiş ve kuklalara tıklanınca hem müzik çalıp hem de görsel etkiler oluşturmuş olacaksınız.



Şekil 4.133. Cassy dance kuklası kodları



Şekil 4.134. Champ99 kuklası kodları



Şekil 4.135. Button1 kuklası kodları

BÖLÜM 4



Şekil 4.136. Radio kuklası kodları

Etkinliğin kodlarına <https://scratch.mit.edu/projects/385728668/> adresinden ulaşabilirsiniz.

ÖLÇME VE DEĞERLENDİRME

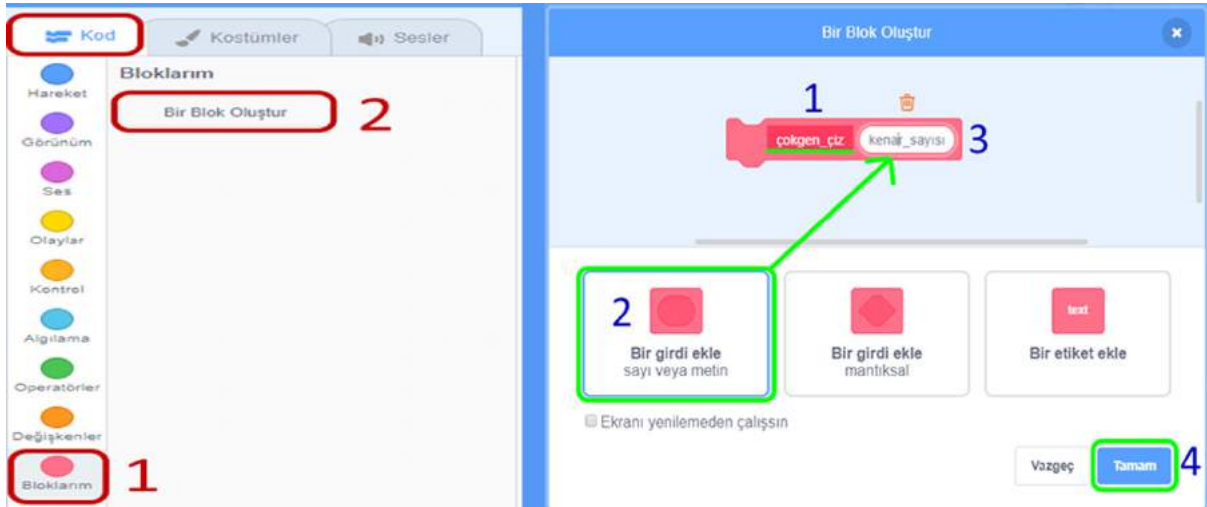
1. Her kuklaya dekor değiştirilmesi için neler yapılmalıdır?

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Döngü kullanma			
İç içe döngü kavramını kullanma			
Eklenti ekleme			
Kostüm çoğaltma			
Kostümlere hareket kazandırma			

4.4.12. Scratchte Fonksiyon Kullanımı

Yazılım geliştirme süreçlerinde bazı kod blokları, tekrarlı bir yapıda aynen veya bazı parametre değişiklikleri ile sıklıkla kullanılmaktadır. Bu kod bloklarının her ihtiyaç duyulduğunda tekrar yazılması, etkili ve işlevsel bir yöntem olmamaktadır. Bu nedenle tekrarlayan bu kod blokları, fonksiyon olarak tanımlanmakta ve yazılım içinde ihtiyaç duyulan her noktada çağrılmaktadır. Bu şekilde fonksiyon olarak tanımlanan kod blokları, çağırma yöntemi ile uygulama içinde tekrar tekrar kullanılabilir. Ayrıca fonksiyon içerisindeki kodlarda bir değişikliğe gidilmesi gerektiğinde yapılan bu değişiklik, uygulamada fonksiyonun kullanıldığı tüm noktalara etki etmektedir.

Fonksiyon kavramı Scratch'te **bloğum** olarak ifade edilmektedir. Scratch uygulamalarına blok eklemek için kategoriler altında bulunan **bloklarım** alanı kullanılmaktadır. Bloklarım alanında bulunan **bir blok oluştur** butonuna tıklanarak blok oluşturma ekranı açılmaktadır. Bu ekranda fonksiyona bir isim verilerek girdiler ve etiketler eklenebilir. Örneğimizde **çokgen_çiz** isimli fonksiyon tanımlanmış ve **girdi ekle** düğmesi kullanılarak **kenar_sayısı** isimli girdi parametresi eklenmiştir. İlgili fonksiyon tanımlama ve girdi ekleme süreci aşağıdaki resimde numaralandırılarak gösterilmiştir.



Şekil 4.137. Blok oluşturma

15. ETKİNLİK: ORKESTRANI KUR

BİLİŞSEL DÜZEY

Uygulama

KAZANIM(LAR)

Bir problemi alt bileşenlere böler.

Bir problemin alt bileşenlerini sıralı olarak çözer.

Değişken tanımlar ve kullanır.

Koşul ifadesi tanımlar ve kullanır.

Döngü yapılarını kullanır.

Blok (fonksiyon) yapılarını kullanır.

İkiz oluşturur ve kullanır.

YÖNTEM/TEKNİKLER

Gösterip yaptıрма, soru cevap, iş birlikli öğrenme

ÖĞRETİM MATERYALLERİ

Bilgisayar, internet bağlantısı, web tarayıcı veya Scratch 3 yazılımı

GİRİŞ

Öğrencilere “Scratch ile müzik uygulaması yapalım mı?” diyerek derse giriş yapılır.

SÜREÇ

- Belirli görevleri bir tanım içerisinde verilebilmek için fonksiyonlar kullanılır. Bu etkinlikte fonksiyonları daha iyi anlayabilmek için müzikte tekrar eden notalar kullanılacaktır. Müzikte notalar, nakarat kısımlarında tekrar etmektedir. Bu kısımları kullanarak fonksiyon oluşturup bilgisayarda kolayca müzik yapabilirsiniz.

