

SCRATCH İLE PROGRAMLAMA

1. BÖLÜM: GİRİŞ VE TEMEL İFADELER

PROGRAMLAMA NEDİR?

Bilgisayarların isteğe uygun özel işlemler yapması için programlanması gerekir. Örneğin, bir şirkette kullanılan stok uygulaması, sipariş uygulaması ya da değişik iş takipleri, hastane otomasyonları ya da eğitim kurumlarının kullandığı öğrenci otomasyonları gibi. Programlamaya çok fazla örnek vermek mümkündür.

Programlama, yaptırmak istediklerimizi bilgisayarın anlayabileceği şekilde komutlar vererek söylemektir.

ALGORİTMA

Belirli bir problemi çözmek ve belli bir sonuca ulaşmak için çizilen yola algoritma denir. Algoritma, bilgisayarın adım adım ne yapması gerektiğini söyleyen bir taslaktır. Her bir yazılım oluşturulmadan önce algoritma hazırlanır. Daha sonra bu algorithmadaki adımlara göre kodlar yazılarak, program oluşturulur.

Bir Algoritmanın;

1. Başlı olmalı
2. Basit olmalı
3. Problemin çözümünü mümkün olan en az adımla, en kısa sürede gerçekleştirmeli
4. Sonu olmalı

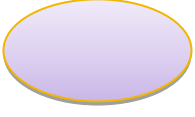
Algoritmalar günlük hayatımıza da uygulanabilirler.

Diş fırçalama algoritması	Ayran yapma Algoritması
1. Başla 2. Diş macununu ve fırçayı yerinden al 3. Diş macununu fırçaya sür 4. Dişlerini fırçala 5. Temizlen 6. Fırçayı temizle 7. Macunu ve fırçayı yerine bırak 8. Bitir	1. Başla 2. Yoğurdu kaba koy 3. Su ekle 4. Çırp 5. Bardağa doldur 6. Bitir

Soru: Tv izleme algoritmasını adım adım yazınız?

AKIŞ DİYAGRAMI (ŞEMASI)

Algoritmanın şekillerle gösterilmesidir. Akış Şeması hazırlanırken bazı şekiller kullanılır.



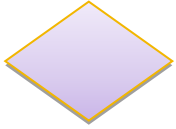
ELİPS: Akış diyagramının başlangıç ve bitiş yerlerini gösterir.



PARALEL KENAR: Klavyeden yapılacak girişler için kullanılır.



DİKDÖRTGEN: Her türlü hesaplama ve atama işlemleri için kullanılır.



EŞKENAR DÖRTGEN: Karar verme ya da karşılaştırma işlemini temsil eder.



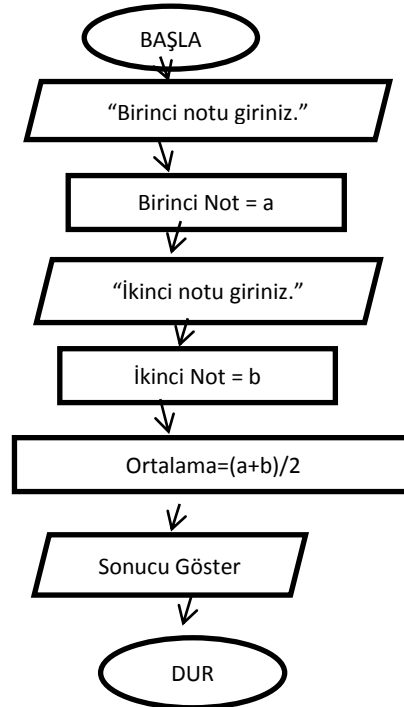
OK İŞARETLERİ: Diyagramın akış yönünü, yani herhangi bir adımdaki işlem tamamlandıktan sonra hangi adıma gidileceğini gösterir.

Örnekler:

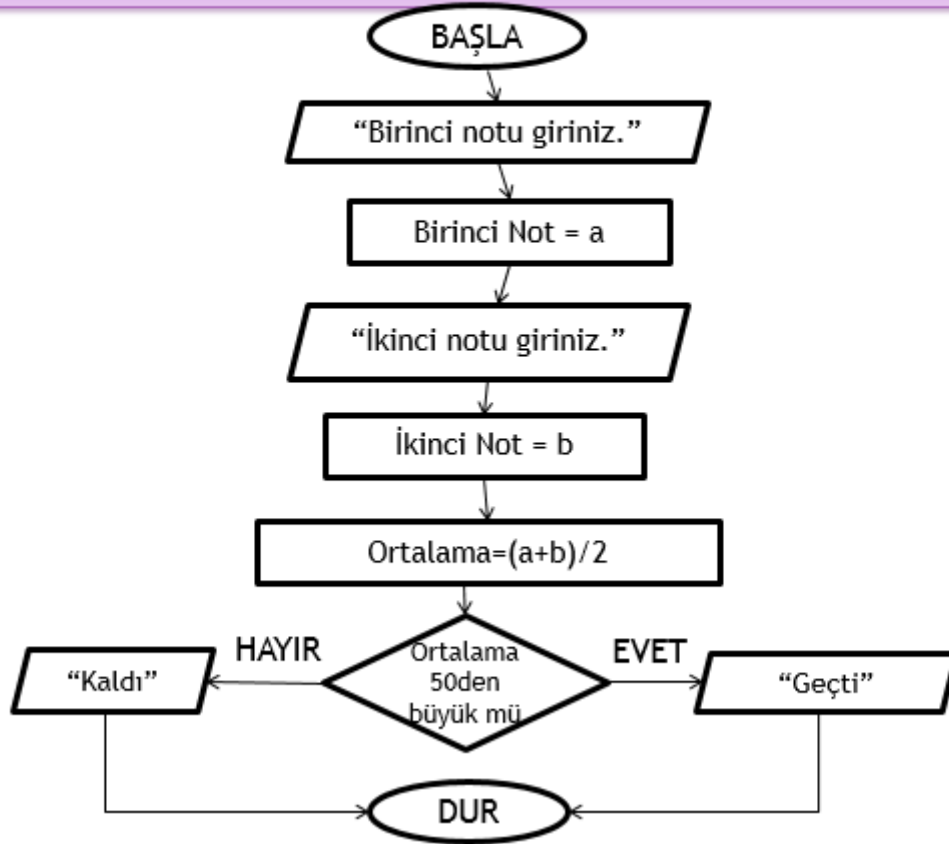
İki Sayıyı Toplayıp Sonucu Ekrana Yazdıran Program

ALGORİTMA	AKIŞ DİYAGRAMI
ADIM 1: Başla. ADIM 2: İlk Sayıyı Gir ADIM 3: İlk Sayı = a ADIM 4: İkinci sayıyı Gir ADIM 5: İkinci Sayı = b ADIM 6: Bu iki sayıyı topla. $a+b$ ADIM 7: Sonucu Göster. ADIM 8: Bitir.	<pre>graph TD; A([BAŞLA]) --> B[/“Birinci sayıyı giriniz”/]; B --> C[a = sayı 1]; C --> D[/“İkinci sayıyı giriniz”/]; D --> E[b = sayı 2]; E --> F[Sonuc= a+b]; F --> G[/Sonucu Göster/]; G --> H([DUR]);</pre>

Bir öğrencinin iki yazılı notu klavyeden girilecektir. Bu notların aritmetik ortalamasını hesaplayan bir programın akış diyagramı:



Bir öğrencinin iki yazılı notu klavyeden girilecektir. Bu notların ortalamasını hesaplayıp, ortalama 50'den büyükse ekrana "Geçti", küçükse ekrana "Kaldı" yazdıran bir programın algoritmasını yapalım.



ÇALIŞMA SORULARI

- 1) Klavyeden girilen 3 sayının toplamını bulup sonucu yazan programın algoritmasını ve akış diyagramını yapınız.
- 2) Klavyeden girilen bir sayının karesini(sayının kendisiyle çarpımını) bulup sonucu yazan programın algoritmasını ve akış diyagramını yapınız.
- 3) Klavyeden girilen iki sayıdan büyük olanı bulup yazan algoritmasını ve akış diyagramını yapınız.

BASLICA PROGRAMLAMA DİLLERİ

- **PHP:** internet tabanlı bir dildir. Web sayfaları ve web uygulamaları geliştirmek için kullanılır. Bu dilde geliştirilen programlar internet üzerinden kullanılır.
- **Visual Basic:** Windows yüklü bilgisayarlarda kullanılır. PC tabanlıdır. Bu dilde geliştirilen programlar bilgisayara yüklenerek kullanılır. 10 sene önce yoğun olarak kullanılırdı.
- **C# :** PC tabanlıdır. Dot net teknolojisi sayesinde yoğun olarak kullanılır. Bu dilde geliştirilen programlar bilgisayara yüklenerek kullanılır.
- **Scratch:** Programlama öğrenmek için geliştirilmiştir. Basit ve görseldir.

