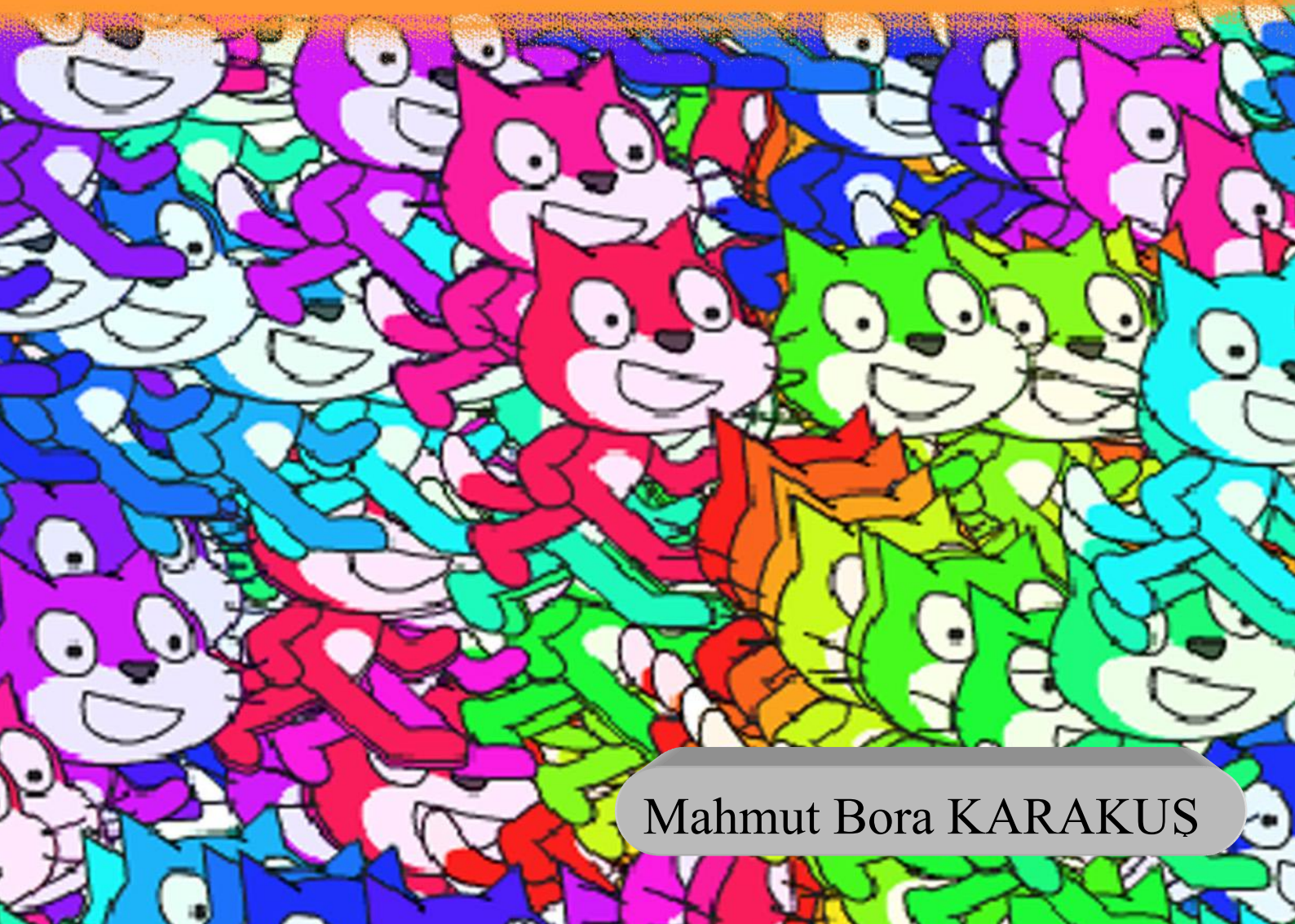




SCRATCH

ile

Programlamaya Giriş



Mahmut Bora KARAKUS



Bütün ümidim gençliktedir.

Mustafa Kemal ATATÜRK



Hayallerinle dünyanı programla



Yazar Hakkında



yapmaktayım.

• İlk ve orta öğrenimimi Mersin'de tamamladım.

• Çanakkale 18 Mart Üniversitesi Bilgisayar ve Öğretim Teknolojileri Öğretmenliği mezunuyum.

• Ahmet Yesevi Üniversitesi Bilgisayar Mühendisliği 3. Sınıf öğrencisiyim.

• Bir devlet okulunda Bilişim Teknolojileri ve Yazılım Dersi öğretmenliği



Scratch ile Programlamaya Giriş E-Kitabı Creative Commons Alıntı-Gayriticari-Türetilemez 4.0 Uluslararası Lisansı ile lisanslanmıştır.

<http://www.bilbilimbilisim.com/blog/scratch-kitabi/>

(Kaynak gösterilmeden paylaşılamaz, Ticari olarak kullanılamaz ve Üzerinde değişiklik yapılamaz)

İÇİNDEKİLER

| | |
|--|-----------|
| BÖLÜM 1: PROGRAMLAMA | 8 |
| Programlama Nedir? | 9 |
| Niçin Programlama Öğrenmeliyiz? | 10 |
| Algoritma Nedir? | 11 |
| Minecraft ile Algoritma Öğreniyorum | 13 |
| BÖLÜM 2: SCRATCH ile PROGRAMLAMA ÖĞRENİYORUM | 23 |
| Scratch Nedir? | 24 |
| Scratch Offline Editör | 25 |
| Hareket ve Görünüm | 34 |
| Ses ve Kalem | 43 |
| Kontrol ve Algılama | 47 |
| Koşu Yarışması | 49 |
| Canım Annem Animasyonu | 55 |
| İşlemler ve Veri | 58 |
| Sınıf Arkadaşlarım Uygulaması | 64 |
| Tıklama Oyunu | 67 |
| Puan Tablolulu Tıklama Yarışması | 75 |
| Mesajlaşma ve Özel Taşlar | 79 |
| Bölümlü Labirent Oyunu | 80 |
| BÖLÜM 3: SCRATCH ONLINE EDITÖR | 92 |
| Scratch İnternet Sitesine Üye Olmak | 94 |
| Scratch Online Editör | 97 |
| Scratcher Olmak | 99 |
| Bulut Değişken | 101 |
| Dünya Rekoru Yapma | 104 |
| Offline Editörde Yaptığımız Çalışmayı Online Editöre Aktarma | 106 |

| | |
|--|------------|
| BÖLÜM 4: SCRATCH İLE OYUN PROGRAMLAMA | 108 |
| İlk Bilgisayar Oyunu | 110 |
| Elma Yakalama Oyunu | 111 |
| Uçan Kedi | 119 |
| Yılan Oyunu | 126 |
| Giydirme Oyunu | 134 |
| Scratch'da Bir Oyunu Hacklemek | 152 |



Robotik çalışmalar için Scratch ile Arduino Programlama kitabımı okumak için [Tıklayınız!](#)

1

Programlama

Programlama, bilgisayarın isteđimiz işleri yapabilmesi için ne yapması gerektiđinin adım adım yazılmasıdır.