Yağ Satarım Bal Satarım melodikaogreniyorum.blogspot.com

Melodica

YAĞ SA TA RIM BAL SA TA RIM US TAM ÖL MÜŞ BEN SA TA RIM
LA LA LA SOL LA LA LA SOL FA SOL FA SOL LA LA LA SOL

Mel.

5 US TAM ÖL DÜ KÜR KÜ VAR SAT MAM ON BEŞ Lİ RA YA
FA SOL FA SOL LA LA SOL FA SOL FA SOL LA LA SOL

Mel.

9 YAĞ SA TA RIM BAL SA TA RIM YAĞ LI CA BAL LI CA DA YAK A TA RIM
LA LA LA SOL LA LA LA SOL FA FA SOL FA FA SOL LA LA LA LA SOL

Şekil 4.138. Notalar

- Notaları La La La Sol La La La Sol Fa Sol Fa Sol olan şarkının kodlamasını yapınız. Notaları kodlamak için Şekil 4.139'daki görselden faydalanabilirsiniz.

Do# Re# Fa# Sol# La#

60 61 62 63 64 65 66 67 68 69 70 71 72

do re mi fa sol la si do

Şekil 4.139. Sayıların nota karşılığı

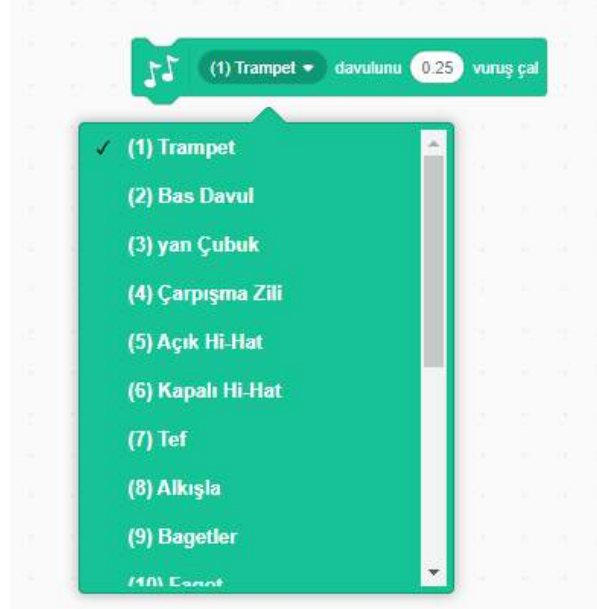
- Scratch eklentiler bölümünden müzik eklentisini ekleyiniz.

BÖLÜM 4



Şekil 4.140. Sprite kuklasına eklenecek kod bloğu

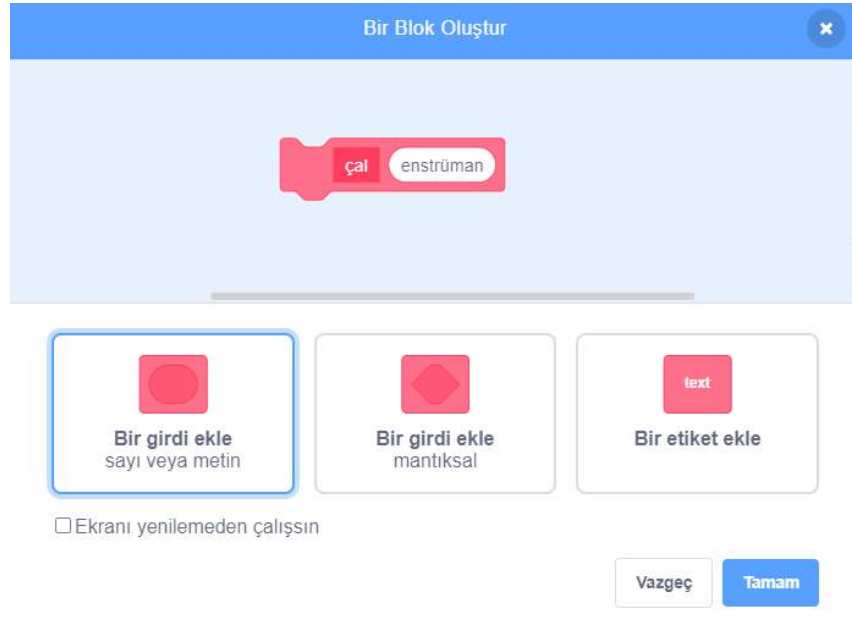
- Şekil 4.140'taki kod bloğunu ismi Sprite olan kedi kuklasının içine müzik eklentisini ekleyerek oluşturunuz.
- Bu durumda farklı enstrümanlar ile aynı notları kullanmak için aynı kodları kopyalamaya gerek kalmadan fonksiyon kullanmak kolaylık sağlayacaktır. Bunun için "Yağ Satarım" isimli çocuk şarkısı bir fonksiyon hâline getirilecektir.