Java Programlama Dili Kod Örneği	C# Programlama Dili Kod Örneği
<pre>for (var count = 0; count < 5; count++) { moveForward(); }</pre>	<pre>public static List<string> uyeler = new List<string>(); [System.Web.Services.WebMethodAttribute(), System.Web Services.ScriptMethodAttribute()]</pre>

2. BÖLÜM: CODE.ORG İLE PROGRAMLAMA

GİRİŞ

[Studio.code.org](https://studio.code.org) adresinde 8 farklı eğitim mevcuttur. Burada *Klasik Labirent* eğitimi anlatılacaktır. Diğer eğitimleri kendinizi geliştirmek için kullanabilirsiniz.

1. Ders: Yaramaz domuzu yakalamama yardım edebilir misin? Oraya gitmem için birkaç "İlerle" bloğunu birleştirmeli ve "Programı Çalıştır" tuşuna basmalısın.

bloklar

Bloklarını burda topla: 2 / 3

ilerle

sola dön

sağa dön

Çalıştır

Çalıştığı zaman

ilerle

Yaramaz domuzu yakalamama yardım edebilir misin? Oraya gitmem için birkaç "İlerle" bloğunu birleştirmeli ve "Programı Çalıştır" tuşuna basmalısın.

3. Ders: Yolu takip et ve beni sersem domuza götür. Patlayıcılardan uzak dur yoksa bu tüylerim havaya uçacak!

bloklar

Çalışma alanı: 2 / 5 bloklar

ilerle

sola dön

sağa dön

Çalıştığı zaman

ilerle

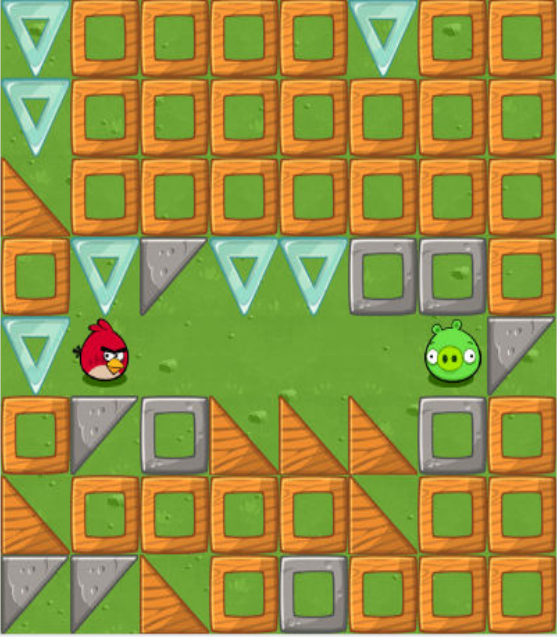
Yolu takip et ve beni sersem domuza götür. Patlayıcılardan uzak dur yoksa bu tüylerim havaya uçacak!

DÖNGÜLER

Program yazarken bazen aynı komutu birden fazla kez vermek gerekebilir. İşte bu gibi durumlarda döngüler kullanılır. Döngüleri kullanarak, bir bilgisayara herhangi bir komut dizisini defalarca tekrar etmesini söyleyebilirsiniz, hatta binlerce ve milyarlarca kez.

Döngülerin içine birden fazla komut eklenebilir. Hatta başka bir döngü bile eklenebilir.

6. Ders: 2 blok(komut) kullanarak yeşil domuzcuğa ulaşabileceğim bir yol var. Bunu çözebilir misin?



bloklar

Blokları burada top

ilerle

sola dön ↺

sağa dön ↻

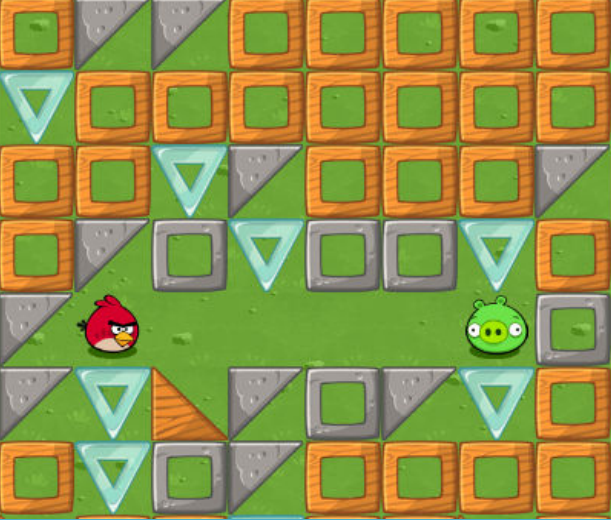
Tekrarla 5 kez tekrarlar yap

Çalıştığı zaman

Tekrarla 5 kez tekrarlar yap

Olana Kadar Tekrarla: Bazen döngümüzü belli bir sayıda değil bir durum gerçekleşinceye kadar tekrarlatmamız gerekebilir. Mesela kapıya kadar kaç adım atacağımızı bilmiyoruz bu durumda kapıya varana kadar ilerle diyebiliriz.

10. Basamak: Tamam, yeni "kadar tekrarlar" bloğunu dene. Bu, ben yeşil domuza ulaşana kadar tekrarlayacak.



bloklar

Blokları burada bu

ilerle

sola dön ↺

sağa dön ↻

kadar tekrarlar  yap

Çalıştığı zaman

EĞER KOMUTU

Bilgisayarlar bizler gibi bazen karar almak zorunda kalabilirler. Eğer komutu da karar almak için kullanılır. Eğer komutu içinde bir şartın gerçekleşme ihtimali vardır.

Örnekler:

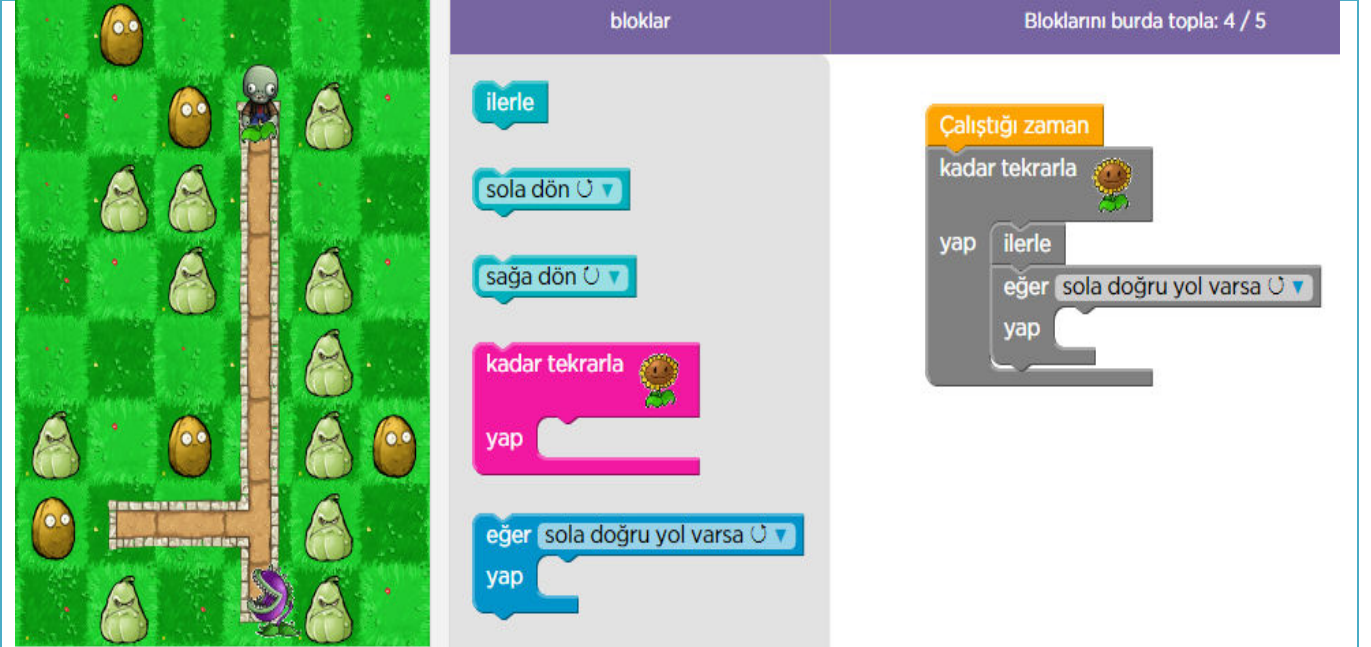
EĞER ÇALIŞIRSAM BÜTÜN DERSLERDEN GEÇEBİLİRİM

YAĞMUR YAĞARSA VE ŞEMSIYEM YOKSA ISLANIRIM

BÜTÜN DERSLERDEN İYİ NOT ALIRSAM TAKDİR BELGESİ ALABİLİRİM

VERİLEN ŞARTA GÖRE OLACAK İŞLEMLER KOŞULLARLA BELİRLENİR.

