

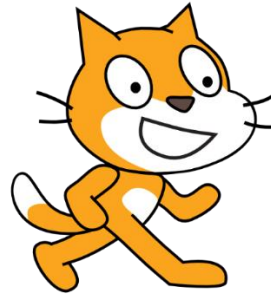
KODLAMA SERÜVENİ 2:

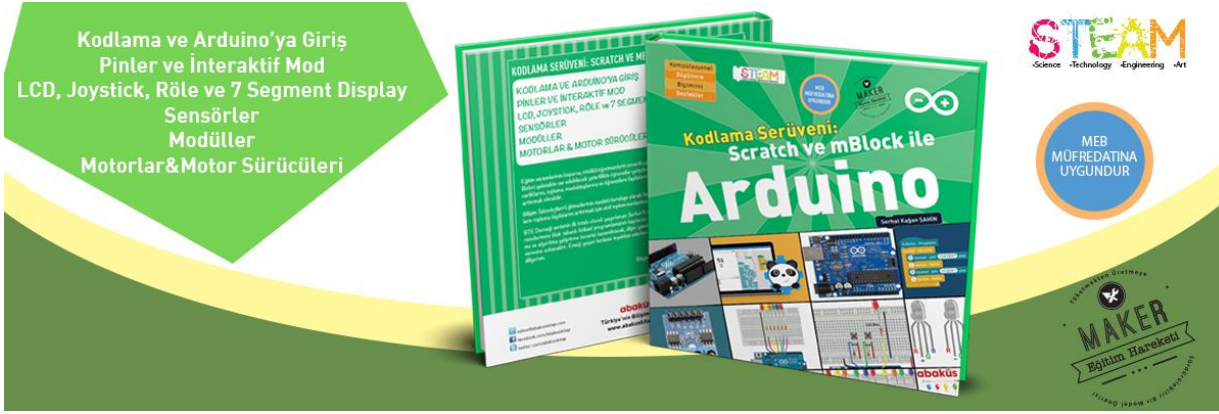
SCRATCH İLE PROGRAMLAMA

Serhat Kağan ŞAHİN

Scratch programında yapılmış kolaydan zora 29 proje

SCRATCH





**Bu Kitap; Düşünme, Problem Çözme Ve Algoritma Geliştirme Becerisi Kazandıracak, Robotlara Ve Cihazlara Hükmederek Üretme Sürecine Sokacaktır.**

### İÇİNDEKİLER

KODLAMA VE ARDUİNO'YA GİRİŞ  
ARDUİNO'YU MBLOCK İLE PROGRAMLAMAK.  
LED  
BUZZER  
BUTONLAR  
RGB LED  
PİNLER VE İNTERAKTİF MOD  
DİJİTAL PİNLER  
PWM PİNLER  
ANALOG PİNLER  
POTANSİYOMETRE (AYARLANABİLİR DİRENÇ)  
LDR (FOTO DİRENÇ)  
SERVO MOTOR  
HC-SR04 ULTRASONİK MESAFE SENSÖRÜ  
JOYSTICK  
LCD EKTRAN  
I2C MODÜLÜ  
ARDUİNO ROLE KULLANIMI  
7 SEGMENT DİSPLAY  
SENSÖRLER  
PIR SENSÖR (HAREKET SENSÖRÜ – DİJİTAL SENSÖR)

ALEV SENSÖRÜ (DİJİTAL SENSÖR)  
SES SENSÖRÜ (ANALOG SENSÖR)  
TOPRAK NEM SENSÖRÜ  
LM35DZ SICAKLIK SENSÖRÜ  
DHT11 ISI VE NEM SENSÖRÜ  
GAZ SENSÖRLERİ  
YAĞMUR SENSÖRÜ  
MODÜLLER  
74HC595 SHIFT REGISTER ENTEGRESİ  
MAX7219 MATRİX MODÜLÜ  
IR ALICI VERİCİ KUMANDA  
DS1302 RTC SAAT MODÜLÜ  
MEMBRAN TUŞ TAKIMI (KEYPAD)  
BLUETOOTH MODÜLÜ  
MOTORLAR & MOTOR SÜRÜCÜLERİ  
STEP MOTOR  
MOTOR SÜRÜCÜLERİ  
L293D ADAFRUIT MOTOR SÜRÜCÜ SHIELD  
STANDART TEKERLEK VE MOTOR  
L298N MOTOR SÜRÜCÜ DEVRESİ  
L9110 ÇİFT MOTOR SÜRÜCÜ KARTI

Scratch programını kullanmak için <https://scratch.mit.edu/> adresine girilerek



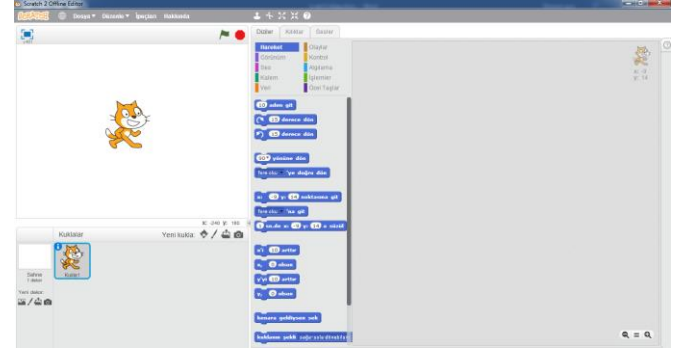
Oluştur seçeneğine tıklanarak hemen programlar yazılabileceği gibi

<https://scratch.mit.edu/download>

adresine girilerek

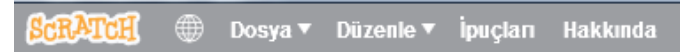


Bilgisayarımızda yüklü işletim sistemi için uygun olan sürüm indirilerek kurulur.



Program kurulduğunda ekrandaki gibi bir ara yüz gelmektedir. Gelen pencerenin bölümleri incelendiğinde:

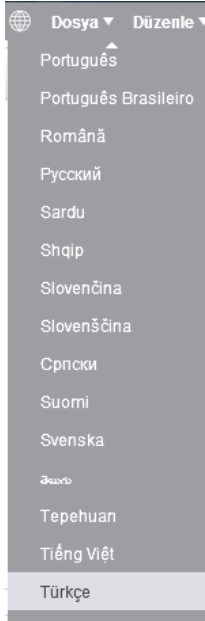
**Menü:**



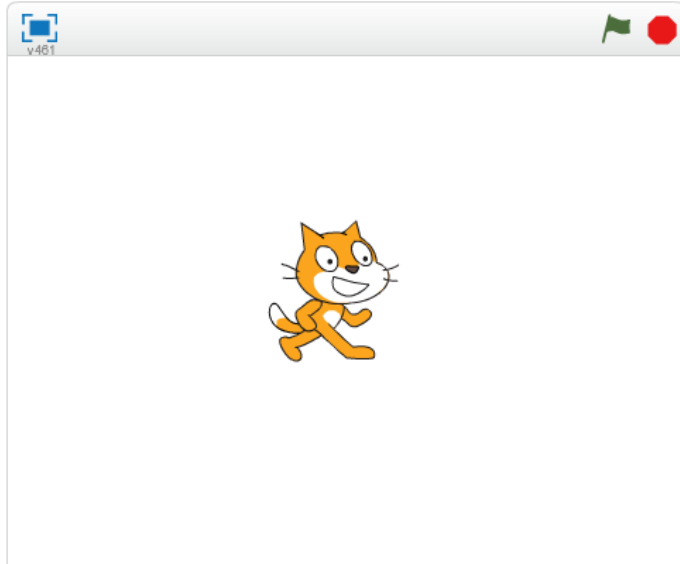
Bölümü ile kaydetme, **yeni proje oluşturma, kaydetme, var olan projeyi açma, son yapılan işlemi geri alma** gibi seçeneklerin bulunduğu menü bulunmaktadır. Şayet dil Türkçe değil de farklı bir dilde program açıldıysa bu bölümdeki,



Simgesine tıklanarak Türkçe seçeneği seçilir

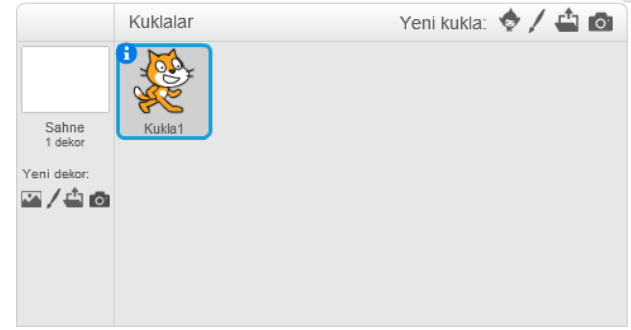


## Sahne:




Program çalıştığında kuklaların hareketlerini gözlemleyebildiğimiz alandır. Standart olarak **scratch 2** programının simgesi olan **kedi** karakteri gelir.

## Kuklalar ve Sahne Dekor:



Projelerimizde kullanacağımız kuklalarla ve sahne dekorlarıyla ilgili ayarlamaları yapabileceğimiz bölümdür. Yeni dekor bölümündeki araçlarla

| Yeni dekor: | İşlevi                   | Açıklama   |
|-------------|--------------------------|--|
|             | Dekoru kütüphaneden seç  | Scratch programıyla beraber gelen sahne tasarımları seçilebilir.   |
|             | Yeni dekor çiz           | Bu seçenek ile scratch editörü ile sahne dekoru çizmeniz sağlanır.   |
|             | Dekoru bilgisayardan seç | Bilgisayarda bulunan bir grafik ya da resim dosyasını sahneye arka plan aktararak sahne oluşturmayı sağlar |

|   |                               |  |
|---|-------------------------------|--|
|  | Kameradan yeni dekor oluştur. | Kamera sayesinde fotoğraf çekilerek sahneye dekor eklemeyi sağlar. |
|---|-------------------------------|--|

Aynı şekilde kukla için

|   |                              |
|---|------------------------------|
| Yeni kukla:     | İşlevi                       |
|    | Kuklayı kütüphaneden seç     |
|    | Yeni kukla çiz               |
|    | Kuklayı bilgisayardan seç    |
|    | Kameradan yeni kukla oluştur |

**Diziler sekmesi:** Karakterleri ve sahneyi kodlamamız için gerekli olan kod bloklarının bulunduğu alandır.

Diziler Kılıklar Sesler

**Hareket** Olaylar

Görünüm Kontrol

**Ses** Algılama

Kalem İşlemler

Veri Özel Taşlar

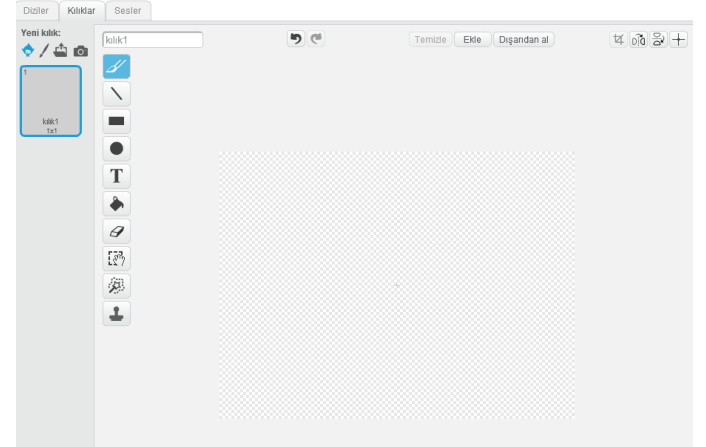
10 adım git

15 derece dön

15 derece dön

Bu bölüm Hareket, olaylar, görünüm, kontrol, ses, algılama, kalem, işlemler, veri ve özel taşlar gibi alt blokları içerisinde barındırır. Her alt blok içinde değişik işlevler için ihtiyaç duyulan kod blokları bulunur.

**Kılıklar Sekmesi:** Bir sahne dekoruna ya da kuklaya ait kılıkları düzenlemek ya da yeni kılık oluşturmak ve düzenlemek için kullanılır. Ayrıca yeni kukla çiz ya da yeni dekoru çiz seçenekleri seçildiğinde düzenlemeler yine bu bölümde yapılmaktadır.



**Sesler Sekmesi:** Bir kuklaya ait ses dosyalarının bulunduğu bölümdür. Buradan yeni ses dosyaları eklenebilir.

## Örnek 1: Kediye Yürütelim

**Anahtar kelimeler:** 10 adım git, kenara geldiysen sek, kuklanın şekli sağa-sola dönebilsin, sonraki kılık, sürekli tekrarla

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/265330451/>

**Neler öğreneceğiz:** Uygulamayı tamamladığımızda yazdığımız bir programı yeşil bayrağa basarak başlatmayı, karaktere hareket vermeyi, sahnenin duvarlarına çarptığında geri dönme işlemi, kukla yön değiştirdiğinde ters dönmesini engelleyerek sağa sola dönme hareketi yapabilmemesini, kostüm değiştirme işlemi, belirli bir süre bekleme işlemi ve yukarıdaki işlemlerin tamamını sürekli olarak tekrarlama işlemlerini öğreneceğiz.

Şimdi hazır olduğumuza göre ilk programımızı yazmaya başlayalım.



İşe, sahnedeki kedi karakterini sağa sola yönlendirerek başlayabiliriz. Kedi kuklasının seçili olduğundan emin olduktan sonra olaylar sekmesindeki



butonu sahneye sürüklenerek programın sahnenin üst tarafında bulunan



yeşil bayrak simgesine tıklanarak başlaması sağlanabilir. Kuklanın yeşil bayrağa tıklandıktan sonra hareket etmesi için hareket sekmesinden



bloğu tıklanınca bloğunun altına sürüklenmelidir.



Bu işlemi yaptıktan sonra yeşil bayrağa tıklanınca kukla kısa bir mesafe hareket edip duracaktır. Kuklanın hareketini sürekli olarak sürdürebilmesi için kontrol sekmesinden sürekli bloğu kullanılmalıdır. Bu işlem sayesinde programımız dışarıdan bir müdahale gelene kadar yani program kullanıcı tarafından sonlandırılana kadar sürekli çalışacaktır.

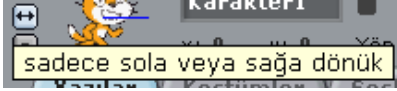


Artık kukla hareketini sürekli bir şekilde tekrarlamakta ancak ekranın sonuna geldiğinde orada kalmaktadır.

Kuklayı ekranın sonuna geldikten sonra geri döndürmek için hareket sekmesindeki **kenara geldiysen sek** bloğu kullanılmalıdır.



Bu işlemden sonra kuklanın kenara değdikten sonra geri döndüğünü gözlemlemiştinizdir. Ancak kula geri dönüş hareketi yaparken ters dönmektedir. Scratch 1.4 versiyonunu kullandıysanız bu işlemi yapmak için



sahnenin üzerinde seçeneğini seçmek gerekiyordu. Scratch programının yeni (2.0) sürümünde bu işlem artık kod bloklarıyla kolaylıkla yapılabilmektedir.

Hareket sekmesi altındaki



bloğu ile

kullanılarak **ters** dönmesi kodblokları ile engellenmiş olur.

### Algoritma:

1. Karakterimiz ters dönmeden sağa sola hareket edebilsin
2. Biraz gitsin
3. Duvarlara çarparsa geri dönsün
4. 2. Adımdan devam etsin



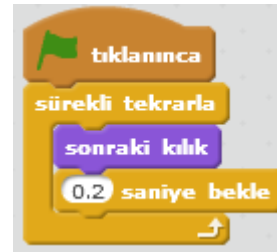
Karakterimiz artık sağa sola doğru sürekli hareket edebilmektedir. Karakterin hızını azaltmak ya da arttırmak için adım sayısı artırılıp azaltılabilir.



Yeşil bayrağa basıldığında yürüme işlemi yerine kayarak hareket etmektedir. Bunun yukarıdaki kod bloğundan farklı olarak

### Algoritma:

1. Karakterimizin kostümünü değiştirelim
2. Kısa bir süre bekleyip
3. 1. Adımdan devam edelim



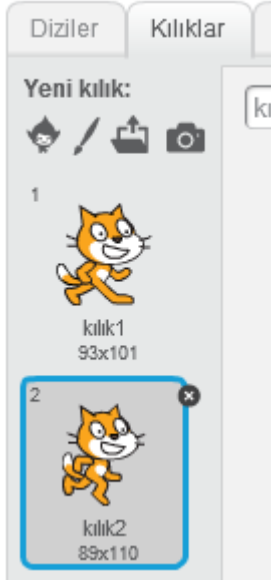
Bloğu oluşturulmalıdır. Burada adım atma eyleminin hareketle daha senkron olabilmesi için

0.2 saniye bekle

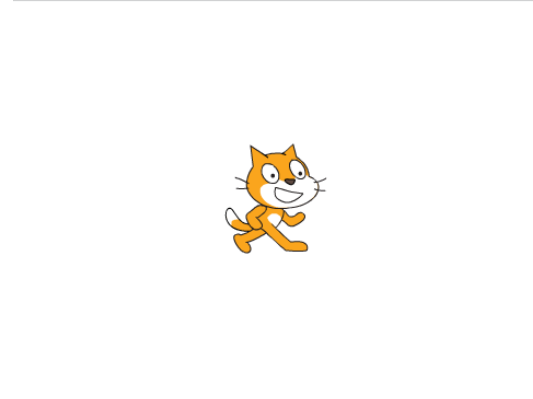
bloğu kullanılmalıdır. Kuklanın bu şekilde hareket ediyor görülebilmemesi sadece yukarıda kullandığımız

sonraki kılık

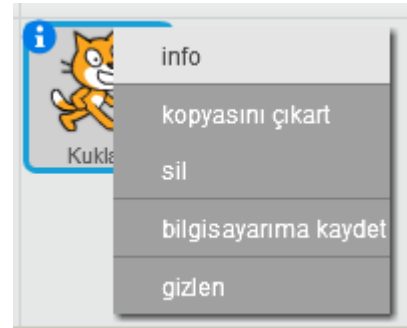
bloğu ile sağlanmamakta ayrıca kılıklar sekmesinde o kuklanın farklı bir görünümünün olması gerekmektedir.



Artık temel işlemleri öğrendiğimize göre bundan sonraki örnekleri kolaylıkla yapabiliriz.

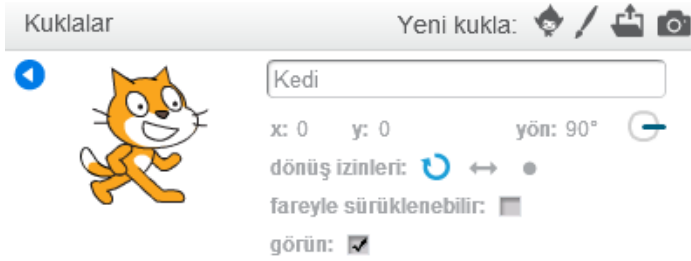


Son olarak kuklanın ismini değiştirmek için kuklanın üzerine sağ tuşa basılarak info seçeneği seçilir,



Kukla1 yazan yere istenilen isim girilebilir.





### Örnek 2: Canlanan Harfler

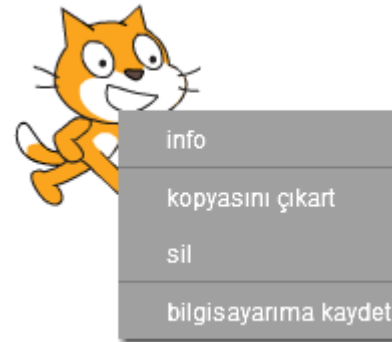
**Anahtar Kelimeler:** Renk etkisini arttır, .. ile .. arasında bir sayı tut.

**Projenin tamamlanmış hali:**

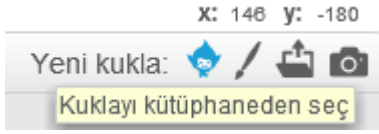
<https://scratch.mit.edu/projects/265330896/#player>

**Neler öğreneceğiz:** Bu bölümde sahnede bulunan herhangi bir kuklayı sahneden silmeyi, sahneye kütüphaneden kukla eklemeyi, renk etkisi özelliği kullanarak kuklanın rengini değiştirmeyi ve belirlenen aralıklarda rasgele bir sayı tutmayı öğreneceğiz.

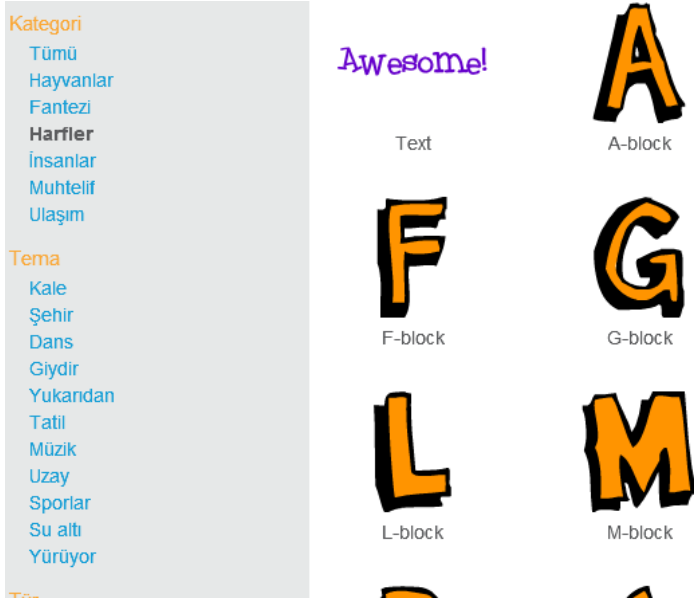
Örneğimizde sahneye eklenen harflerin renklerini rasgele değiştirerek farklı tonlarda görünmesini sağlayacağız. Sahneden kedi kuklası üzerine sağ tuşla tıklanarak silinir



Kütüphaneden yeni kukla seç düğmesine basılır



Harfler sekmesinden harflerin bulunduğu sekme tıklanır ve istenilen harfler seçilere sahneye aktarılır.



Kuklalar sırayla seçilerek aşağıdaki bloklar yazılır.



Görünüm sekmesinden



Bloğu yardımı ile kodlar eklenir.



Bu şekilde yapılan bir uygulama ile harflerin hepsi aynı renk yanar. Farklı renk kombinasyonları için işlemler sekmesindeki



Bloğu yardımı ile

**Algoritma:**

1. Rasgele bir renk seçelim
2. Kısa bir süre bekleyelim
3. 1. Adımdan devam edelim



Kodlarımız düzenlenir. Bu sayede yeşil bayrağa tıkladığında program sürekli çalışarak karakterin renk etkisini 1 – 255 arasında rasgele olarak 0.3 saniye aralıklarla değiştirecektir.

CAN

### Örnek 3: Geometrik Şekiller

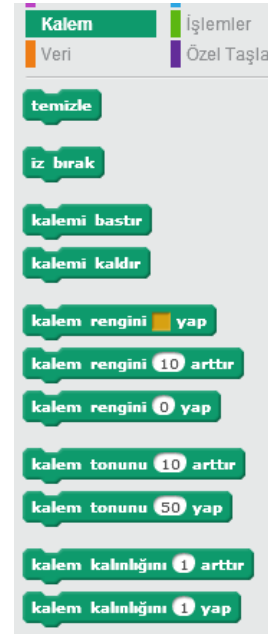
**Anahtar Kelimeler:** Temizle, kalemi bastır, kalemi kaldır, kalem kalınlığını .. yap, .. derece dön, .. defa tekrarla. X .. y.. noktasına git

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/265332656/>

**Neler Öğreneceğiz:** Bu bölümde sahneyi temizlemeyi, kalemi bastırarak ekrana çizgi çizmeyi, kalem kalınlığını ayarlamayı, karakterin yönünü değiştirmeyi, döngüyü belirli bir sayıda sınırlandırmayı öğreneceğiz.

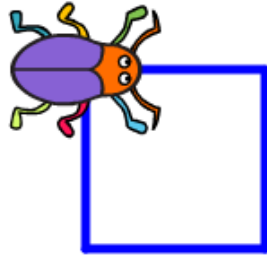
Örneğimizde kalem sekmesindeki araçlar yardımı ile çeşitli geometrik şekiller çizme işlemi yapacağız. Öncelikle sahneye herhangi bir kukla eklenmelidir. Ardından **scratch**'taki kalem sekmesindeki özellikler kullanılacaktır.



Yeşil bayrağa her tıkladığında ekranın temizlenmesi için temizle komutu en üste alınmalıdır.



Kalem bastır komutu ile karakterin iz bırakması sağlanır. Kalem kalınlığından çizilecek çizginin kalınlığı ayarlanır. Daha sonra karakterin belirli bir mesafe giderek 90 derece bir yöne dönmesi sağlanır. Bu işlem 4 defa tekrarlanarak kare şekli çizilebilir. Çizimleri net görebilmek için araya 1 saniyelik beklemler eklenmelidir.



Bloklarımızı baktığımızda aynı işlemlerin 4 defa yapıldığı ve programın gereksiz uzadığı görülmektedir. Daha önce

kullandığımız **sürekli tekrarlar** komutu yerine **kontrol** sekmesinde bulunan



Bloğu ile istenilen sayıda çalışacak döngü oluşturulabilir.

### Algoritma:

1. Sahneyi temizleyelim
2. Kalem yazmaya başlasın
3. Kalem kalınlığını ayarlayalım
4. Biraz ilerlesin
5. 90 derece sağa dönsün
6. Biraz beklesin
7. 4. Adımdan devam etsin

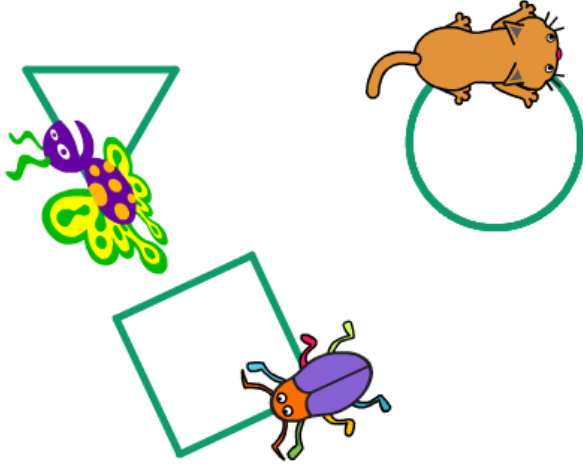


Görüldüğü gibi aynı işlem çok daha az blok kullanılarak yapılabilmektedir.

**Üçgen** ve **daire** çizmek için gerekli bloklar.

```
tıklanınca
üçgen de 2 saniye
temizle
kalemi bastır
kalem kalınlığını 5 yap
3 defa tekrarla
100 adım git
120 derece dön
1 saniye bekle
```

```
tıklanınca
ÇEMBER de 2 saniye
temizle
kalemi bastır
kalem kalınlığını 5 yap
36 defa tekrarla
10 adım git
10 derece dön
```



x: -208 y: 168

## Örnek 6: İz Bırakma

Anahtar kelimeler: iz bırak, fare oku'ya dön, kılığı kütüphaneden seç, renk etkisini .. arttır.

Projenin tamamlanmış hali:

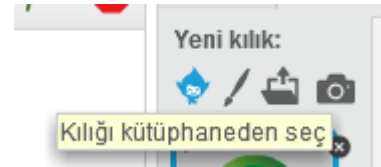
<https://scratch.mit.edu/projects/270008333/>

Neler Öğreneceğiz: Bu bölümde karakterimize yeni kılıklar eklemeyi, karakterin yönünü fare imlecine çevirmeyi, karakterin rasgele bir açıyla dönmesini, klavyeden tuşa basarak kılığın değiştirme işlemlerini öğreneceğiz.