Algoritma, programlardaki komutların yapılış sıralamasıdır. Programlar yazılırken algoritmalar kullanılır.

Bu bölümde programlama ve algoritma nedir? konularına değineceđiz ve code.org internet sitesine girerek minecraft oynayacađız ve ilk algoritmalarımızı oluřturarak programlarımızı yazacađız.

PROGRAMLAMA

Programlama Nedir?

Programlama, bilgisayarın ne yapması gerektiğinin adım adım söylenmesidir. Örneğin; fareye tıklandığında sayfayı kapat, boşluk tuşuna basıldığında oyundaki karakteri zıplat. Bir bilgisayar, bu işleri nasıl yapması gerektiğini bilmez. Bu yüzden bilgisayarın bir işi yapabilmesi için programlanması gerekmektedir.

Bilgisayarlarda çalışan birçok program vardır. Resimleri açmak veya internete bağlanmak için bilgisayardaki bu programları kullanırız. Bu programlar bilgisayarın çalışması için gereken kılavuzlardır. Bilgisayardaki bu programları oluşturmak için programlama dillerini kullanırız. İnsanların günlük hayatta iletişim kurmak için nasıl bir dil kullanmaları gerekiyorsa bizim de bilgisayarlar ile haberleşebilmemiz için bir dil kullanmamız gerekir. Bu dillerden bir tanesi de Scratch programlama dilidir.

Programlama dillerinin, bizim kullandığımız dillerden farkı, yazılı olmalarıdır. Programlama dillerini kullanarak adım adım bilgisayarın yapması gereken işleri yazarız. Bilgisayarlar da bu işleri sırasıyla bizim belirttiğimiz şekilde yaparlar.



Scratch MIT (Massachusetts Institute of Technology) üniversitesi tarafından geliştirilen bir görsel programlama dilidir. Bu dil özellikle 9-16 yaş grubu öğrenciler için oluşturulmuştur. Bu dili kullanarak kendi oyununuzu programlayabilir, kendi animasyonlarınızı oluşturabilir ve yaratıcılığınız dahilinde birbirinden güzel çalışmalar üretebilirsiniz.

Scratch'da program yazmak çok kolaydır. Scratch'da bilgisayara verilecek komutlar, bloklar şeklindedir. Size düşen ise lego yapar gibi bu blokları birleştirmektir.

Niçin Programlama Öğrenmeliyiz?

Dünya çok hızlı bir şekilde gelişmekte ve bu gelişme her alanda kendini göstermektedir. Bugünün öğrencileri, yarın meslek hayatına atıldığında karşılaşacakları dünya çok farklı olacaktır. Belki de sizin uğraş edindiğiniz meslek bugün olmayan bir meslek olabilir. İngiliz Hükümetince yapılan bir araştırmaya göre, 2030 yılının en popüler meslekleri içerisinde "**vücut parçası imalatçılığı**" da bulunuyor. Bu gelişim sonucunda yeni meslekler ortaya çıkacağı gibi mevcut olan mesleklerde bu değişimden etkileneceklerdir. Örneğin; cerrahi operasyonlar robotlarla yapılacak ve robotik cerrahi alanında uzman olmayan doktorlar cerrahi operasyonlar yapamayacaklar.



Dünyadaki gelişmelerin zorunlu kılması üzerine bugün gelişmiş ülkelerin birçoğunda programlama eğitimi ilkokuldan itibaren verilmeye başlanmıştır. Ülkemizde de kodlama dersinin ilkokullardan itibaren zorunlu olması planlanmaktadır. Bu derslerin ilkokuldan itibaren verilmesinin amacı, öğrencilerin hepsinin birer bilgisayar mühendisi olması değildir. Bu derslerin öğrencilere verilmesinin nedeni 21. Yüzyılın zorunlu kıldığı yeteneklerin öğrencilere kazandırılmasının sağlanmasıdır.

Bu yeteneklerden bazıları şunlardır:

- Bilgisayarca düşünme

Bilgisayarları üretim amaçlı olarak hayat problemlerinin çözümünde kullanabilmek için gerekli olan bilgi, beceri ve tutumlara sahip olmak.

- Analitik düşünme

Bir problemi çözmek için, bilgileri küçük parçalara ayırmak ve küçük parçalar üzerinden problemi çözmek.

- Yaratıcı düşünme

Elde mevcut bulunan bilgi ve imkanlarla yeni ve işe yarar bir şeyi meydana getirmek.

- Problem çözme

Hayatta karşılaşılan güçlük, zorluk ve problemlere çözüm getirebilme becerisi.

Facebook, Twitter, Apple gibi büyük şirketlerin kurucularının başarılı olmalarının altında yatan ortak etkenlerden biri de küçük yaşlarda programlama öğrenmeleridir. Bu programlama yetenekleri onları diğer insanlardan daha avantajlı kılmıştır. Bu nedenden ötürü bugün öğrencilere küçük yaşlardan itibaren programlama öğretilmektedir. Programlama öğrenmek, olaylar arasındaki ilişkiyi görme becerisini, sistematik düşünme becerisini ve problem çözme becerisini artırır. Bilgisayarda program yazarken algoritmalar geliştirilir. Algoritma problemin adım adım çözülme sürecidir. Bu da bize olayları detaylı düşünebilme ve çözebilme yeteneği kazandırır.

“Bu ülkedeki herkes bilgisayar programlamayı öğrenmeli çünkü insana düşünmeyi öğretir.”

Steve JOBS

Algoritma Nedir?

Algoritma, bir işin yapılması için izlenecek adımlardır. Bilgisayarda programlar yazılırken algoritmalar kullanılmaktadır. Algoritmalarla biz bir işi parçalara böleriz ve sırası ile o parçaları çözeriz. Bu durum bizim olayları detaylı bir şekilde düşünmemizi ve kolayca çözmemizi sağlar.

Hepimiz her gün sabah kahvaltı yaparken, dişimizi fırçalarken ya da uyumaya giderken algoritmaları kullanırız. Algoritma bu yaptığımız işlerin yapılış sıralamasıdır.

Hayatımızdaki Algoritmalar

Ayakkabı Giyme Algoritması:



1. Ayakkabıyı Raftan Al



2. Ayakkabıyı Giy



3. Bağcıklarını Baęla

Diş Fırçalama Algoritması:



1. Kirli Dişler



2. Fırçaya Diş Macunu Sür



3. Dişlerini Fırçala



4. Temiz Dişler

Çiçek Dikme Algoritması:



1. Saksıyı

Toprakla

Doldur



2. Topraęı

Karıştır



3. Tohumu

Dik



4. Tohumu

Toprakla

Karıştır



5. Çiçeęi

Sula



6. Güneşli

Bir

Yere

Koy

Algoritmalarla olayları basamaklara böleriz ve sırasıyla o basamakları izleriz. Algoritmalar programlamada önemli bir yer tutar. Yapacağımız işin algoritmasını çıkarmak o işi detaylı düşünmemizi ve küçük parçalar halinde daha kolay yapmamızı sağlar.

Minecraft ile Algoritma Öğreniyorum



MINECRAFT

Steve

Saat Kodu

Steve ya da Alex'i Minecraft dünyasında bir maceraya çıkarmak için kod blokları kullan.