14. Basamak: Ne zaman döneceğime karar vermem için yeni "eğer" bloğunu kullan. İpucu: Sadece bir bloğa daha ihtiyacın var ama bunu nasıl kurduğumuzu öğren ki bir dahaki sefer kullanabilesin.



The image shows a Scratch interface. On the left is a game scene with a character on a path, surrounded by various obstacles. On the right is the code editor. The code starts with a 'when green flag clicked' event block, followed by a 'say' block, a 'repeat' block, and an 'if' block. The 'if' block contains a 'say' block and a 'repeat' block. The 'repeat' block contains a 'say' block and a 'repeat' block. The 'if' block is labeled 'eğer sola doğru yol varsa' (if the way to the left is correct).

Eğer – Değilse Komutu: Eğer yapısında koşulumuz gerçekleşirse verdiğimiz komutlar yapıliyordu. Eğer/değilse yapısında ise koşul gerçekleştiğinde yapılacak komutlar ve gerçekleşmediğinde yapılacak komutlar vardır. Örneğin ödevimiz varsa ödevimizi yapalım yoksa oyun oynayabiliriz.

18. Basamak: "Eğer-Değilse" bloğu koşulu kontrol eder ve birini ya da diğerini yapar. Meşe palamuduna ulaşmam için bu yeni bloğu kullanın.



Studio.code.org sitesinde bulunan diğer eğitimleri de alarak programlamayı daha iyi kavrayabilirsiniz.



Star Wars

Droidleri programlamayı ve çok çok uzaklardaki bir galakside kendi Yıldız Savaşları oyununu yaratmayı öğren.



Minecraft

Bir Minecraft dünyasını kodlamayla keşfet



Donmuş

Buzun büyüü ve güzelliğini keşfetmek için Anna ve Elsa'ya katılıp kodlama yapalım.



Klasik Labirent

Temel bilgisayar bilimlerine milyonlar bir şans verdi, sen de dene.



Flappy Kod

10 dakikadan az bir süre içinde kendi oyununu yazmak ister misin? Flappy Kod eğitimimizi deneyin!



Sınırsız oyun la...

Bir hikaye ya da Disney Infinity karakteriyle bir oyun yaratmak için Play Lab'i kullanın.



Oyun Laboratu...

Oyun Laboratuvarı ile hikaye yarat veya oyun yap!



Aktör

Sanatçı ile harika resimler ve tasarımlar yap!

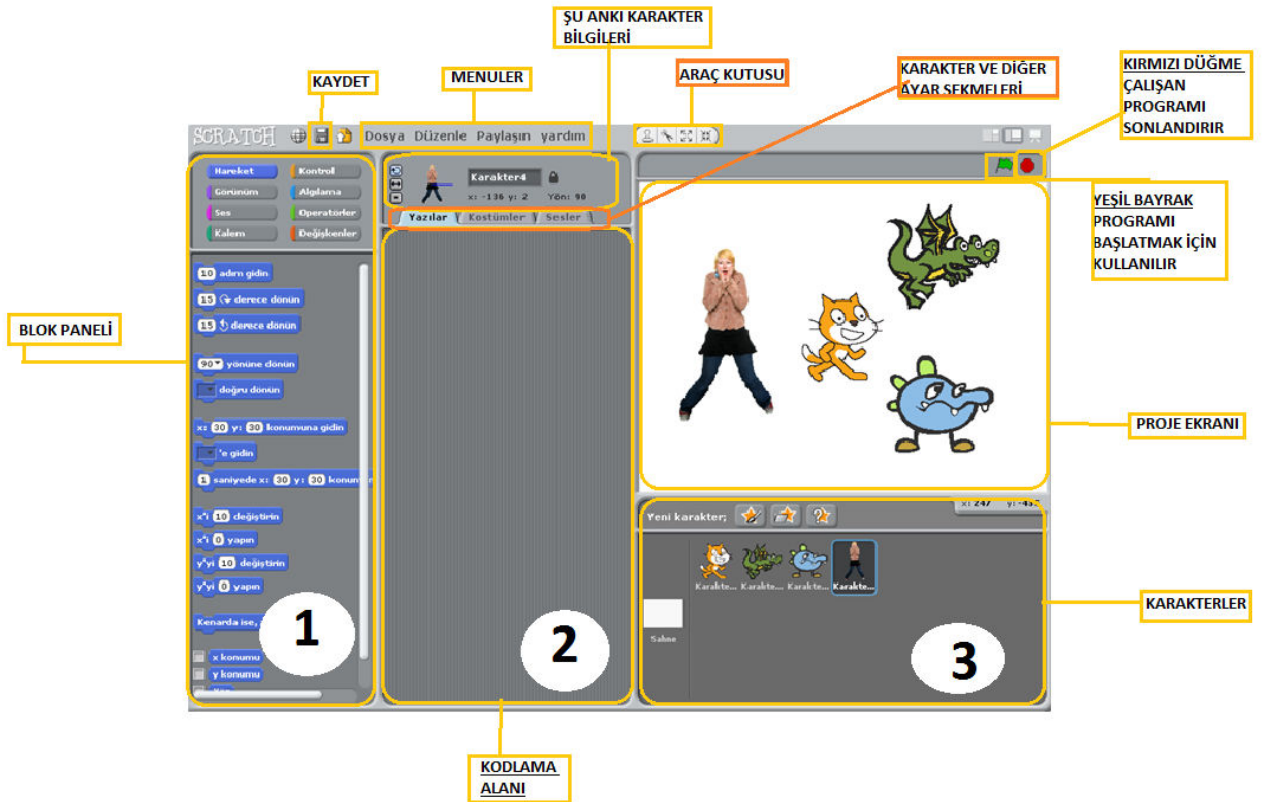
3. BÖLÜM: SCRATCH İLE PROGRAMLAMA

SCRATCH EKRANI

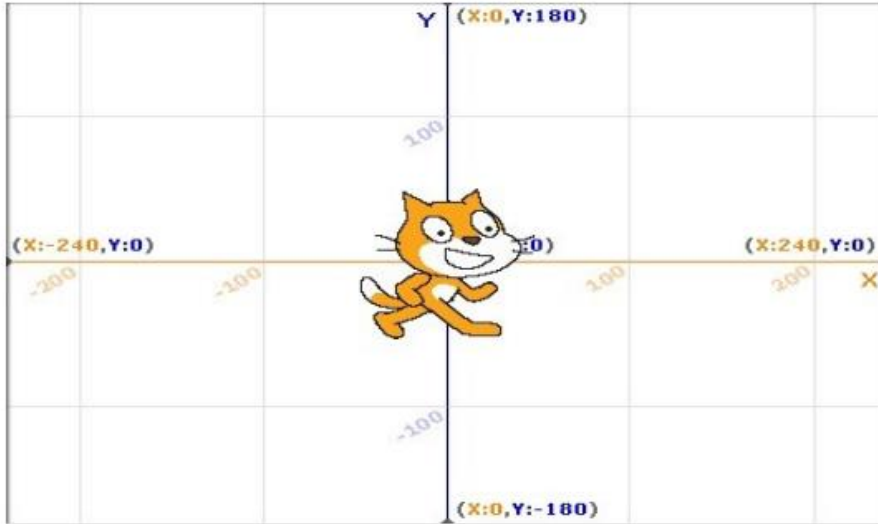
Scratch, kendi interaktif hikâyelerinizi, oyunlarınızı ve animasyonlarınızı kolaylıkla yaratabilmeniz ve bunları web ortamında diğerleriyle paylaşabilmenizi sağlayan yeni bir programlama dilidir.

Scratch'in gelişmiş ve kullanması kolay bir arayüzü vardır. Scratch arayüzü 3 ana bölmeden oluşur.

1. Blok Paletinde karakterleri programlamak için kullanılan bloklar vardır.
2. Kodlama alanı bilgisayar programını yazdığımız yerdir. Buraya blokları çekip bırakarak programımızı yazarız.
3. Proje ekranı bizim sahnemizdir. Tasarladığımız her şey Scratch'in sahnesinde hayat bulur.