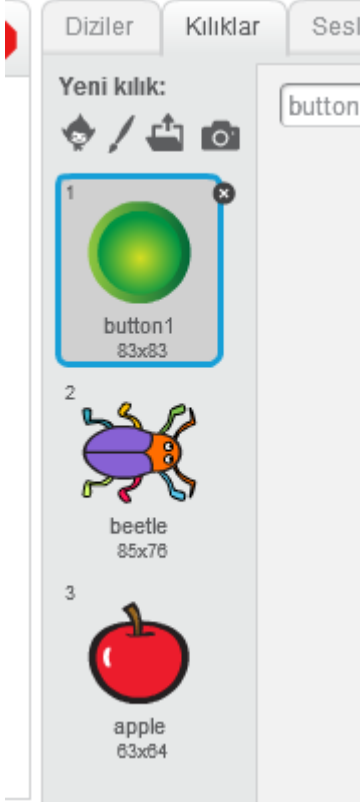
Scratch programında iz bırakma özelliği sayesinde kuklaların gölgeleri yaratılabilmektedir. Örneğimizde karakterin fareye doğru hareket etmesini bu işlemi yaparken klonlarını ekrana bırakmasını sağlamak için sahneye bir karakter eklenir



Daha sonra kılıklar sekmesinde girilerek



Seçeneği seçilerek farklı karakterler de kılığın içine eklenir.



Karakterin içine

## Algoritma:

1. Sahnenin belli bir noktasına gidelim
2. Sahneyi temizleyelim
3. Arkada iz bırakalım
4. Biraz ilerleyelim
5. Renk etkisini arttıralım
6. Yönümüzü fare imlecine çevirelim
7. Rasgele bir tarafa dönelim
8. Duvarlara çarparsak geri dönelim



Blokları eklenerek yeşil bayrağa basıldığında, kuklanın belli bir yerden başlaması ve ekranın temizlenmesi, programın sürekli çalışarak iz bırakmaya başlaması, 5 adım giderek renk etkisini değiştirmesi, fareye doğru

fareyi takip etmesi ve ekrana klonunu bırakması sağlanır. Ayrıca boşluk tuşuna basıldığında kılığın değişmesi için karaktere

## Algoritma:

1. Klavyenin boşluk tuşuna bastığında
2. Kılık değiştirelim



Bloklarının da eklenmelidir.



## Örnek 5: Labirent Oyunu

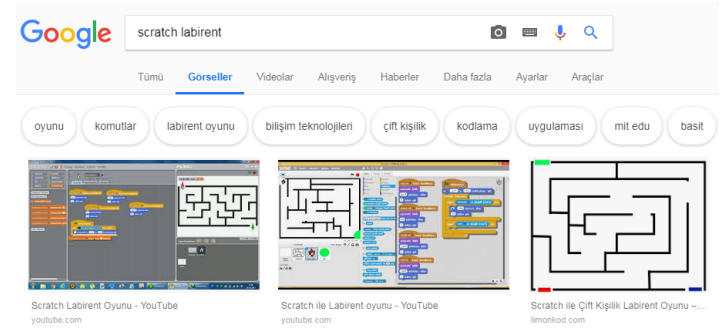
Anahtar kelimeler: x: .. y: .. noktasına git, büyüklüğü %.. yap, ... tuşu basılınca, dekoru bilgisayardan seç, x'i .. arttır, y'yi .. arttır.

Projenin tamamlanmış hali:

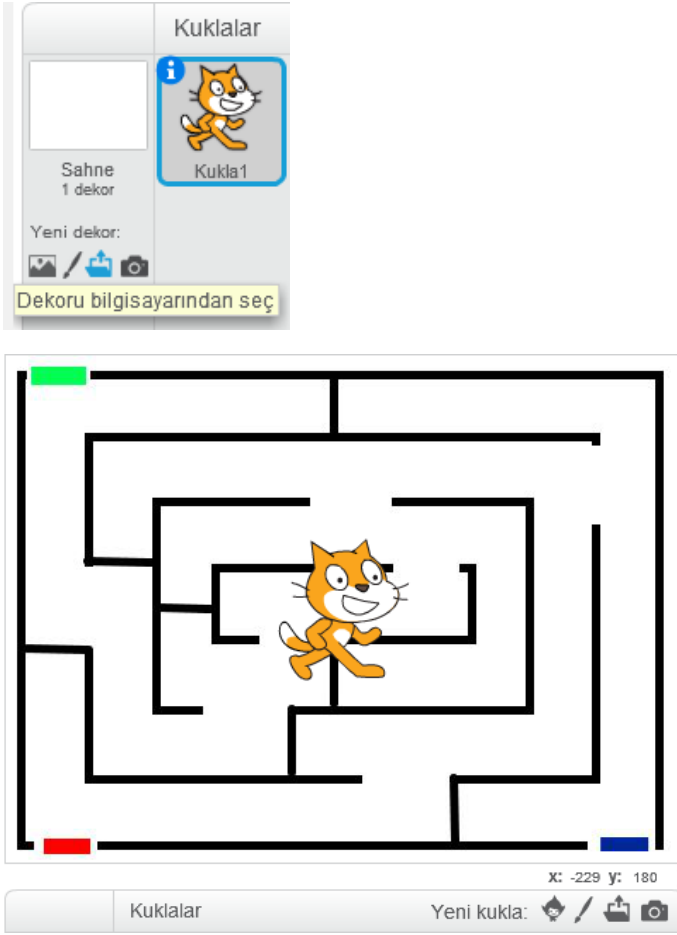
<https://scratch.mit.edu/projects/269973852/#player>

**Neler Öğreneceğiz:** Bu bölümde sahneye kütüphane haricinde internetten bulduğumuz resmi eklemeyi, karakteri klavye tuşları ile hareket ettirmeyi, programın bir renge değiştiğinde bunu algılamasını öğreneceğiz.

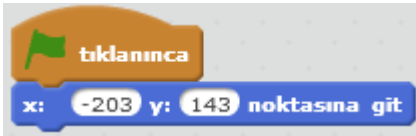
Örneğimizde sahneye labirent ekleyerek istediğimiz kuklayı labirent içinde hareket ettirebiliriz. Oyunu oynayabilmek için yeni sahne dekoru oluşturabilir ya da internet üzerinden labirent dekoru bulabiliriz.



Bulunan resim sağ tıklanarak bilgisayara(masaüstüne) indirilir. Saha sonra dekoru bilgisayardan seç seçeneği ile sahneye aktarılır.



Program çalıştığında karakterin harekete sahenin sol üstteki yeşil alandan başlaması için



Bloğu eklenir. Ayrıca dikkat edilirse karakter labirente göre biraz büyüktür. Boyutunu küçültmek için görünüm

sekmesindeki 'büyüklüğü % 100 yap' (size 100%) bloğu kullanılmalıdır. Karakterin boyutu % 30 yapılarak karakterin ekranın sol üst köşesinden başlaması sağlanır.



Artık karakterimiz labirent içinde rahatlıkla hareket edebilecektir. Bunun için karakter içine yazılacak kod blokları

## Algoritma:

1. Klavye tuşlarına bastığımızda
2. Karakterimiz yürütelim
3. Kostüm değiştirelim





Şeklinde olmalıdır. Bu sayede klavyedeki ok tuşlarına basılarak karakterin hareket etmesi sağlanabilir. Ayrıca sonraki kılık bloğu sayesinde karakterin yürüme hareketi yapması sağlanabilir.

Karakterin siyah çizgiler üzerine geldiğinde ileri gitmemesi yani duvarlardan geçmesini önlemek için karaktere ilk eklenen bloklara



Bloğu eklenmelidir. Artık karakter duvarlara(siyah renge) çarptığında geri dönecek yani olduğu yerde kalacaktır. Karakterin kırmızı bölgeye geldiğinde oyunun sonlanması için ise karaktere ait bloklar

### Algoritma:

1. Sahnenin belli bir noktasına gidelim
2. Boyutu küçültelim
3. Siyah renge değdiğimizde geri gidelim
4. Kırmızı renge değdiğimizde, tebrikler diyip oyunu bitirelim.



Şeklinde düzenlenmelidir.



## Örnek 7: Uçan Yarasalar

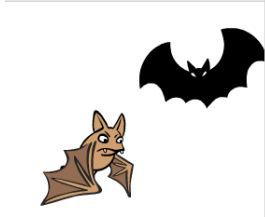
**Anahtar kelimeler:** sağ-sol-yukarı-aşağı yönüne dön, .. tuşu basılı mı?

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/270126816/#player>

**Neler Öğreneceğiz:** Bu bölümde karakterin fare imlecinin bulunduğu konuma gitmesini, karakterin yön tuşları ile hareket etmesini, yönüne dön özelliği ile karakterin yönüne değiştirmeyi öğreneceğiz.

Örneğimizde yarasalardan birini klavye tuşlarıyla diğerini farenin hareketine göre yöneteceğiz.



**Algoritma:**

1. Fare imlecinin bulunduğu yere gidelim

**Algoritma:**

1. Kostüm değiştirilim
2. Biraz bekleyelim
3. 1. Adıma dönelim

Siyah yarasanın içine aşağıdaki kod bloklarını ekliyoruz.



Bu sayede karakter hem kendi kendine kanat çırpma

hareketi yapmakta hem de komutuyla farenin bulunduğu konuma gitmektedir.

Daha önceki labirent örneğinde karakterin, klavye tuşlarıyla hareketini sağlamak için “**olaylar**” sekmesindeki



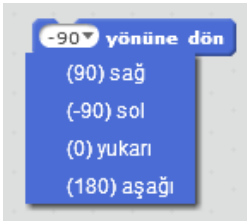
Bloklarını kullanmıştık.



Ancak bu şekildeki kullanımda yön tuşlarına basılı tutulduğunda karakter önce bir adım hareket edip duracak sonra hareketine devam edecektir. Yani karakter hareketinde ufak bir takılma olacaktır. Bu bloklar yerine “kahverengi yarasa” karakterine aşağıdaki bloklar eklenerek kesintisiz bir şekilde hareket etmesi sağlanır.



Ayrıca sağa ve sola hareket etmesi için eklenecek



“Yönüne dön” bloğu sayesinde ise karakterimiz sağa ve sola dönme hareketi yapabilmektedir. Yön değiştirdiğinde karakterin ters dönmemesi için ise .

kuklanın şekli sağa-sola dönebilsin

Bloğu kullanılmalıdır.

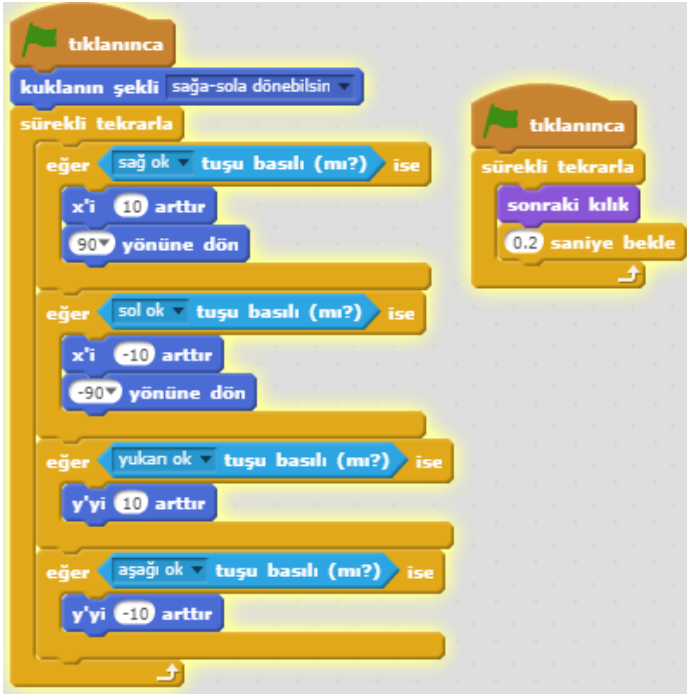
Aynı şekilde bu işlem yukarı aşağı tuşları içinde eklenerek gideceği yönün açısı ayarlanabilir.

Son olarak kuklanın blokları

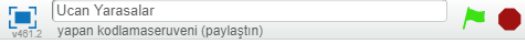


**Algoritma:**

1. Karakterimiz ters dönmeden sağa sola hareket edebilsin
2. Klavyenin yön tuşlarına basıldığında
3. Basılan yöne doğru dönelim
4. Basılan tuşun yönüne doğru hareket edelim
5. Kostüm değiştirelim
6. Biraz bekleyelim



Şeklinde olmalıdır.



## Örnek 8: Hayaleti yakala

Anahtar kelimeler: bir değişken oluştur, görün, gizlen, değişkeni .. arttır, değişken 0 olsun, bu kukla tıklandığında

Projenin tamamlanmış hali:

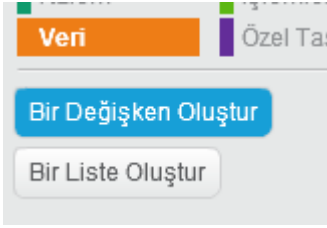
<https://scratch.mit.edu/projects/270154290/>

Neler Öğreneceğiz: Bu bölümde değişken kavramı ve değişken oluşturma, değişkenin değerini değiştirme, değişkenin değerini 0 yapma, rasgele konuma gitme özelliklerini öğreneceğiz.

Örneğimizde sürekli sahnede yeri değişen hayaleti yakalamaya çalışacağız.

Bir adet değişken oluşturarak hayaletin kaç kere tıklandığını hesaplayacağız. Öncelikle veri bloğuna girip değişken oluşturmamız.

Değişken nedir: Bir işlemi gerçekleştirmek için yapılması gereken ilk şey o veriyi hafızaya almaktır. İşlem yapmak istediğimizde de veriyi hafızadan çağırıp gerekli işlemleri yerine getirmektir. Hafızadaki verileri ifade etmek için programlama dillerinde değişkenleri kullanırız. **Özet vermek gerekirse; programlama dilinde işlediğimiz verileri bilgisayarın hafızasında tutmak için yapmış olduğumuz tanımlamalardır.**



Skor isimli bir değişken oluşturalım.



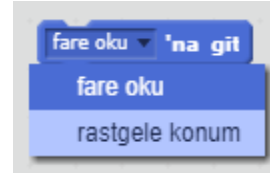
Hayalet karakterinin içine eklenecek kodlar aşağıdaki gibi olmalıdır.

### Algoritma:

1. Karakterin boyutunu ayarlayalım
2. Karakterimiz sahnenin ortasına konumlansın
3. Puanımız 0 dan başlasın
4. "hadi beni yakala " desin ve başlasın
5. Karakterimiz görünmez olsun
6. Biraz bekleyelim
7. Karakterimiz ekranın herhangi bir yerine gitsin
8. Karakterimiz görünsün
9. Biraz bekleyelim
10. 5. adımdan devam edelim



Rasgele konuma git bloğunu kullanmak için, hareket sekmesinden "fare oku'na git" bloğu altındaki

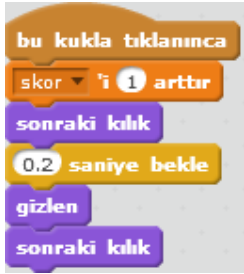


Seçeneği seçilmelidir.

Ayrıca aşağıdaki bloklarla, karakter üzerine tıkladığında;

### Algoritma:

1. Kuklanın üzerine tıkladığımızda
2. Puanımız 1 artsın
3. Kostüm değiştirelim
4. Biraz bekleyelim
4. Karakterimiz görünmez olsun
5. Kostüm değiştirelim



Son olarak karakterin kod blokları aşağıdaki gibi olmalıdır.



## Örnek 9: Fare Yakalama Oyunu

**Anahtar kelimeler:** fare oku'a mesafe, operatörler(<=>), fare oku'ye doğru dön, karakter'e doğru dön, rasgele konum

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/270289904/>

**Neler Öğreneceğiz:** Bu bölümde karakterin fare oku ile mesafesine göre konumlanmasını, büyük operatörünü, programın iki karakterin birbirine değip değmemesini algılamasını sağlamayı öğreneceğiz.

Örneğimizde sahneye ekleyeceğimiz fare karakterinin imleç'in peşinden gitmesini, kedi karakterinin ise fare karakterini kovalamasını sağlayacağız. Sahneye kedi ve fare kuklaları eklenir.



Fare karakterine aşağıdaki bloklar eklenerek, imleç'e belli bir mesafeye kadar yaklaşması bu sayede karakterin yönünü şaşırması, Sürekli olarak fare okuna doğru yönünü dönmesi, 15 adım gitmesi, Sürekli olarak kılık değiştirmesi sağlanır.

## Algoritma:

1. Karakterimizin fare imleci ile arasında belli bir mesafe olduğunda
2. Fare imlecine doğru dönelim
3. Biraz gidelim
4. Kılık değiştirelim
5. Biraz bekleyelim
6. 1. Adımdan devam edelim



Ayrıca kedi karakteri değiştiğinde tepki vermesi için fare karakterine

## Algoritma:

1. Kedi karakteri bizi yakaladığında
2. Çığlık atalım



Blokları eklenmelidir.

Kedi karakterinin fare karakterini takip etmesi için gerekli blokları.

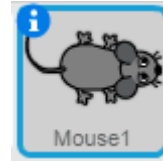


## Algoritma:

1. Fareye doğru dönelim
2. Biraz ilerleyelim



Şeklinde olmalıdır.



```
Scratch Script 1:
tıklanınca
sürekli tekrarla
eğer fare oku a mesafe > 20 ise
  fare oku 'ye doğru dön
  15 adım git
  sonraki kılık
  0.05 saniye bekle
  ↻

Scratch Script 2:
tıklanınca
sürekli tekrarla
eğer Cat2 a değdi (mi?) ise
  eek de 2 saniye
  ↻
```



```
Scratch Script 3:
tıklanınca
sürekli tekrarla
Mouse1 'ye doğru dön
3 adım git
  ↻
```



## Örnek 11: Pervane Döndürme

Anahtar kelimeler: eğer koşulu, bu kukla tıklanınca, .. derece dön

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/270531937/>

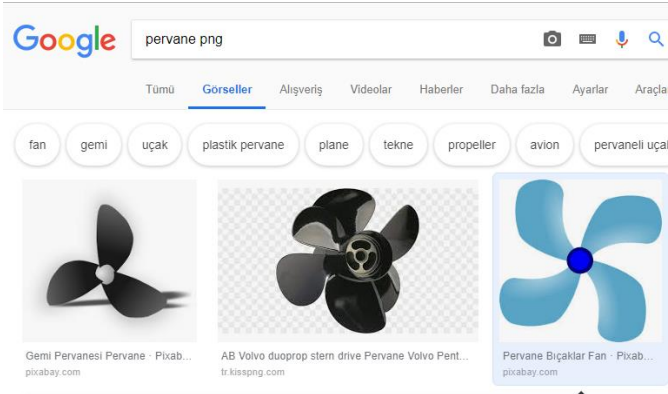
Neler Öğreneceğiz: Bu bölümde karakter tıklanınca karakterin içindeki blokların çalışmasını, sahneye internetten bulduğumuz bir karakteri eklemeyi öğreneceğiz.

Örneğimizde basit bir pervane yapıp hızlı ya da yavaş bir şekilde dönmesini sağlayacağız. Öncelikle sahneye arka plan ekleyelim



Daha sonra internette pervane resmi aratılarak arka plansız bir pervane resmi buluyoruz.



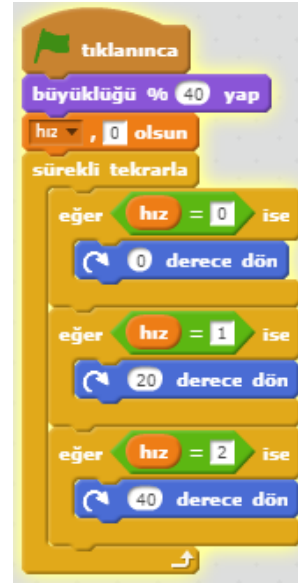


Resmin büyük halini bilgisayara indirip, kuklayı bilgisayardan seç seçeneği ile ekrana ekliyoruz. Pervanenin içine yazacağımız kodlar



## Algoritma:

1. Karakterimizin boyutunu ayarlayalım
2. Hız değişkeninin değerine göre
3. Hız değişkeninin değeri 0 olduğunda pervanemiz dursun
4. 1 olduğunda yavaş dönsün
5. 2 olduğunda hızlı dönsün
6. 3. Adımdan devam etsin



Şeklinde olmalıdır.

Pervanenin büyüklüğü ayarlanır,

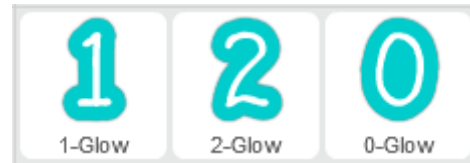
Hız isimli değişken 0 yapılır,

Eğer hız değişkeninin değeri 0 ise pervanenin dönüş hızı 0

Hız değişkeninin hızı 1 ise dönüş hızı 20

Hız değişkeninin hızı 2 ise dönüş hızı 40 olmalıdır.

Pervanenin hızını ayarlamak için ise sahneye kütüphane içinde bulunan



Kuklalarını ekliyoruz.. hız isimli bir değişken ile pervanenin dönme hızını rakamlar üzerine tıklanarak değişmesini sağlıyoruz.

### Algoritma:

1. Hangi rakam tıklanırsa
2. Onun hızını alalım



## Örnek 10: Sahne Değiştirme

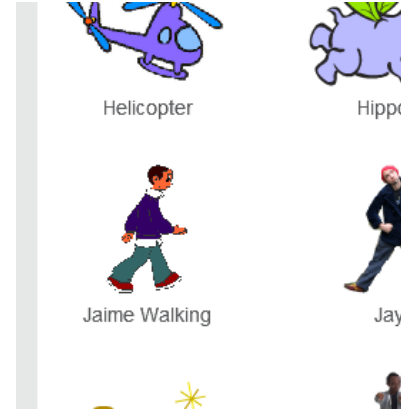
Anahtar kelimeler: sonraki dekor, kenar değdi (mi?), ..haberini sal

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/270373742/>

**Neler Öğreneceğiz:** Bu bölümde programın karakterin duvara değip değmediğini algılamasını, karakterin duyuru yapmasını, duyuru yapıldığında sahne dekorunu değiştirmeyi öğreneceğiz.

Öncelikle sahneye yürümesi için bir karakter eklenir. Jaime Walking isimli karakter bu iş için çok uygundur.



Karakterin kılıklarına bakılırsa yürümesi için gerekli karakterlerin olduğu görülmektedir.



Karakterin yürümesi için aşağıdaki bloklar eklenmelidir.

## Algoritma:

1. Kostüm deęiřtirelim
2. Biraz bekleyelim
3. 1. Adımdan devam edelim



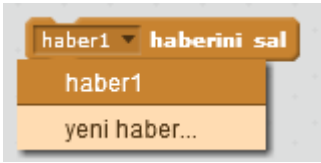
Karakterimize yürüme efekti ekledik. Şimdi ise kenara geldiğinde sahne dekoru deęiřtirmemiz gerekiyor. Bunun için karakterin ekranın sonuna geldiğinde yani kenara geldiğinde bir duyuru yapmasını saęlayacaęız.

## Algoritma:

1. Biraz gidelim
1. Duvarlara çarptığımızda
2. Duyuru yapalım
3. Bařlangıçtaki konuma gidelim
4. 1. Adımdan devam edelim

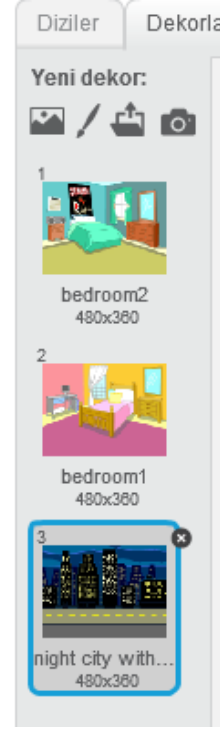


Karakter kenara geldiğinde olaylar sekmesinde bulunan



Bloğunun altındaki haber1 haberini sal ya da yeni haber tıklanarak oluşturulacak haber ismi belirlenebilir. Kenar isimli bir haber oluşturarak yukarıdaki bloklar oluşturulmalıdır.

Sahnenin dekorunun değişmesi için sahneye birkaç dekor eklemeliyiz.



Son olarak sahne dekorunun içine

## Algoritma:

1. Duyuru yapınca
2. Sahne değiştirilim



Blokları eklenmelidir. Artık karakterimiz ekranın sonuna geldiğinde hem sahnenin başına gidecek hem de sahne dekoru değişecektir.



### Örnek 4: Pong Oyunu

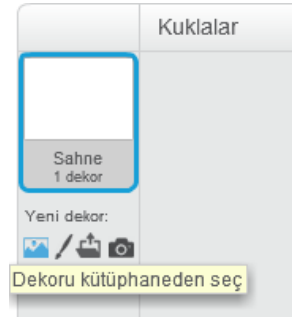
**Anahtar Kelimeler:** Dekorü kütüphaneden seç, x farenin x'i olsun, yönüne dön, karaktere değdi mi?, rengine değdi mi?, hepsini durdur, koşul – eğer, yönü

Projenin tamamlanmış hali:

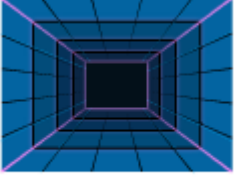
<https://scratch.mit.edu/projects/265335457/#player>

**Neler Öğreneceğiz:** Bu bölümde sahneye kütüphaneden dekor eklemeyi, kuklanın farenin sağa ve sol hareketi ile beraber sağa sola hareket etmesini sağlamayı, bir kuklanın başka bir kuklaya değdiğinde programın bunu algılamasını, kuklanın bir renge değdiğinde programın algılamasını, karakterin belirli bir koordinata gitmesini, koşullu ifadeleri ve operatör kavramlarını göreceğiz

Örneğimizde basit bir top sektirme oyunu yapacağız. Bunun için önceki örneklerden farklı olarak sahne dekorunu değiştireceğiz. Öncelikle sahneden kedi karakteri silinmelidir. Ardından yeni dekor sekmesindeki **dekorü kütüphaneden seç** seçeneği seçilerek



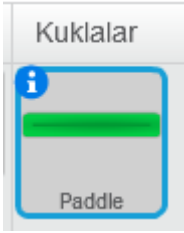
neon tunnel isimli dekor seçilmelidir. (farklı dekor da seçilebilir)



neon tunnel

Bu işlemden sonra sahne dekorumuz değişecektir.

Kuklayı kütüphaneden seç seçeneği seçilerek paddle isimli kukla sahneye eklenir.



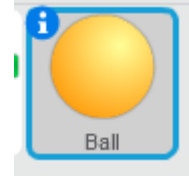
Oyun esnasında karakterimizin fare hareketiyle sadece sağa ve sola hareket etmesini sağlamak için

### Algoritma:

1. Farenin sağa sola hareketi ile karakterinki aynı olsun.



Bloğu eklenmelidir. Bu işlem sayesinde karakter fare hareketiyle sadece sağa ve sola(x eksenini) hareket edecek yukarı aşağı hareketi yapmayacaktır. Ekranı ayrıca bir top karakteri ekleyelim




Karakterin kod blokları

### Algoritma:

1. Sahnenin ortasına gidelim
2. Rasgele bir yöne dönelim
3. Biraz ilerleyelim
4. Duvarlara çarparsak geri dönelim
5. 3. Adımdan devam edelim



Yukarıdaki  bloğu sayesinde oyun her baştan başladığında top karakteri ekranın ortasından hareket etmeye başlayacak,



bloğu sayesinde program her başladığında karakter farklı bir yöne yönelecektir. Yeşil bayrağa basılırsa topun rasgele bir yöne yönelerek hareket ettiği görülecektir. Ancak paddle isimli değişkene çarptığında yönünün değişmediği daha doğrusu karakteri algılamadığı görülmektedir. Bunun için top(ball) karakterine ayrıca aşağıdaki bloklar eklenmelidir.