Çoklu dil | Modern tarayıcılar ve tabletler | 6 ve üzeri yaş

[Şimdi dene](#)

Minecraft hemen hemen herkesin çok sevdiği bir oyun. Biz de algoritmaları kullanarak Minecraft oynayacağız.

<https://code.org/mc> adresine girip şimdi dene butonuna tıklıyoruz.

Hadi başlayalım.

Karakterinizi seçin.

Steve



Seç

Alex



Seç

Bir karakter seçiyoruz.

Minecraft Bölüm 1

MINECRAFT

blok

Çalışma alanı: 2 / 3 bloklar

ilerle

sola dön ↶

sağa dön ↷

1

Karakterimizin verilen görevleri yapabilmesi için kod bloklarını çalıştığı zaman bloğunun altına sürükleyip bırakıyoruz

2

Çalıştığı zaman

ilerle

Burada karakterimizin koyuna ulaşması gerekmektedir. Bunun için bir ilerle bloğunu çalıştığı zaman bloğunun altına ekliyoruz.

3

Programı test etmek için çalıştırı tıklıyoruz

Çalıştır

Koyunlara ulaşmak için ikinci bir "ilerle git" komutu ekleyin.

Karakterimizin koyuna ulaşabilmesi için iki adım ilerlemesi gerekmektedir. Algoritma kurarken yapacağımız işleri adımlara böleriz ve sırasıyla o adımları izleriz. Burada da yapacağımız işleri adımlara bölelim. Karakterimizin önce bir adım ilerlemesi ve sonra bir adım daha ilerlemesi gerekmektedir. Çalıştığı zaman bloğunun altına bir adet ilerle bloğu ekleyelim ve çalıştır diyelim. Gördüğümüz gibi algoritma oluşturmak bu kadar basit.

Minecraft Bölüm 2



Minecraft'ta bir şeyler inşa edebilmemiz için odun parçalarına ihtiyacımız var. Burada da ağaca doğru ilerledikten sonra "bloğu yok et" komutunu kullanarak odun parçaları elde edeceğiz. Algoritmamızı oluştururken dikkat etmemiz gereken nokta adım adım düşünmek ve yapacağımız işleri adım adım yapmak.

Çalıştığı zaman

ilerle

ilerle

bloğu yok et

Minecraft Bölüm 5



Bu bölümde karakterimiz aynı işi birden fazla kez yapmak zorundadır. Bazen bilgisayarlarda programlar yazılırken aynı işin tekrar tekrar yapılması gerekebilir. Bu durumda aynı komutları tekrar tekrar yazmak hem yorucu hem de sıkıcı olacaktır. Bilgisayarlar tekrar eden komutları tekrarla bloğuyla kolayca tekrarlayabilir.



Karakterimiz dört kez **yerleştir huş kalaslar** ve **ilerle** komutunu uygulamalıdır. Bu komutları alt alta ve tekrar tekrar yazmak yerine bir **tekrarla** bloğu ile bu komutları tekrarlayabiliriz. Bu bizim aynı işi daha az kod bloğuyla ve daha az zahmetle yapmamızı sağlar.

Minecraft Bölüm 7



Bazen tekrar eden blokların tekrarını bitirdikten sonra yine tekrar etmesi gerekebilir. Bu durumda iç içe tekrar blokları kullanırız. Karakterimiz altı kez **hasat et** ve **ilerle** komutunu uyguladıktan sonra **sağa dön**, **ilerle**, **ilerle**, **sağa dön**, **ilerle** komutlarını uygulamalıdır. Döngü bittikten sonra tekrar başa dönerek aynı işleri bir kez daha yapmalıdır. Bu yüzden komutlarımız iki kez tekrarla bloğu içine alınmıştır.

Eğer ise Yap Bloğu

Bilgisayarların bir işi yapıp yapmayacaklarına ya da ne zaman yapacaklarına dair karar almaları gerekebilir. Biz de zaman zaman karar almak zorunda kalırız. Örneğin; telefonun şarjı bitmişse telefonu şarja tak. Burada bir koşulumuz var, telefonun şarjının bitmesi bu koşul gerçekleşirse bir eylem yapıyoruz yani telefonu şarja takıyoruz. Bilgisayarlarda buna benzer kararları almak zorundadırlar. Eğer blokları karar bloklarıdır. Karar blokları programlamada çok önemli bir yer tutar. Kullandığımız bilgisayarlarda, akıllı telefonlarınızda hatta Mars'a gönderilen robotta bile bu karar yapıları kullanılır.

Facebook internet sitesine giriş yaptığınızda sizden kullanıcı adı ve şifre ister. Bu kullanıcı adı ve şifre sistemde kayıtlı olan kullanıcı adları ve şifrelerle karşılaştırılır ve eğer kullanıcı adı ve şifre doğruysa sisteme giriş yaparsınız. Burada siteye giriş yapmak için yapılan kontrol, eğer yapısı kullanılarak yapılır.

Çarpışmayan Arabalar



Lunaparklarda çarpışan arabalar en sevilen oyuncakların başında gelir. Çarpışan arabalarda amaç mümkün olduğunca çok araca çarpmaktır. Trafikte ise maddi hasarlara hatta ölümlere neden olan çarpışmalar istenmeyen bir durumdur. Bu durumu çözmek için

mühendisler arabalar arasındaki mesafeyi ölçmekte ve yazdıkları bir programla mesafeye göre arabanın otomatik olarak fren yapmasını sağlamaktadırlar. Arabada bulunan bilgisayar bir karar alarak eğer mesafe kısaysa fren yap komutunu uygulamaktadır. Bu çözüm, kaza sayısında ciddi oranda düşüş sağlamıştır.



Minecraft Bölüm 11



Bu bölümde karakterimizin yolunda bazı kısımlarda lav dolu boşluklar var. Bu lav dolu boşluklara düşmemek için bu boşlukların doldurması gerekmektedir. Burada karakterimiz önünde lav dolu boşluk olup olmadığını kontrol etmeli ve eğer lav dolu boşluk varsa doldurmalıdır.