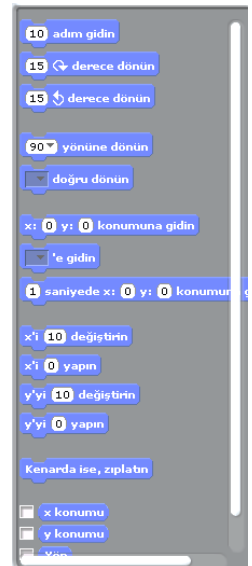


Scratch sahnesi aslında bir koordinat düzlemidir. Bütün animasyonlarımız ve bilgisayar oyunlarımız bu koordinat düzlemi üzerinde tasarlanır. Scratch karakteri olan kedi program açıldığı zaman (0,0) noktasındadır. Ekranın üstünde karakterlerimizin x ve y değerlerini görebiliriz.



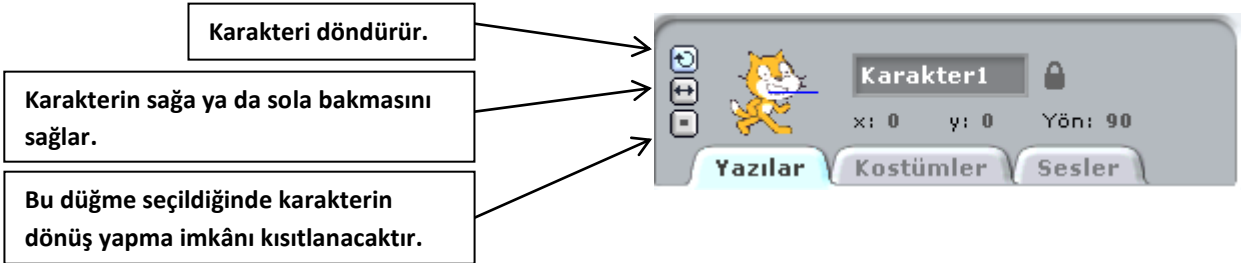
1. Kod Bölümü

Kod bölümü ana ekranın en solunda bulunan bölümdür. Bu bölümün üst kısmında, bu bölümde kullanabileceğimiz kodların grupları yer almaktadır. Genel olarak kodlar, yeteneklerine göre gruplanmışlardır. Aşağıdaki şekilde de görüldüğü gibi 8 farklı grup olarak sunulan bu bölümdeki sekmelere tıkladığınızda, tıkladığınız gruba ait tanımlamalar ekranda belirir.



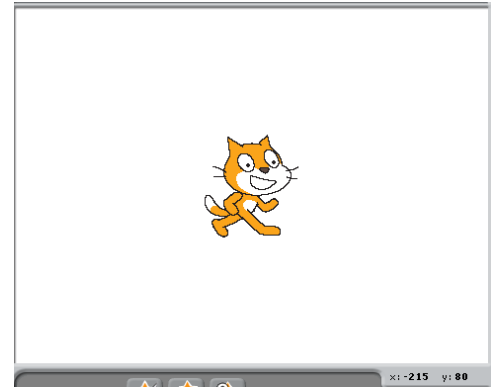
2. Kodlama Paneli

Bu bölüm ana ekranın tam ortasında bulunan kısımdır. Temel olarak bu bölümde yine iki ayrı pencereden oluşmuştur. Üst pencerede bulunan bilgiler, kullandığınız karakterin ismi, yeri, yönü ve karakterin dönebilme yeteneğini bildiren bilgileri barındırır.



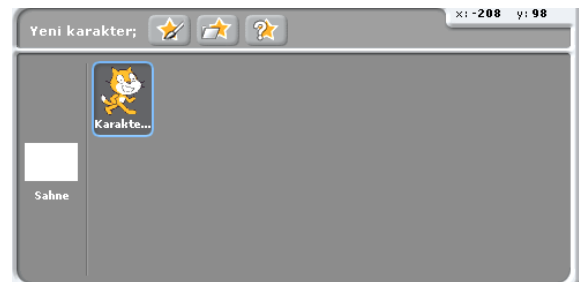
3. Sahne ve Karakterler

Ana ekranın sağ üst tarafında bulunan bölüm, sahne olarak adlandırılır. Karakterlerinizin ve olaylarınızın gerçekleşeceği yer burasıdır. Sahne x ye koordinatlarıyla kontrol edilir. İşaretçinizi sahne üzerinde gezdirdiğinizde, sahne bölümünün sağ alt köşesinde x ve ye koordinatlarının değiştiğini görebilirsiniz.



Sahnenin sağ üst köşesindeki yeşil bayrak programı çalıştırmanızı sağlar. Kod bloğunda yazmış olduğunuz komutlar çalışmaya başlar. Yine sahnenin sağ üst köşesinde bulunan kırmızı yuvarlak ise çalıştırmış olduğunuz bir programı durdurmanızı sağlar.

Sahnenin altında yer alan karakterler paneli hazırladığımız ya da hali hazırda Scratch ile beraber gelen karakterlerin bulunduğu kütüphaneden seçtiğimiz, karakterlerin bulunduğu paneldir.



Karakterler panelinde sahneye tıkladığımızda açılan kod paneli sahne (arkaplan) a ait düzenlemeler yaptığımız bölümdür.

DEĞİŞKEN KAVRAMI

Değişken; verilerin saklanması için gerekli olan alanlardır.

Başka bir anlatımla değişkeni bir kutu olarak düşünebiliriz. Kutu içine eşyalar koymaya yarar. Gerektiğinde ise kutudaki eşyaları çıkartırız.

Programlama dillerinde değişkenler bir veya daha fazla kelimedenden oluşur. Bu kelimelerin yazılmasında belli kurallar vardır.

Değişkenlerin isimlendirilmesi ile ilgili kurallar

1. Değişken bir harfle başlamalıdır. Örneğin; -a4- değişken olabilir ancak -4a- değişken olamaz.
2. Programlama dilinde kullanılan komutların aynısından değişken olamaz.
3. Bir isim, harflerin, sayıların ve altçizgilerin herhangi bir kombinasyonundan oluşabilir. Örneğin; toplanacak_sayi_44 şeklinde bir değişken kullanılabilir.
4. Değişkenleri anlamlı bir şekilde isimlendirmek faydalıdır – değişkenler istendiği kadar uzun olabileceği için, amaçlarını açıklayan değişken isimleri kullanın.

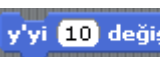
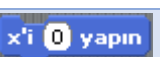
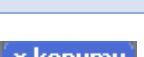
Değişken örnekleri:

Değişken olabilir	Değişken olamaz
sayi	sayı → Türkçe karakter kullanılmaz
pc11	11pc → Rakamla başlayamaz
Toplama_islemi	toplama işlemi → Boşluk olamaz

4. BÖLÜM: SCRATCH KOD BLOKLARI

BLOK AÇIKLAMALARI

Scratch blokları sekiz farklı renkte kategorize edilmiş olup her renk farklı bir kod özelliğini belirtmektedir.

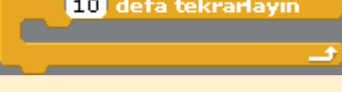
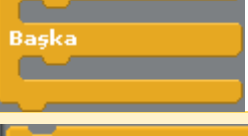
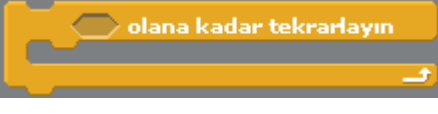
HAREKET BLOKLARI	
	Karakteri öne ya da arkaya doğru hareket ettirir.
	Karakteri saat yönünde döndürür.
	Karakteri saat yönünün tersi yönünde döndürür.
	Karakteri belirtilen yöne döndürür. 0=up, 90=right, 180=down, -90=left
	Karakteri diğer karaktere veya fare imlecine yöneltir.
	Karakteri belli bir x ve y pozisyonuna hareket ettirir.
	Karakteri diğer karaktere veya fare imlecine hareket ettirir.
	Karakteri belli bir sürede belli bir noktaya yavaşça hareket ettirir.
	Belirtilen miktarda karakterin x pozisyonunu değiştirir.
	Karakterin y pozisyonunu belli miktarda değiştirir.
	x pozisyonunu belli değerde olmasını sağlar.
	y pozisyonunun belli değerde olmasını sağlar.
	Ekranın köşesine dokunduğunda karakteri ters yöne döndürür.
	Karakterin x pozisyonu bilgisini verir. (ranges from -240 to 240)
	Karakterin y pozisyonu bilgisini verir. (ranges from -180 to 180)
	Karakterin yönüyle ilgili bilgi verir. 0=up, 90=right, 180=down, -90=left

SES BLOKLARI	
	Müzik çalmaya başlar (menüden seçilmiş), and hemen bir sonraki bloğa gider (müziğin hala çalıyor olmasına rağmen)
	Müziği çalar ve sonraki bloğa geçmek için çalmayı bitirmesini bekler.
	Belirlenmiş bir süre için (saniye) davul sesi çalar. (menüden seçilmiş)
	Belirlenmiş bir süre için (saniye) nota çalar. (yüksek deceler için yüksek değerler)
	Karakterin nota çalma bloklarını kullanabilmesi için çalgı aleti ayarlar.(Her karakterin kendi çalgı aleti bulunur.)