#### Algoritma:

1. paddle'a değdiğimizde
2. Geldiğimiz yöne geri dönelim



Burada dikkat edilirse eğer isimli bir koşul ifadesi kullanılmıştır. “Eğer ise” ifadesi bir koşulun sağlanıp sağlanmadığını belirlemek için kullanılır. Burada ise karakterin bir karaktere ya da belirlenen bir renge değip değmediğini algılamak için koşul ifadelerinden faydalanılmaktadır.

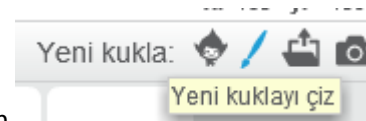
Yapı olarak incelenirse

#### Eğer koşulu gerçekleşirse

##### Yapılacak işlemler

Şeklinde çalışır. Şayet koşul sağlanmazsa eğer koşulu içindeki işlemler yapılmayacaktır.

Artık top(ball) karakteri paddle değişkenine her değdiğinde yönünü değiştirecektir. Son olarak ekranın altına kırmızı bir çizgi eklenerek kırmızı çizgiye çarptığında oyunun sonlanması sağlanmalıdır.



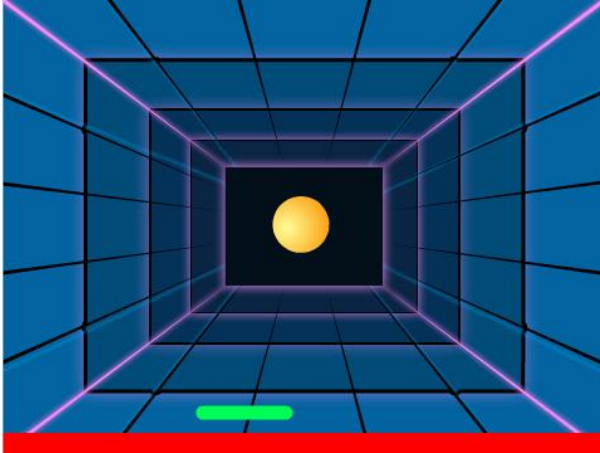
Bunun için yeni kukla çiz seçeneği seçilerek dikkörtgen aracıyla



Kırmızı ve içi dolu bir çizim oluşturularak



Sahneye eklenmelidir. Son durumda sahne



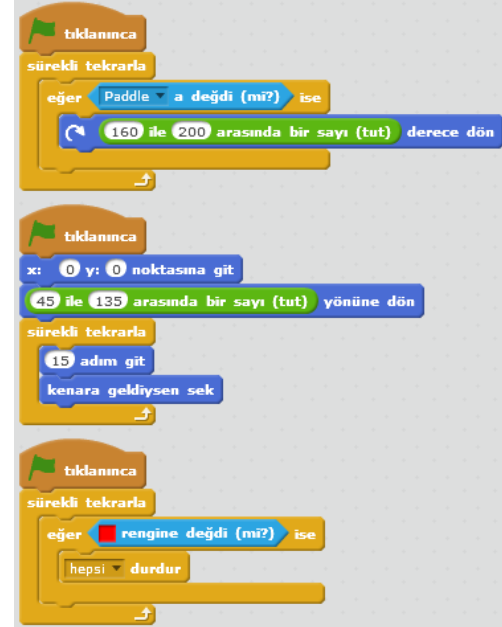
Şeklinde olmalıdır. Top(ball) karakterinin kırmızı çizgiye değdiğinde oyunun sonlanması için

## Algoritma:

1. Kırmızı renge değdiğimizde
2. Oyun bitsin



Blokları eklenmelidir.





## Örnek 12: Elma Toplama Oyunu

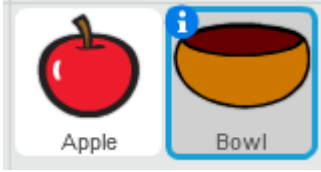
Anahtar kelimeler: y konumu, x konumu, süre değişkeni

Projenin tamamlanmış hali:

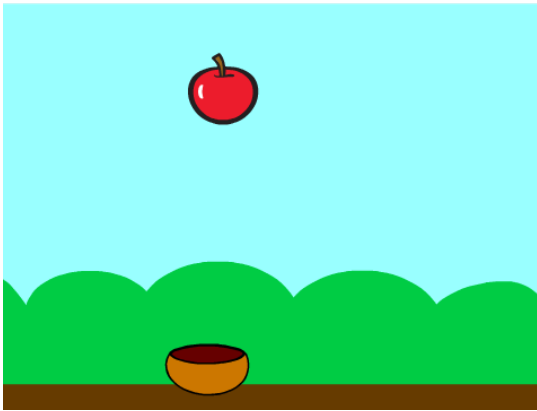
<https://scratch.mit.edu/projects/270539047/>

Neler Öğreneceğiz: Karakterin y konumunu ayarlamayı, y konumu belli bir seviyeye geldiğinde programın algılamasını, süre değeri oluşturmayı, süre belli bir değer olunca programı sonlandırmayı öğreneceğiz.

Örneğimizde ekranın üstünden farklı aralıklarla düşen elmaları toplama etkinliği yapacağız.



Öncelikle sahneye kütüphaneden yukarıdaki karakterleri ekliyoruz. Dilersek sahne dekorunu da değiştirebiliriz.



Elma karakterine hareket bloğundaki



Bloğuna tıklanarak



rasgele konum seçeneği seçilir.

Elma karakterine eklenecek kod blokları



şeklinde oluşturularak karakterin rasgele bir konuma gitmesi ancak y konumu yani yüksekliğinin 180 olması sağlanır. Yani bu iki blok çalıştığında karakter x ekseninde

herhangi bir yere y ekseninde ise 180 konumuna gitmiş olur. Hatta aynı işlem

x: -220 ile 220 arasında bir sayı (tut) y: 180 noktasına git

Bloğu ile de yapılabilir.

Karakterin y eksenindeki hareketi sürekli -5 azaltılarak y konumu -160 dan küçükse yani zemine değdi ise tekrar sahnenin üst tarafına konumlanarak aynı işlemi tekrar etmesi sağlanır.

Elma karakterinin sepete değdiğinde oluşturulan değişkenin değerinin sürekli artması ve tekrar ekranın üst kısmında konumlanması için

### Algoritma:

1. Puanımız 0 dan başlasın
2. Sahnenin en üstünden başlayarak yavaş yavaş aşağı düşelim,
3. Zemine değerse tekrar sahnenin üstüne gidip düşmeye başlayalım,
4. Sepetin içine düşerse puanımız



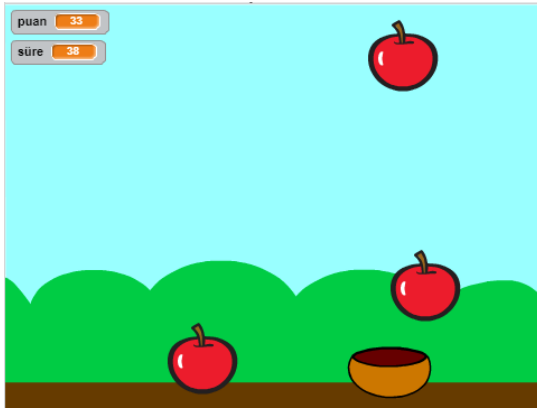
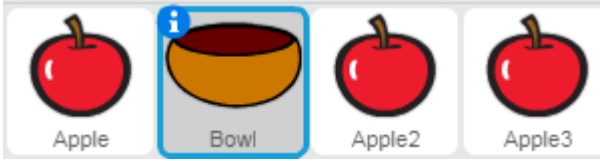
Şeklinde düzenlenmelidir.

Sepet karakterine eklenecek bloklar





Şeklinde olmalıdır. Bu sayede sepet karakteri klavye tuşları ile sağa sola doğru hareket edecek, süre değişkeni her saniye 1 artacak, süre 60 olduğunda tüm blokların hareketi sonlanacaktır. Elma karakterinin kopyası çıkarılarak oyunun zorluğu artırılarak daha eğlenceli hale getirilebilir.



## Örnek 13: Pinpon Oyunu

**Anahtar Kelimeler:** ... tuşuna basılı mı?, fare'nin y'si olsun, yeni kukla çiz,

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/270564794/>

Daha önce yaptığımız pong oyununda top kullanarak top sektirme işlemi yapmıştık. Bu örneğimizde ise karşılıklı iki karakter kullanarak topumuzu sektireceğiz. Öncelikle yeni kukla çiz seçeneği ile kırmızı renkli bir çubuk çizilir. Karakterin kopyası çıkarılarak rengi maviye boyanır ve 2a. Kukla oluşturulur.



Kırmızı kuklanın fare imleci ile beraber yukarı aşağı hareket etmesi için içerisine



Blokları eklenmelidir. Bu sayede karakterin y eksenindeki hareketi fare'nin y hareketine göre konumlanacaktır.



Kukla2

Mavi renkli kukla ise klavyedeki yukarı aşağı ok tuşlarıyla konumlanacaktır. Bunun için karaktere

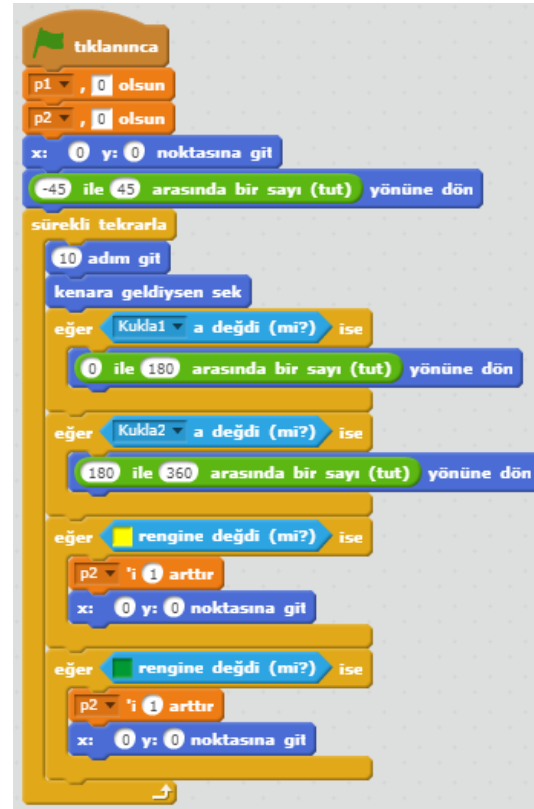
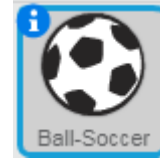


Blokları eklenmelidir.



Sahne dekoru çiz seçeneği seçilerek sahnenin kenarlarına sarı ve yeşil renkte kaleler çizilir.

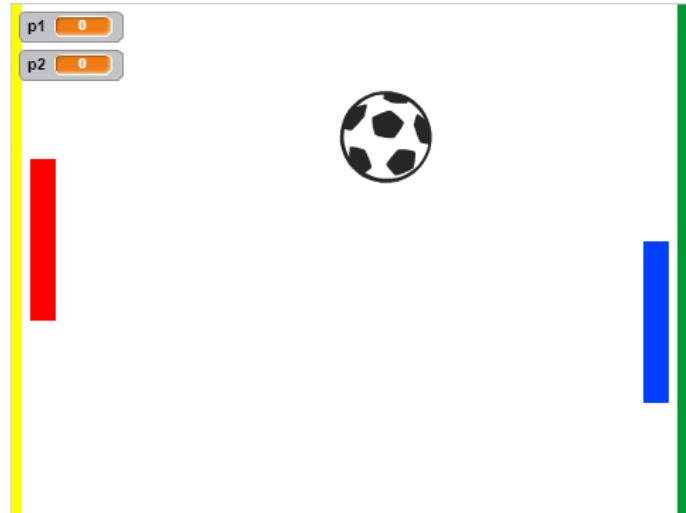
Ayrıca skor değerini tutmak için p1 ve p2 adında 2 adet değişken oluşturulmalıdır. Son olarak kütüphaneden ekleyeceğimiz top karakterine ekleyeceğimiz kodlar:



Şeklinde olmalıdır. Bu sayede yeşil bayrağa tıklanınca: p1 ve p2 skor değişkenleri 0 olacak, top orta noktadan başlayacak,

-45 – 45 arasında bir açı belirleyerek ona göre rasgele hareketine başlayacak  
Sürekli olarak hareket ederek kenara geldiyse sekecek,  
Kukla1'e değerse 0-180 arası bir değer üreterek o yöne dönecek,  
Kukla2 ye değerse 180-360 arası bir değer üreterek o yöne dönecek (kukla1 ve kukla2 ye değdiğinde farklı yönlere dönmesinin nedeni topun dönme açısı ve kuklaların farklı taraflarda olmasından dolayıdır.)  
Sarı renge değerse p2 1 artacak ve top karakteri sahne ortasına konumlanacak,  
yeşil renge değerse p1 1 artacak ve top karakteri sahne ortasına konumlanacaktır.

Oyun Ekranı



### Örnek 14 Akvaryum

**Anahtar kelimeler:** görün, gizlen, kılığı düzenle, veya operatörü

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/270655605/>

Örneğimizde akvaryumda kendi başlarına gezen minik balıkları yönettiğimiz köpekbalığı ile yakalamaya çalışacağız

Öncelikle su altı ile ilgili sahne yüklenir

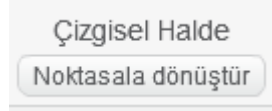


Ardından sahneye balık karakterleri eklenir. Şayet boyutları büyük gelirse

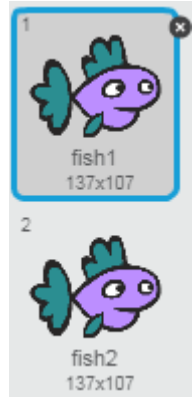
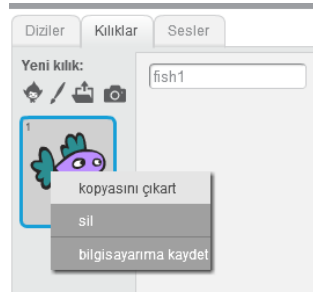


Komutları ile boyutu küçültülebilir.

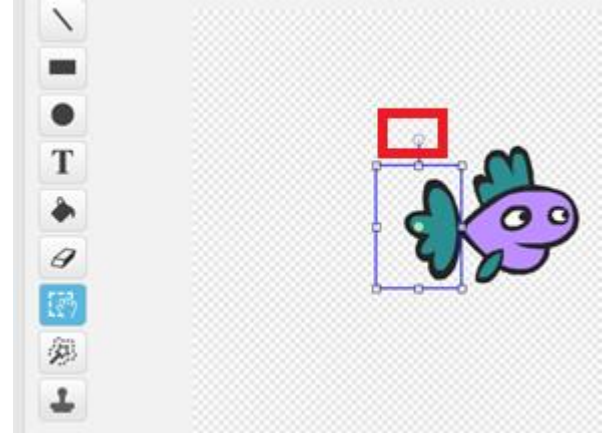
Kılıklar sekmesinde balığın 2. Kostümü yoktur. Hareket özelliği kazandırmak için üzerinde küçük bir düzenleme yapmak gereklidir. Kılıklar sekmesinde karakter seçili iken sağ alt tarafta bulunan **noktasala dönüştür** seçeneği seçilir.



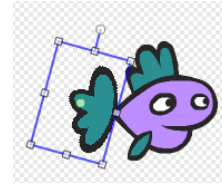
Ardından karaktere sağ tıklanarak kopyası oluşturulur.



Şimdi oluşturduğumuz kopya üzerinde değişiklik yapalım. Seçim aracı seçilerek ikinci kılığın kuyruk kısmı seçilir.



Seçilen alanın üst kısmındaki kırmızı çizgi içindeki alan seçilerek sağa ya da sola doğru oynatılır gerekirse ortalama yapılır.



Artık iki farklı kılığa sahip kuklamız oluşmuştur.



Karakterin içine yazılacak kod blokları

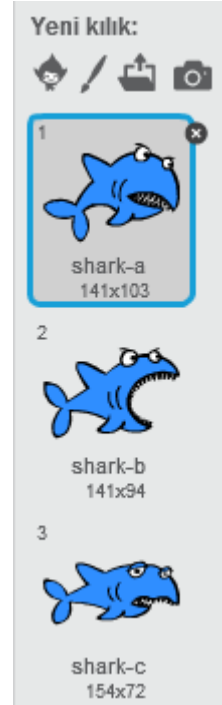


Şeklinde olmalıdır. Yeşil bayrağa tıkladığında:  
15 derece dön bloğu ile karakterin düz bir şekilde değil belirli bir açıyla hareket etmesi,  
Boyutunun küçültülmesi,  
Sürekli olarak ilerleyerek kenara gelince sekmesi  
Kuklanın şekli sağa sola dönerek ters dönmesinin önüne geçilmesi,  
0.2 saniye aralıklarla kılığının değişmesi sağlanır.

Aynı işlemler farklı balık karakterlerine de yapılabilir.



Şimdi ise sahneye bir köpek balığı ekleyerek balıkları avlamasını sağlayalım.



Eklediğimiz karakterin 3. Kılığını silelim.

Köpek balığının fare ile aynı konuma gitmesi ve balıkların üzerine gelince ağzını açıp kapama hareketi yapması için



Blokları eklenmelidir. Bu sayede karakter fare ile aynı konuma gidecek, balık karakterine değdiği zaman ağzını açıp kapatarak yeme hareketi yapacaktır. Sağ taraftaki blok her balık için ayrı ayrı oluşturulabileceği gibi,



Şeklinde bir yapı da kullanılabilir.

Artık köpek balığı yeme işlemi yaptığımıza göre balıklara kaybolma efekti ekleyebiliriz.

Her bir balığa aşağıdaki kodlar eklenerek köpek balığına değdikten 1 saniye sonra gizlenmesi 2 saniye sonra ise sahnenin herhangi bir yerinde tekrar görünmesi sağlanabilir. Bunun için her bir balık karakterine,



Blokları eklenmelidir. Bu sayede karakter köpekbalığı karakterine değdiği zaman 1 saniye bekleyip gizlenecek, 2 saniye sonra ise sahnenin herhangi bir yerinde tekrar görünecektir.





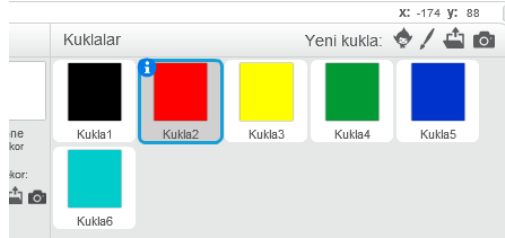
## Örnek 15: Paint uygulaması

Anahtar kelimeler: haberini sal, haberi gelince, fareye basılı mı , fare'nin x'i, küçüktür - büyüktür operatörleri, ve operatörü

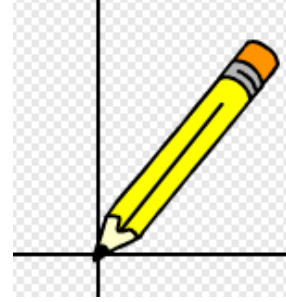
Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/270978041/>

Örneğimizde scratch programında paint uygulaması yapacağız. Öncelikle kullanacağımız renkler için renkli kuklalar oluşturmalıyız.



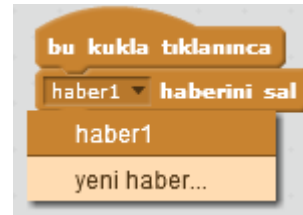
Ardından kalem karakteri ekleyip kılık merkezi olarak kalem ucunu belirliyoruz.



Kalem tıklanan rengi algılayıp o renkte çizim yapması için her bir renk paletine olaylar sekmesinden



Blokları kullanılacaktır. Her bir kukla için **yeni haber sal** özelliği eklenerek



## kodlama serüveni

Karaktere basılıp basılmadığı bu şekilde kontrol edilecektir. Yeşil kukla için



bloğu eklenerek her karakter(renk) için aynı işlem yapılır.

Kalem içine eklenecek kodlar



Şeklinde hazırlanarak verilen haber durumlarına göre renkler ayarlanır. Ayrıca kaleme basıldığında sahneye tıklanan rengin çizilmesi için



Blokları eklenir. Bu sayede yeşil bayrağa tıklandığında

ekran temizlenir,

kalem kalınlığı 5 yapılır,

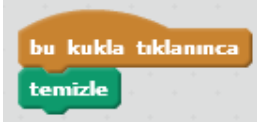
fareye basılı ise **ve** fare'nin x konumu 90 küçükse çizim işlemine başlanır. Bu sayede ekranın sağındaki renk kutucuklarının olduğu yerde kalem konumlanmayacak ve çizim yapmayacaktır.

Ve operatörü ile iki durumun aynı anda gerçekleşmesi sağlanmaktadır.

Sahne renk kuklalarının olduğu yere bir arkplan ve çizilebilir.

Ayrıca eklenecek bir butonla ekranın temizlenmesi sağlanabilir.





Bloğu eklenir.



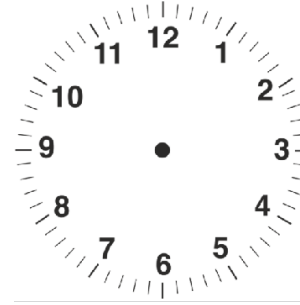
## Örnek 16: Duvar Saati

Anahtar kelimeler: Şimdiki saniye, şimdiki dakika, şimdiki saat, üste çık

Projenin tamamlanmış hali:

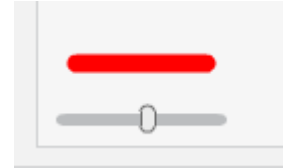
<https://scratch.mit.edu/projects/271002235/>

Scrath programında gerçek zamanlı saat uygulaması yapılabilir. Bunun için sahne dekoru olarak stage.png isimli sahne dekoru olarak yüklenmelidir.



Sahneye farklı büyüklük ve renklerde 3 adet ok çizilir. Saniye uzun, saat kısa olmalıdır.

Çizginin kalınlığı artırılmalıdır.



Öncelikle eklenen kuklaların kılık merkezi kontrol edilmeli, her bir kuklanın kılık merkezleri



Şeklinde ayarlanmalıdır.



Her birine isim verilmesinin faydası olacaktır.



Saat karakterine eklenecek kodlar



Şeklinde olmalıdır. Burada kullanılan şimdiki saat bloğu saat bilgisini bilgisayardan alıp ekranda gösterilmesi için kullanılır. Kullandığımız kuklanın saat verisine göre dönmesi için



İşlemine yapıyoruz. Analog saatimizde toplam 12 saat olduğuna göre, ayrıca kuklanın 360 derece döndüğü varsayılırsa  $360/12=30$  sonucu bulunur. Bu nedenle yukarıdaki kod bloğu kullanılarak şimdiki saat verisi ile 30 çarpılarak saatin doğru yeri göstermesi sağlanır.



Dakika isimli kuklaya ise



## kodlama serüveni

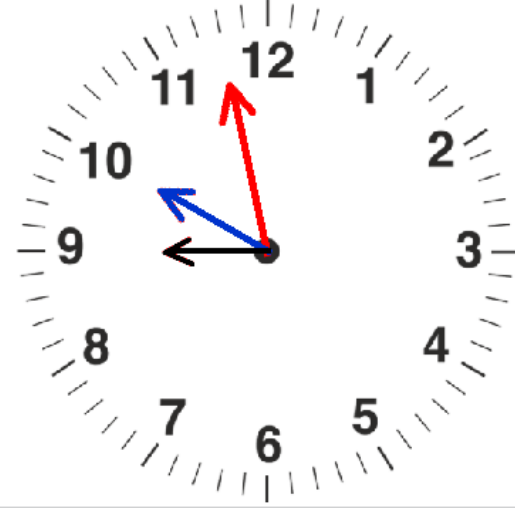
Blokları eklenir. Saat üzerinde 60 dakika olduğu düşünürse  $360/60=6$  formülü ile her dakika için dakika değeri için 6 ile çarpmak gereklidir.



Saniye isimli kukla için



Blokları eklenerek aynı dakika kuklasındaki gibi  $360/60=6$  formülü ile her saniye için 6 ile çarpmak gereklidir.



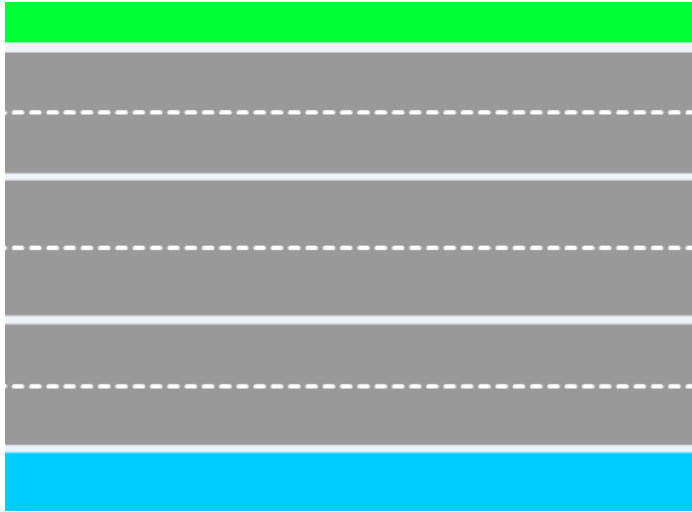
## Örnek 17: Karşıya Geçme Oyunu

Anahtar Kelimeler: olana kadar tekrarla, hepsi durdur.

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/271298845/>

Örneğimizde karakterin akan trafikte karşıya geçmesini sağlayacağız. Sahneye dekor eklenir



Scratch kütüphanesinde bulunan arabalardan istenilen kadarı kukla olarak eklenir.



Her birinin içerisine aşağıdaki bloklar eklenir.



Her araç için



Bloğu ile başlangıç noktaları farklı olarak ayarlanır.

Ayrıca



bloğu ile her aracın hızı farklı olarak ayarlanır.



Kütüphaneden greenflag isimli karakter eklenerek sahnenin alt tarafına yerleştirilir.





Kütüphaneden kurbağa karakteri eklenerek içerisine

```
when green flag clicked
  set can to 3
  set size to 50%
  go to x: 5 y: 167
  loop
    if Car-Bug is clicked or Convertible1 is clicked
      increase can by 1
      go to x: 5 y: 167
    if yellow key is pressed
      increase y by 3
    if down key is pressed
      increase y by -3
    if right key is pressed
      increase x by 3
    if left key is pressed
      increase x by -3
    if Green Flag is clicked
      wait 1 seconds
      stop all
```

İçerisine blokları eklenir.  
Bu sayede can isimli değişken 3 yapılır,  
Büyüklüğü yüzde 50 yapılarak küçültülür,

Başlangıç noktası ayarlanır,  
Herhangi bir araba karakterine değdiği zaman can  
değişkeni 1 azalarak başlangıç noktasına döner,  
Yön tuşlarıyla hareketi sağlanır,  
Greenflag isimli bayrağa değdiğinde 1 saniye boyunca  
kazandın diyerek tüm bloklar durdurulur.

Her bir araç karakteri içine veya bloğu ile

```
if Bus is clicked or Car-Bug is clicked or Convertible2 is clicked
```

Blokları yerleştirilir. Bu sayede karakterin araçlara değdiği  
zaman başlangıç noktasına dönmesi sağlanır.

Şayet karakter araçlara 3 defa değerse oyun kaybedildi  
şeklinde uyarı verilir, karakter yeşil bayrağa(greenflag)  
ulaştığında kazandın şeklinde uyarı vererek oyun sonlanır.



## Örnek 18: Düşen toplar

Anahtar kelimeler: Kopyasını çıkart, .. dekoruna geç

Projenin Tamamlanmış Hali

<https://scratch.mit.edu/projects/273465076/>

Oyunumuzda havadan düşen toplardan kaçmaya çalışacağız.