Şekil 4.141. Seçilebilecek enstrümanlar

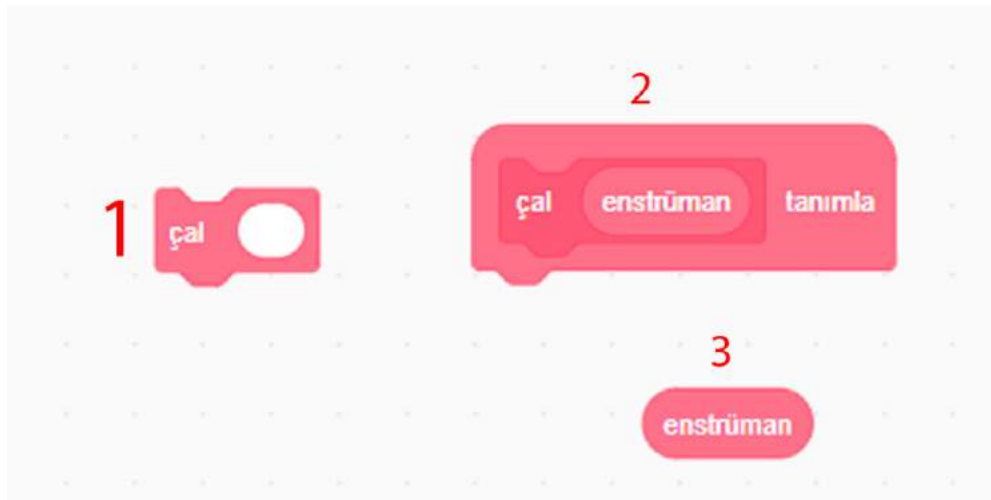
- Farklı enstrümanlar sayılar ile ifade edilmektedir. Örneğin, 19 girilen değerde ksilofon ile Yağ Satarım şarkısı çalmaktadır. 2 değeri ile bas davul ile notalar çalınmaktadır.
- Bu sayı değerlerini farklılaştırarak sırayla farklı enstrümanlardan Yağ Satarım şarkısını dinleyebilirsiniz. Bunu yapmak için kodları kopyalamak yerine fonksiyon oluşturmak gerekir.
- Bloklarım bölümündeki **Bir Blok Oluştur**'a tıklayınız.

BÖLÜM 4



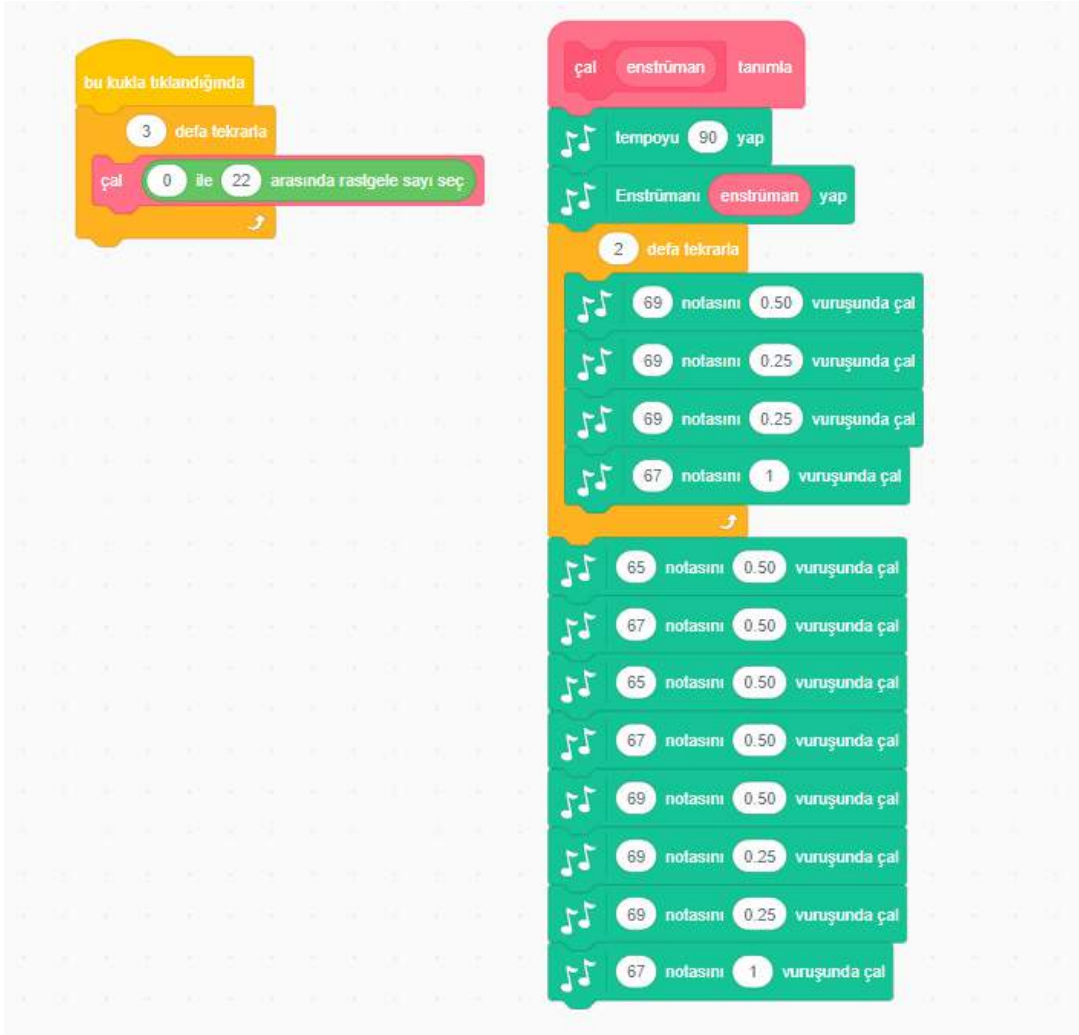
Şekil 4.142. Yeni blok oluşturma

- Bir girdi alıp fonksiyonu çalıştıracığınız için ilk seçeneği seçmeniz gerekir.



Şekil 4.143. Fonksiyon oluşturma

- Şekil 4.143'te 1 ile gösterilen blokta, fonksiyonu çalıştırmak için kullanılacak olan fonksiyon ve sayı değerinin gönderileceği alanı, 2 ile gösterilen blok fonksiyonu, 3 ile gösterilen blok ise 1 ile gösterilen blokta girilen sayı değerinin kullanılacağı kod bulunmaktadır.



Şekil 4.144. Fonksiyon kod blokları

- Fonksiyonlar yerine konulduktan sonra 0 ile 22 arasında en az 3 kere tekrarlanacak bir enstrüman seçimi yapılır. Bu enstrüman seçimi, enstrüman değişkenine aktarılır. Aktarılan değer farklı enstrümanlar ile Yağ Satarım şarkısının çalınmasını sağlar.