DEĞİŞKEN BLOKLARI	
	Yeni bir değişken yaratıp adlandırmanızı sağlar. Bir değişken yarattığınızda, 3 blok otomatik olarak oluşturulur. (aşağıya bakın). Değişkenin bütün karakterler mi (global) yoksa sadece birisi için mi (local) olduğunu belirtir.
	Değişkenle bağlantılı 3 bloğu da siler
	Belirtilen miktarda değişkenleri değiştirir
	Belirtilen sayıda değişken yerleştirir.
	Değişkenin değerini belirtir.

KALEM BLOKLARI	
	Ekrandan bütün kalem izlerini siler.
	Karakterin kalemini bırakır, böylece, hareket ettiği gibi çizer.
	Karakterin kalemini alır, böylece, hareket ettiği gibi çizmez.
	Renk paletindeki renge göre kalemin rengini belirler.
	Belirtilen miktarda kalemin rengini değiştirir.
	Kalemin rengini belirtilen değer için ayarlar (kalem rengi = 0 renk tablosunun kırmızı sonu, kalem rengi = 100 ise mavi sonudur.)
	Belirtilen miktarda kalemin gölgelemesini değiştirir.
	Kalemin gölgelemesini belirtilen değer için ayarlar (kalem-gölgesi = 0 çok koyu, kalem-gölgesi = 100 is çok açık)
	Kalemin kalınlığını değiştirir.
	Kalemin kalınlığını ayarlar.
	Karakterin imajını Ekranı yapıştırır.

GÖRÜNÜMLER	
	Farklı kostümler değiştirerek karakterin görünümünü değiştirir.
	Karakterin kostümünü kostüm listesindeki bir sonraki kostüme değiştirir. (Eğer liste sonundaysa en başa döner)
	Belirlenmiş bir süre için karakterin konuşma balonunu gösterir.
	Karakterin konuşma balonunu gösterir. (yazı olmadan bu bloğu çalıştırarak konuşma balonunu silebilirsiniz.)
	Belirlenmiş bir zaman için karakterin düşünme balonunu gösterir.
	Karakterin düşünme balonunu gösterir.
	Karakterdeki görsel efekti belli bir süre için değiştirir. (Efekt seçmek için menüyü kullanın.)
	Verilmiş bir sayı için görsel efekt oluşturur. Birçok görsel efekt 0 dan 100 e doğru çeşitlenir.
	Karakter için bütün grafik efektleri temizler.
	Belli miktarda karakterin boyunu değiştirir.
	% olarak belirlenmiş oranda boyunu değiştirir.
	Karakterin büyüklüğünü % olarak bildirir.
	Karakteri ekranda görünür hale getirir.
	Karakteri Ekranda görünmez yapar. Karakter gizlendiğinde diğerleri dokunarak onu bulamaz.
	Karakteri diğerlerinin önüne hareket ettirir.
	Karakterin belli katman miktarınca geri gitmesini sağlar, böylece diğer katmanların arkasına gizlenebilir.

KONTROL BLOKLARI	
	Yeşil bayrağa tıklandığında aşağıdaki kodu çalıştırır.
	Belirtilmiş düğmeye basıldığında aşağıdaki kodu çalıştırır.
	Karaktere tıklandığında aşağıdaki kodu çalıştırır.
	Belirlenmiş süre kadar bekler ve sonraki blokla devam eder.
	İçinde bulunan blokları defalarca çalıştırır.
	İçinde bulunan blokları belirtildiği kadar çalıştırır.
	Bütün karakterlere hareket etmeleri için mesaj gönderir ve sonraki bloğu çalıştırmaya devam edebilmek için bu hareketlerini bitirmelerini bekler
	Bütün karakterlere hareket etmeleri için mesaj gönderir. (bu hareketlerini bitirmelerini beklemez.)
	Özel yayımla mesajı aldığı anda aşağıda bulunan kodları çalıştırır.
	Sürekli olarak koşulun doğru olup olmadığını kontrol eder. Doğru olduğunda içeride bulunan kodu çalıştırır
	Eğer koşul doğruysa içerideki blokları çalıştırır.
	Eğer koşul doğruysa "if" kısmındaki kodu çalıştırır, eğer yanlışsa, "else" kısmındaki kodu çalıştırır.
	Koşul doğru oluncaya kadar bekler, sonra, aşağıdaki blokları çalıştırmaya başlar.
	Koşulun doğru olup olmadığını kontrol eder, eğer doğruysa, içerideki bloğu çalıştırıp koşulu tekrar kontrol eder, eğer yanlışsa, takip eden bloklarla devam eder.
	Kodu durdurur.
	Bütün karakterlerdeki bütün kodları durdurur.

ALGILAMA BLOKLARI	
	Farenin x konumunu belirtir
	Farenin y konumunu belirtir.
	Faredeki düğmeye basıldığını doğrular
	Belirtilmiş düğmeye basıldığını doğrular
	Karakterin belirtilmiş diğer karaktere, Ekranın köşesine veya fare imlecine dokunduğunu doğrular
	Karakterin doğru renge dokunduğunu doğrular. Renk tablosuna tıklayın, sonra, renk seçmek için damlalığı kullanın.
	Karakterdeki rengin arka fona ya da başka karakterdeki renge dokunduğunu doğrular. Renk tablosuna tıklayın, sonra, renk seçmek için damlalığı kullanın.
	Belli bir karaktere ya da fareye olan uzaklığı belirtir.
	Sayacı sıfırlar
	Sayacın değerini saniye olarak belirtir. Not: Sayaç sürekli ilerler.
	Bilgisayar mikrofonu tarafından bulunan sesin miktarını belirtir.
	Bilgisayar mikrofonunun bulduğu sesin miktarı 30'dan fazla olup olmadığını belirtir (tabloda 1-100 arasında)
	Spesifik bir sensörün değerini belirtir. Bu bloğu kullanabilmek için bilgisayarınıza bağlı bir "Scratch Tahtası" olması gerekir. http://scratch.mit.edu/scratchboard adresine bakabilirsiniz.
	Spesifik sensöre basıldığını doğrular. Kullanabilmek için bilgisayarınıza bağlı bir "Scartch Board" olması gerekir

OPERATÖR BLOKLARI		
		İki sayıyı toplar.
		Bir sayıyı diğerinden çıkartır
		İki sayıyı çarpar
		Bir sayıyı diğerine böler.
		Belirtilen değerde rastgele bir tamsayı alır.
		Birinci sayının ikinciden az olup olmadığını belirtir.
		İki sayının birbirine eşit olup olmadığını belirtir.
		Birinci sayının ikinciden büyük olup olmadığını belirtir
		Her iki koşulunda doğru olup olmadığını belirtir.
		Koşullardan birisinin doğru olup olmadığını belirtir
		Koşul yanlışsa bu durumun doğru olduğunu, koşul doğruysa durumun yanlış olduğunu belirtir
		Bir sayının istenen değerini belirtir.
		Birinci sayının ikinciye bölümünden kalanını belirtir
		Bir sayıya en yakın tamsayıyı belirtir.

5. BÖLÜM: SCRATCH UYGULAMALARI

AKVARYUM UYGULAMASI

İlk önce sahne arka planını değiştirelim. Bunun için ekranın sağ alt kısmındaki sahne resmini tıklarız. Böylece sahnenin düzenlendiği kontrol panelleri ve sahne senaryolarımızı yazdığımız kısım açılacaktır.



Daha sonra Arka planlar düğmesini tıklayalım. Açılan bölümden "İçeri Aktar" düğmesine tıklayarak proje için uygun olan arka plan üzerine tıklayarak sahne arka planını değiştirebiliriz.



Şimdi sıra projemiz için uygun olan karakterleri projemiz içine aktarmakta. Bunun için ekranın sağ alt kısmındaki Kedinin üzerine tıklayalım.

Kostümler ve daha sonra İçeri Aktar düğmesine tıklayarak projemizin karakterlerini değiştirebiliriz. İçeri Aktar düğmesine bastıktan sonra projemiz için gerekli olan balık karakterlerini sahnemize ekleyebiliriz. Sahnedeki Kedi karakterini Kostümler bölümünden silebilirsiniz.





Birden fazla karakterle çalışırken karakterlerimize isim vermek işlerimizi kolaylaştırır. Bunun için ekranın üstündeki karakter bilgilerini gösteren bölüme gelerek karakterlerimizin isimlerini değiştirebiliriz.



Şimdi sahnemiz yandakine benzer bir durumdadır. Bundan sonraki adım balıkların ve ahtapotun hareketlerini düzenlemektir.