Öncelikle

Sahneye bir adet dekor ekleyelim

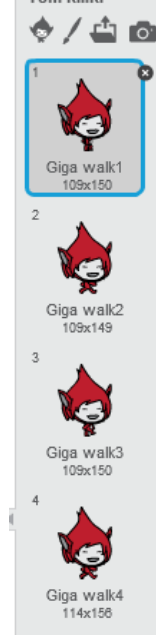


sahneye karakterimizi ekleyelim



Giga walking

Kılıklar sekmesinde karakterin yürüme efekti için gerekli kılıklar mevcuttur.



Karakterin klavyedeki yön tuşlarıyla sağa sola hareket edebilmesi ve yürümesi için gerekli kod bloklarını ekleyelim.





Sahneye top karakteri ekleyelim.

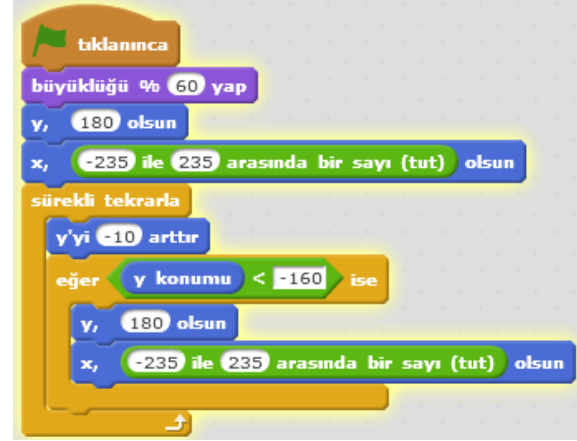


Karakterin ekranın üst konumuna yerleşmesi ve rasgele bir konumdan düşmesi için



Blokları eklenmelidir.

Top karakterinin yere düştüğünde tekrar ekranın üstünden düşmesini sağlamak için bloklar



Şeklinde düzenlenmelidir.

Kuklaya can vererek 3 defa top değdiğinde oyunun sonlanması için can isimli bir değişken oluşturulup bloklar

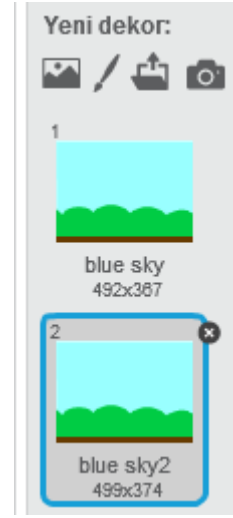
```
when clicked on the green flag
  can = 3
  size = 60
  y = 180
  x = random number between -235 and 235

  repeat (forever)
    y = y - 10
    if (y < -160)
      y = 180
      x = random number between -235 and 235

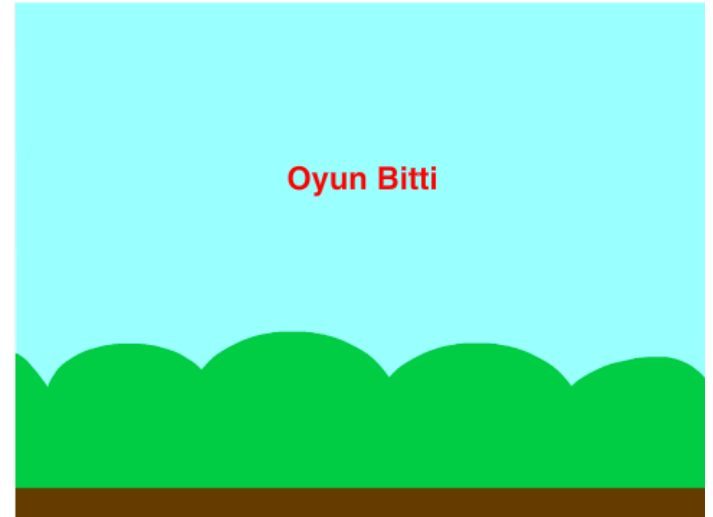
    if (Giga walking2 is changed)
      can = can - 1
      y = 180
      x = random number between -235 and 235

    if (can = 0)
      say goodbye
      stop all
```

Şeklinde düzenlenmelidir. Yandın haberi gelince sahne dekoru çoğaltılarak



Kopyalanan dekora

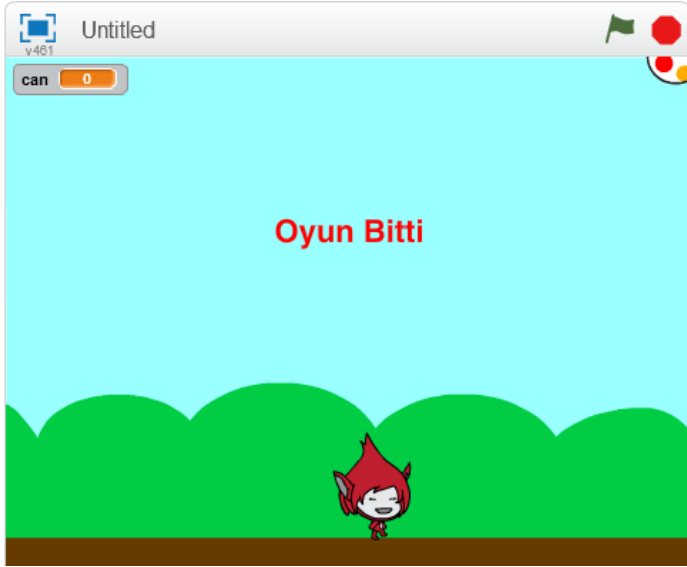


Yazısı yazılmalı, ayrıca sahneye



Blokları eklenmelidir. Oyun başladığında 1. Dekorla başlayacak can bittiğinde ise 2. Dekora geçilecektir.

Artık karakterin canı bittiğinde ekranda **oyun bitti** mesajı yazacaktır.



Şeklinde düzenlenmelidir.

Top kuklasının üzerine sağ tuşa basarak **kopyasını çıkart** seçeneği ile top sayısını 3'e çıkararak oyun zorluğunu artırabiliriz.



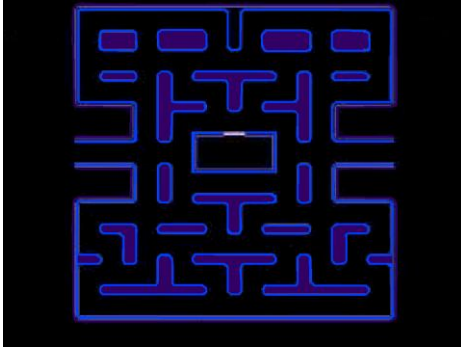
## Örnek 19: Pacman oyunu

**Anahtar Kelimeler:** rengi rengine değdi mi,

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/271920796/>

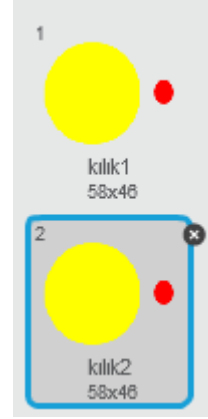
Bu örneğimizde pacman oyunu yapacağız. Arama motoruna pacman scratch yazılarak boş ve uygun bir sahne seçilir. Skor değişkeni olacağı için seçeceğimiz sahenin kenarlarında bir miktar boşluk olmasının faydası vardır.



Kaydettiğimiz arka planı sahneye ekliyoruz. Ardından karakterimizi **yeni kukla çiz** seçeneği ile çiziyoruz. Çizdiğimiz karakterin önüne minik kırmızı bir daire çiziyoruz.



İçi dolu sarı bir daire oluşturmamız yeterlidir. Oluşturduğumuz kırmızı daire ile kuklanın duvarlara çarpıp çarpmadığını kontrol edeceğiz. Ardından kuklanın kılığının kopyasını çıkarıp



2. kılığ silgi ile düzenleyerek ağız yapıyoruz.



Çizdiğimiz kukla muhtemelen labirente göre büyük geleceğinden boyutunu biraz küçültmemiz gerekmektedir. Ayrıca yeşil bayrağa tıkladığında belirli bir konumdan başlaması için ve duvarlara değmediği sürece hareket etmesi için kuklanın kodlarını



Blokları eklenir. Bu sayede karakterin önündeki kırmızı nokta labirentin siyah zeminine temas ettiği süreçte karakter hareket edecek mavi duvarlara değdiği zaman duracaktır.

Yön tuşlarına basıldığında ise kuklanın yönünün dönmesi için blokların



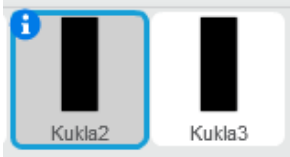
Şeklinde düzenlenmesi gereklidir. Artık kuklamız hem duvara çarpmadığı sürece hareket edebilecek, hem de bastığımız tuşun yönüne dönebilecektir. Ayrıca karaktere ağız açma kapama hareketi eklemek için



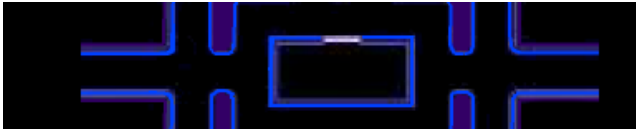
Blokları eklenmelidir.



Bu bölümde karşı tarafa geçmesi için ise 2 tane karakter çizip

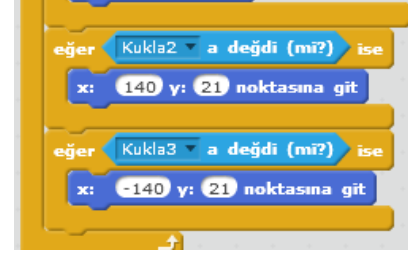


Karşılıklı kenarlara koymak gereklidir.

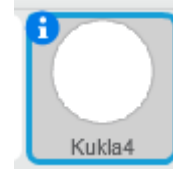


oluşturduğumuz karakterler burada

Kuklanın oluşturduğumuz karaktere değince ekranın diğer tarafından çıkması için gerekli olan bloklar ana bloğun altına eklenmelidir.



Karakterin ekrandaki hedefleri toplaması için



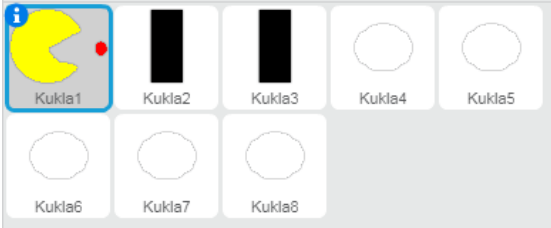
Yeni bir kukla çizilerek içine

```

tıklanınca
  puan, 0 olsun
  görün
  büyüklüğü % 40 yap
  sürekli tekrarla
    eğer (Kukla1 a değdi (mi?)) ise
      gizlen
      puan, 1 arttır
  
```

Blokları eklenmelidir. Bu sayede kukla 1 karakteri hedefe değdiği zaman hedef ekrandan kaybolacak ve puan değeri 1 artacaktır.

Skor koyduğumuz hedef sayısı olduğunda programın sonlanması için son olarak kukla1 in içine(5 adet hedef olduğunu varsayarsak)



Kukla1 karakterine

```

x: -140 y: 21 noktasına git
eğer (puan = 5) ise
  kazandın de 1 saniye
  hepsi durdur

```

Bloğu eklenmelidir.

```

tıklanınca
  büyüklüğü % 20 yap
  x: 20 y: -136 noktasına git
  sürekli tekrarla
    eğer (renge rengine değdi (mi?)) ise
      2 adım git
    eğer (yukarı ok tuşu basılı (mi?)) ise
      0 yönüne dön
    eğer (aşağı ok tuşu basılı (mi?)) ise
      180 yönüne dön
    eğer (sol ok tuşu basılı (mi?)) ise
      -90 yönüne dön
    eğer (sağ ok tuşu basılı (mi?)) ise
      90 yönüne dön
    eğer (Kukla2 a değdi (mi?)) ise
      x: 140 y: 21 noktasına git
    eğer (Kukla3 a değdi (mi?)) ise
      x: -140 y: 21 noktasına git
  sürekli tekrarla
    sonraki kılık
    0.2 saniye bekle
    eğer (puan = 5) ise
      kazandın de 1 saniye
      hepsi durdur

```



## Örnek 20: İkizini oluşturma

Anahtar kelimeler: ikizini oluştur, ikiz olarak başladığında

Projenin Tamamlanmış Hali:

<https://scratch.mit.edu/projects/272257753/>

Yapacağımız oyun ile ikizini oluştur özelliğini kullanacağız.

Öncelikle sahneye bir arka plan yükleyelim.



Sahneye karakterimizi yüklüyoruz,



Karakterimizin boyutunu küçülterek sürekli koşma hareketi veriyoruz



Karakterimizin önüne engeller geldiği zaman üzerinde zıplaması için



Bloklarını eklemliyiz. Burada kullandığımız



Blokları sayesinde karakterimizin yavaş bir şekilde yükselmesini ve alçalmasını sağlarız. Bunun yerine



Bloğunu kullansaydık karakter bir anda yükselecek ve biz zıplama hareketini göremeyecektik.

Sahneye bir ağaç kuklası ekleyelim.





Burada yapacağımız işlemle kuklanın klonu oluşturularak belirli sürelerle karakterin üzerine girmesini sağlayacağız. Bunun için



Bloklarını ekliyoruz. Bu sayede oyun başladığında karakterin gizlenmesini, karakterin boyutunun yüzde 60 olmasını, 1 -10 saniye arasında karakterin ikizinin oluşmasını sağlıyoruz.

Oluşan ikizin görünerek ekranın sağından soluna doğru hareket etmesi için



Bloklarını eklemeliyiz. Bu sayede karakter belirli aralıklarla görünecek ve pico walking karakterinin üzerine doğru gidecektir. Pico karakterinin ağaç karakterine 3 kere değdiğinde oyunun sonlanması için bir değişken oluşturarak



```

tıklanınca
  can , 3 olsun
  sürekli tekrarla
    eğer Tree1 a değdi (mi?) ise
      can 'i -1 arttır
      1 saniye bekle
    eğer can = 0 ise
      kaybettin de
      hepsi durdur

```

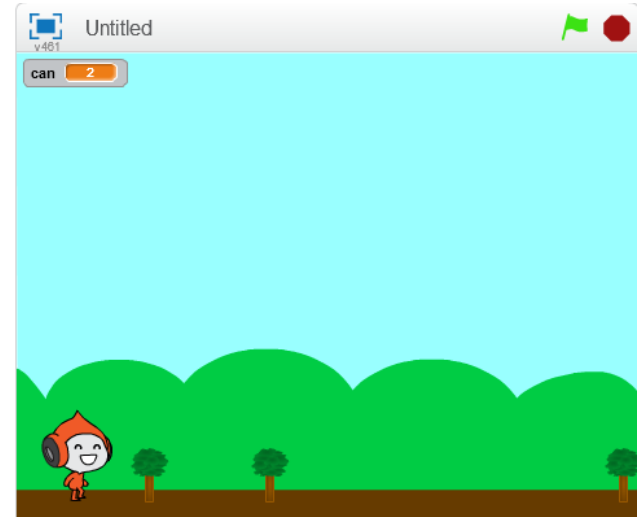
Blokları eklenmelidir.



```

tıklanınca
  x: -194 y: -130 noktasına git
  büyüklüğü % 50 yap
  sürekli tekrarla
    sonraki kıkık
    0.1 saniye bekle
  sürekli tekrarla
    eğer boşluk tuşu basılı (mi?) ise
      25 defa tekrarla
        y'yi 4 arttır
      25 defa tekrarla
        y'yi -4 arttır
tıklanınca
  can , 3 olsun
  sürekli tekrarla
    eğer Tree1 a değdi (mi?) ise
      can 'i -1 arttır
      1 saniye bekle
    eğer can = 0 ise
      Kaybettin de 1 saniye
      hepsi durdur

```



## Örnek 21: Engelli Yol

Anahtar kelimeler: eğer ise - değilse

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/273427192/>

Örneğimizde sahnede hareket halindeki aracımızı engellerden kaçırarak puan toplamaya çalışacağız.



Sahneye karakterlerimizi ekledikten sonra araba karakterinin içine



Blokları eklenir. Bu sayede sırayla:

Karakterin boyutu küçültülür,

Başlangıç noktasına gitmesi sağlanır.

90(yukarı) yönüne dönmesi sağlanır,

Sol ve sağ ok tuşuna basılınca x değeri artırılıp azaltılır ve sağa dola gitmesi sağlanır. 80,100 yönüne dön komutu ile dönüşlerde aracın kavis alması sağlanır.

Sağ ve sol kolona değdiğinde x değerleri değiştirilerek yoldan çıkması engellenir.

Engellere değdiğinde ise oyununu sonlanması sağlanır.



Engel1 karakterinin içine ise



Blokları eklenerek;

Büyüklüğü %60 yapılır,

Burada kullanılan eğer ise – değilse bloğu ile oyunun 1. Eğer koşulunda 1 ile 3 arası bir sayı tutması sağlanır, tuttuğu sayı 3 ise 0-173 konumuna gitmesi sağlanır, eğer 3 sayısı tutulmadı, 2 sayısı tutuldu ise -50 – 173 konumuna gitmesi, 1 sayısı tutuldu ise 50 – 173 konumuna gitmesi sağlanır.

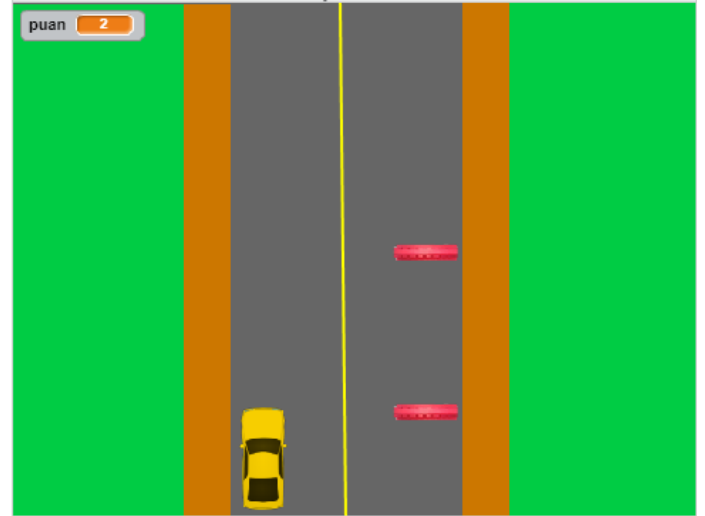
Kenara değene kadar karakterin y değeri -8 azaltılır ve ekranın en altına gelip kenara değdiğinde yukarıdaki işlemler tekrarlanır.

2. engele ekstra olarak



Blokları eklenerek 1. Engelden 1 saniye sonra başlaması sağlanır.

Ayrıca araba karakterinin içine süre isimli değişken oluşturularak puan toplanması sağlanır. Oyunu zorlaştırmak için ekstra engeller eklenebileceği gibi engellerin hızları da değiştirilebilir.



## Örnek 22: Özel Taşlar

**Anahtar kelimeler:** özel taşlar, bir taş oluştur, rakamsal girdi ekle

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/271928930/>

Scratch programlarının daha sade olması için özel taşlar kullanılmaktadır. Bu sayede bloklarımızı daha işlevsel hale getirebiliriz.

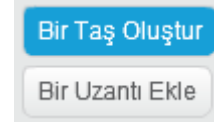
Programımızda basit bir oyun yaparak karakterimizin üzerine gelen engellerin üzerinden atlamasını sağlayacağız. Öncelikle kedi karakterimize yürüme efekti vermek için



Bloğunu ekleriz. Özel taşı oluşturmak için özel taşlar bloğu seçilerek



Bir taş oluştur seçeneği seçilir.



Oluşturulacak taşın ismi girilir



Artık yeni taşımız oluşmuş ve kodlama sahnesine gelmiştir. Şu an için içerisinde herhangi bir komut olmadığı için çalışmaz. Ayrıca içerisinde komut olsa bile çalışması için çağırılması gereklidir.



Taşımız çağırıldığında karakterimizin zıplama hareketi yapmasını sağlayalım.

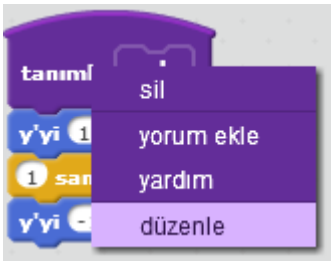
Bunun için yukarı ok tuşuna basıldığında yapılacak işlem için özel taşlarda oluşturduğumuz taşın bloğunu ekleyelim



Artık yukarı tuşuna basıldığında **zıpla** isimli özel taş çağırılacaktır. Son olarak özel taşımıza zıplama efekti vermek için gerekli blokları ekleyelim.



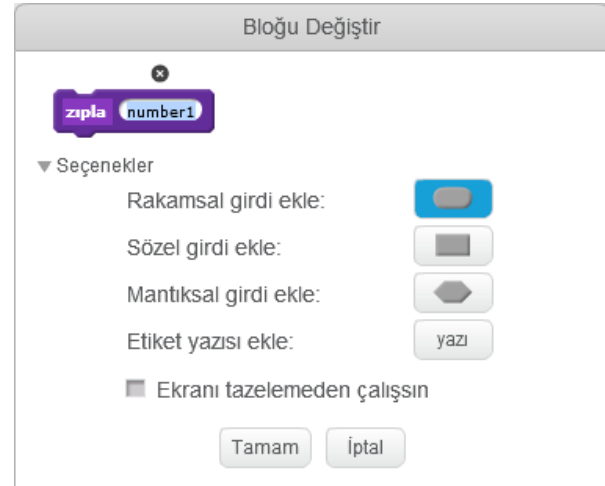
Oluşturduğumuz taşlara yeni özellikler ekleyerek daha işlevsel hale getirebiliriz. Bunun için taşın üzerine sağ tuşa basarak düzenle seçeneği seçilir.



Açılan pencereden **seçenekler** seçeneği seçilir.



**Rakamsal girdi ekle** seçeneğinin yanındaki kutucuğa tıklanır.



Artık taşımızın yanında **number1** isimli bir değer alanı oluşmuş oldu. Bu alana **yükseklik** yazarak **tamam** düğmesine basalım


zıpla yükseklik

Artık taşımızın yeni bir özelliği olmuş oldu.



Şimdi y değeri için taşımızın yeni özelliğini kullanalım. Yükseklik yazan bloğu sürükleyerek y'yi .. arttır bloğunun içine yerleştirelim



düşme hareketi için ise  bloğunu kullanalım.



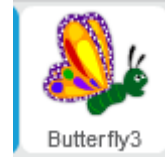
Yukarı ok tuşuna basılınca bloğunu da düzenlemeyi unutmamalıyız.



Peki bu durumun bize ne faydası olacak. Çok kısa eklemeler yaparak farklı tuşlara farklı özellikler ekleyebiliriz.



Artık sadece değer göndererek farklı birimlerde zıplama yapmasını sağlayabiliriz. Şimdi karakterimizin üzerine bir engel gönderelim zıplamasını sağlayalım. Sahneye herhangi bir karakter ekleyelim



İçine yazacağımız kodlar



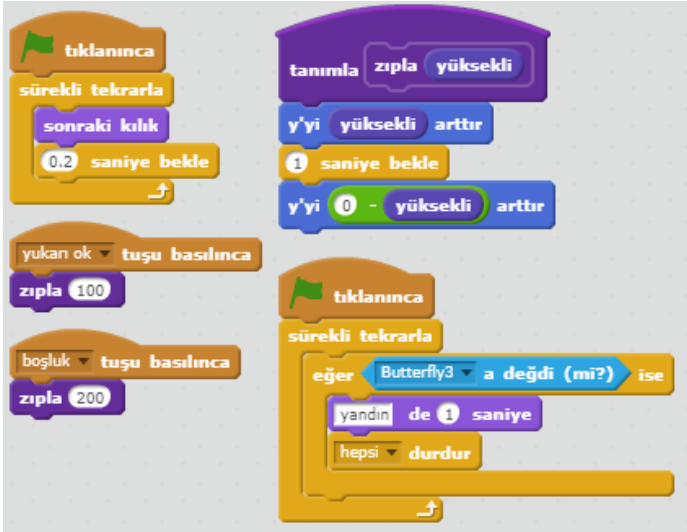
Şeklinde olmalıdır. Karakter devamlı kedi karakterine doğru gidecek ekranın sonuna ulaştığında başa dönecektir.

Kedi karakterinin üzerine gelen diğer karaktere değdiğinde oyunun bitmesi için ise



```
when green flag clicked
  loop
    if Butterfly3 is touching a
      wait 1 second
      stop everything
```

Blokları eklenmelidir.



```
when green flag clicked
  loop
    next tick
    wait 0.2 seconds
    set jump to 100
    set jump to 200
    define jump as high
    increase jump by high
    wait 1 second
    increase jump by 0 - high
    if Butterfly3 is touching a
      wait 1 second
      stop everything
```





## Örnek 23: Flappy Bird

**Anahtar kelimeler:** Fareye basılı (mı), ... değil olana kadar bekle, kendim'in ikizini yarat, ikiz olarak başladığında, bu ikizi sil

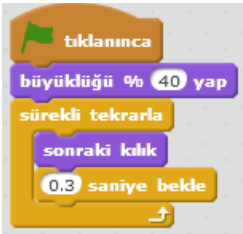
**Projenin Tamamlanmış Hali:**

<https://scratch.mit.edu/projects/272115276/>

Bu oyunumuzda herkesin bildiği flappy bird oyununu yapacağız. Öncelikle sahnemize yarası karakteri ekleyelim.



Karakterimize kanat çırpma hareketi verelim. Karakterimizin boyutunun büyük olduğunu varsayarak boyutunu küçütelim.



Karakterin sahnenin ortasından başlayarak yavaş yavaş aşağı doğru süzülmesini sağlamak için



Bloklarını ekleyelim. Artık karakterimiz yavaş yavaş ekranın altına doğru kayacaktır. Fareye basıldığında bir miktar yükselmesini sağlamak için



Bloğu eklenmelidir. Ancak burada şöyle bir durum vardır ki fareye basılı tutulursa karakter sürekli yükselme hareketi yapmaya çalışacaktır. Bu durumun önüne geçmek için bloklarımızı



Şeklinde düzenlemeliyiz. Bu sayede karakter fare'ye tıklanınca bir miktar yükselecek daha sonra kullanıcının fare'den elini çekmesi beklenecektir.

Sahneye boru kuklası eklenerek,



Yarasa karakterinin borulara veya ekranın altına düştüğünde oyunun sonlanması için bloklar,



Şeklinde düzenlenmelidir.

Ekrandaki boruların belirli sürelerle ekranda görünmesi için boru kuklasının içine



Blokları eklenmelidir. Bu sayede kodlarımız çalıştığında borular kuklası nerede olursa olsun gizlenecek bu sayede sahnenin boş başlaması sağlanacaktır. Ayrıca her 8 saniyede bir yeni bir ikiz oluşacaktır. Kontrol sekmesindeki



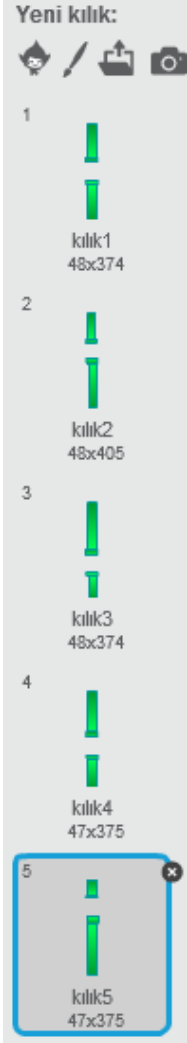
Bloğunu kullanarak



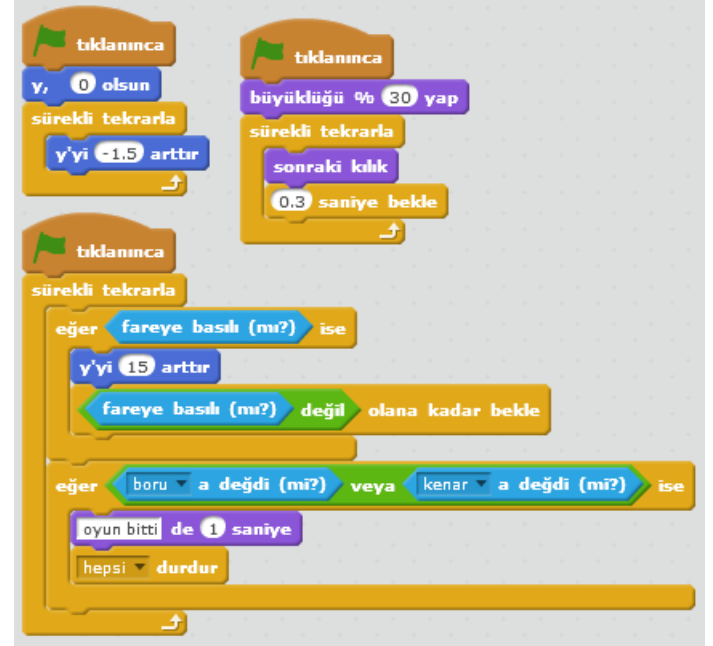
Bloğu oluşturulmadır. Bu sayede boru karakterinin 5 kılığından biri rasgele seçilecek, ekranın sağından

başlayarak sola doğru kayacak ekranın en soluna geldiğinde ise kaybolacaktır.