BÖLÜM 4

ÖLÇME VE DEĞERLENDİRME

Farklı enstrümanlar ve kuklalar ekleyerek aynı notaları birlikte çalabilecek bir orkestra için kod bloklarını oluşturunuz.

Ölçütler	Geliştirilmeli (1 puan)	İyi (3 puan)	Çok iyi (5 puan)
Yeni blok oluşturma			
Fonksiyon oluşturma			
Eklenti ekleme			
Kuklalara fonksiyon tanımlama			
Müzik notalarını bilme			

BÖLÜM 5

KODLAMA PLATFORMLARI VE EĞİTSEL ROBOTİK KİTLER

Kullanılan kodlama araçlarında planlanan kodların fiziksel olarak kullanılabileceği elektronik platformlar, fiziksel kodlama ve eğitsel robot setleri mevcuttur. Bu setler aracılığı ile bilgisayarda yapılan bazı uygulamaların fiziksel olarak karşılıklarını, kullanıcıların bu platformlar aracılığı ile görmesi mümkün olmaktadır.

5.1. Arduino

Arduino, kullanımı kolay donanım ve yazılıma dayanan açık kaynaklı bir elektronik platformdur. Arduino kartları, elektronik devre kontrolünü kolay bir şekilde kodlamayı sağlayabilmektedir. Arduino kartları, robotik projelerinden akıllı ev sistemlerine kadar birçok alanda yer alır. Arduino kartını programlamak için Arduino programlama dili (Wiring diline dayalı) ve Arduino Yazılımı (IDE) kullanılır (Arduino, 2020).



Şekil 5.1. Arduino uno 3 Kaynak: <https://store.arduino.cc/usa/arduino-uno-rev3>

BÖLÜM 5

Arduino kartları, elektronik ve programlama bilgisi olmayan öğrencilere yönelik hızlı prototipleme için kolay bir araç olarak tasarlanmıştır. Arduino'nun basit 8 bit elektronik kartlar, IoT (nesnelerin interneti) uygulama kartları, giyilebilir teknoloji, 3D baskı ve gömülü sistemler gibi ürünleri bulunabilir. Tüm Arduino kartları açık kaynak kodludur ve kullanıcıların bağımsız olarak kullanmalarına, kendi ihtiyaçlarına göre uyarlamalarına olanak verir. Arduino yazılımı da açık kaynak kodludur. Açık kaynak olması nedeniyle birçok topluluk ve platform tarafından desteklenen Arduino, çok zengin bir kaynak ve kod desteğine sahiptir. Arduino UNO (Şekil 5.1), elektronik ve kodlamaya başlamak için en sık kullanılan kartlardan biri olarak önerilebilir.

Arduino ile uyumlu geniş ürün yelpazesi bulunmaktadır. Bir Arduino projesi yapılmak istendiğinde gerekli elektronik modüllere ve cihazlara kolay bir şekilde erişmek mümkündür. Arduino kartları kodlamak için çok sayıda blok kodlama platformu da (Scratch for Arduino, mBlock, ArduBlock, ArduinoBlocks vb.) bulunmaktadır. Arduino kartları, elektronik ve kodlama kolaylıkları (özellikle blok kodlama) ve geliştirme olanaklarının zenginliği sayesinde farklı eğitim kademelerinde yaygın bir şekilde kullanılabilir.

5.2. Raspberry Pi

Raspberry Pi, bilgisayar monitörüne veya televizyona takılabilen, standart bir klavye ve fare kullanan, düşük maliyetli, kredi kartı boyutunda bir bilgisayar olarak tanımlanabilir. Bilgisayarları keşfetmek ve özellikle Scratch, Python gibi dillerde programlamayı öğrenmek için tasarlanmış küçük bir bilgisayardır. Bir masaüstü bilgisayardan beklenen işlevlerin hemen hepsini yerine getirebilir (Raspberry Pi, 2020).

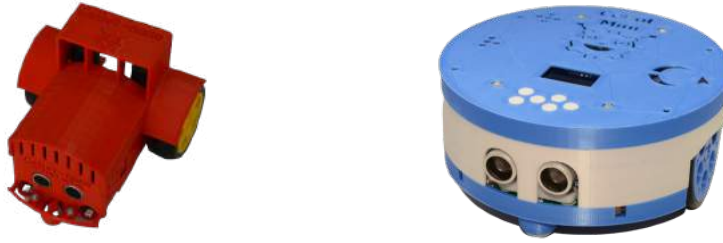


Şekil 5.2. Raspberry Pi 4 B model **Kaynak:** <https://www.raspberrypi.org/products/>

Raspberry Pi kullanılarak müzik kutularından hava ölçüm istasyonlarına, robotlara, otonom cihazlara ve yüz tanıma sistemlerine kadar çeşitli projeler yapılabilir. Raspberry Pi ile uyumlu çok sayıda elektronik modül, cihaz ve elektronik devre elemanı sayesinde gelişmiş projeler yapılabilir.

5.3. CeBot

CeBot, Arduino tabanlı programlanabilir bir robot eğitim aracıdır. CeBot, donanım ve yazılım yapısı ile okul öncesi eğitimden yükseköğretime kadar geniş bir alanda kullanılabilir. CeBot projesi Cezeri Yeşil Teknoloji Mesleki Teknik Anadolu Lisesi tarafından açık kaynak donanım ve açık kaynak yazılıma dayalı olarak geliştirilmiştir. CeBot eğitim seti, birçok sensörü ve robotik eklentisini desteklemektedir. CeBot ve CeBot mini olmak üzere iki robot eğitim seti mevcuttur.