Karakterlerimizin davranışlarını Kontrol panelindeki bloklarla kontrol ederiz. Bu projede "Yeşil Bayrak tıklandığında" ve "sürekli" bloklarını kullanacağız. Bu blokları çekip senaryo yazdığımız kısma bırakalım isteyin.



Balıkları hareket ettirmek için Hareket panelinden bloklarını yandaki gibi "tıklandığında" ve "sürekli" bloklarının içine çekip bırakalım. 10 adım gidin bloğunun içerisine tıklayarak 10 rakamını 1 yaptım.

Şimdi balığımız sürekli olarak 1 adım gidecek ve sahnenin sonuna geldiğinde geri dönüp hareketine devam edecektir. Scratch ile bir karakteri kontrol etmenin değişik yolları vardır. Yeşil bayrak çok kullanışlıdır. Ama karakterlerimizi daha da interaktif bir hale getirebiliriz. Kontrol Paneline gelerek "balık 1 tıklandığında" blokunu yazı kısmına çekip bırakalım. Daha sonra bu bloğun altına Görünüm panelinden "söyle merhaba süre 2 saniye" blokunu çekip bırakalım. Adım blokunun içindeki rakam gibi bu bloğun içindeki yazıları da değiştirebiliriz ve balığa istediğimiz bir şeyi söyletebiliriz. Diğer balıklar ve ahtapot için senaryo önerileri şunlardır:



Her balığın senaryosunun yazıldığı bölümün ayrı olduğunu unutmayın. Senaryosunu yazmak ya da değiştirmek istediğimiz karakterin yazı bölümüne o karakterin üzerine tıklayarak ulaşabiliriz.



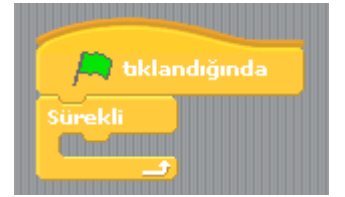
Not: Balıkları sahne kenarında zıplattıktan sonra baş aşağı döndüklerini görebiliriz. Balıkların nasıl döneceklerini karakter bilgilerinin bulunduğu yerden düzenleyebiliriz.

PAPAĞAN UYGULAMASI

1. Yeni karakter ekle butonundan Animals klasöründen papağan "papağan1-a" karakterini sahnemize ekliyoruz.
2. Sahnemizde olan kedi karakterini sağ tıklayıp sahneden siliyoruz.
3. Papağan karakterimize uçma görünümü vermek için papağanımızın 2. bir resmi olan "papağan1-b" yi papağanımız seçili iken kostüm bölümünde "içeri aktar" butonundan ekliyoruz.
4. Kodlarımızı papağana vereceğimiz için papağanımız seçili iken kod bloklarından "Kontrol" bloğunu açıyoruz ve kod başlangıcı için "**yeşil bayrak tıklandığında**" kodunu alıyoruz. (bu kodun altına yeşil bayrak butonuna tıklandığında ne olacağını gösteren kodları ekleyeceğiz.)



5. Papağanımızın sürekli uçmasını istediğimiz için süreklilik içeren bir hareketlilik için kontrol kod bloğundan "**sürekli**" kodunu aşağıdaki gibi ekliyoruz.



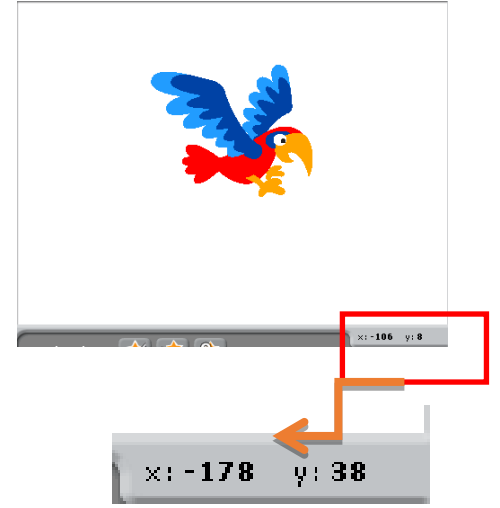
6. Resimde de görüldüğü üzere "**sürekli**" kodunun içi boş, sürekli ne olacağını bu kısma ekleyeceğiz. Hatırlarsanız 2 papağan resmimiz vardı kanat çırpıp. Bu iki görünüm arasında geçiş yapacağımız için "**Görünüm**" kod bloğundan "**..... kostümüne geçin**" kodunu sürekli kodunun içine alıyoruz. Boşluk kısmında gördüğünüz gibi papağan1-a yazıyor. İlk görünümümüz bu sonraki görünümün papağan1-b olmasını istiyoruz. Bunun için son eklediğimiz kodundan bir tane daha altına ekliyoruz.



7. Şimdi sahnemizi yeşil bayrak butonuna basarak çalıştırıp baktığımızda papağanımızın çok hızlı olarak kanat çırttığını göreceğiz. Bunu belli bir süre aralığında yapabilmesi için papağanın iki görünümünden sonrası için süre ekleyeceğimiz bunun için kontrol kod bloğundan "**1 saniye bekleyin**" kodunu resimdeki gibi ekliyoruz. Dilersek beyaz alanlarda yazan 1 sayısını tıklayarak değiştirebiliriz.



8. Yeşil bayrak butonuna basıp sahnemizi izlediğimizde görüyoruz ki papağanımız durduğu yerde kanat hareket ederken sahnenin sağ alt köşesinde gösterilmektedir.



9. Yeni eklediğimiz “**yeşil bayrak tıkladığında**” kodunun altına hareket kod bloğundan başlangıç konumunu belirten “**x y konumuna gidin**” kodunu altına ekliyoruz.



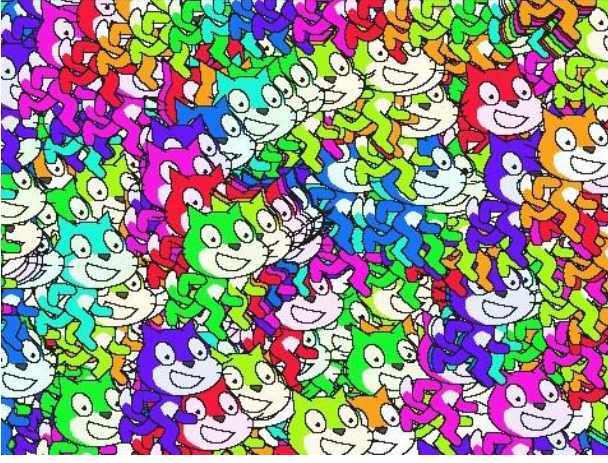
10. Bundan sonrasında kuşun gideceğini konumlarına kaç saniyede gideceğini belirten kodlar ekliyoruz. Bu hareket kod bloğunun altındaki “**1 saniyede x y konumuna gidin**” kodu eklenerek yapılır.



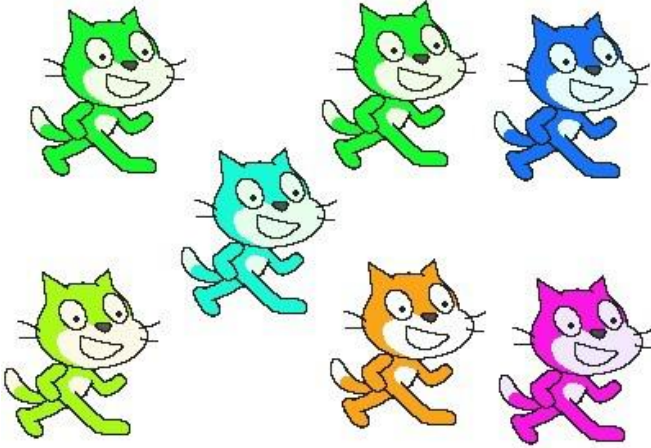
11. Son olarak isterseniz sahne arka planınızı değiştirip konuya uygun bir manzara yapabilirsiniz.