Boru karakterinin kılıklarına bakıldığında içerisinde



Kılıklarının olduğu görülür.





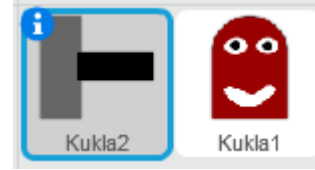
## Örnek 24: Çekiç oyunu

Anahtar kelimeler: Üste çık, 10 katman alta al

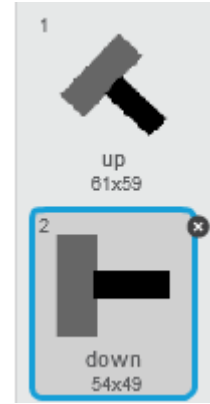
Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/272201326/>

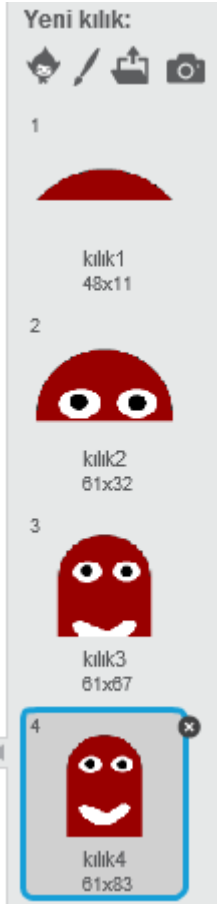
Kuklayı bilgisayardan seç seçeneği ile kuklalarımızı sahneye ekliyoruz.



Çekiç karakterinin 2 adet kılığı



Kukla1 karakterinin ise 4 adet kılığı bulunmaktadır.



Çekiç karakterine eklenecek kod blokları:



Şeklinde yapılarak karakterin fare ile aynı konuma gitmesi sağlanır.



Bloğuyla çekiç karakterinin kukla1 karakterinden alt katmanında olmaması sağlanır. Aksi halde çekiç karakteri kukla karakterinin altında kalabilir.



Ayrıca fareye tıkladığında çekiç karakterine aşağı inme hareketi vermek için



Bloğu eklenir. Bu sayede fareye tıklanmadığı zaman küçük karakteri up kılığına tıkladığında ise down kılığına geçer. Fare basılı olduğu süreçte karakter down kılığında kalacaktır. Ayrıca tıklama işlemi yapıldığında bas haberi alınır.



Kukla 1 karakterine eklenecek kodlar.



Şeklinde olmalıdır. Karakter 5 saniyelik aralıklarla görünecek ve gizlenecektir. Ayrıca yukarı ve aşağı isiminde iki özel taş oluşturulmuştur.

**yukarı** isimli özel taş



Özel taş çağırıldığında karakter ekranın herhangi bir noktasına konumlanarak, görünecek ancak karakterin 4 adet kılığı olduğu için bu işlemi aşağıdan yukarı çıkıyormuş gibi yapacaktır.

**Aşağı** isimli özel taş çağırıldığında



Karakter ekrandan kaybolacaktır.

Ayrıca çekiç karakterinden gönderilen bas haberi gelince



Yani çekiç karakteri ile kuklaya vurulunca puan değışkeni 1 artacak ve aşağı isimli özel taş çağırılarak karakter gizlenecektir.



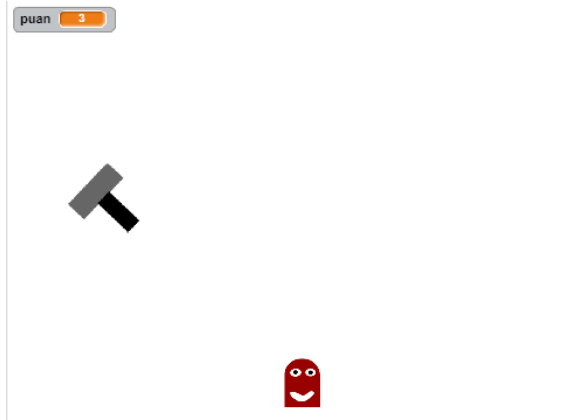
```

    tıklanınca
    büyüklüğü % 50 yap
    gizlen
    1 ile 5 arasında bir sayı (tut) saniye bekle
    yukarı
    1 ile 5 arasında bir sayı (tut) saniye bekle
    aşağı

    bas haberi gelince
    eğer Kukla2 a değdi (mi?) ise
    puan '1 arttır
    aşağı
    1 ile 5 arasında bir sayı (tut) saniye bekle
    yukarı

    tanımla aşağı
    0.1 saniye bekle
    kalık3 kılığına geç
    0.1 saniye bekle
    kalık2 kılığına geç
    0.1 saniye bekle
    kalık1 kılığına geç
    gizlen

    tanımla yukarı
    rastgele konum 'na git
    kalık1 kılığına geç
    görün
    0.1 saniye bekle
    kalık2 kılığına geç
    0.1 saniye bekle
    kalık3 kılığına geç
    0.1 saniye bekle
    kalık4 kılığına geç
    
```



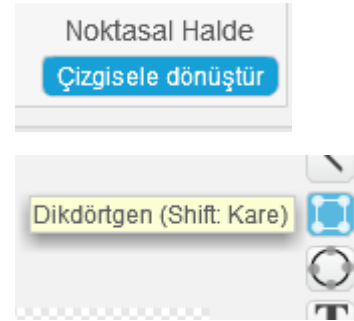
## Örnek 25: Yılan Oyunu

**Anahtar kelimeler:** hayalet etkisi .. olsun, hayalet etkisini .. arttır.

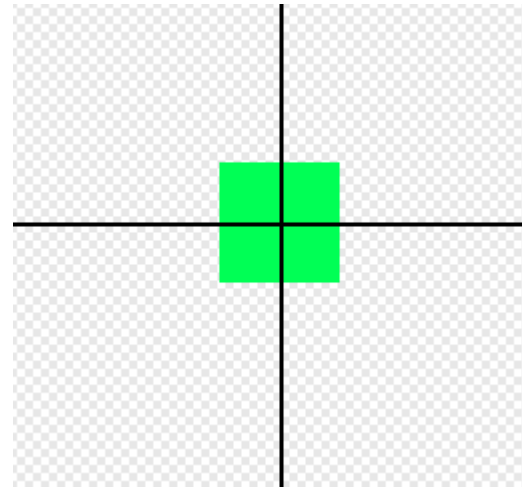
**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/273068743/>

Öncelikle yeni bir kukla oluşturulur. Bunu yaparken çizgisele dönüştür seçeneği seçilir.



Kalık merkezi olarak çizilen karakterin ortası seçilir







Karakterin ismini **“yılan”** olarak değiştirelim. Karakterin klavyeden basılan yön tuşlarının yönüne dönerek sürekli hareket etmesi için ayrıca karakterin hızını değişken ile belirlemek için hız isimli bir değişken oluşturulur.



Blokları eklenir.

Karakterin ilk hareket ettikten belli bir süre arkasında klonunu oluşturup karakteri takip etmesini sağlamalıyız. Bunun için



Bloğunu kullanarak karakterin ikizini yaratmalı ardından,



Kodlarıyla ikizin oluşturulma süresini belirlemeliyiz. Buradaki saniye bekle komutu oluşturulacak ikizin uzunluğunu belirler.



Klon değişkeninin süresi ne kadar uzun tutulursa karakterin kuyruğu da o kadar uzun olur.

Ekranın rasgele bir konumunda yiyecek çıkıp karakterin yiyecekleri toplaması için yeni bir karakter oluşturmalıyız.



Yılan karakteri yiyecek isimli karaktere değdiği zaman yiyeceğin ekranda başka bir konuma yönelmesi ve yılan karakterinin hızının artması ayrıca büyüklüğünün artması için yiyecek karakterine



Blokları eklenmelidir.

Yılan karakterinin kenarlara değdiği zaman oyunun sonlanması için **yılan** karakteri içine



Blokları eklenmelidir.

Bitti haberi geldiğinde ekranın kararması ve oyunu kaybettiğini bildirmesi için yeni bir kukla oluşturulur.

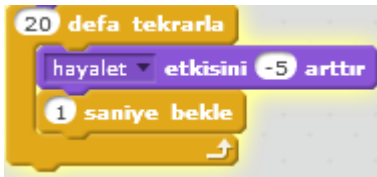
Oluşturulacak kukla kova aracıyla komple siyaha boyanmalı ve içine **“oyun bitti”** yazılmalıdır..



Yılan karakterinin kodlarında kenara değdiğinde **bitti haberi sal** komutu eklemiştik kenara değdiğinde ekranın yavaş bir şekilde kararması için yeni oluşturduğumuz siyah karakterin içine



Blokları eklenmelidir. Buradaki hayalet etkisi 100 olsun bloğu ile karakterin görünmemesi sağlanır

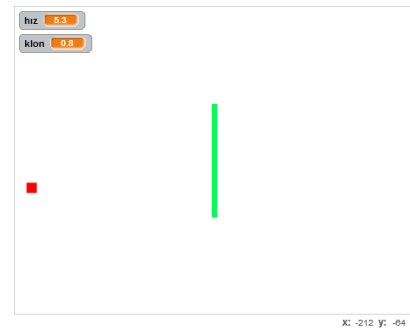


Bloklarıyla ise karakterin yavaş bir şekilde ekranda belirmesi sağlanır.

Yılan karakteri blokları



Şeklinde olmalıdır.



## Örnek 26: Tank oyunu

**Anahtar kelimeler:** yönü değeri ... yönüne dön

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/273083199/#player>

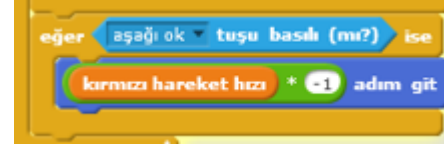
Örneğimizde bir adet tank yaparak karşısına gelen düşmanları yok etmesini sağlayacağız. Sahneye karakterimizi yükleyelim.



Karakterin içine



Bloklarını ekleyelim. Dönüş hızı isimli değişkenle karakterin sağa sola dönüş hızını hareket hızı ile de karakterin ileri geri hareket hızını belirleyebiliriz.



Komutu ile aşağı ok tuşuna basıldığında geri gitmesini sağlarız.



Silah1 karakteri ile tankın ateş etmesini sağlayacağız bunun için,

Silah1 karakterinin içine aşağıdaki bloklar eklenir.



Bu sayede boşluk tuşuna basıldığında karakterin ikizi oluşturulur. Yükleme süresi isimli değişken oluşturularak her 1 saniyede bir atış yapabilmesi sağlanır. Süre azaltılarak daha sık aralıklarla ateş etmesi sağlanabilir. Ayrıca silah1 karakterine

```
ikiz olarak başladığımda
görün
redtank 'na git
yönü değeri redtank'ın yönüne dön
kenar a değdi (mi?) olana kadar tekrarla
10 adım git
bu ikizi sil
```

Blokları eklenerek silah1 karakterinin tank karakteri ile aynı konuma gidip

```
dekor * değeri Sahne 'in
```

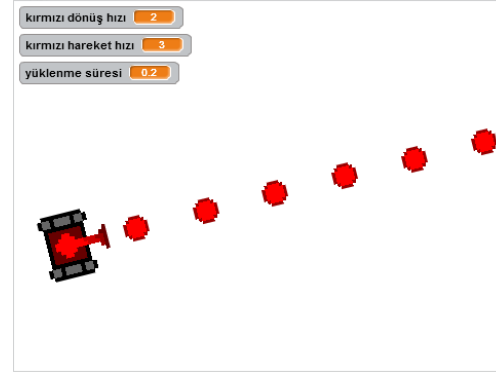
Komutu sayesinde

```
yönü değeri redtank 'in yönüne dön
```

Silah isimli karakterin redtank isimli karakter ile aynı yöne dönmesi yani ileri doğru ateş etmesi sağlanır. Aksi halde silah karakteri redtank karakteri ile farklı yönlere gidebilir.

```
kenar a değdi (mi?) olana kadar tekrarla
10 adım git
bu ikizi sil
```

Bloğu oluşturularak, klonun kenara değene kadar hareket etmesi sağlanır.



Ayrıca mavitank isimli karakter eklenerek, redtank isimli karaktere yönelmesi redtanka değdiği zaman oyununu sonlanması sağlanmalıdır.

```
tıklanınca
puan , 0 olsun
gizlen
sürekli tekrarla
kendim in ikizini yarat
0.5 saniye bekle
```

Mavitank isimli kuklaya blokları eklenerek, her 0.5 saniyede karakterin ikizinin yaratılması sağlanır.

```

ikiz olarak başladığımda
görün
x: 230 y: 180 ile -180 arasında bir sayı (tut) noktasına git
sürekli tekrarla
  redtank 'ye doğru dön
  3 adım git
  eğer silah1 a değdi (mi?) ise
    puan 'i 1 arttır
    bu ikizi sil

```

Ayrıca mavitank isimli kuklaya yukarıdaki kodlar eklenerek karakterin sahnenin sağ tarafından oluşması(x: 230), sürekli olarak redtank isimli kuklaya dönerek hareket etmesi, silah isimli kuklaya değer ise puan değişkeni 1 arttırılarak ikizin silinmesi sağlanır.

Son olarak mavitank isimli kukla redtank isimli kuklaya değdiğinde oyunun sona ermesi için redtank isimli kuklaya



```

tıklanınca
sürekli tekrarla
  eğer mavitank a değdi (mi?) ise
    oyun bitti de 1 saniye
    hepsi durdur

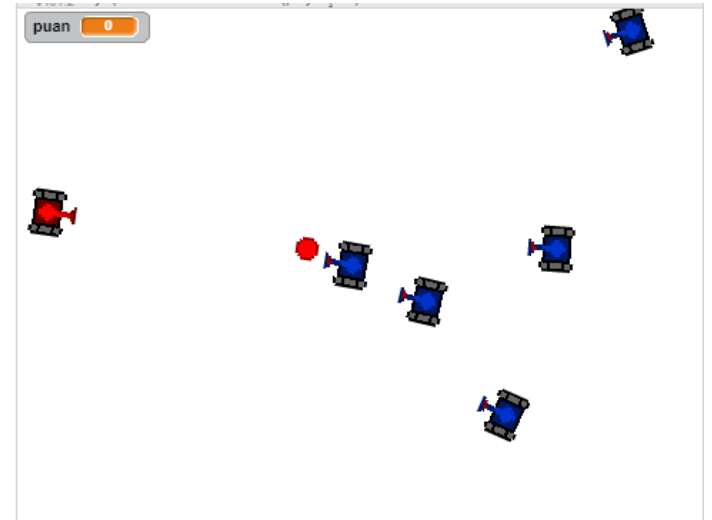
```



```

tıklanınca
  dönüş hızı 2 olsun
  hareket hızı 3 olsun
sürekli tekrarla
  eğer sağ ok tuşu basılı (mi?) ise
    dönüş hızı derece dön
  eğer sol ok tuşu basılı (mi?) ise
    dönüş hızı derece dön
  eğer yukarı ok tuşu basılı (mi?) ise
    hareket hızı adım git
  eğer aşağı ok tuşu basılı (mi?) ise
    hareket hızı +1 adım git

```



## Örnek 27: Uzay oyunu

**Anahtar kelimeler:** Kayan karakter, sürekli yenilenen ekran

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/273283994/#player>

Örneğimizde uzay oyunu yaparak karakterimizin kendisine gelen düşmanları yok etmesini, ayrıca arkaplanın kayarak sürekli olarak hareket etmesini sağlayacağız.

Sahneye kütüphaneden spaceship isimli kukla eklenir.



Kuklanın klavye tuşlarıyla hareket etmesi için daha önceki örneklerde



Özellikliğini kullanmıştık. Bu sayede yön tuşlarına basıldığında karakterin sağa dola hareket etmesini sağlıyorduk. Karakterin daha efektif ve kayar gibi hareket etmesini, yavaşlayarak durmasını sağlamak için hız isminde bir değişken tanımlayarak,



Blokları eklenir. Bu sayede; Karakterin en üst katmana çıkması , karakterin boyutunun %40 olması, başlangıçta x: 0 y: -127 konumuna gitmesi, sağ ok ve sol ok tuşlarına basınca hız değişkeninin değerinin 5 ve -5 olması, karakterinin x konumunun hız değişkeninin değerini alması,



Bloğu ile tuşa basma işlemi bitince karakterin yavaşlayarak durması sağlanır. Buradaki 0.9 değeri ne kadar küçültülürse duruşlar o kadar ani olur.

Sahneye BackCostume karakteri eklenerek



Karaktere sürekli kayma efekti verilmelidir. Bu sayede ekran sürekli bir bütünmüş gibi kayacaktır. Karakterin içine,



Blokları eklenmelidir. Bu sayede Karakter öncelikle sahnenin ortasına konumlanacak, ikizini yaratacak(tek ikiz) Karakterin hızını ayarlamak için kayma isimli değişken 4 yapılacak(ekranın kayma hızı) Y konumu -340 olana kadar karakterin y değeri azaltılacak bu sayede karakter sahnenin ortasından aşağı doğru hareket edecektir.

Aynı şekilde karakterin içine ekleyeceğimiz



Blokları ile, karakterin ikizi yaratılarak bu sefer sahnenin en üstüne konumlanacak ve aşağı doğru kayma hareketi yapacak, her iki blok y: -340 konumuna geldiklerinde eski konumlarına döneceklerdir.

Karakterin karşısına çıkan engellere ateş etmesi için,

Yeni kukla çiz seçeneği ile yuvarlak bir top çizilerek silah karakteri oluşturulması sağlanır.



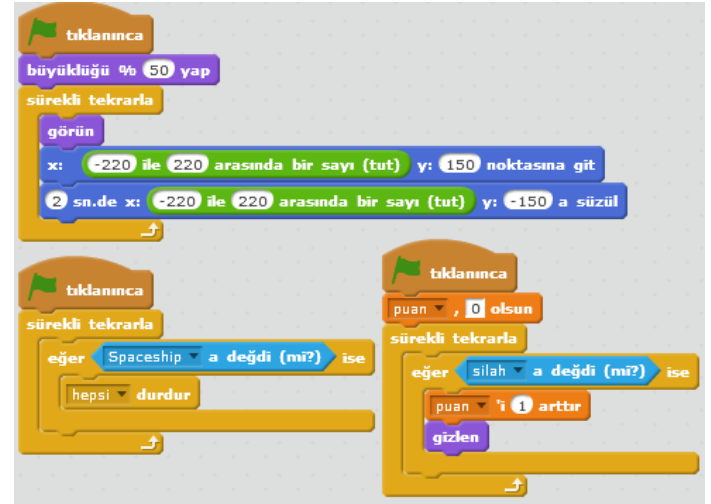
Karakterin içine





Blokları eklenmelidir. Bu sayede boşluk tuşuna basıldığında 1 saniyelik aralıklarla karakterin ikizinin oluşturulması sağlanır. Karakterin ikizi spaceship isimli karakterle aynı konuma giderecek, kenara değene kadar y değeri artar. Bu sayede karaktere ateş etme efekti verilmiş olur.

Sahneye bir düşman karakteri eklenerek



Blokları eklenir. Bu sayede düşman karakterinin ekranın üstünden altına doğru hareket etmesi sağlanarak uzay gemisine değdiğinde oyunun bitmesi, silah karakterine değdiği zaman kaybolması ve puanın artması sağlanır. Sahnenin kenarına değdiğinde yine karakter kaybolacaktır.

## Örnek 28: Araba Parkuru

**Anahtar Kelimeler:** olana kadar bekle, değil olana kadar bekle

**Projenin tamamlanmış hali:**

<https://scratch.mit.edu/projects/273436892/>

Uygulamamızda parkur oyunu arabanın yaparak parkuru tamamlamasını sağlayacağız.



Aracın sahnenin üstünden başlaması ve boyutunun bir miktar küçülmesi için



Blokları eklenmelidir,

Hız isimli bir değişken oluşturularak yukarı aşağı tuşlarıyla hızını ayarlamak, sol ve sağ ok tuşlarıyla yönünü değiştirmek için,



Blokları eklenir. Yukarıdaki bloklar sayesinde aracın hareketini klavye tuşlarından bağımsız olarak hız değişkenine bağlayarak, yukarı aşağı ok tuşlarıyla hızını ayarlayabileceğiz, ayrıca tuşlara basılmasa bile araç engele çarpana kadar mevcut hızını koruyarak hareketine devam edecektir.

Karakterin yoldan çıkarak yeşil alana girmesi durumunda, durması(hız değişkeni 0 olacak) ve yerinde spin atma hareketi yapması için,



Blokları eklenmelidir.

Karakterin belirli sayı döndükten sonra turu tamamlaması için



Karakterin kodlarına



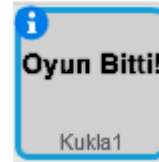
Blokları eklenir. 5 defa başlangıç noktasına ulaştığında oyun sonlanır. Buradaki



bloklarıyla siyah çizgiye değip değmediği tespit edilir. Siyah çizgiye değdikten sonra sayacın sürekli artmaması için yani siyah çizgi üzerinden karakter ayrıldıktan sonra sayacın artması için



Bloğu kullanılmıştır. 5 defa etap tamamlandıktan sonra oyun bitti haberi salınarak içerisinde oyun bitti yazan kukla oluşturulur.



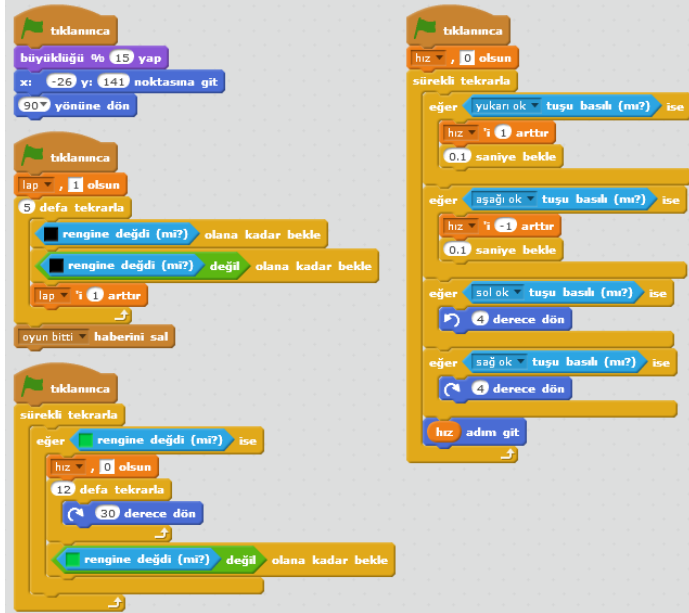
Kuklanın içine



Blokları eklenerek oyun bittiğinde



Ekranının gelmesi sağlar.



İstendiği takdirde süre değişkeni eklenerek aracın turu tamamlama süresi hesaplanabilir.

## Örnek 29: Trone Oyunu

Anahtar Kelimeler:

Projenin tamamlanmış hali:

<https://scratch.mit.edu/projects/273459029/>

Oyunumuzda iki adet oyuncu araçlarıyla hareket ederek, birisi mavi diğeri yeşil iz bırakarak hareket edecek, diğer oyuncunun çıkardığı renge değmemeye çalışacak ve diğer oyuncuyu kendi çıkardığı renge değıdirmeye çalışacaktır.

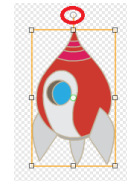
Kütüphaneden spaceship isimli kostüm eklenir.



Öncelikle karakterin yönünün değıştirilmesi gereklidir. Bunun için kılıklar sekmesinden karakter seçilerek

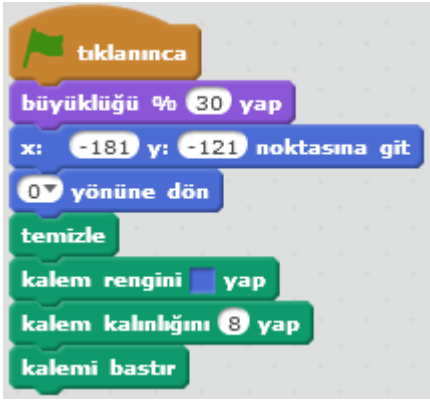


Karakterin üst tarafındaki yuvarlak vasıtasıyla yönünün sağa doğru çevrilmesi gereklidir.





Karakter 90 derece döndürüldükten sonra içerisine



Blokları eklenerek karakterin boyutu küçültülür, başlangıç noktası seçilir, yönü ayarlanır, kalem rengi, kalem kalınlığı ayarlanarak kalemin bastırılarak çizmesi sağlanır. Karakter hareket ettiğinde sürekli olarak mavi iz bırakması sağlanır

Ayrıca yukarıdaki blokların altına aşağıdaki bloklar eklenerek

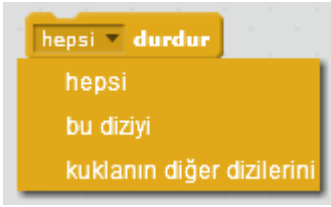


Karakterin sürekli olarak hareket etmesi, klavyeden a tuşuna basıldığında sola, d tuşuna basıldığında sağa gitmesi sağlanır. Ayrıca diğer karakterin çıkardığı yeşil çizgiye değdiğinde p2 kazandı haberi salınması sağlanır.

Kuklaya eklenecek aşağıdaki bloklar sayesinde

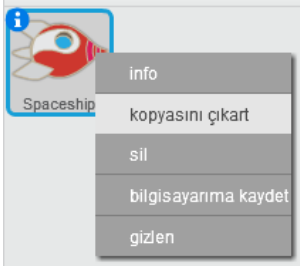


p2 kazandı haberi geldiğinde kukla

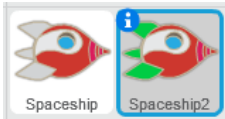


Kuklanın diğer dizileri durdurulacak, kaybettim diyecek ve tüm bloklar durdurulacaktır. **Kuklanın diğer dizilerini durdur** komutunun avantajı bu kuklaya ait diğer diziler(bloklar) çalışmayı durduracak yani karakterin hareketi sonlanacak ancak bu blok dizisinin altındaki bloklar çalışmaya devam edecektir.