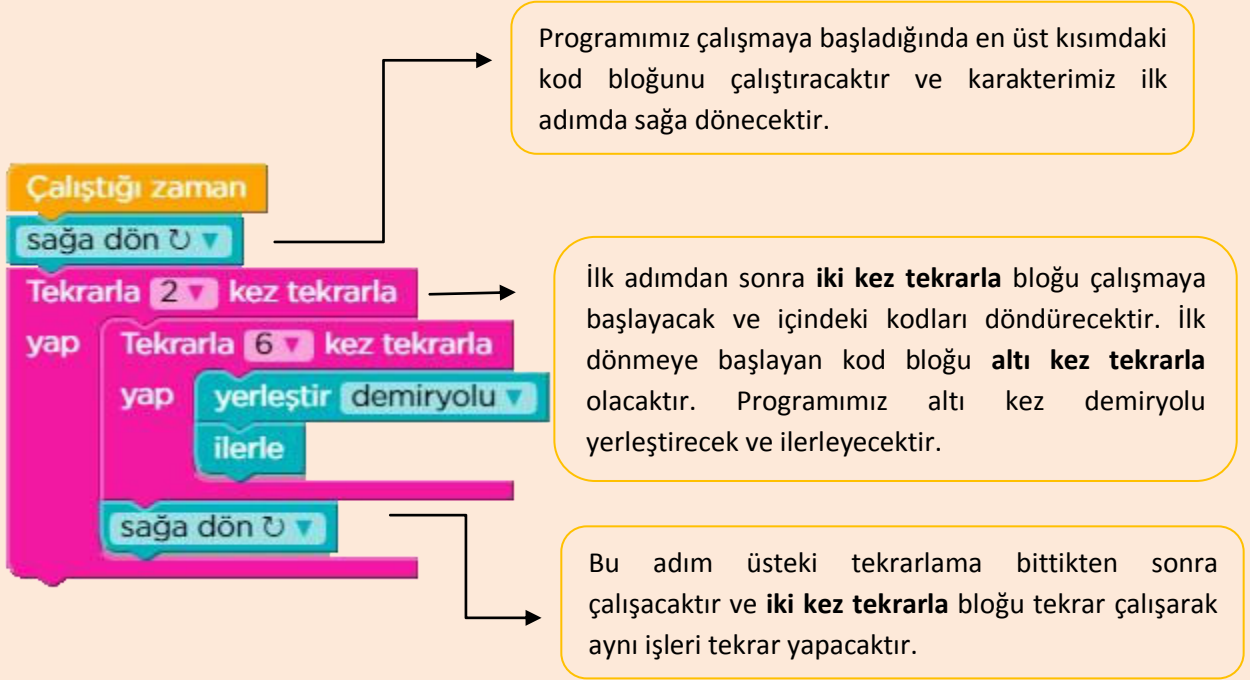
Programımız yedi kez önündeki bloğu yok edecek, eğer ise yapısı ile eğer ileride lav varsa **yerleştir kırıktaş ileri** bloğunu çalıştırıp boşluğu dolduracak ya da lav yoksa bu bloğu çalıştırmayarak pas geçecek ve ilerleyecektir.

Minecraft Bölüm 13

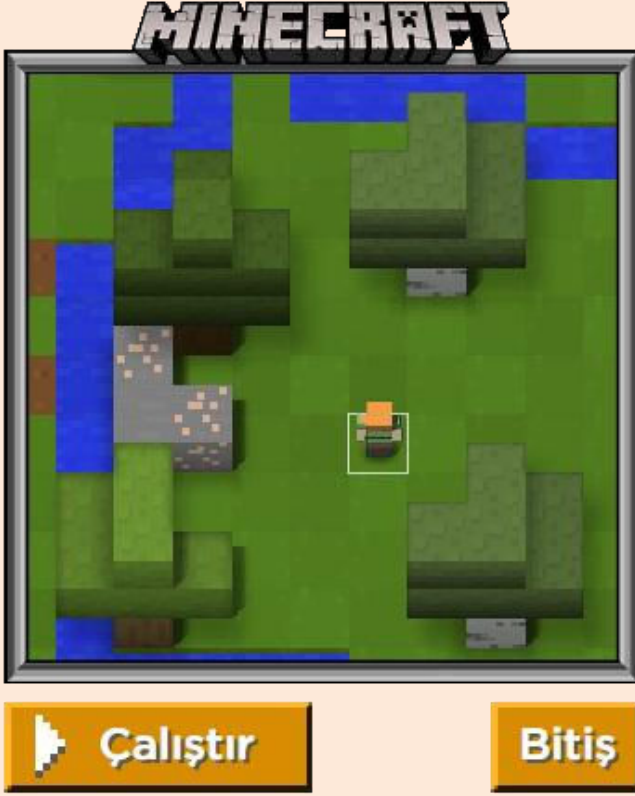
Şimdiye kadar birçok malzeme topladık şimdi bu malzemelerle bir tren yolu inşa edelim.



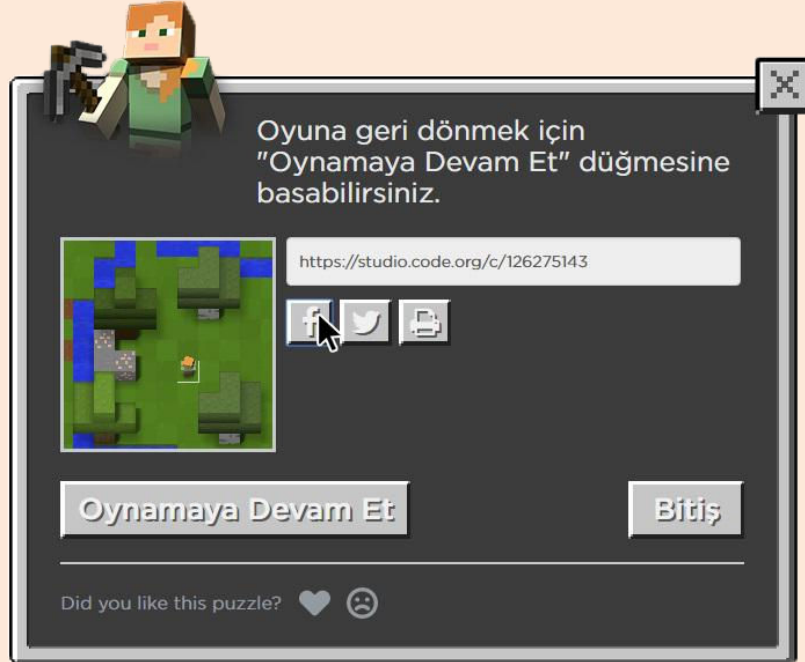
Bu bölümde karakterin yapması gereken işlemleri önce düşünmelisiniz. Programınızı yazmaya başlamadan önce algoritmasını çıkarmalısınız. Algoritma ile yapacağınız işleri sıralayıp, hangi işi hangi sırayla yapmanız gerektiğini belirlemelisiniz. İşler sıraya dizildiği için yazacağınız programı en kolay yoldan yazabilirsiniz. Böylece fazla kod yazma zahmetinden de kurtulmuş olursunuz.



Minecraft Bölüm 14



Son bölümde dünyanızı inşa etmekte serbestsiniz. Dünyanızı istediğiniz gibi inşa edebilirsiniz. Kullanacağınız kod bloklarını ve yapacağınız işleri siz belirleyin. Programınızı test etmek için çalıştır butonunu kullanın. Eğer programınız bittiyse bitiş butonunu kullanarak çalışmanızı arkadaşlarınızla paylaşabilirsiniz.



Bitiş butonuna bastıktan sonra dilerseniz çalışmanızı Facebook'ta arkadaşlarınızla paylaşabilirsiniz.









Code.org

Code.org internet sitesi Google, Microsoft, Facebook, ve Twitter mühendislerinin yardımıyla oluşturulan 7'den 70'e herkesin programlamanın temellerini öğrenebileceği bir platformdur. Bu platformda sadece Minecraft temalı oyunlar değil aynı zamanda flappy Bird, Angry Birds, Star Wars temalı oyunları da bulabilirsiniz.

“Tüm okullardaki bütün öğrencinin, bilgisayar bilimlerini öğrenmek için fırsatı olmalıdır.”