DAMGALA UYGULAMASI

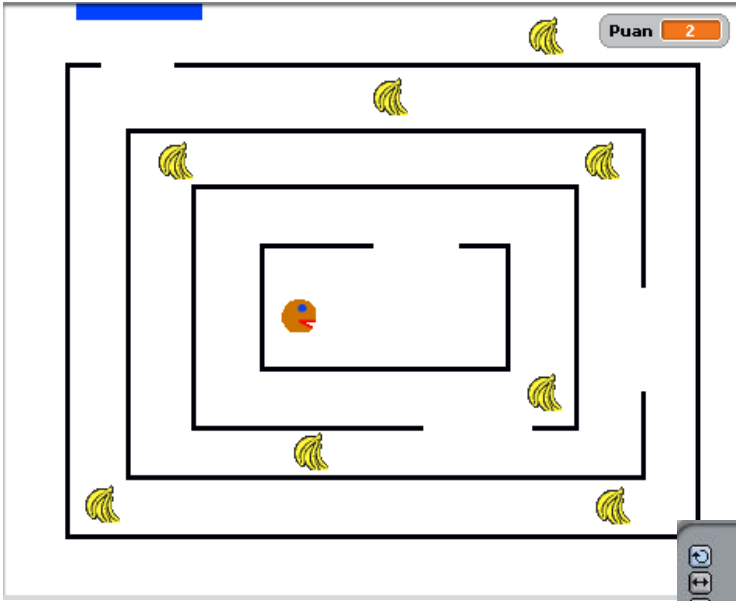
Bu uygulamada damga blokunu kullanarak görsel tasarımlar yaratabiliriz. Farklı karakterler kullanarak ya da kendi karakterlerimizi yaratarak uygulamalarımızı çeşitlendirebiliriz. Aşağıdaki projede kedi sürekli olarak fare işaretçisini takip eder ve sürekli olarak renk değiştirerek damgalar. Yeşil bayrak tuşuna basıldığında ekran temizlenir ve aynı işlemler tekrarlanır.



Bu proje kumbara projesinin benzeridir. Kedi sürekli olarak fareyi takip eder ve renk değiştirir. Kedi sadece üzerine tıklandığında damgalar.



PACMAN LABİRENT



Bu oyunda Pacman karakteri çizilen labirent çizgilerine çarpmadan mavi çizgiye ulaşmalıdır. Çizgilere değerse başa döner. Muz yerse puan kazanır.

- 1- Sahneye labirent şekli çizilir.
- 2- Pacman karakteri iki kostümlü olarak çizilir. Ağız açık ve kapalı.
- 3- Pacman'ın komutları aşağıdaki gibidir.

A- Oyun başladığında Pacman'ın tam ortada ve sağa dönük olmasını sağlar.

B- Pacman'ın sürekli ağızını açıp kapamasını sağlar.

C- Yukarı ok tıklanınca Pacman yukarı hareket eder.

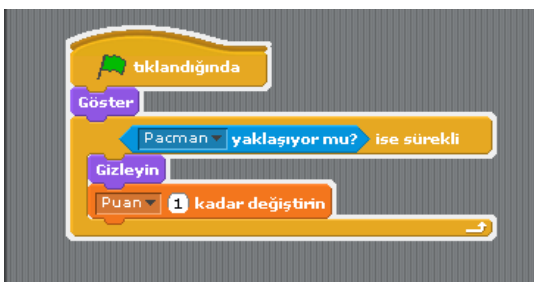
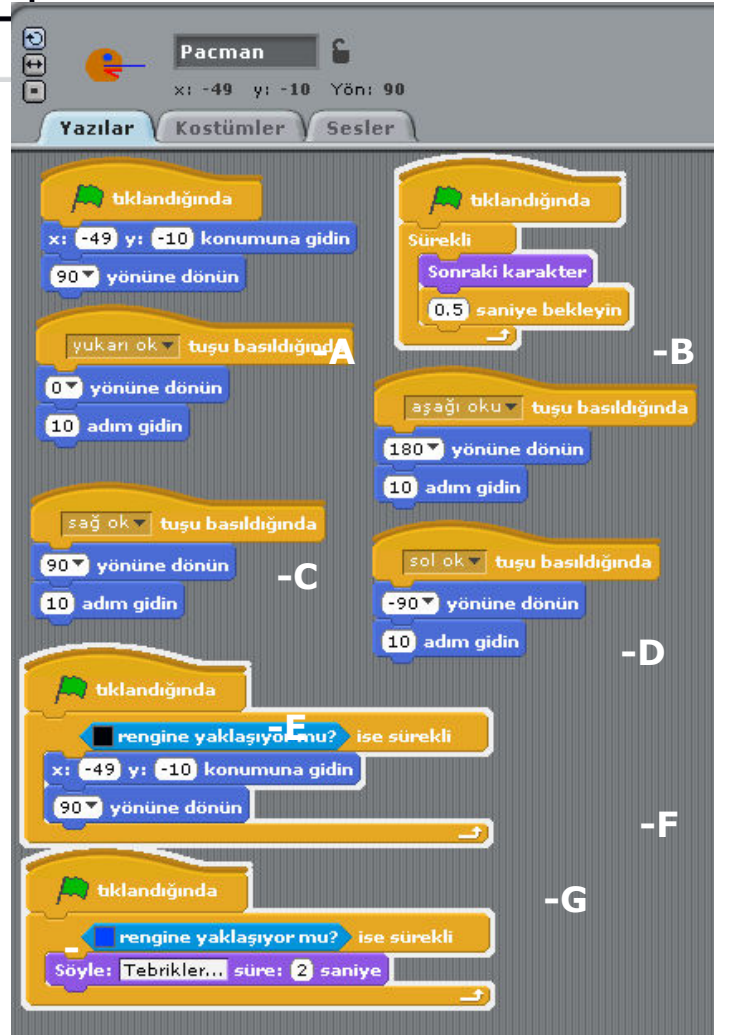
D- Aşağı ok tıklanınca Pacman aşağı hareket eder.

E- Sağ ok tıklanınca Pacman sağa hareket eder.

F- Sol ok tıklanınca Pacman sola hareket eder.

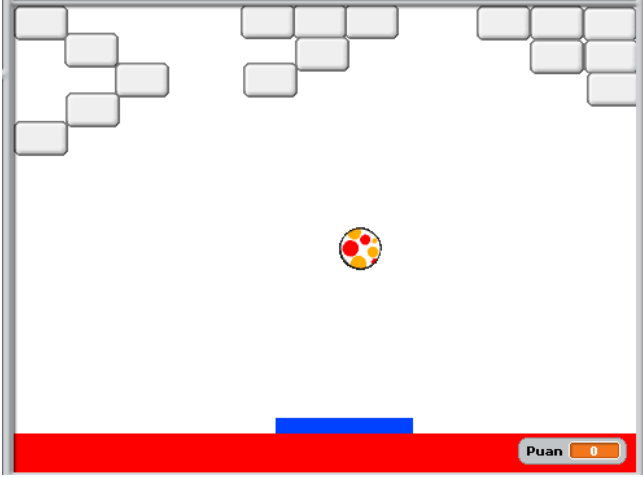
G- Siyah renge değince(çizgilere çarpınca) başa dönmesini sağlar.

H- Mavi renge değince (Bitişe geldiğinde) "Tebrikler" yazar.



Muzlara yazılacak komut yandaki gibidir. Pacman muzlara değdiğinde Muzlar gizlenir. Her muza yazılmalıdır.

DXBALL (MAUSE İLE OYNANAN TUĞLA KIRMALI)



Bu oyunda mavi çubuk Fare ile hareket ettirilir topa vurmaya çalışılır. Kırılan tuğlalar puan kazandırır. Kırmızıya değince puan kaybedilir.

Topa Yazılan Komutlar

- A-** Topun sahnedeki hareketi sağlanır.
- B-** Mavi renge yani çubuğa değince tam tersi istikamete dönüp, ilerler.
- C-** Kırmızıya değince tam tersi dönüp ilerler, puan kaybeder.



Mavi Çubuğa Yazılan Komut

Yandaki komut mavi çubuğa yazılır bu şekilde farenin sağ ve sol yöne hareketleri(Fare x yönünde) sağlanmış olur.



Tuğlalara Yazılacak Komut



Top tuğlalara değince tuğlalar gizlenir. Puan kazanılır.