Şekil 5.3. CeBot ve CeBot Mini

CeBot gövdesi, 3 boyutlu yazıcı ile üretilebilir. CeBot eğitim setleri ile engelden kaçan, çizgi izleyen, ışık izleyen, ışıktan kaçan, masadan düşmeyen, el hareketleri, buton, uzaktan kumanda ve bluetooth ile kontrol edilebilen robot uygulamaları yapılabilir. CeBot "mBlock" ve "Arduino IDE" ile programlanabilir. Bu sayede hem blok tabanlı hem de metin tabanlı kodlama olanağı vermektedir (MTEDOSE, 2020).

5.4. LEGO

LEGO; robotik çalışmaları için mekanik, elektronik ve programlanabilir yapılar hâlinde robot setleri sunmaktadır. Birleştirilebilir parçalar, öğrencilerin kendi tasarımlarını yapmalarına olanak vermektedir. Öğrenciler yaptıkları robotları veya diğer tasarımları Lego blok kodlama aracı kullanarak programlayabilir (Lego, 2020).



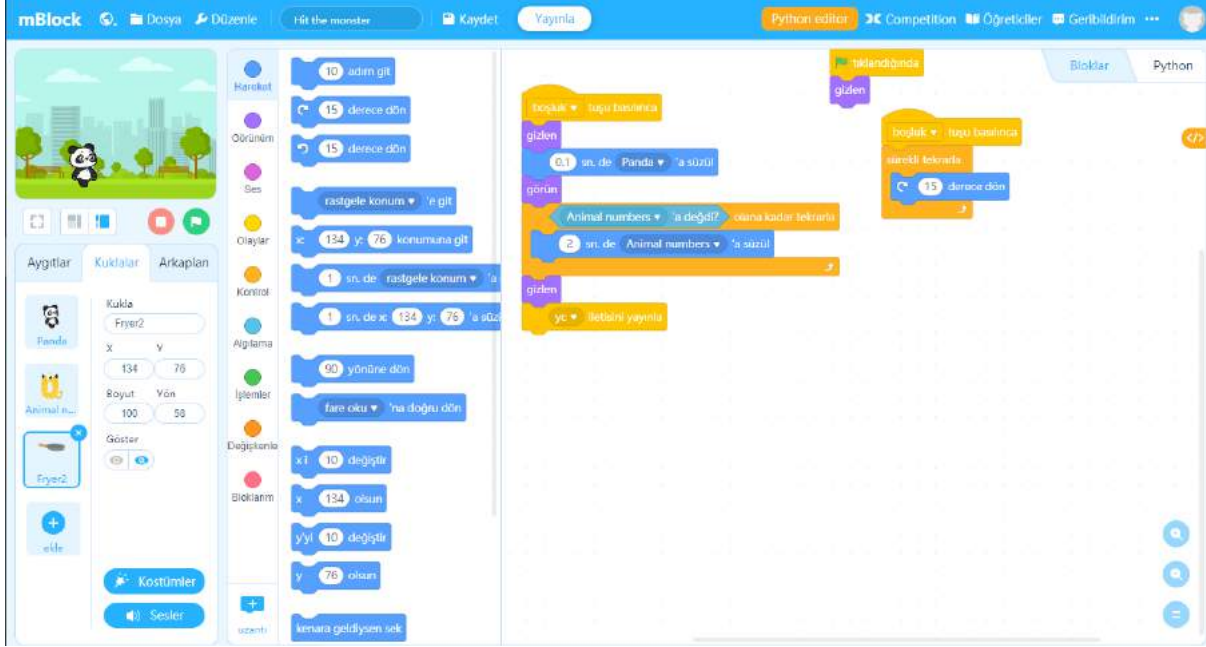
Şekil 5.4. Lego Mindstroms **Kaynak:** <https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313>

Şekil 5.4'te Lego setlerinden biri kullanılarak yapılan bir robot görülmektedir. Kodlama yapmak için tablet veya bilgisayar kullanılabilir. <https://education.lego.com/en-us/lessons> eğitim adresinde ders planları ve eğitimler yer alır.

5.5. Mblock

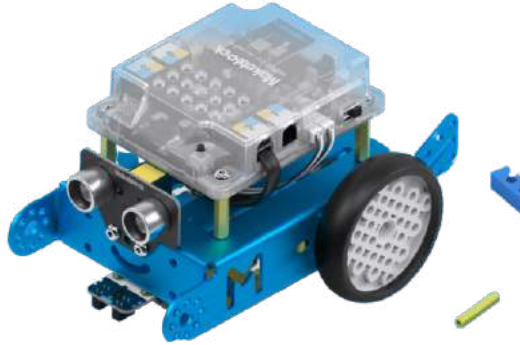
MBlock, çocuklara yönelik olarak tasarlanmış (ama geniş bir kitle tarafından kullanılan) bir STEAM (bilim, teknoloji, mühendislik, sanat ve matematik) programlama yazılımı aracıdır. Mblock, Scratch ve Arduino kodlarına göre geliştirilmiştir. Blok tabanlı ve metin tabanlı programlama desteği vermektedir.

MBlock ile blok kodlama veya Python kodu kullanılarak Scratch'teki gibi oyun ve animasyonlar oluşturulabilir. Mblock sayesinde yapay zekâ ve IoT gibi teknolojiler kullanarak projeler de oluşturulabilir. Çevrim içi topluluklar tarafından çok sayıda mBlock projesi, kaynak ve kod paylaşılmaktadır.



Şekil 5.5. Çevrim içi mBlock kodlama sayfası **Kaynak:** <https://ide.mblock.cc/#/>

MBlock ile aynı zamanda robotlar ve elektronik kartlar da programlanabilir. MBlock'un başta Makeblock firmasının ürettiği kartlar ve robotik kitleri olmak üzere çeşitli fiziksel kodlama modülleri bulunmaktadır. Şekil 5.6'da görülen robot, mBlock ile blok tabanlı kodlama kullanılarak programlanabilir. Özellikle Arduino programlamak için de kolay modüller sunan mBlock, farklı eğitim kademelerinde kodlama ve robotik çalışmalarında kullanılabilir (mBlock, 2020).