Code.org

<https://studio.code.org> adresine girdikten sonra istediğiniz bir oyunu seçebilirsiniz.

| | | | |
|--|--|---|--|
|  <p>Star Wars</p> <p>Droidleri programlamayı ve çok çok uzaktaki bir galakside kendi Yıldız Savaşları oyununu yaratmayı öğren.</p> |  <p>Minecraft</p> <p>Bir Minecraft dünyasını kodlamayla keşfet</p> |  <p>Donmuş</p> <p>Buzun büyüü ve güzelliğini keşfetmek için Anna ve Elsa'ya katılıp kodlama yapalım.</p> |  <p>Klasik Labirent</p> <p>Temel bilgisayar bilimlerine milyonlar bir şans verdi, sen de dene.</p> |
|  <p>Flappy Kod</p> <p>10 dakikadan az bir süre içinde kendi oyununu yazmak ister misin? Flappy Kod eğitimimizi deneyin!</p> |  <p>Sınırsız oyun la...</p> <p>Bir hikaye ya da Disney Infinity karakteriyle bir oyun yaratmak için Play Lab'i kullanın.</p> |  <p>Oyun Laboratu...</p> <p>Oyun Laboratuvarı ile hikaye yarat veya oyun yap!</p> |  <p>Aktör</p> <p>Sanatçı ile harika resimler ve tasarımlar yap!</p> |

2

Scratch ile Programlama Öğreniyorum

Scratch, kendi oyunlarınızı, çizgi filmlerinizi ve programlarınızı yapabileceğiniz görsel bir programlama dilidir. Kolay olduğu kadar eğlencelidir.

Bu bölümde Scratch programını ve Scratch bloklarını tanıyacağız ve örnek uygulamalar geliştireceğiz.

SCRATCH ile PROGRAMLAMA ÖĞRENİYORUM

Scratch Nedir?

Scratch, programlarınızı görsel olarak yazabileceğiniz bir programlama dilidir. Scratch ile programlama öğrenmek hem basit hem de zevklidir. Scratch'da kod yazmak için blokları kullanırız. Blok sistemi sürükle bırak yöntemine dayandığı için programlarınızı hızlı ve kolay bir şekilde oluşturabilirsiniz. Scratch'da oyun programlamak lego oynamaya benzer. Lego oyunundaki gibi blokları birleştirirsiniz. Eğer doğru blokları birleştirirseniz ortaya çok güzel bir program, animasyon veya oyun çıkar.



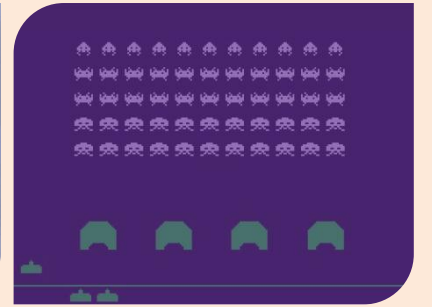
Scratch, günümüzde programlama eğitiminde en çok kullanılan programlama dillerinden biridir. Dünyanın en saygın üniversitelerinden biri olan Harvard'da bilgisayar bilimine giriş dersinde Scratch kullanılmaktadır. Scratch çok kullanılan ve sevilen bir programlama dili olduğu için Scratch kullanırken destek bulmakta zorlanmazsınız. Sürekli güncellenen ve gelişen yapısı ile de çağa ayak uyduran Scratch, programlama öğrenirken sizin en büyük yardımcınız olacaktır.



Hayal Et



Programla



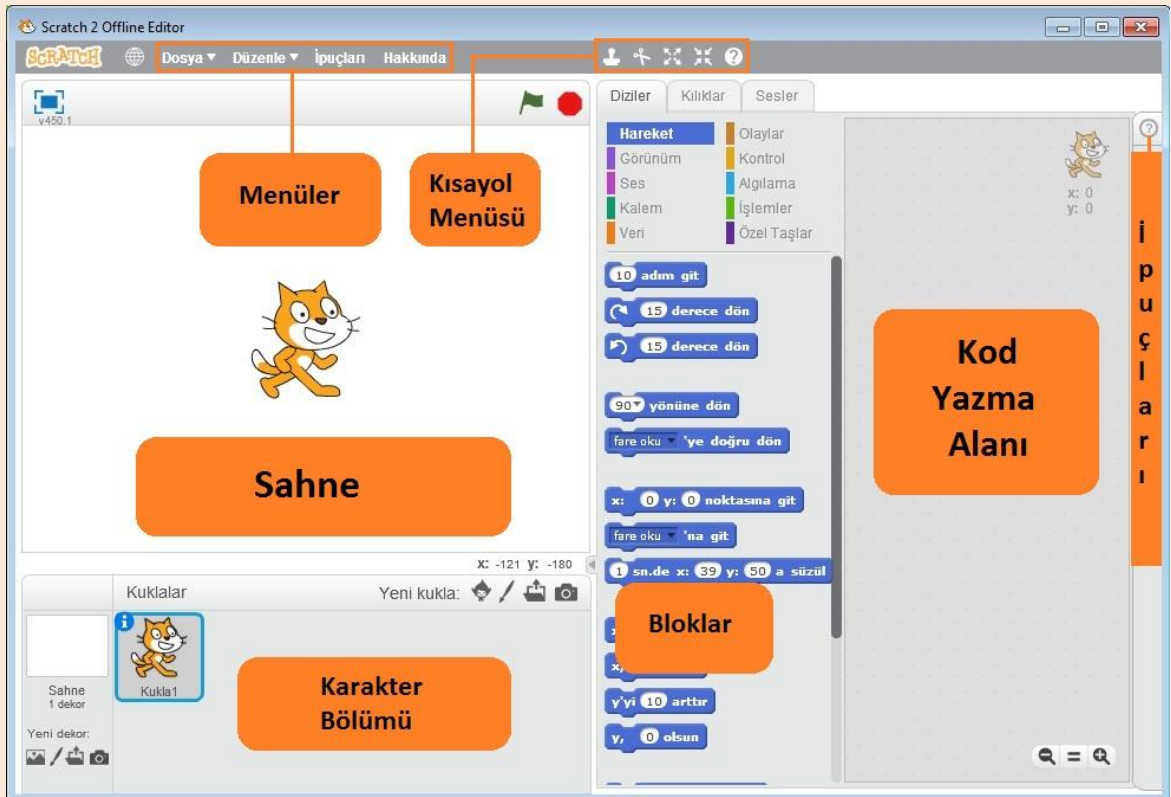
Paylaş

Scratch Programına Giriş

Scratch programını internete bağlı olmadan (offline) ya da internete bağlı olarak (online) kullanabilirsiniz. İnternet üzerinden program yazmak için <https://scratch.mit.edu> adresini ziyaret edebilirsiniz.

Scratch programını indirip bilgisayarınıza kurmak için, <https://scratch.mit.edu/scratch2download> adresini ziyaret ederek programı kurabilirsiniz. Scratch programını kurmadan önce Adobe AIR programını kurunuz.

Scratch Offline Editör



Scratch'da bloklarımızı sürükleyip bırak yöntemiyle kod yazma alanına taşıyoruz. Yazdığımız kodlar etkisini sahne kısmında gösterir.