Karakterin kopyası çıkarılarak



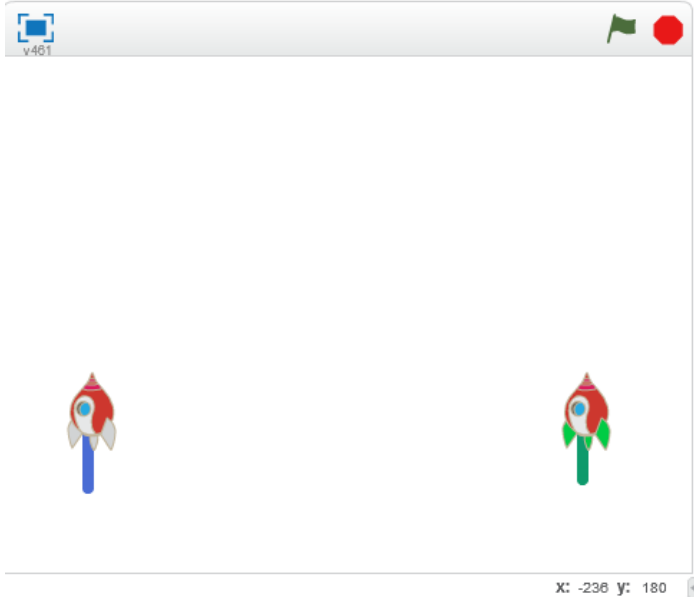
Karakterlerin renklerinin karışmaması için kılıklar sekmesinden ikinci karakterin rengi değiştirilebilir.



İkinci karakterin içine



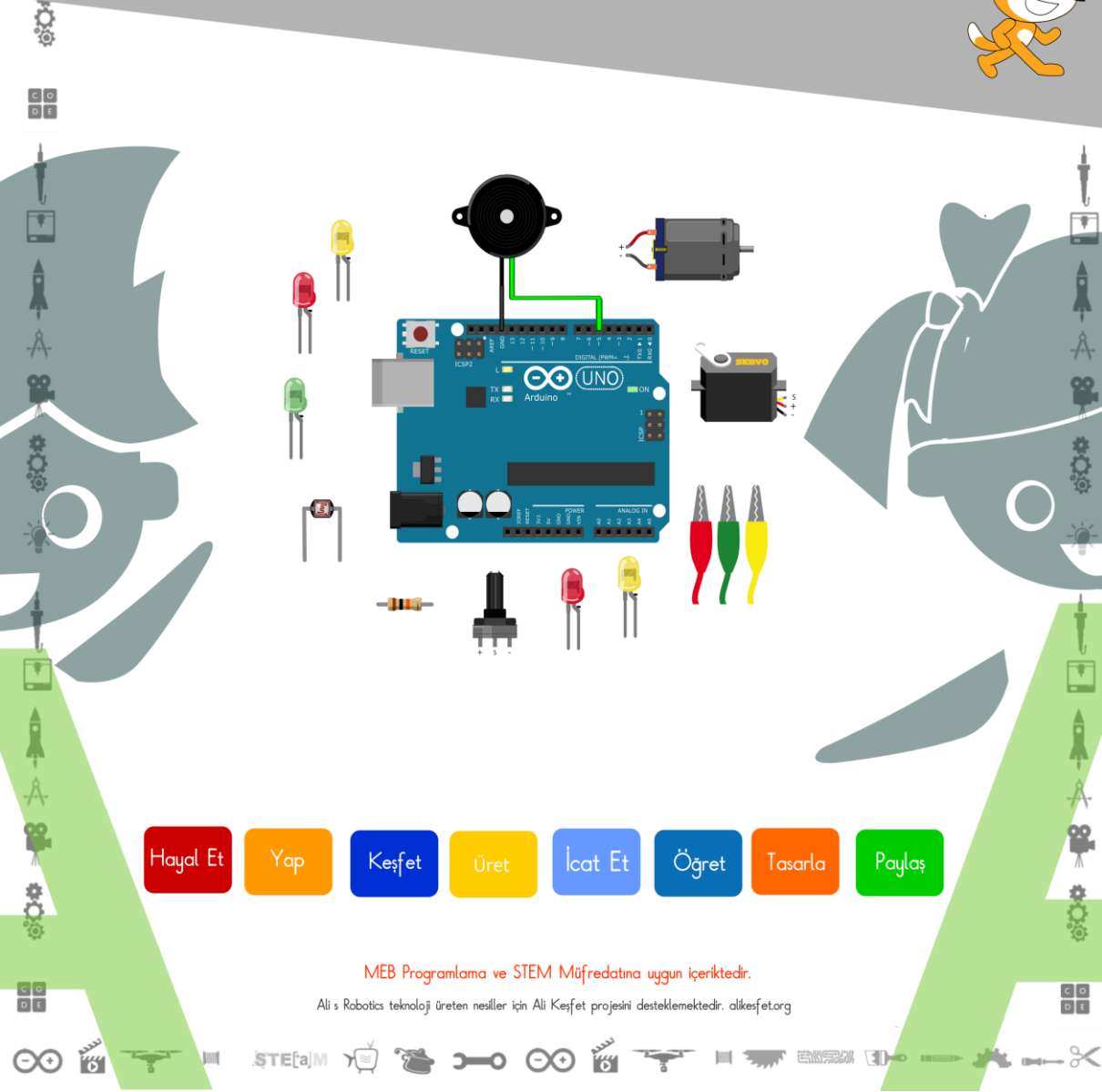
Blokları eklenmelidir. 1. Kukladan farklı olarak karakter yeşil çizgi çizmeli mavi renge değdiği zaman **p1 kazandı haberi salmalı** ve **p1 kazandı** haberi gelince ve **p2 kazandı** haberi gelince kodlarının ismi değiştirilmelidir. Ayrıca yön değiştirmek için a ve d tuşlarının yerine **sol ok** ve **sağ ok** tuşları kullanılmalıdır.



Çocuklar İçin

# Robotik Kodlama

Bilgisayarca Düşünmeyi Destekler



Hayal Et

Yap

Keşfet

Üret

İcat Et

Öğret

Tasarla

Paylaş

MEB Programlama ve STEM Müfredatına uygun içeriktedir.

Ali's Robotics teknoloji üreten nesiller için Ali Keşfet projesini desteklemektedir. alikesfet.org



## Bilgisayarca Düşünme

“Computational Thinking (Bilgisayarca Düşünme) terimini bilgisayarları üretim amaçlı olarak hayat problemlerinin çözümünde kullanabilmek için gerekli olan bilgi, beceri ve tutumlara sahip olmak diye tanımlamak mümkündür.”

### M. Yaşar ÖZDEN

Bilgisayarca düşünmenin temelinde problem çözme vardır. Günümüz öğrencileri teknolojiyi etkin bir biçimde kullanmaktadırlar. Bilgisayarca düşünmede ise amaç teknolojiyi ve bilgisayarları problem çözmeye kullanmaktır. Bilgisayarca düşünme, bilgisayarların işlem gücü ile insanın düşünme yeteneklerinin birleşmesidir. Bu ikisi, yaşam kalitemizi arttırmak ya da yaratıcı çözümler üretmek için kullanılır.

Bilgisayarca düşünme öğrencilerin sahip olması gereken 21.yy yeteneklerini de içeren bir düşünme biçimidir.

Bu yetenekler aşağıdaki gibidir.

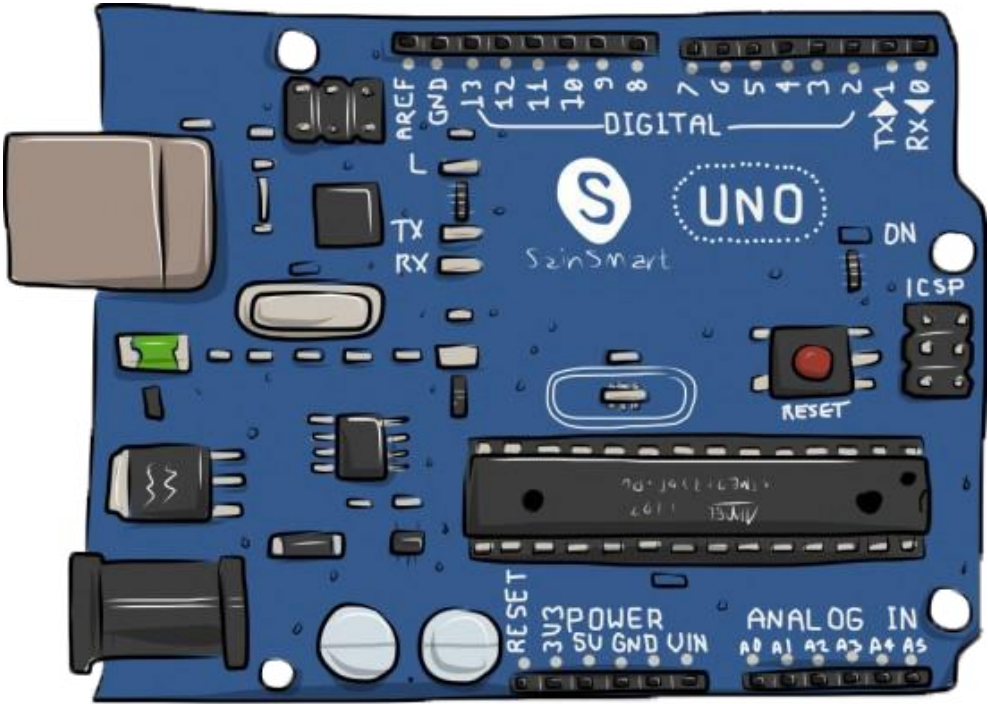
- ⊗ Kodlama
- ⊗ Algoritmik düşünme
- ⊗ Eleştirel düşünme
- ⊗ Yaratıcı düşünme
- ⊗ Problem çözme
- ⊗ İletişim
- ⊗ İş birliği

Günümüz öğrencileri meslek hayatına atıldığında karşılaşacakları dünya bugünkü dünyadan çok farklı olacaktır. Bu değişimin sonucu olarak bu yüzyılın gerektirdiği yeteneklere sahip olan bireyler, her zaman bir adım önde olacaklardır.

## Robot Nedir?

Robotlar, verilen görevleri yerine getirmesi için programlanmış, elektronik bir beyni ve hareket yeteneği olan cihazlardır.

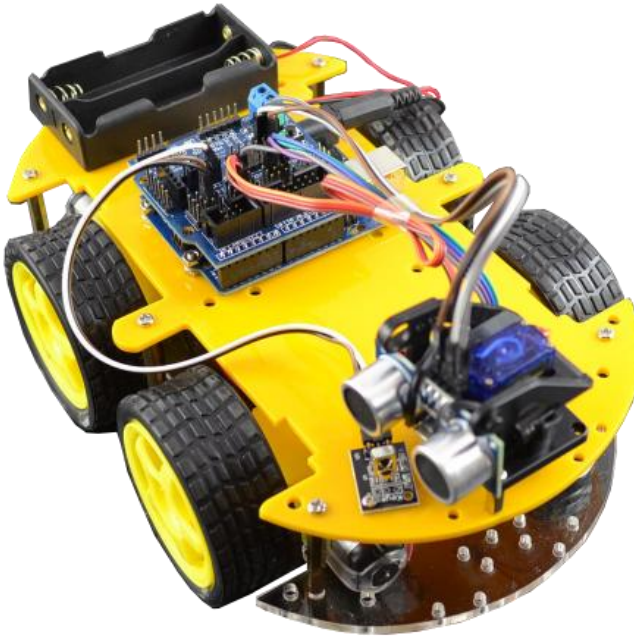
Robotumuzun verilen görevleri yerine getirebilmesi için programlanabilir bir beyni olması gerekmektedir.



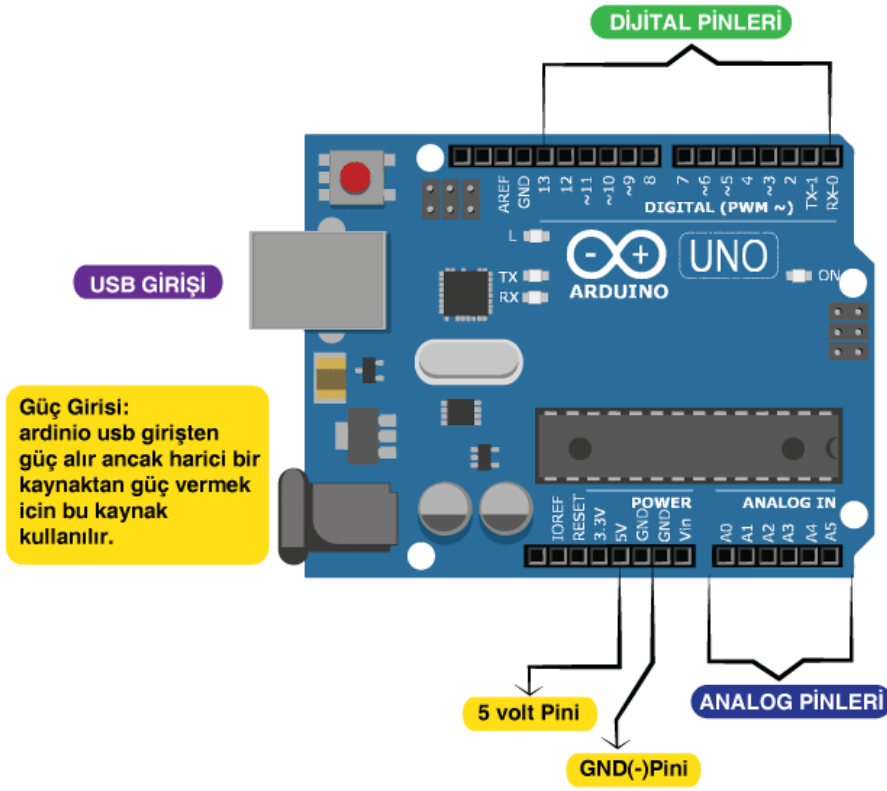
Arduino Uno, öğrencilerin kolaylıkla programlayabileceği elektronik bir beyindir. Bu kartı bilgisayara bağlayarak robotik sistemlerimizi rahatlıkla kontrol edebiliriz. Kartın üzerinde bilgisayarımızdaki işlemciye benzeyen programlanabilir bir parça vardır. Bu kısmı bilgisayarda yazdığımız kodlarla programlarız ve istediğimiz görevleri yerine getirmesini sağlarız.

## Arduino Nedir?

Arduino, bir mikro kontrol ünitesidir. Bilgisayarımızda bulunan işlemciye benzer. Üzerinde elektronik cihazlarımızı kontrol etmemize yarayan bir mikrodenetleyici vardır. Biz bu mikrodenetleyiciyi programlayarak istediğimiz işleri yapmasını sağlarız. Arduino'muz ile yapabileceğimiz hayli geniş ve sizin hayal gücünüze kalmıştır. Örneğin; engelleri algılayıp aşan bir robottan, toprağın kurduğunu algılayıp çiçeğinizi sulayan bir sisteme kadar geniş bir yelpazede çalışmalar yapabilirsiniz. Arduino'nun birçok modeli mevcuttur. Biz kitabımızda en çok kullanılan model olan uno r3 modelini kullanacağız.



Engelden Kaçan Robot

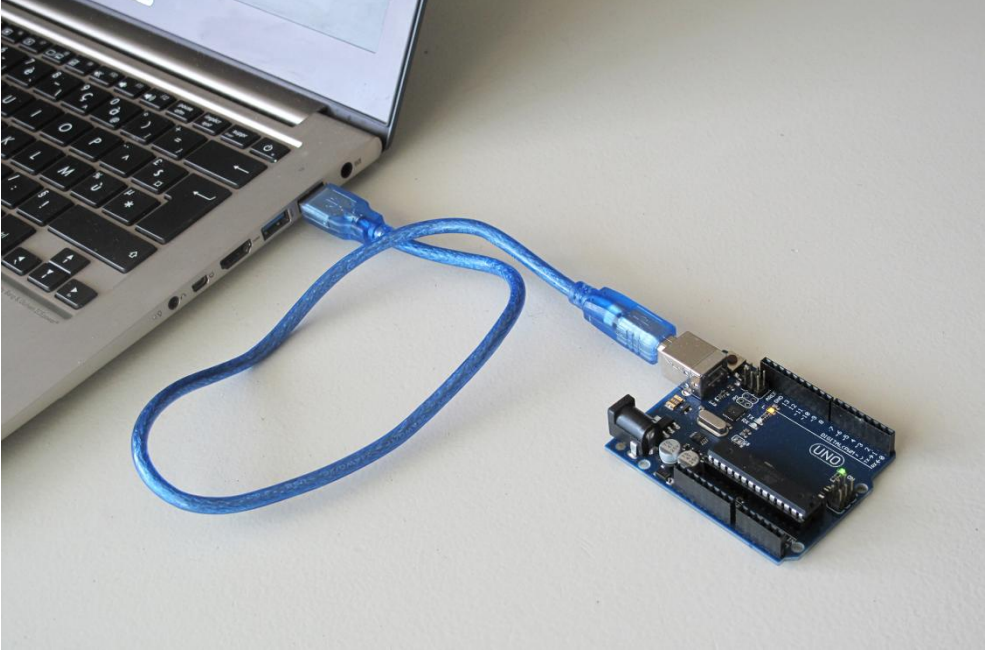


**Dijital Pinler:** Arduino'muzu çevresiyle etkileşim halinde tutabilmek için pinleri kullanırız. Dijital pinleri Arduino'dan dışarıya akım vermek ya da gelen akımları akımak için kullanırız. Örneğin; Arduino'ya bağladığımız bir Led'i yakmak için dijital pinleri kullanırız.

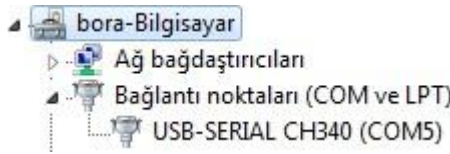
**Analog Pinler:** Arduino'ya bağladığımız sensörleri okumak için analog pinleri kullanırız.

**Usb Girişi:** Arduino'yu bilgisayara bağlamak için kullandığımız giriş.

## Arduino'yu Bilgisayara Bağlama



Arduino'yu bilgisayarımıza usb kablosu aracılığıyla bağlarız. Bilgisayarınızın Arduino'yu tanıyıp tanımadığını anlamak ya da nasıl bağlandığını görmek için bilgisayarına sağ tıklayın, özellikler menüsünden **aygıt yöneticisini** açın.



Arduino bilgisayara Com bağlantı noktasından bağlanır. Bende bağlantı noktası Com5 olarak gözükmemekte. Sizde farklı bir Com noktası gözükebilir.

## Arduino'yu Nasıl Programlarız?

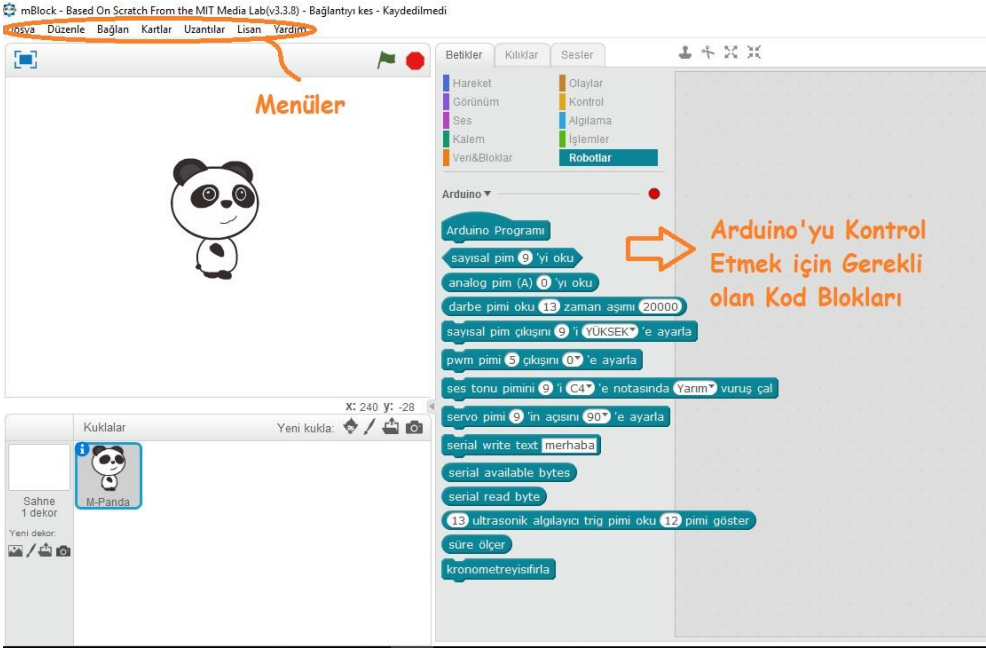
**Programlama**, bilgisayarın ya da robotların ne yapması gerektiğinin adım adım söylenmesidir. Örneğin; engel algılandığında sola dön, ileri tuşuna basıldığında motorlara güç ver. Bir robot, bu işleri nasıl yapması gerektiğini bilmez. Bu yüzden robotların bir işi yapabilmesi için programlanması gerekmektedir.

Arduino'yu programlamak için bir programlama diline ihtiyacımız var. Programlama dilleri, robotlara gerekli komutları anlayacağı bir şekilde vermek için kullanılır. İnsanların günlük hayatta iletişim kurmak için nasıl bir dil kullanmaları gerekiyorsa bizim de robotlar ile haberleşebilmemiz için bir dil kullanmamız gerekir. Biz kitabımızda Arduino'yu programlamak için Scratch programlama dilini kullanacağız.

Scratch MIT (Massachusetts Institute of Technology) üniversitesitarafından geliştirilen bir görsel programlama dilidir. Bu dil özellikle 9-16yaş grubu öğrenciler için oluşturulmuştur. Bu dili kullanarak kendi oyununuzu programlayabilir, kendi animasyonlarınızı oluşturabilir ve yaratıcılığınız dahilinde birbirinden güzel çalışmalar üretebilirsiniz. Biz Arduino'yu programlamak için Scratch programlama dili üzerine inşa edilmiş olan **Mblock** programını kullanacağız.

Mblock programını [www.mblock.cc](http://www.mblock.cc) adresine girerek indirebilirsiniz.

Mblock internet sitesine girdikten sonra sırasıyla **download>Mblock for PC>Mblock 3** yolunu takip edelim. İşletim sistemi seçimini yaptıktan sonra **Previous Versions** kısmını seçelim. Bizim kitapta kullandığımız versiyon 3.4.11'dir.

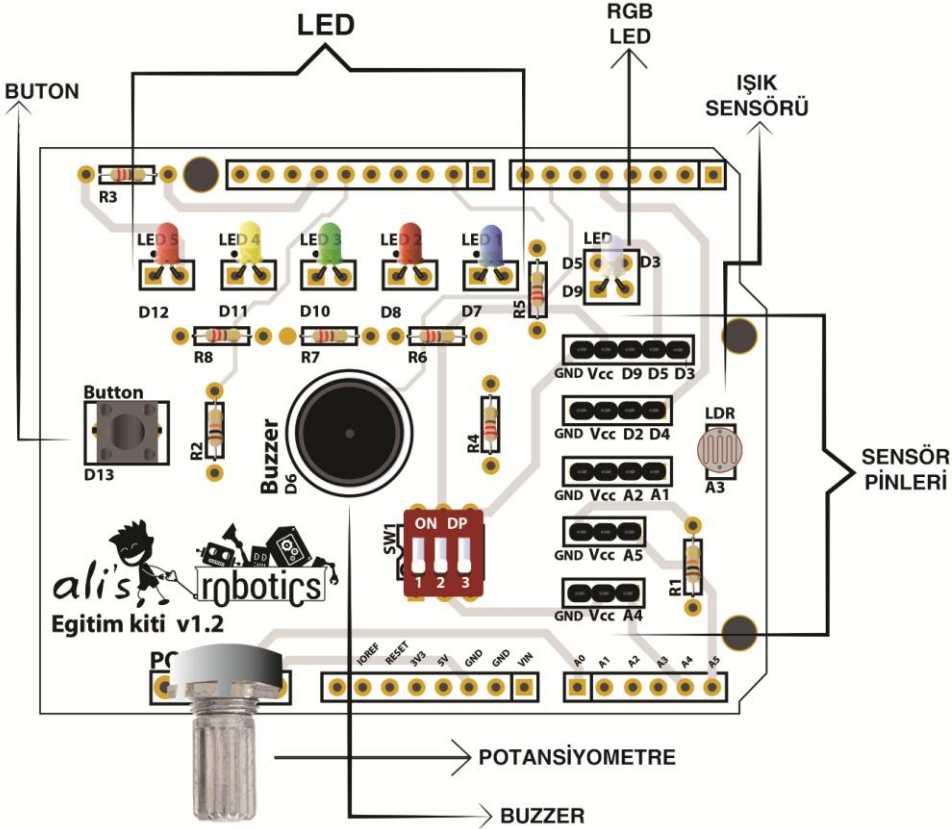


Mblock programının arayüzü Scratch ile aynıdır. Scratch'e ek olarak Arduino'yu programlamak için robotlar kısmında bazı kod blokları mevcuttur.

Mblock programını yükledikten sonra **Language** menüsünden Türkçe'yi seçebilirsiniz. Dil ayarını yaptıktan sonra **Kartlar** menüsünden Arduino Uno seçilmelidir.



## Scratch Eğitim Kartı



Scratch eğitim kartı, öğrencilerin kolay ve güvenli bir şekilde projelerini gerçekleştirebilmeleri için hazırlanmıştır. Üzerinde öğrencilerin robotik konuları öğrenebilmeleri için gerekli elektronik parçalar mevcuttur. Öğrenciler bu kartı bilgisayara bağlayarak hemen kendi projelerini kodlayabilir, arkadaşlarıyla paylaşabilir ve hazırlanmış projeleri dilediklerinde hackleyebilirler.

Youtube Videosu İçin Tıklayınız!





Scratch eğitim kartı Arduino kartın üzerine takılacak şekilde tasarlanmıştır. İstenildiğinde sökülerek Arduino tek başına kullanılabilir.

Kartın üzerinde Arduino'nun dijital pinlerine bağlı Ledler, Rgb Led (Her renkte yanabilen Led), Buzzer ve bir adet programlanabilir buton mevcuttur. Kartın üzerindeki Potansiyometre ve LDR (ışık sensörü) ise analog pinlere bağlıdır. Öğrencilerin değişik sensörleri ve elektronik parçaları kullanabilmeleri için de ayrıca sensör pinleri mevcuttur.

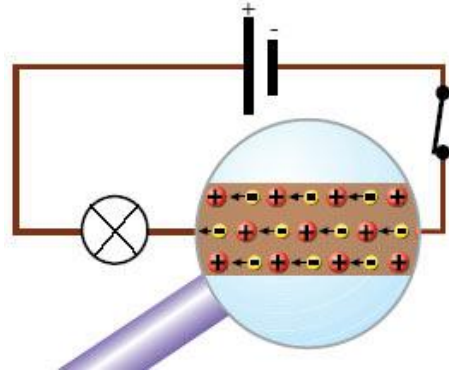
## Temel Elektronik

Arduino'yu programlarken pek fazla elektronik bilgisi gerekmez ancak elektronik hakkında temel bilgiler edinmekte fayda var.

**Elektronik**, elektronik aygıtları çalıştırmak için küçük elektrik akımlarının nasıl kumanda edilmesi gerektiğini inceleyen bilim dalıdır. Kullandığımız bilgisayar, televizyon, akıllı telefonlar hepsi elektronik bilimi sayesinde hayatımıza girdi. Bizde kitabımızda bazı elektronik kavramlarını ve elektronik devre elemanlarını inceleyeceğiz. İnceleyeceğimiz devre elemanları, kullandığımız bütün elektronik aletlerde bulunan parçalardır.