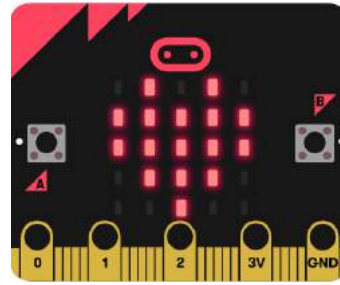


Şekil 5.6. MBlock eğitsel robot **Kaynak:** <https://www.makeblock.com/mbot>

BÖLÜM 5

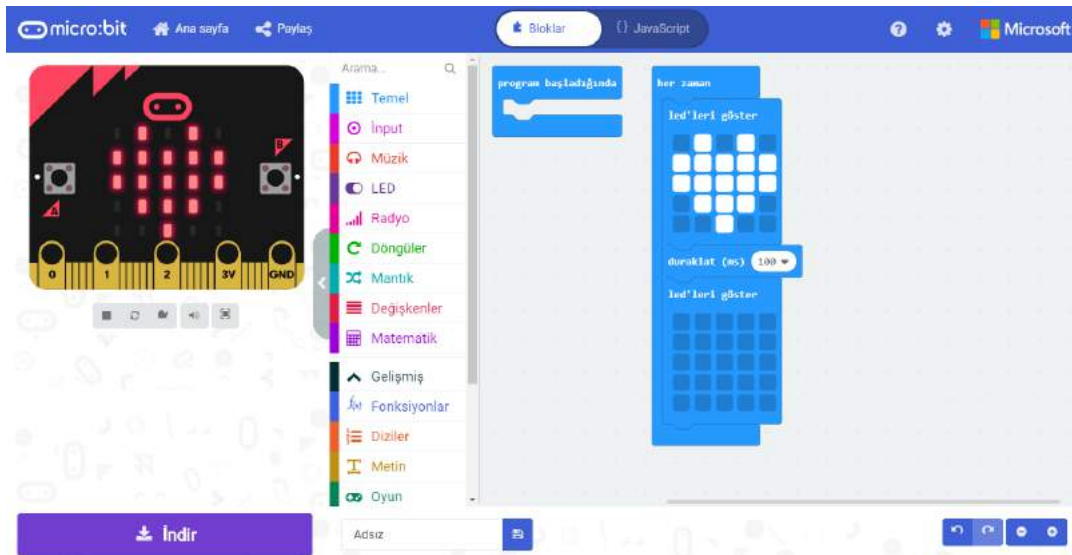
5.6. Micro:bit

Micro: bit, kodlanabilir elektronik bir karttır. Çevrim içi kodlama desteği kullanılarak bir program oluşturulabilir ve bilgisayara “.hex” dosyası olarak kaydedildikten sonra micro: bit’e yüklenerek çalıştırılabilir. Elektronik özellikleri ve blok kodlama desteği ile öğrencilerin fiziksel kodlama, IoT ve robotik eğitimlerinde kullanılabilir (micro:bit, 2020).



Şekil 5.7. micro:bit kartı **Kaynak:** <https://microbit.org/projects/make-it-code-it/emotion-badge/>

Micro:bit kodlamak için kullanılacak çevrim içi editör (<https://makecode.microbit.org/#editor>) aynı zamanda bir simülatöre sahiptir. Editörü kullanarak fiziksel olarak micro:bit olmasa da kodların çıktılarını simülatörde görülebilir. Web sayfasında micro:bit ile yapılan projeler, eğitim videoları, kodlar ve diğer kaynaklar yer almaktadır.



Şekil 5.8. Çevrim içi microbit editörü **Kaynak:** <https://makecode.microbit.org/#editor>

5.7. Vex

Vex'te mekanik düzeneklerin tasarımına olanak veren, elektronik ve robotik modüllerden oluşan eğitsel robotik setler bulunur. Farklı eğitim kademelerine göre farklı robotik setleri sunulmuştur (Vex, 2020).

Vex setleri; modüler yapılar, elektronik ve programlanabilir bir kart ve birleştirilebilir parçalar içerir. Bu setler, robot çalışmalarında mekanik tasarımları kolaylaştırmakta ve programlanabilir robotlar (Şekil 5.9) tasarlamaya olanak vermektedir. Vex, robotik setlerinde kodlama yapmak için blok tabanlı kodlama araçlarını sunmaktadır. Tablet veya bilgisayarda çalışabilen blok kodlama araçları ile tasarlanan robotlar istenilen şekilde programlanabilir. Farklı platformlarda Vex kodlama araçları için geliştirilmiş kodlama araçları ve simülatörler de (<https://www.robotmesh.com/>) yer alır.