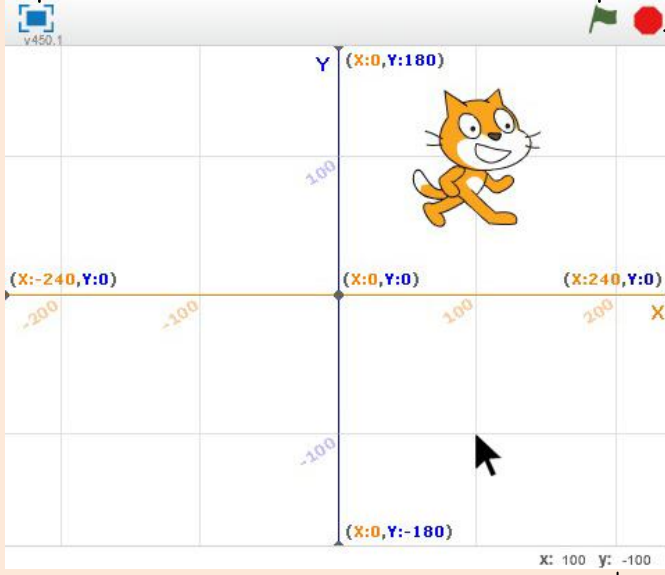


Sahne

Tam Ekran Butonu

Başlat Butonu

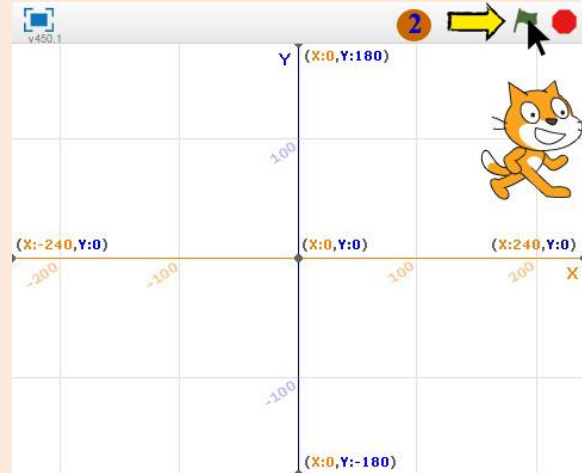
Durdur Butonu



Fare imlecinin o anki koordinatı

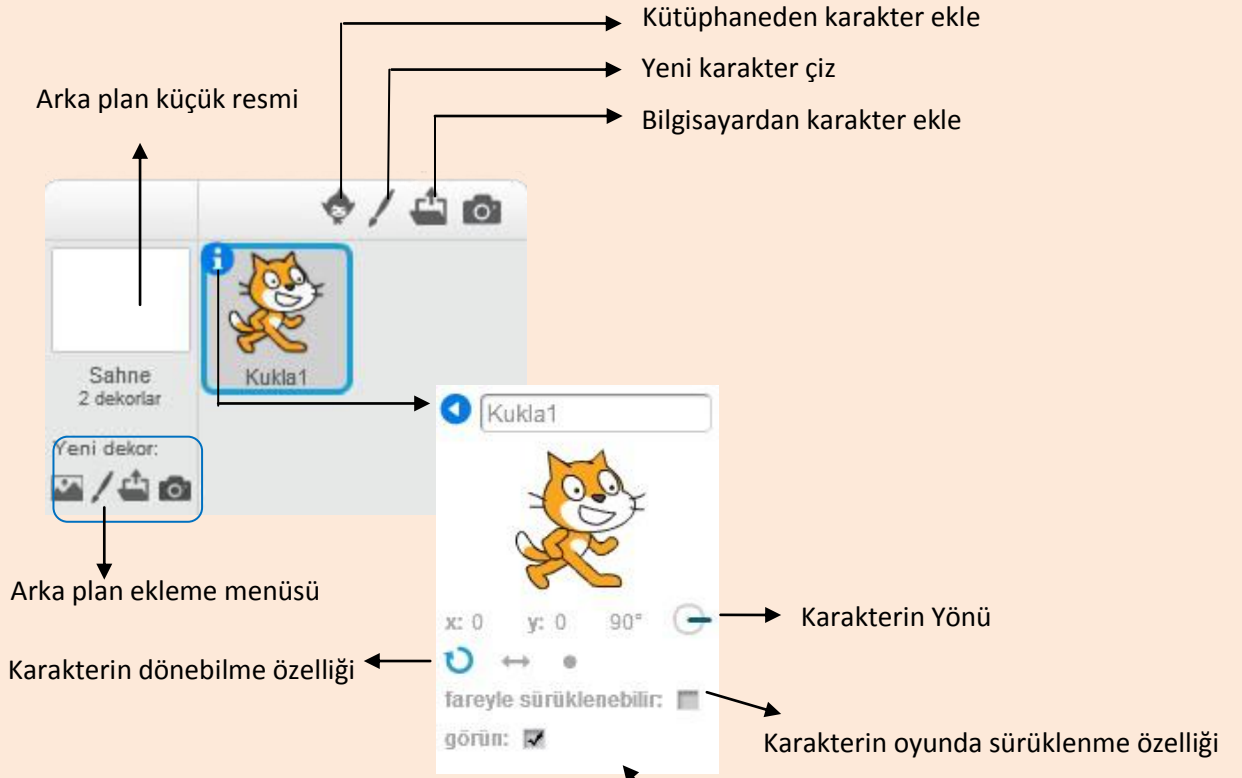
Sahne üzerindeki karakterin bulunduğu konumu kontrol etmek için bir koordinat sistemi vardır. Bu koordinat sistemini, yatay ve dikey, iki doğru olarak düşünebiliriz. Yatay doğruya X doğruyu, dikey doğruya ise Y doğruyu diyeceğiz. X doğrusunda 480 nokta, Y doğrusunda ise 360 nokta vardır.

Karakterin bulunduğu konum X ve Y doğrularının kesişim noktasıdır. Burada karakterimiz X=100, Y=100 noktasındadır.



Koordinat sistemini programlarımızda sıkça kullanacağız. Koordinat sistemi ile karakterimizin hareketlerini kontrol ederiz. Karakterin ileriye doğru hareket etmesi aslında X konumunun artması, geriye doğru hareket etmesi ise aslında X konumunun azalmasıdır. Benzer bir durum Y koordinatı içinde geçerlidir. Bir oyunda karakter zıpladığında aslında Y konumu önce artar daha sonra azalarak eski konumuna gelir.