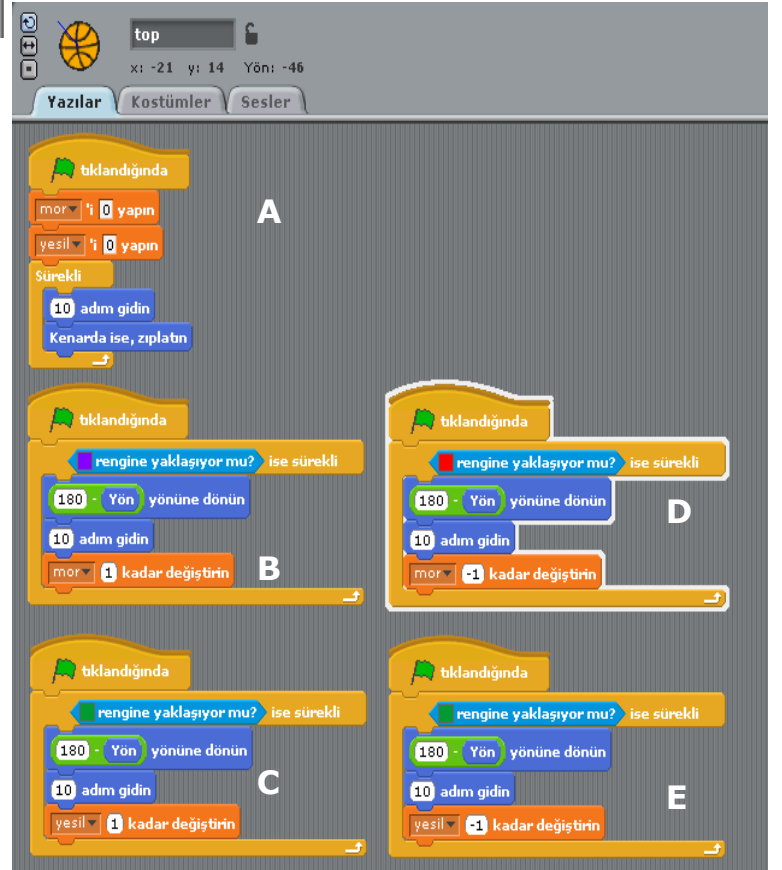
DXBALL (2 OYUNCULU)



Bu oyunda iki oyuncu vardır 1. Oyuncu(mor çubuk) sağ ve sol ok ile ikinci oyuncu(yeşil çubuk) a ve s tuşları ile oynar. Topa vurunca puan kazanırlar. Vuramayınca puan kaybederler.

Topa Yazılacak Komutlar

- A-** Topun sahnedeki hareketi sağlanır. Oyuncuların puanları oyun başında 0'lanır.
- B-** Mor renge yani mor çubuğa değince tam tersi istikamete dönüp, ilerler. Mor puan kazanır.
- C-** Yeşil renge yani yeşil çubuğa değince tam tersi istikamete dönüp, ilerler. Yeşil puan kazanır.
- D-** Kırmızıya değince tam tersi dönüp ilerler, Mor çubuk puan kaybeder.
- E-** Mavi değince tam tersi dönüp ilerler, yeşil çubuk puan kaybeder.



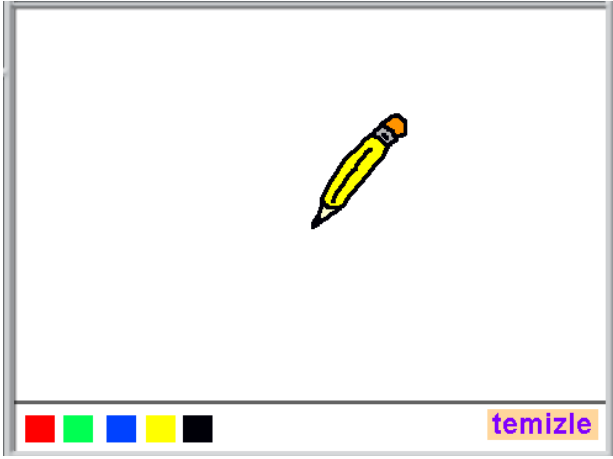
Çubuklara Yazılan Komutlar



1- Sağdaki şekilde Mor çubuk sağ oka basılınca x'i 10 değişir yani sağa ilerler. Sol oka basılınca X'i -10 değişir yani sola ilerler.

2- Soldaki şekilde yeşil çubuk "s" tuşuna basılınca x'i 10 değişir yani sağa ilerler. "a" tuşuna basılınca X'i -10 değişir yani sola ilerler.

PAİNT PROGRAMI



Bu program renklere tıklanınca kalemın ren-
gi tıklanan renk olur. Kalem o renkte yazar.
Temizleye basılınca çizim alanı temizlenir.

- 1- Karakter olarak 5 adet renk kutusu oluşturulur.
- 2- Temizle kutusu oluşturulur.
- 3- Kalem karakteri eklenir.

Renklere Yazılacak Komut

Yandaki komutlar tüm renk karelerine yazılır. O renk kutusuna tıklanınca o rengin alarmı verilir.



Kaleme Yazılacak Komut

A- Kalemi program başlayınca tam ortada tutar.(0,0) noktasında. Sahneyi temizler. Kalem boyutunu 5 yapar. Fare basılı iken kalemın yazmasını basılı değilken yazmamasını sağlar.

B- Mavi alarmı verildiğinde Kalem rengini mavi yapar.

C- Siyah alarmı verildiğinde Kalem rengini siyah yapar.

D- Yeşil alarmı verildiğinde Kalem rengini yeşil yapar.

E- Kırmızı alarmı verildiğinde Kalem rengini kırmızı yapar.

F- Sarı alarmı verildiğinde Kalem rengini sarı yapar.

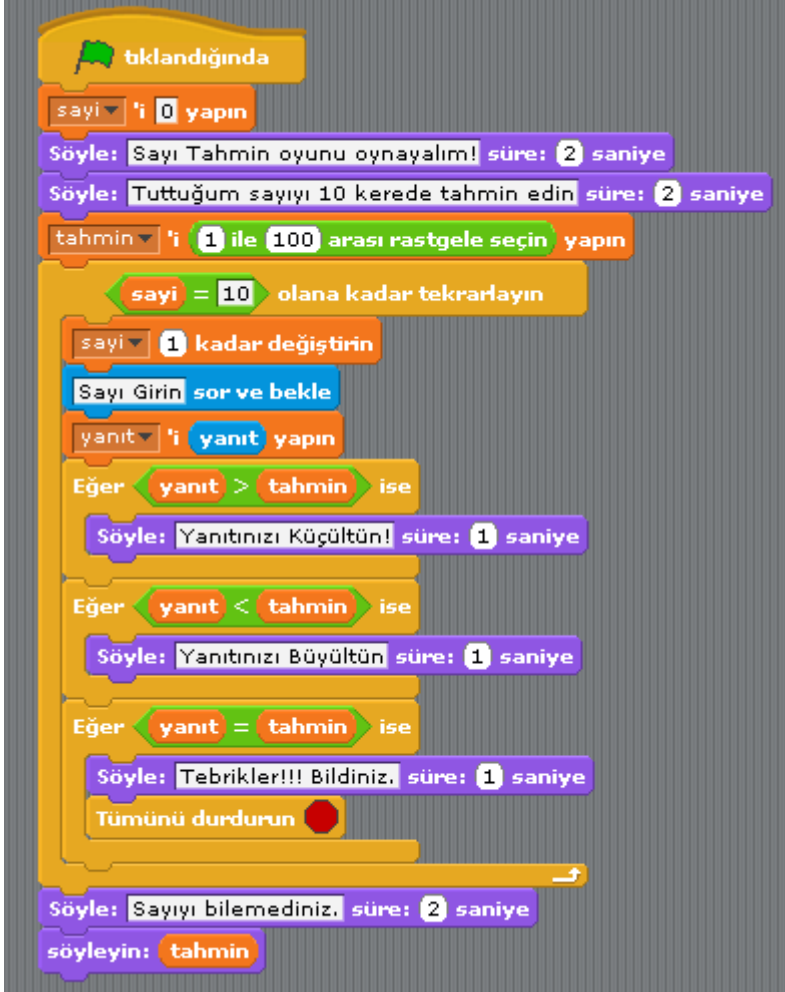
Temizle'ye Yazılacak Komut

Temizle kutucuğuna yazılacak komut yandaki gibidir. Tıklandığında tüm sahneyi temizler.



SAYI TAHMİN OYUNU

Bu programda bilgisayar 1 ile 100 arasında bir sayı tutar ve kullanıcının bu sayıyı 10 kerede tahmin etmesi istenir. Sahneye herhangi bir karakter eklemek yeterlidir. Komutlar bu karaktere yazılmalıdır.



A) Sayı, tahmin ve yanıt isimli 3 farklı değişken oluşturulur.

B) Tahmin değişkenine 1 ile 100 arasında rastgele tutulan bir sayı atanır.

C) Sayı değişkeni 10 oluncaya kadar bir döngü açılır.

D) Döngü içine tahmini girmesi istenir ve girilen tahminin büyük, küçük veya eşit olup olmadığı kontrol ettirilir.

E) 10 tahmin içinde tutulan sayı bilinirse program duracaktır.

F) 10 tahminde de tutulan sayı bilinemezse oyun biter ve tutulan sayı gösterilir.

BÜYÜK – KÜÇÜK UYGULAMASI

Girilen iki sayıdan hangisinin büyük ya da küçük olduğunu söyleyen programdır. Program komutları karaktere girilmelidir.



- A) Kullanıcıdan iki sayı girmesi beklenir.
- B) Girilen sayılar farklı iki değişkene aktarılır.
- C) Eğer komutu ile değişkenler karşılaştırılır.

NOTLAR

A series of horizontal dotted lines spanning the width of the page, intended as a guide for handwriting practice.