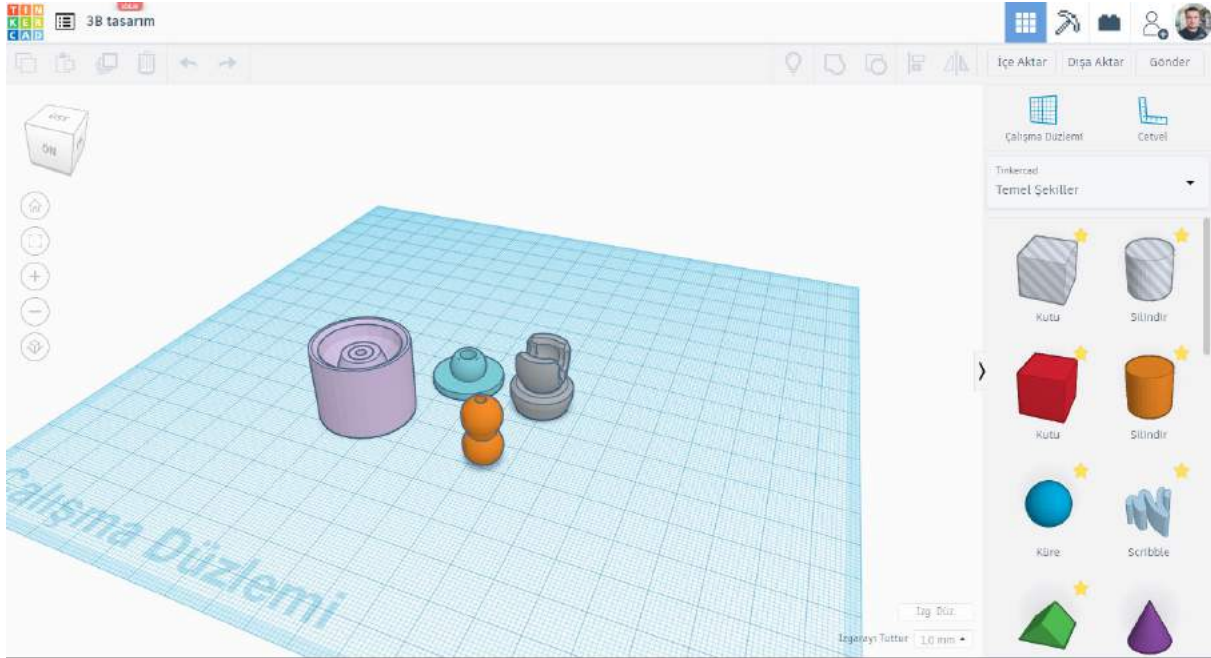
Şekil 5.9. Vex clawbot IQ Kaynak: <https://www.vexrobotics.com/228-3193.html>

5.8. Tinkercad

Robotik ve kodlama eğitimlerinde, 3 boyutlu tasarım, devre simülasyonu, Arduino programlama ve simülasyonu gibi işlevler sunan gelişmiş bir platformdur. Çevrim içi olarak <https://www.tinkercad.com/> adresinden erişilebilir. Ücretsiz üyelik ile platform kullanılabilir. Ayrıca platformda eğitimciler, öğrenciler için sınıfl oluşturabilir, öğrencilerin çalışmalarını takip edebilirler. Öğrenciler aynı proje üzerinde ortak çalışmalar yapabilirler. Tinkercad ile öğretmenler ve öğrenciler Şekil 5.10'daki gibi 3B tasarımlar yapabilirler. Tinkercad hazır 3B şekiller bulundurmakta ama aynı zamanda kullanıcıların

BÖLÜM 5

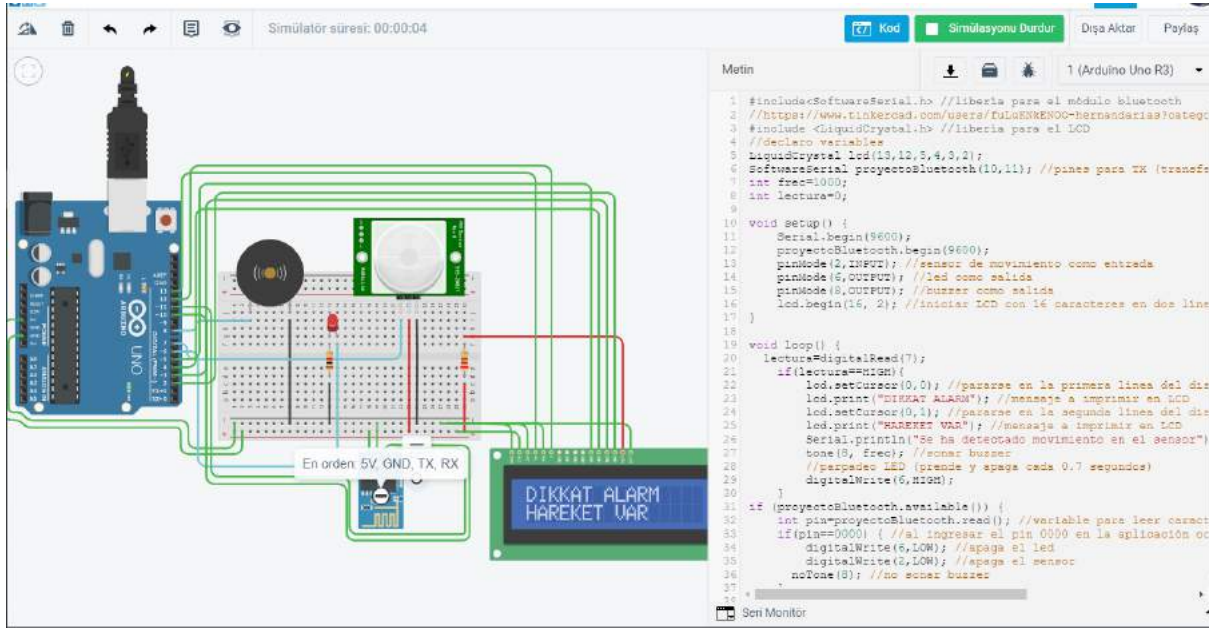
da kendi 3B cisimlerini ve şekillerini oluşturabilecekleri temel yapılar sunmaktadır. Tinkercad içinde programlama yoluyla 3B tasarım yapmak için blok kodlama seçeneği de bulunur. Yapılan 3B şekiller paylaşılabilir, 3B yazıcıdan çıktı almak üzere uygun formatta (stl) kaydedilebilir. Kullanıcılar, yaptıkları projeleri yayınlatabilir ve arama seçenekleri ile yapılan projeleri bularak kullanabilirler.