Karakter Bölümü



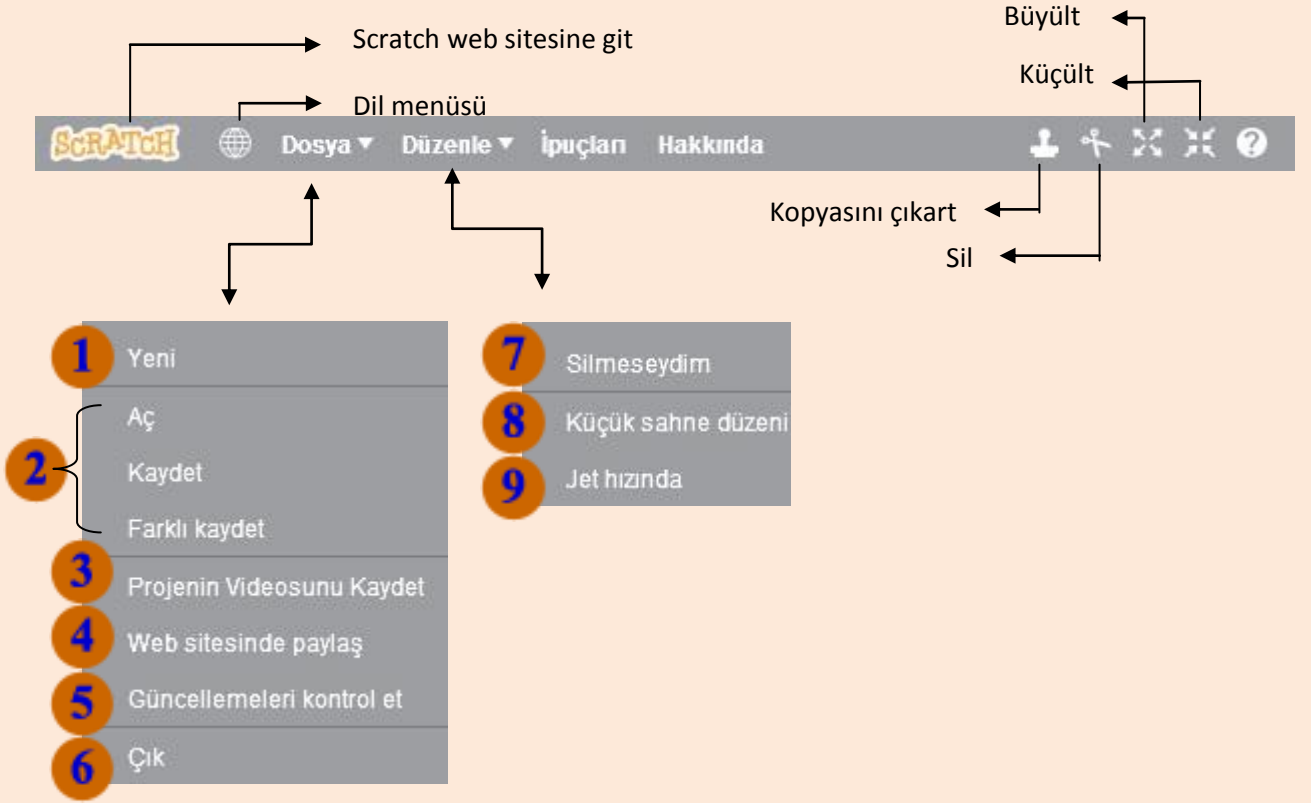
Scratch'da oyunlarımızı yaparken karakterler kullanacağız. Bu karakterleri programlayarak projelerimizi hayata geçireceğiz.

- 1 Karakterimizin içindeki tüm kodlar ve içeriklerle birlikte bir kopyasını çıkartır.
- 2 Karakterimizi siler.
- 3 Karakterimizi bilgisayara kaydeder.
- 4 Karakterimiz sahnede üzerinde görünüp gizlenebilir. Bu menü ile karakterimizi gizleyebiliriz.

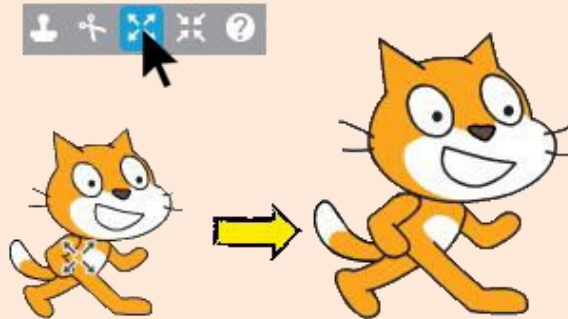


Karaktere sağ tıklandığında açılan menü

Menüler



Menüleri kullanarak 1 yeni bir çalışma sayfası açabiliriz. 2 Çalışmamızı kaydedebilir ya da mevcut olan bir çalışmayı açabiliriz. 3 Yaptığımız projenin videosunu kaydedebiliriz. 4 Web sitesinde herkese açık bir şekilde paylaşabiliriz. Bunun için Scratch web sitesine kayıt olmamız gerekmektedir. 5 Program için güncelleme olup olmadığını kontrol edebiliriz. 6 Programdan çıkış yapabiliriz. 7 Yanlışlıkla sildiğimiz blokları geri getirebiliriz. 8 Sahneyi küçültüp kod yazdığımız alanı büyütebiliriz. 9 Programımızın çalışma hızını arttırabiliriz.



Büyült butonu kullanımı

Diziler



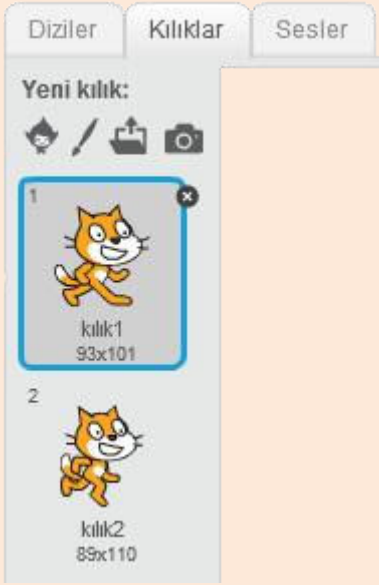
Scratch'da programlarımızı yaparken blokları kullanacağız. Bu bloklar Diziler menüsü altındadır. Bloklar kendi içlerinde gruplara ayrılmıştır. Örneğin; hareket menüsü karakterin sahnede hareketini sağlayan komutları içeren blokları barındırırken, Kontrol menüsü eğer ise gibi kontrol bloklarını bünyesinde barındırır.

Scratch'da kod blokları en üstten çalışmaya başlar. En üstte **Olaylar** menüsü blokları bulunur. Daha sonra sırasıyla alt kısımdaki kodlar çalışmaya başlar. Kod bloğu çalıştırdıktan sonra sıra bir alttakine geçer ve son bloğa kadar bu şekilde çalıştırılır.



Kodlar adım adım çalıştırılır. ❶ Programlar genelde yeşil bayrağa basılınca başlar. Sırasıyla diğer adımlara geçilir. Yeşil bayrağın altındaki ilk kod bloğu **10 adım git** olduğu için karakter ❷ 10 adım ileri gider. Daha sonra bir alttaki kod bloğu çalıştırılır. ❸ Karakter ekrana "Merhaba" yazar.