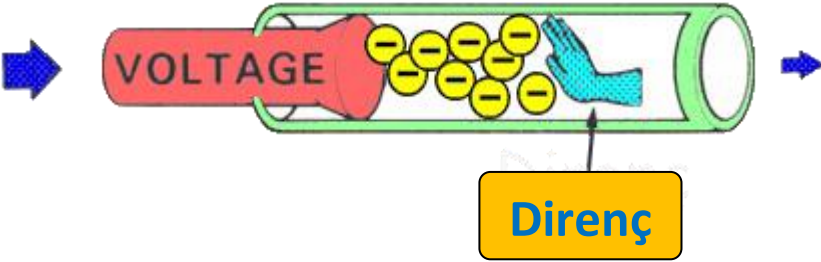
### Elektrik Akımı

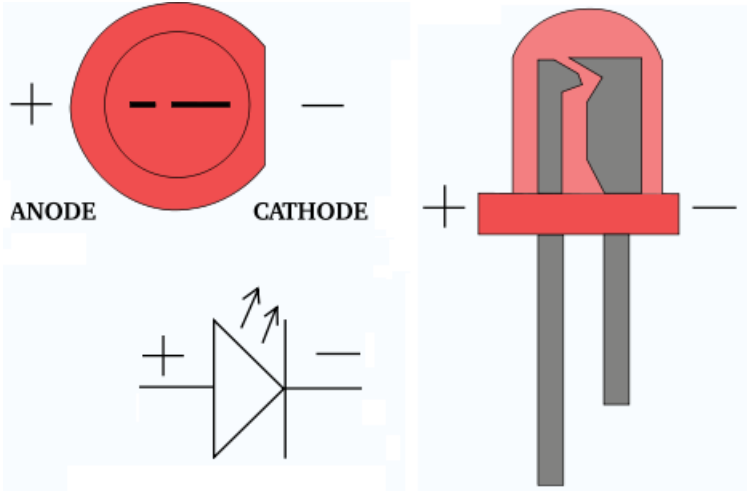
Elektrik akımı, elektron adı verilen çok küçük parçacıkların bir tel boyunca akışıdır. Bütün maddeler elektron içerir. Elektronlar her şeyi oluşturan atomların bir parçasıdır.



[Detaylı Bilgi için Tıklayınız!](#)

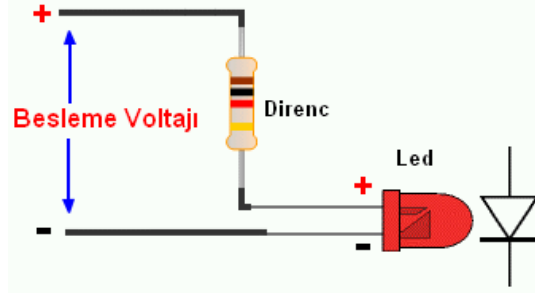
Elektronlar pilin (+) ucundan (-) ucuna doğru hareket ederler. Bu hareket sonucu bir gerilim oluşur, biz bu gerilime Voltaj diyeceğiz ve V ile göstereceğiz. Elektronların (+) uctan (-) uca doğru hareketine akım diyeceğiz ve I ile göstereceğiz. Akım (I) ve gerilim (V) bir devrenin temel bileşenlerinden ikisidir. Üçüncü temel bileşen dirençtir. Direnç, kimi maddelerin elektron akışına yani akıma karşı gösterdiği zorluktur.





**Ledler**, üzerinden akım geçtiğinde ışık yayan devre elemanlarıdır. Uzun ucu (+) kısa ucu (-). Yalnız dikkat etmemiz gereken husus: Asla Led'imizi direnç olmadan devremize bağlamayacağız. Çünkü Ledler hassas devre elemanlarıdır, uzun süre yüksek akıma maruz kaldığında yanabilir direnç akıma zorluk göstererek akımı azaltacak ve Led'imize uygun hale getirecektir.

Kartın üzerinde ledler direnç ile bağlanmıştır.



**Dirençler** akıma zorluk gösteren devre elemanlarıdır. Arduino ile pinlerden vereceğimiz akım Ledler için yüksek değerdedir. Dirençler bu değeri düşürerek Led'e uygun hale getirirler.

## Bir Led Yak

Ledler elektrik enerjisini ışık enerjisine çeviren elektronik parçalardır. Bilgisayarımızın ya da televizyonumuzun üzerinde Ledler bulunmaktadır. Bu Ledler bize daha çok o anki durumu anlatan uyarıcılar olarak kullanılmaktadır. Scratch eğitim kartımızın üzerinde de 5 adet Led bulunmaktadır. Bu Ledler sırasıyla Arduino'nun üzerinde bulunan 12,11,10,8 ve 7 no.lu pinlere bağlıdır.

Haydi, hep birlikte bu Ledleri yakmayı öğrenelim.

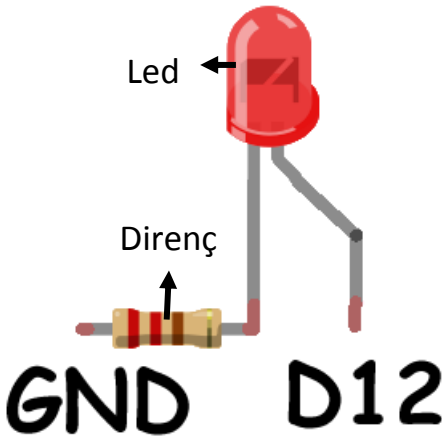
- 1 Üzerinde Scratch eğitim kartı takılı olan Arduino kartımızı usb kablosu aracılığıyla bilgisayara bağlayalım ve Mblock programımızı çalıştıralım.
- 2 Mblock programında **Bağlan> Seri Kapı** menü yolunu izleyerek Arduino kartımızın bağlı olduğu seri kapı numarasını seçelim.
- 3 Bağlan menüsünden **Aygıt Yazılımı Güncellemesi** kısmını seçelim. Bu menü ile Arduino karta bir yazılım yüklenecek. Bu sayede Mblock programı ile Arduino kartımız iletişime geçecek.
- 4 Aşağıdaki kodları yazalım ve yeşil bayrağa tıklayalım.



```
tıklandığında
sürekli tekrarla
12 sayısal pini YÜKSEK yap
1 saniye bekle
12 sayısal pini DÜŞÜK yap
1 saniye bekle
```

Bu kod bloklarıyla kartımızın üzerinde bulunan ve Arduino'nun 12 no.lu pinine bağlı Led'i yakıp söndüreceğiz.

Not: Arduino üzerinde bulunan dijital pinleri, karttan dışarıya elektrik akımı vermek ya da gelen elektrik akımlarını okumak için kullanırız.



Ledler elektrik enerjisini ışık enerjisine çevirirler. Kartımızın üzerinde bulunan Ledleri yakabilmemiz için Ledlerin bağlı bulunduğu pinlerden elektrik akımı vermemiz gerekmektedir.

9 sayısal pini YÜKSEK yap

**9 sayısal pini YÜKSEK yap** kod bloğuyla arduino kartın dijital pinlerinden dışarıya elektrik akımı verebiliyoruz. 9 yazan yeri değiştirebilirsiniz.

Arduino kartımızın dijital pinlerinden elektrik akımı gönderdiğimizde her bir dijital pin 5V elektrik gönderir. Çoğu Led için bu değer yüksektir. Bu yüksek olan değeri düşürmek için gerilim bölücü yani direnç kullanırız. Direnç pinden gelen Voltaj değerinin bir kısmını üzerine çekerek elektrik akımını led için uygun hale getirir. Eğer direnç kullanmazsak Led akıma dayanamaz ve yanar.



```
when green flag clicked
  set digital pin 10 to LOW
  set digital pin 7 to LOW
  loop
    set digital pin 12 to HIGH
    wait 1 seconds
    set digital pin 12 to LOW
    wait 1 seconds
```

Yeşil bayrağa tıkladığınızda bazı Ledlerin fazladan yandığını görebilirsiniz. Bunun nedeni Mblock programının çalışma sistemidir. Bu Ledlerin yanmasını engellemek için 1,2,3 no.lu düğmeleri aşağıya indirerek yan tarafta bulunan kod bloklarını ekleyebilirsiniz.



## RGB Led Uygulaması

RGB Ledler, Kırmızı, Yeşil ve Mavi olan üç Led'in birleşiminden oluşur. Bu üç Led'i istediğimiz gibi kontrol ederek tüm renkleri elde edebiliriz.

```

k tuşu basılınca
3 pwm pini 0 yap
5 pwm pini 255 yap
9 pwm pini 255 yap
    
```

```

y tuşu basılınca
3 pwm pini 255 yap
5 pwm pini 0 yap
9 pwm pini 255 yap
    
```

```

m tuşu basılınca
3 pwm pini 255 yap
5 pwm pini 255 yap
9 pwm pini 0 yap
    
```



RGB Led'imizin üzerinde 3 adet pin vardır. Bu pinlere elektrik akımı vererek Led'i istediğimiz renkte yakabiliriz.

RGB Led'imiz Arduino kart üzerindeki 3, 5 ve 6 nolu dijital pinlere bağlıdır. Bu pinlerin başka bir özelliği daha vardır. Bu pinler PWM pinlerdir. PWM pinlerin değer dijital pinlerden farkı 0-255 arası değer gönderebilmesidir. Dijital pinlerden sadece 0V ya da 5V akım gönderilebilirken PWM pinlerden 0V-5V arası akım gönderilebilir.

```

5 pwm pini 100 yap
    
```

Örneğin; 5 nolu PWM pinden 100 değerini gönderdiğimizde yaklaşık 2V bir elektrik akımı gönderilir.

Şimdi RGB Led'imizi rastgele renklerde

yakalım. Bunun için kırmızı, yeşil ve mavi adlarında üç değişken oluşturalım. Bu oluşturduğumuz değişkenlere 0-255 arasında rastgele değerler atayalım. Bu değişkenlerden gelen değerleri de RGB Led'in bağlı olduğu PWM pinlere gönderelim.

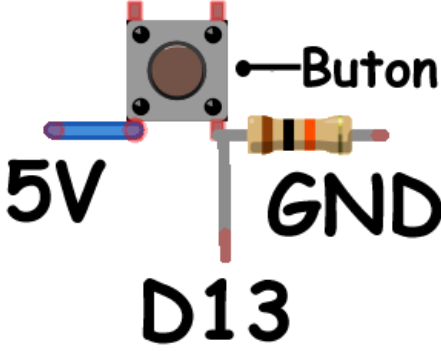


PWM pinlerden 0-255 arası değerler girip, 0V-5V arası akım göndererek RGB Led'imizi farklı renklerde yakabiliyoruz. Bu örnekte ise değerleri değişkenler aracılığı ile girdik. Oluşturduğumuz değişkenlere rastgele değerler atadığımız için de RGB Led'imiz rastgele renklerde yanacaktır.

Youtube Videosu İçin Tıklayınız!



## Buton Kullanımı



Scratch eğitim kartının üzerinde kullanmanız için bir adet buton bulunmaktadır. Bugün birçok dijital aletin, oyun kolunun üzerinde bu butonların benzerleri bulunmaktadır.

Butona basıldığında 13 nolu dijital pin ile 5V pini birbirine bağlanır. 5V pininden çıkan elektrik akımı 13 no.lu dijital

pine ulaşır. Bu elektrik akımı 13 no.lu dijital pine ulaştığında butona basıldığını anlarız.

Butona basılmadığında ise Arduino kart bazı nedenlerden dolayı bazen akım gelmiş bazen de akım gelmemiş gibi davranır. Bu nedenle biz butona basmasak da sanki butona basılmış gibi olur. Bu karışıklığı önlemek için butonu bir direnç yardımıyla GND pinine bağlarız. Bu dirence **Pull-Down** direnci denir. Bu direncin görevi buton basılı olmadığında karışık değerler okunmasını engellemektir.

**Dijital pinlerden dışarıya elektrik akımı verdiğimiz gibi gelen elektrik akımlarını da okuyabiliriz.** Şimdi butona basıldığında gelen elektrik akımını okuyarak butona basıldığını anlayalım.

```
when clicked
  loop
    if 13 digital pin read = 1 then
      12 digital pin HIGH
    else
      12 digital pin LOW
```

13 nolu dijital pinin değerini okuyoruz. Butona basılı değilken 13 nolu pinden, bağladığımız direnç sayesinde sabit 0 değeri okunur. Butona bastığımızda ise 1 değeri okunur. Biz yazdığımız bir eğer ise değilse bloğu ile 13 no.lu dijital pinin değeri 1 olduğunda yani butona basıldığında 12 nolu dijital pine bağlı ledi yakıyoruz. Basılı olmadığında ise söndürüyoruz.

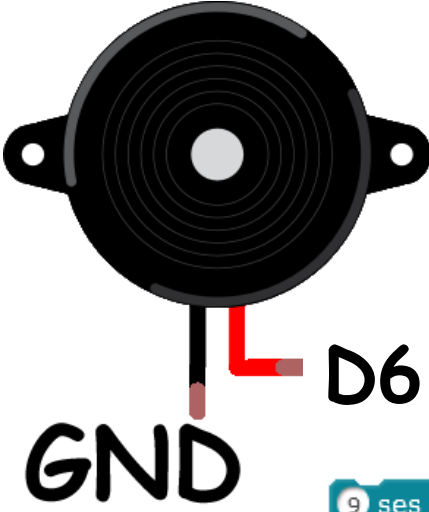
9 sayısal pini oku



Pin numarasını değiştirerek 13 yapınız!

## Buzzer Kullanımı

Buzzerlar elektronik devrelerde uyarı sesleri vermek için kullanılır.



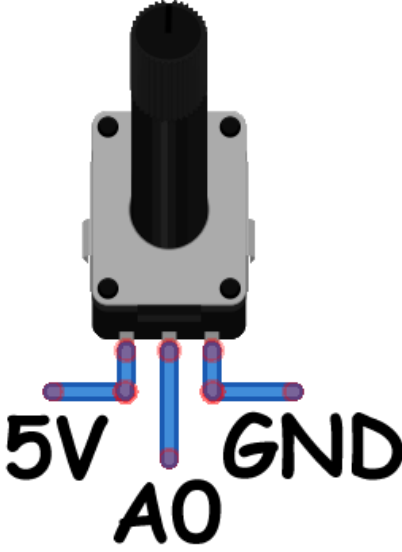
Scratch eğitim kartımızın üzerinde de dijital 6 pinine bağlı bir Buzzer bulunmaktadır. Bu Buzzer'ı ilgili pine akım vererek kullanabileceğimiz gibi özel oluşturulmuş bloğunu da kullanarak istediğimiz notada ses üretebiliriz.

9 ses tonu pini C4 notasında Yarım vuruş çal



Pin Numarasını buradan değiştirerek 6 yapınız!

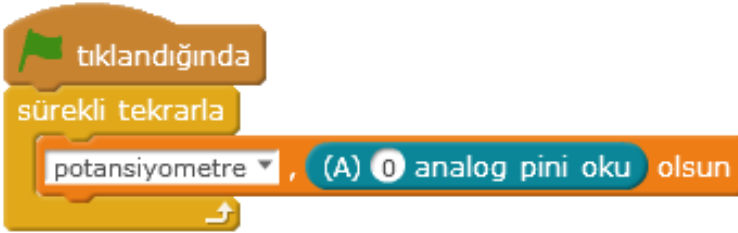
## Potansiyometre



Potansiyometre, ayarlanabilir dirençtir. Elektronik cihazların üzerinde döndürülebilir ses düzeyini ayarladığımız düğmeler potansiyometrelerle yapılır.

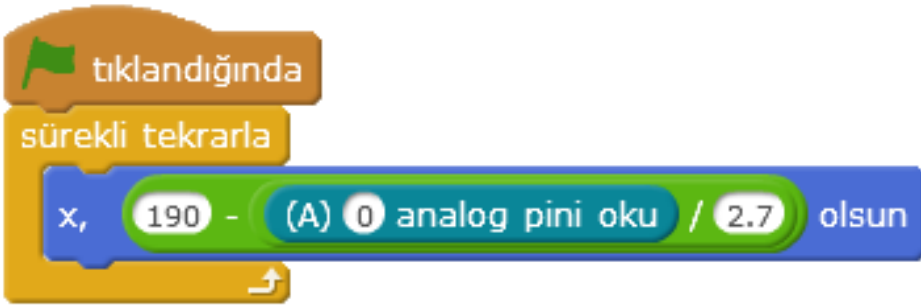
Potansiyometremizin üç ayağı vardır. Bir ayağını 5V , bir ayağını GND, orta ayağını da analog 0 pinine bağlıyoruz.

Bir değişken oluşturuyoruz ve analog 0 pininden okunan değeri bu değişkene aktarıyoruz.



## Potansiyometre ile Karakter Kontrolü

Potansiyometre ile Analog pinden 0-1023 arası bir değer okuruz. Potansiyometreyi döndürdükçe okunan değer değişecektir ve bu değişen değere göre Potansiyometre'nin hangi yöne döndüğünü ya da hangi konumda olduğunu belirleyebiliriz. Biz bu değeri kullanarak karakterin X konumunu kontrol etmekte kullanacağız.



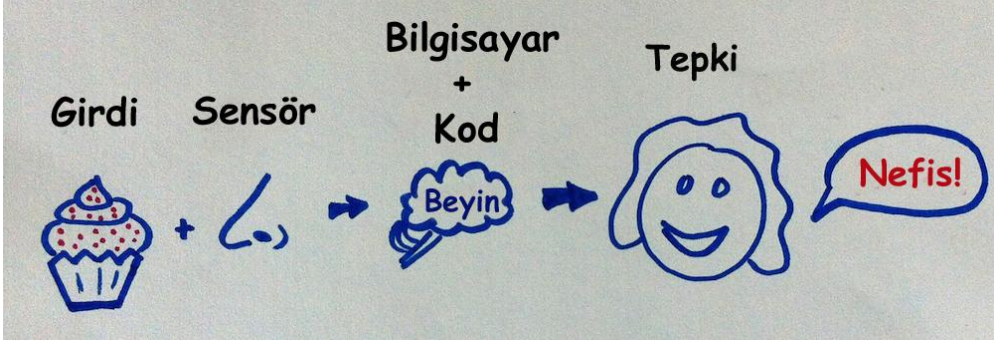
Potansiyometre'nin konumunu en sola getirdiğinizde Analog pinden 1023 değeri okunurken en sağa getirdiğinizde ise 0 değeri okunur. Analog pinden okunan değeri 2.7'e bölüp 190'dan çıkarıyoruz. Potansiyometre en soldayken Analog pinden 1023 değeri okunur 1023 değerini 2.7'e böldüğümüzde 379 gibi değer gelecektir. 190'dan 378 değeri çıkarıldığında yaklaşık -190 değeri bulunur. Karakterin X konumu -190 olur. Potansiyometre sağa doğru çevrildikçe Potansiyometreden gelen değer azalacaktır ve karakter sağa doğru hareket edecektir.

Not: Kitabın başında verilen linkten indirdiğiniz kaynak dosyalar içinde bulunan "potansiyometreoyun.sb2" dosyasını açınız. Bu oyun potansiyometre ile karakter kontrolüne örnek oluşturmaktadır.

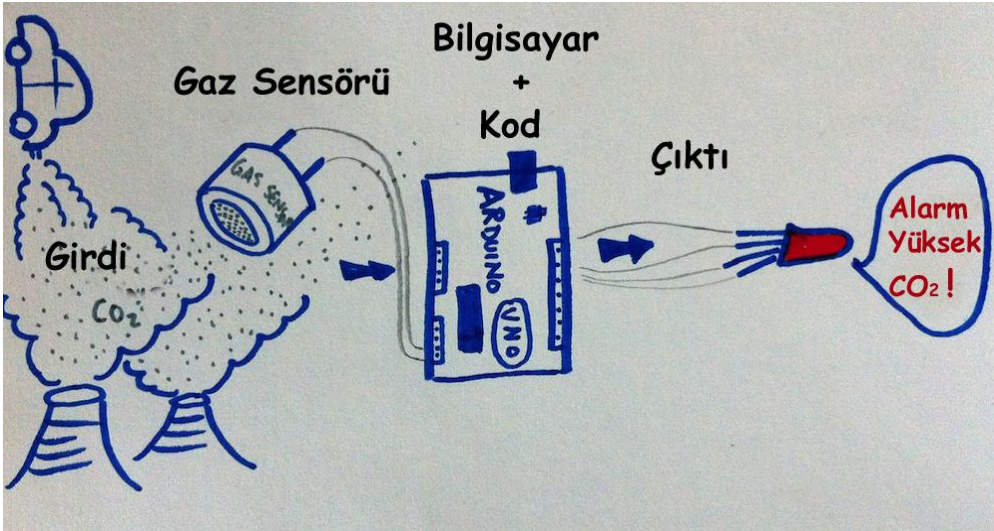
Youtube Videosu İçin Tıklayınız!



## Sensörler



İnsanlar çevrelerindeki dünyayı ve çevrelerinde olup bitenleri algılamak için duyu organlarını kullanır.



Makineler (Bilgisayarlar, Robotlar) ise çevrelerinde olup bitenleri algılamak için sensörleri kullanır.



## Işık Sensörü



**LDRler** ışığa duyarlı dirençlerdir. Aydınlıkta dirençleri minimum, karanlıkta ise maksimumdur. Biz bu özelliğini kullanarak gece olduğunda yani karanlıkta yanan Led yapacağız. Hava kararınca Ldr'nin direnci maksimum olacak ve yazacağımız kod ile bu durum gerçekleştiğinde Led yanacak.

Ldr'nin bir ayağını 5V pinine diğer ayağını da Analog 3 pinine bağlıyoruz. Analog 3 pinine bağladığımız bacağını da bir Pull-Down direnci ile GND pinine bağlıyoruz ki sensörden karışık değerler okunmasın.

## Hava Kararınca Yanan Led

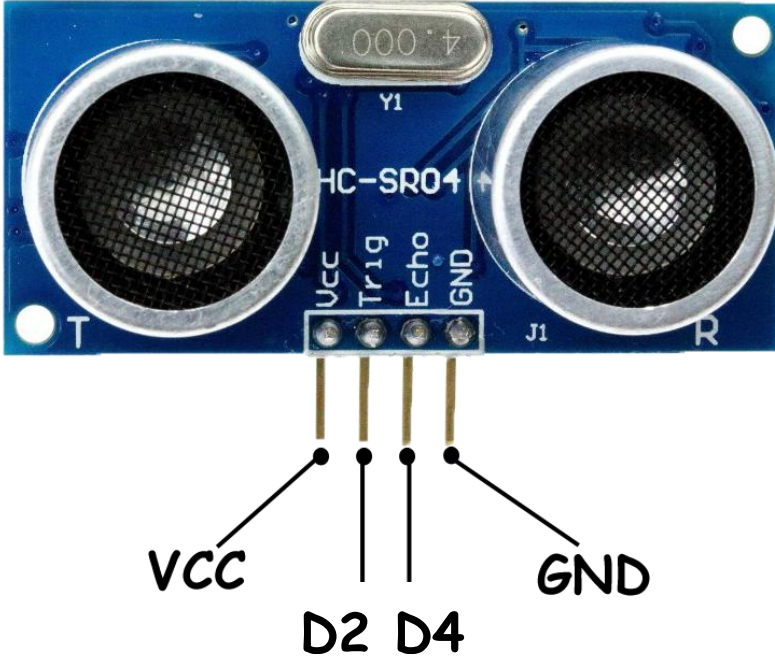


Analog 3 pinine baęlı olan ışık sensöründen 0-1023 arası deęerler okuyoruz. Aydınlıkta 1023'e yakın deęerler okunurken karanlıkta 0'a yakın deęerler okunur.

Bu örneğimizde bir deęişken oluşturarak Analog 3 pininden okunan deęeri bu deęişkene aktarıyoruz. Bir eęer ise deęilse bloęuyla da ışık deęeri belli deęerin altına düştüğünde 12 nolu pine baęlı Led'i yakıyoruz. Eęer ışık deęeri belirttiğimiz deęerin altında deęilse 12 nolu pine baęlı Led'i söndürüyoruz.

**Not:** Biz bu örneğimizde 12 no.lu pine baęlı Led'i yaktık. Siz isterseniz RGB Led ya da istediğiniz başka devre elemanlarıyla uygulamalar geliştirebilirsiniz.

## Mesafe Sensörü



HC-SR04 ultrasonik mesafe sensörü, yarasalarinkine benzer bir sistemle ses dalgası yollayarak ve ses dalgasının dönüş süresini hesaplayarak cisimler arasındaki mesafeyi ölçmemize yarar.

HC-SR04 ultrasonik mesafe sensörünü Scratch Eğitim Kartına bağlamak için **dişi-dişi** bağlantı kablolarını kullanacağız. Ultrasonik sensör üzerinde VCC yazan yeri kart üzerindeki sensör pinlerinden 4'lü olan yerdeki VCC pinine, TRIG pinini D2 yazan pine, ECHO pinini D4 yazan pine, GND pinini de GND yazan pine bağlayacağız.

**Not:** Sensör pinlerini kullanabilmek için kartın üzerinde kırmızı olan 2 ve 3 no.lu anahtarları yukarıya kaldırınız. Eğer RGB ledi kullanmayacaksanız 1 no.lu anahtarı aşağıya indirebilirsiniz.

## Ultrasonik Sensörü Okuma

Bağlantıları kurduktan sonra Mblock programını açalım gerekli ayarları yaptıktan sonra aşağıdaki kodları yazalım ve sensör ile mesafe ölçmeye başlayalım.

Mesafe adında bir değişken oluşturalım.



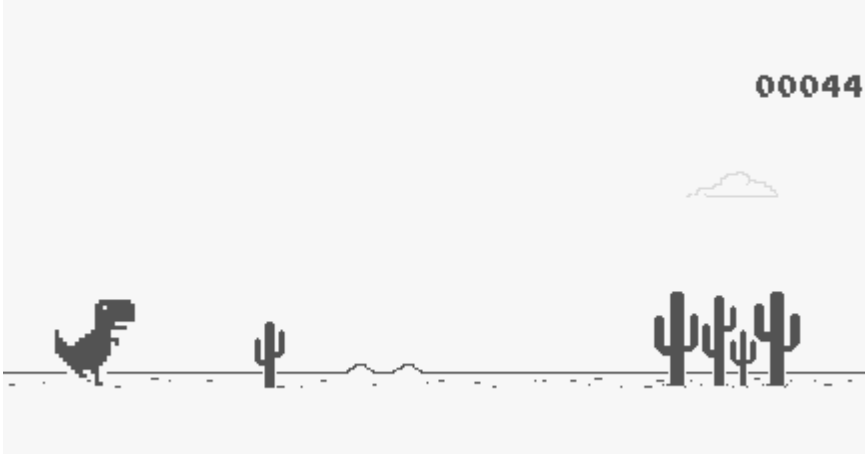
Ultrasonik mesafe sensörünün değerini okumak için gerekli olan kod bloğunu sürükleyerek mesafe değişkeninin ilgili yerine bırakalım.

13 ve 12 yazan yerleri 2 ve 4 olarak değiştirelim.



Mesafe değişkenini bir sürekli bloğunun içine alalım. Artık sensörümüz mesafe ölçümü için hazır.

## Zıplama Oyunu



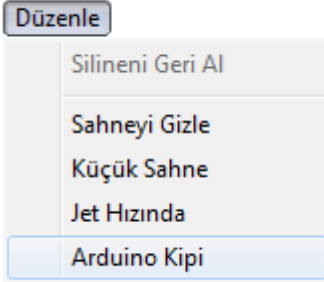
Google Chrome web tarayıcı programında internet olmadığında gelen bir dinazor oyunu vardır. Gelin hep birlikte bu oyunun değişik bir versiyonunu yapalım. Ultrasonik mesafe sensörünü zıplama mesafesine koyalım ve biz zıpladığımızda sensörden okunan değerin düşmesine göre oyundaki karakteri de zıplatalım.

Ultrasonik mesafe sensörünün bağlantılarını yapalım ve kitabın başında verilen kaynak dosyalarının içindeki **dinazorgamemegauno.sb2** dosyasını Mblock programında açalım. Mesafe sensörünü zıplama mesafesine sabitleyelim ve oyunun keyfini çıkaralım.