Şekil 5.10. Tinkercad ile 3B tasarım

Tinkercad'in devre tasarım ve simülasyon programı Şekil 5.11'de görüldüğü üzere karmaşık elektronik devrelerin yapılmasına olanak vermektedir. Tinkercad, devre tasarımında Arduino kullanımını da olanak vermekte ve Arduino kartın programlanabilmesi için blok ve metin tabanlı kodlama araçları sunmaktadır. Simülasyonun çalıştırılması ile yapılan devre ve yazılan kodlar test edilebilir. Özellikle donanım kullanmadan ve elektrikle ilgili riskleri almadan kolay ve maliyetsiz bir devre tasarımı ve kodlama olanağı sunar. Bu kısımda yapılan projeler de paylaşılabilir ve proje havuzunda yer alan projelerde kullanılabilir (Tinkercad, 2020).

BÖLÜM 5



Şekil 5.11. Tinkercad Devre Tasarım, Kodlama ve Simülasyon Arduino, 2020. Arduino Web Sayfası. <https://www.arduino.cc/en/guide/introduction> adresinden 15.05.2020 tarihinde alınmıştır.

KAYNAKÇA

- Vatansever, F., (2009). *Algoritma Geliştirme ve Programlamaya Giriş.*, 7. baskı, (pp. 33-44), Ankara: Seçkin Yayıncılık.
- Futschek, G. (2006). *Algorithmic Thinking: The Key for Understanding Computer Science*. R. T. Mittermeir içinde, *Informatics Education – The Bridge between Using and Understanding Computers* (s. 159-168). Vilnius, Lithuania: Springer.
- Google Education (2017b). *Exploring Computational Thinking*. Retrieved July 19, 2020, from <https://edu.google.com/resources/programs/exploring-computational-thinking/>
- Gülbahar, Y, Kert, S, Kalelioğlu, F . (2019). *Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği: Geçerlik ve Güvenirlik Çalışması*. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 10 (1), 1-29 . DOI: 10.16949/turkbilmat.385097
- Gülbahar, Y., Kalelioğlu, F . (2018). *Bilişim Teknolojileri ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci*. *Milli Eğitim Dergisi*, 47 (217), 5-23 . Retrieved from <https://dergipark.org.tr/tr/pub/milliegitim/issue/39632/462315>
- Gülbahar, Y., (2018). *Bilgi İşlemsel Düşünmeden Programlamaya*. Ankara: Pegem Akademi.
- ISTE (2019), *Computational Thinking Links and Resources*, Retrieved Jan 13, 2019 | [Url](https://www.iste.org/Computational-Thinking-Links-and-Resources) | Exploring Computational Thinking
- <https://docs.google.com/document/d/1WJ1s2TNEX-qGvwaFcua7zpRjelewmFWO00DibEXYypw/edit>
- ISTE (2016). *Operational Definition of Computational Thinking for K–12 Education*. Retrieved April 13, 2020, from <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>
- ISTE (2016). *The ISTE National Educational Technology Standards (NETS•S) and Performance Indicators for Students*. Retrieved April 20, 2020 from <https://www.iste.org/standards/for-students>
- ISTE and CSTA (2011). *Computational Thinking Teacher Resources*. Retrieved April 13, 2020 from https://id.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2
- ISTE and CSTA (2011). *Computational Thinking For All*. Retrieved April 13, 2020 from <https://www.iste.org/explore/Solutions/Computational-thinking-for-all>

- Kaleliođlu, F., & Gülbahar, Y. (2015, September). Bilgi işlemsel düşünme nedir ve nasıl öğretilir? Paper presented at the 3. In 3th International Instructional Technology and Teacher Education Symposium. Trabzon, Türkiye.
- LEGO Education (2018). WeDo 2.0 Computational thinking teachers guide. Retrieved April 15, 2020, from <https://le-www-live-s.legocdn.com/sc/media/files/user-guides/wedo-2/computational-thinking-teacher-guides/computationalthinkingteacherguide-en-us-v1-47bac5566f2966a04c1b77caf4c7e9f5.pdf>
- MEB. (2018). Yenilik ve Eğitim Teknolojileri Genel Müdürlüğü 5. ve 6. Sınıf Bilişim Teknolojileri ve Yazılım Dersi Kodlama Kılavuzu. Retrieved April 15, 2020, from <http://www.eba.gov.tr/>
- MEB. (2019). 5. Sınıflar Bilişim Teknolojileri ve Yazılım Dersi Öğretmen Rehberi. Retrieved April 15, 2020, from <http://www.eba.gov.tr/>
- MEB. (2019). 6. Sınıflar Bilişim Teknolojileri ve Yazılım Dersi Öğretmen Rehberi. Retrieved April 15, 2020, from <http://www.eba.gov.tr/>
- MEB. (2019). Bilgisayar Bilimi Kur 1, 1. Bölüm. Retrieved April 15, 2020, from <http://www.eba.gov.tr/>
- ÖYGM, MEB. (2020). İşlemsel Düşünme Becerisinin Disiplinlerarası Yaklaşım İle Öğretimi. Retrieved April 15, 2020, from <http://www.eba.gov.tr/>
- SERT, S. B., (2017). Bilgisayar Bilimi Eğitime Giriş. Bilgi İşlemsel Düşünmeden Programlamaya (pp.1-22), Ankara: Pegem Akademi.
- Üzümcü, Ö., (2019), Bilgi İşlemsel Düşünme Becerisine Yönelik Program Tasarımının Geliştirilmesi ve Etkililiğinin Değerlendirilmesi. Yayınlanmış doktora tezi, Gaziantep Üniversitesi, Eğitim Bilimleri Enstitüsü.
- Wing, J. M. (2011). Research Notebook: Computational Thinking—What and Why? The Link Magazine, Spring 2011. Carnegie Mellon University, Pittsburgh. Retrieved April 13, 2020 from <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>
- Wing, J. M., (2016, March) Computational Thinking <http://www.cs.cmu.edu/~wing/publications/Wing06.pdf>

ROBOTİK VE KODLAMA EĞİTİMİ

KODLAMA EĞİTİMİ TEMEL DÜZEY

ROBOTİK

KODLAMA