## Yazdığımız Kodları Arduino'ya Yükleme

Mblock programı ile yazdığımız programlar normalde bilgisayar üzerinden çalışır. Program çalışmayı durdurduğunda ya da bilgisayarı kapattığımızda Arduino programı da durur. Eğer bilgisayar açık değilken de Arduino programının çalışmasını istiyorsanız yazdığınız kodları Arduino'ya yüklemeniz gerekmektedir.

Arduino'ya kodları yüklemek için öncelikli olarak düzenle menüsünden Arduino kipini seçelim.



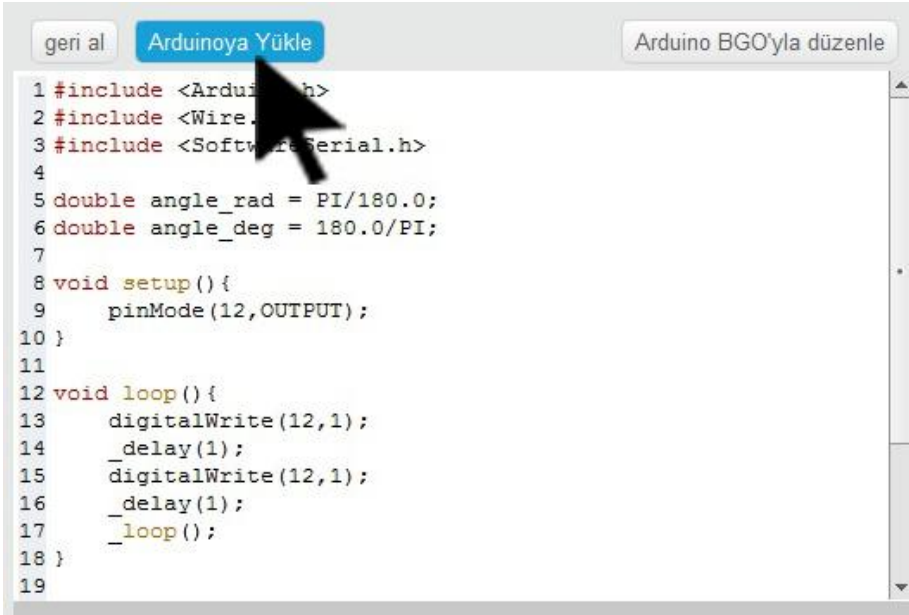
Arduino kipini seçtiğinizde bazı kod bloklarının aktif olmadığını görebilirsiniz. Arduino kipi seçiliyken bütün kod bloklarını kullanamıyoruz. Sadece aktif olan kod bloklarını kullanabiliyoruz.

Arduino kipinde program yazarken başlangıç olarak yeşil bayrak yerine **Arduino programını** kullanıyoruz.



**NOT:** Yazacağımız bütün programlar tek bir Arduino programı altında olmalıdır.

Programımızı yazdıktan sonra **Arduino'ya yükle** butonu ile Arduino karta yüklüyoruz.



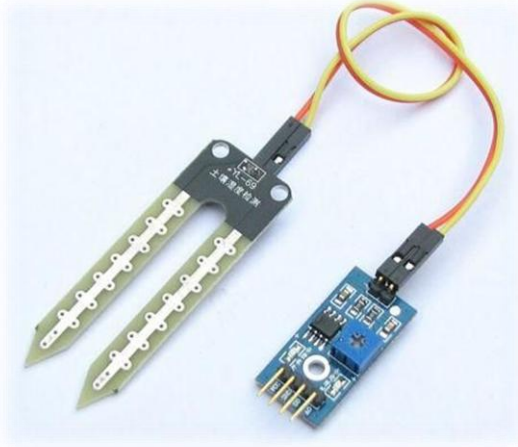
```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 double angle_rad = PI/180.0;
6 double angle_deg = 180.0/PI;
7
8 void setup() {
9     pinMode(12, OUTPUT);
10 }
11
12 void loop() {
13     digitalWrite(12, 1);
14     _delay(1);
15     digitalWrite(12, 1);
16     _delay(1);
17     _loop();
18 }
19
```

Arduino'ya yükle dediğimizde Arduino'nun kendi diline çevrilmiş programı Arduino'ya yüklenecektir.

**NOT:** Arduino'ya program yüklerken bağlan menüsünden kartın bağlı ve kartlar menüsünden Arduino Uno seçili olduğundan emin olun!

**NOT:** Arduino'ya program yüklerken kullandığınız değişken isimlerinde ve özel blok isimlerinde Türkçe'ye has karakterler (ş, ç, ğ, ö, ü, ı) kullanmayınız!

## Toprak Nem Sensörü



Toprak nem sensörünü topraktaki su miktarı ölçmek için kullanırız.

Sensörün uç kısmı toprağa saplanacaktır. Uç kısmına iki kablo aracılığıyla bağladığımız kontrol devresi uç kısımdan aldığı değerleri değerlendirerek arduino karta yollar.

Kontrol kartının üzerindeki GND pinini Scratch eğitim kartının üzerindeki GND pinine, VCC pinini VCC pinine, A0 yazan pinide **A2** pinine bağlayınız.

Eğer topraktaki nem oranı yüksekse 500'e yakın bir değer okunur. Nem oranı düştükçe okunan değer artar. En fazla 1023 değeri okunur.



Sensörden okunan değere göre. Nem oranı iyi olduğunda RGB Led'in yeşil, orta olduğunda RGB Led'in mavi, kötü olduğunda da RGB Led'in kırmızı yandığı ve Buzzer'ın öttüğü bir program yapalım.

Bu program için iç içe eğer ise yapıları kullanacağız. İlk eğer ise bloğunda sensör değerinin 550'de küçük olmasını kontrol edeceğiz. Eğer değer 550'den küçükse nem oranı iyidir. RGB Led yeşil yanar. Değilse kısmına (değer 550'den küçük değilse büyüktür) bir eğer ise değilse bloğu daha koyuyoruz. Bu eğer ise değilse bloğuyla da değer 800'den küçük olup olmadığını ölçüyoruz. Eğer değer 550'den büyük 800'den küçükse RGB Led mavi yanar.

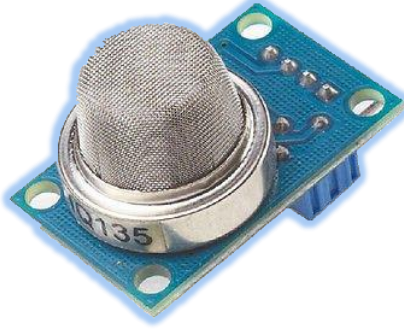


Bu programı Arduino karta yüklüyoruz.

**NOT:** Arduino'ya bir program yükledikten sonra tekrar Arduino kartı bilgisayar üzerinden Scratch ile programlamak için **Aygıt Yazılımı Güncellemesini** tekrar yapınız!

**NOT:** Programdaki değerleri kendi kullandığınız toprağa göre düzenleyebilirsiniz.

## Hava Kalitesi Sensörü (MQ135)



MQ135 hava kalite sensörü havadaki duman ve CO<sub>2</sub> miktarını ölçen bir sensördür.

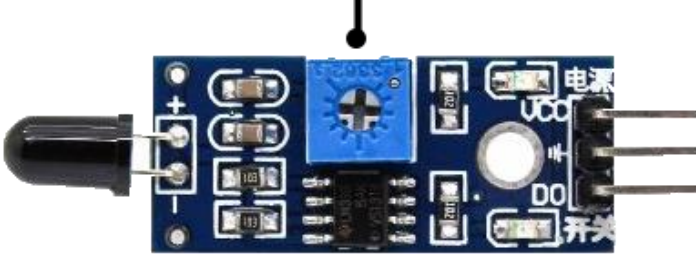
Üzerinde A0, D0, GND, VCC pinleri mevcuttur.

GND ve VCC pinlerini bağladıktan sonra A0 yazan pini A2 ya da A1 pinine bağlayınız ve sensörün değerini okuyunuz.

**NOT:** Sensörü bağladıktan sonra ısınması için biraz bekleyiniz. Okunan değer sabitlendiğinde artık sensörü kullanabilirsiniz. Havadaki CO2 miktarı arttığında ya da duman algılandığında sensörden okunan değer artacaktır. Artan değere göre gerekli kodları yazarak istediğiniz gibi kullanabilirsiniz.

## Alev Sensörü

### Hassasiyet Ayarı



Alev sensörü ateşin dalga boyunu algılayabilir böylece menzil mesafesinde ateş olup olmadığını algılar.

Üzerinde 3 adet pin vardır. GND, VCC, D0.

GND ve VCC pinlerini bağladıktan sonra D0 pinini Scratch eğitim kartının üzerindeki **D2** pinine bağlayınız.

Sensör alev algıladığında 0 değeri algılamadığında ise 1 değeri gönderir.

```
Scratch Script:
- tıkladığında
  - 10 sayısal pini DÜŞÜK yap
  - 7 sayısal pini DÜŞÜK yap
- sürekli tekrarla
  - Sensor 2 sayısal pini oku olsun
  - eğer Sensor = 0 ise
    - 3 pwm pini 0 yap
    - 5 pwm pini 255 yap
    - 9 pwm pini 255 yap
    - 6 ses tonu pini E6 notasında Yarım vuruş çal
  - değilse
    - 3 pwm pini 255 yap
    - 5 pwm pini 0 yap
    - 9 pwm pini 255 yap
```

Alev yokken RGB Led yeşil, alev algılandığında ise RGB Led kırmızı olan ve alarm çalan bir devre yapalım.

Sensörü bağladığımız dijital pinden sensörün değerini okuduğumuz kod bloğu

## Servo Motor



Servo motorlar açılı dönebilen motorlardır. Servo motoru 0-180 arasında istediğimiz bir açığa ayarlayarak döndürebiliriz.

Servo motor üzerinde üç adet kablo bulunur. Kahverengi kablo GND pinine, Kırmızı kablo VCC pinine, Sarı kablo ise D9 pinine bağlanmalıdır.

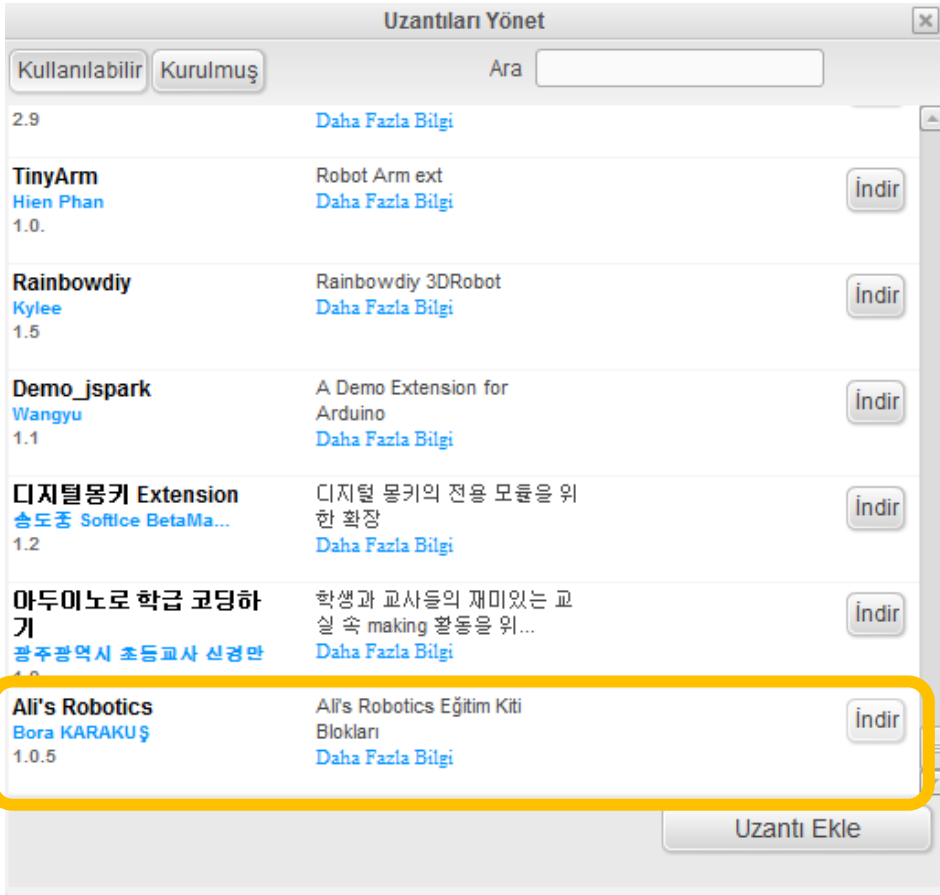
9 servo pini açısını 90° yap

Yukarıdaki kod bloğuyla da servo motoru istediğimiz bir açığa ayarlayabiliriz.

Not: RGB led de 9 no.lu pine bağlı olduğu için kartın üzerindeki kırmızı anahtarlardan 1 no.lu anahtarı aşağıya indirerek RGB ledi kapatınız.

## Uzantı Ekleme

Mblock'un kendi kod bloklarından başka kod blokları kullanmak için uzantı ekleyebiliriz. Uzantılar menüsüne girdikten sonra uzantıları yönet kısmından istediğiniz uzantıyı indirebilirsiniz. İsterseniz eğitim kartının daha kolay programlanması için kartın pinlerine göre ayarlanmış uzantımızı indirebilirsiniz.

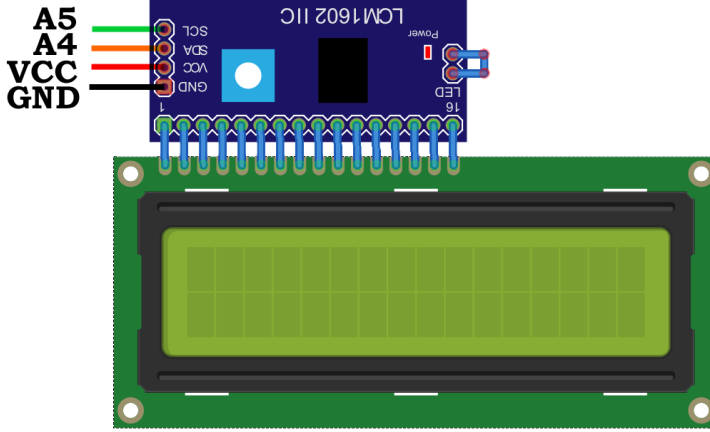


İndir butonuna tıkladıktan sonra uzantı eklenecektir.

Youtube Videosu İçin Tıklayınız!



## Lcd Ekran (I2C)



Lcd ekranları arduinodaki verileri kullanıcıya aktarmak için kullanırız. Burada kullandığımız lcd ekran 16x02 bir lcd ekran. Ekranda 2 satır 16 sütun bulunmakta. Bu lcd ekranı Arduino karta bağlamak için **I2C** bağlantılı olanı kullanıyoruz.

Lcd ekranımızı programlamak için Mblock programındaki kod bloklarını kullanamıyoruz. Ekranı kullanabilmek için bir uzantı eklememiz gerekmektedir. Uzantıyı eklemek için uzantılar menüsünden uzantıları yönet diyerek aşağıdaki resimde bulunan uzantıyı ekleyelim.

LCD1602 with I2C  
Interface

EditBy AiYing|QQ:757447286  
1.01

Using LCD 1602 and 2004  
with I2C interface in mBlock.

[Daha Fazla Bilgi](#)

İndir

İndir butonuna tıkladığımızda kod blokları eklenecektir.



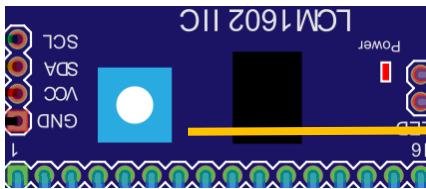
Lcd ekran üzerindeki SDA pinini A4 pinine SCL pinini A5 pinine bağlayınız.

```
Arduino Programı
1 Set: LCD at 0x27 has 2 lines and 16 characters per line
2 Clear LCD at 0x27
3 Turn ON the backlight of LCD at 0x27
4 LCD at 0x27 Line 1 Col 1 Show Hello World!
```

- 1 İlk önce Lcd ekranımızı tanıtıyoruz.
- 2 Ekrandaki yazıları temizliyoruz.
- 3 Ekranın arka plan ışığını ayarlıyoruz.
- 4 Ekrana Hello World! yazdırıyoruz.

**NOT:** Lcd ekran kod bloklarını sadece Arduino Mode ile kullanabiliyoruz.

Lcd ekranı kullanırken Türkçe'ye has karakterleri (ş,ç,ğ,ü,ı,ö) kullanmayınız.



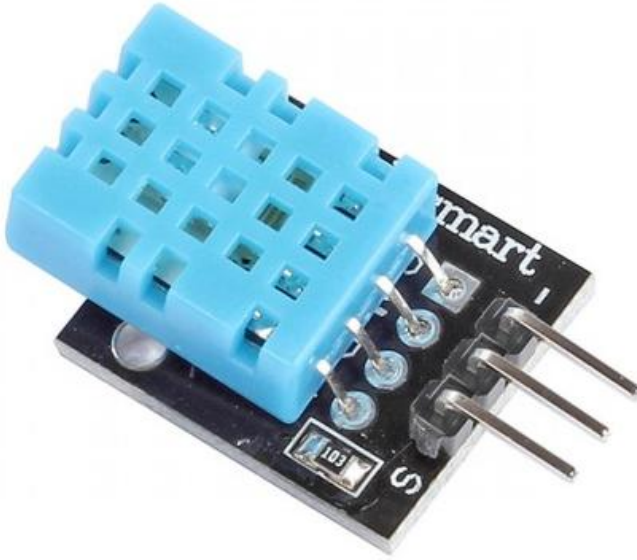
Ekran ışığı ayarı

Ekranında yazılar gözüküyor ya da silik gözüküyorsa I2C modülü üzerinde bulunan potansiyometre ile ekran ışığını ayarlayabilirsiniz.

## DHT11 Sıcaklık ve Nem Sensörü

DHT11 sıcaklık ve nem sensörü ortamdaki sıcaklık ve nem değerlerini ölçerek dijital pin üzerinden Arduinoya aktarabilen, kullanımını ve Arduino bağlantısı oldukça basit olan bir sensördür.

DHT11 sıcaklık ve nem sensörü 0-50°C arasında ve  $\pm 2$  ° C hassasiyetle sıcaklık, % 20-80 arasında nem değeri ölçebilmektedir.



Sensörün üç adet pini bulunmakta. S pini standart olarak 2 no.lu dijital pine, (-) pini GND pinine diğer pin de VCC pinine bağlanmalıdır.

DHT11 sıcaklık ve nem sensörünü kullanabilmek için bir uzantı indirmemiz gerekmektedir.

**DHT Extensions**  
Bora KARAKUŞ  
1.0

DHT Extension for mBlock  
[Daha Fazla Bilgi](#)

İndir

Uzantıyı indir dediğimizde uzantılar kısmına gerekli kod blokları gelecektir.

DHT No 1 , Pin 2 , Version DHT11

Yukarıdaki kod bloğuyla DHT11 sensörümüzü tanımlıyoruz. No kısmından sensörümüze numara verebiliriz. Sadece bir DHT bağlıysa 1 olarak kalabilir.

Pin kısmından sensörün bağlı olduğu pini seçiyoruz. Standart olarak 2 gelir.

Versiyon kısmında DHT sensörümüzün türünü seçiyoruz. DHT11, DHT21 ya da DHT22 seçilebilir.

DHT No 1 , Temperature

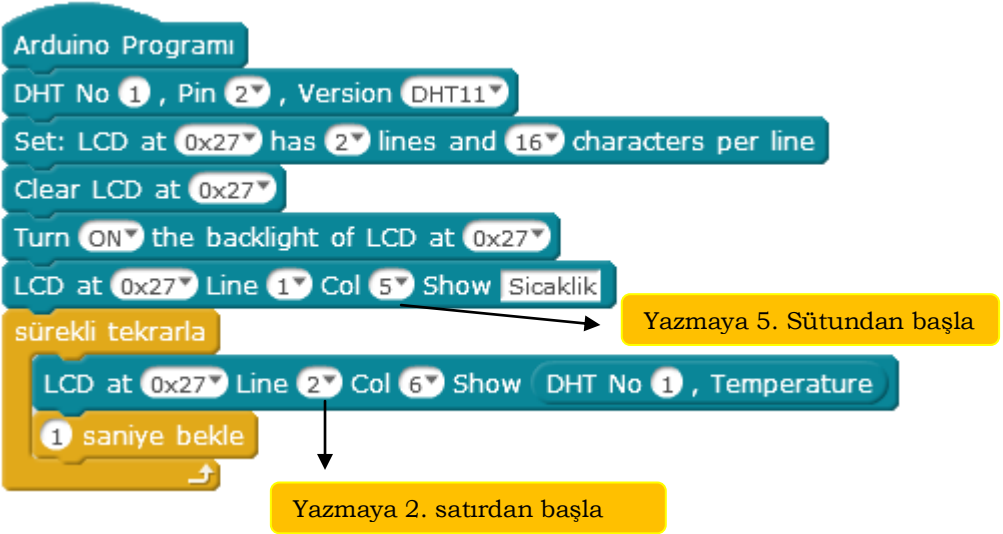
DHT No 1 , Humidity

Temperature kod bloğuyla sensörün sıcaklık değerini, Humidity kod bloğuyla da sensörün nem değerini elde ediyoruz.

## DHT11 ve LCD Ekran

DHT sensörümüzü sadece Arduino Mode ile kullanabildiğimiz için aldığımız değerleri yazdırmak için Lcd ekran kullanacağız.

Öcelikli olarak DHT sensörümüz ve Lcd ekranımızı kitapta verilen bağlantı şemasıyla eğitim kartımıza bağlayalım. Daha sonra aşağıdaki kod bloklarını yazalım ve Arduino'ya yükleyelim.



```
Arduino Programı
DHT No 1 , Pin 2 , Version DHT11
Set: LCD at 0x27 has 2 lines and 16 characters per line
Clear LCD at 0x27
Turn ON the backlight of LCD at 0x27
LCD at 0x27 Line 1 Col 5 Show Sıcaklık
sürekli tekrarla
LCD at 0x27 Line 2 Col 6 Show DHT No 1 , Temperature
1 saniye bekle
```

Yazmaya 5. Sütundan başla

Yazmaya 2. satırdan başla

Not: Uzantıyı ekledikten sonra kitabın başında verilen kaynak dosyaların içindeki örnek kodu açıp yükleyebilirsiniz.

## Uzaktan Kumanda ile Led Kontrolü



Uzaktan kumanda infrared(kızılötesi) sistemini kullanarak haberleşir. Kumanda üzerinde herhangi bir tuşa bastığımızda bir kod gönderir. Arduino'ya bağlayacağımız modül ise bu kodu alarak kullanmamızı sağlar.

Modülü kullanmak için aşağıdaki uzantıyı indirmemiz gerekmektedir.

**IRremote**  
Abaddon  
0.1.3

Function from library IRremote  
(Offline only)  
[Daha Fazla Bilgi](#)

İndir

Her kumandanın göndereceği kod farklı olabilir. Bizim kullandığımız (resimdeki) model için aşağıdaki kodlar yollarır.

Modülün bağlantıları (-) pini → GND, S pini → D2, ortadaki boşta kalan pin de VCC pinine bağlanmalıdır.



Öncelikli olarak **veri/blok** menüsü altında bulunan bir blok oluştur kısmına tıklayalım ve TuslariTanimla adında bir blok oluşturuyoruz ve gelen Tanımla kısmının altına tuşların kodlarını yazıyoruz. Her tuşun kodunu oluşturduğumuz bir değişkene aktarıyoruz.

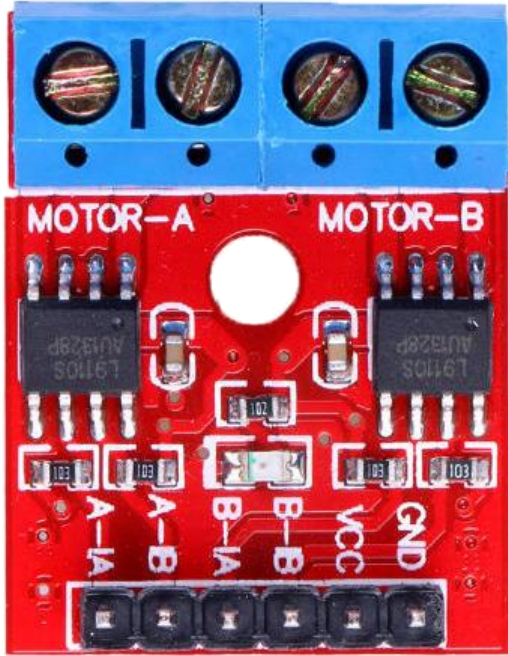
```
Arduino Programı
IRremote pin ( 2 )
TuşlarıTanımla
sürekli tekrarla
  eğer boolean result receive ise
    seri porta Value Results yaz
    eğer Value Results = Tus1 ise
      12 sayısal pini YÜKSEK yap
    eğer Value Results = Tus0 ise
      12 sayısal pini DÜŞÜK yap
  Resume IR receive
```

- 1 Modülü bağladığımız pini seçiyoruz.
- 2 Oluşturduğumuz blokla tuşların kodlarını tanımlıyoruz
- 3 Sürekli kod bloğunun içine bir eğer ise kod bloğu koyuyoruz ve **boolean result receive** kod bloğu ile kumandadan mesaj gelip gelmediğini kontrol ediyoruz.
- 4 Kumandadan gelen mesajı görebilmek için seri porta yazdırabiliriz.
- 5 Mesaj geldiğinde gelen mesaj Value Results kod bloğuna kaydedilir. Bu gelen mesajı karşılaştırıyor ve eğer ise bloğuyla istediğimiz görevi veriyoruz.
- 6 Mesajı alıp işlemleri yapıp mesajı sonlandırıyoruz.

## Motor Sürücü Kartları



Oyuncak araba motoru

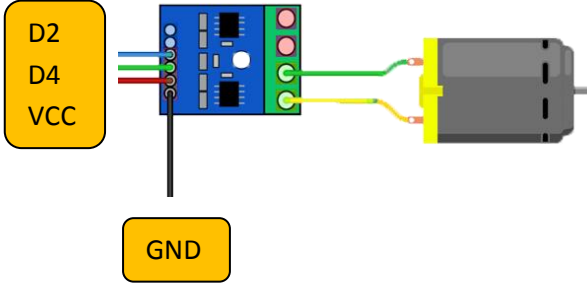


Robotları hareket ettirmek için kullandığımız motorları, direkt Arduino karta bağlayıp çalıştıramıyoruz. Arduino'nun gücü buna yetmiyor. Motorları çalıştırmak için ara devrelere ihtiyacımız var. Motor sürücü kartlar bu ihtiyacımızı karşılar.



Çok güç gerektirmeyen işlerinizde L9110 çift motor sürücü kartı kullanabilirsiniz. 2.5V – 12V arası Voltaj değerlerinde çalışır. Oyuncak araba motorlarını bağlayabilirsiniz. Toplamda iki adet motor bağlayabilirsiniz.

A-IA ve A-IB girişleri Motor-A için, B-IA ve B-IB girişleri Motor-B kontrolü için kullanılır. İki motor girişini de düşük yaparsanız motor durur. İki motor girişinden birini yüksek diğerini düşük yaparsanız motor bir yöne döner. Girişlerin yüksek, düşük sırasını değiştirirseniz diğer yöne döner.



```
when clicked
  loop
    2 digital pin HIGH
    4 digital pin LOW
    1 second wait
    2 digital pin LOW
    4 digital pin LOW
    1 second wait
    2 digital pin LOW
    4 digital pin HIGH
    1 second wait
    2 digital pin LOW
    4 digital pin LOW
    1 second wait
```

Motor sürücü kartının ve motorun bağlantılarını yaptıktan sonra yukarıdaki kodlarla bir saniye bir yöne bir saniye ara bir saniye diğer yöne bir saniye ara ile döndürebiliriz.