Kılıklar



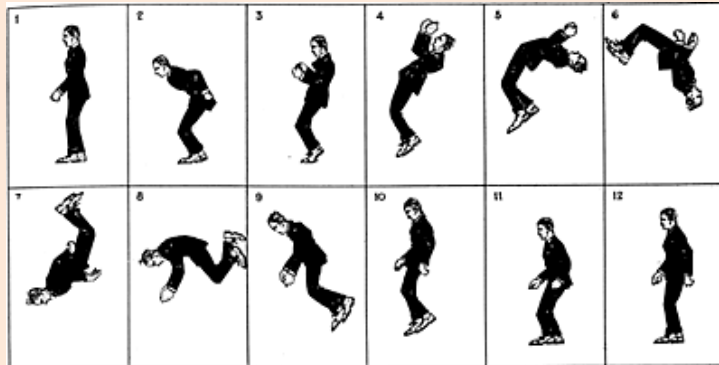
Karakterimize efekt vermek için kılıkları kullanırız. Karakterimize uçma ya da yürüme efekti verebiliriz. Kedi karakterimizin standart olarak iki kılığı vardır. Kılık1 ve kılık2 arasında geçiş yaparak yürüme efekti veririz. Karakterimiz önce kılık1 sonra kılık2'de gözüktüğünde, gözümüz onu yürüyor olarak görür.

Çizgi Filmler Nasıl Yapılır?

Kağıtlarla çizgi filmler oluşturulurken bir çok resim çizilir. Her resimde karakterimiz bir hareketin devamını yapar. Bu resimler arka arkaya eklenir ve daha sonra hızlı bir şekilde aralarında geçiş yapılır. Gözümüz karakteri gerçekten hareket ediyormuş gibi görür.



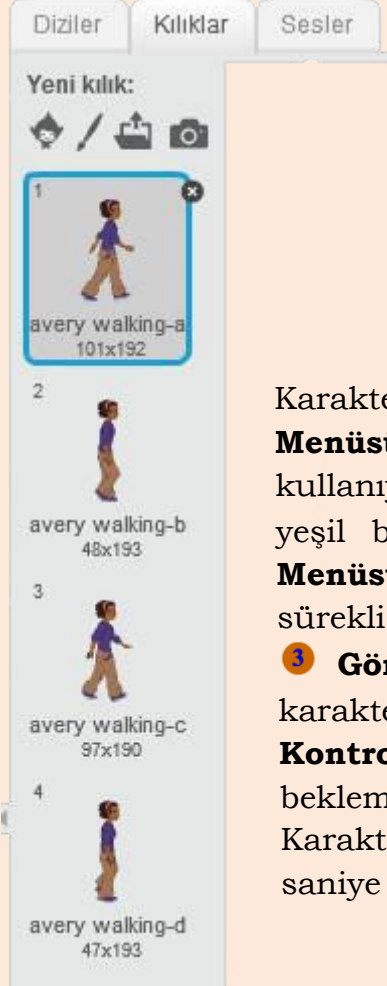
Scratch'da buna benzer bir şekilde kılıklar arasında geçiş yaparak karakterimize yürüme, zıplama ya da uçma efekti verebiliriz.



Ters takla atan bir karakterin kılıkları

Kütüphaneden karakter ekle butonu ile **Avery Walking** karakterini ekleyelim.

Karakterimiz, yürüme efekti verebilmek için dört adet kılığa sahip.



Karakterimize yürüme efekti vermek için **1 Olaylar Menüsü** altındaki “Yeşil bayrak tıklanınca” komutunu kullanıyoruz. Bu programımızın başlangıcı. Program biz yeşil bayrağa tıkladığımızda başlayacaktır. **2 Kontrol Menüsü** altındaki **sürekli tekrarla** bloğu ile efektimizin sürekli devam etmesini sağlayan bir döngü oluştururuz. **3 Görünüm Menüsü** altındaki **sonraki kılık** bloğu karakterimizin bir sonraki kılığa geçmesini sağlar. **4 Kontrol Menüsü** altındaki **1 saniye bekle** bloğu ile araya bekleme koyarak efektin çok hızlı gerçekleşmesini önleriz. Karakter bir sonraki kılığa geçerken her seferinde 0.2 saniye bekler.

Bu kısımdaki yazıları ya da rakamları değiştirebilirsiniz!

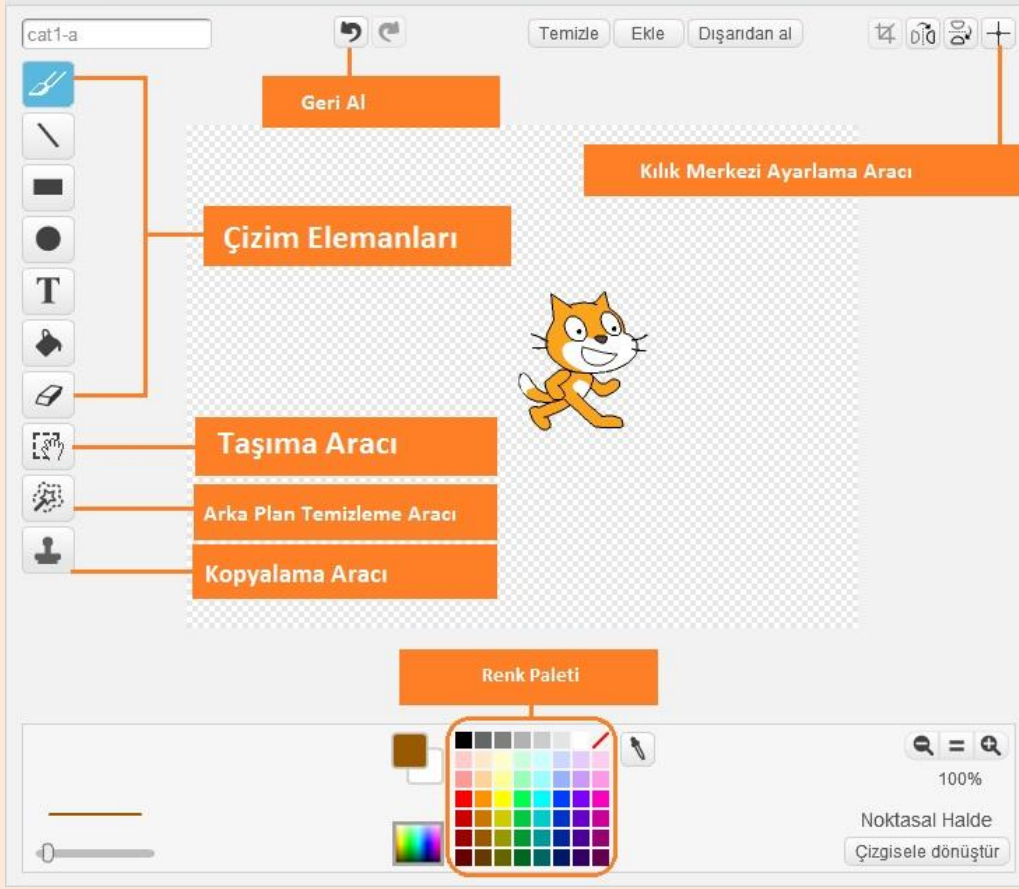


Olaylar

Kontrol

Görünüm

Resim Düzenleme Aracı



Scratch karakterlerinin kılıklarını düzenlemek için Windows işletim sistemindeki Paint programına benzer bir Resim Düzenleme Aracı mevcuttur. Bu aracı kullanarak kılıklarımız rengini ve şeklini değiştirebilir ya da yazı ekleyebiliriz.

Resim düzenleme aracının noktasal ve çizgisel olarak iki türü mevcut. İsteğimiz dahilinde birini kullanabiliriz.



Karakterin rengini değiştirme

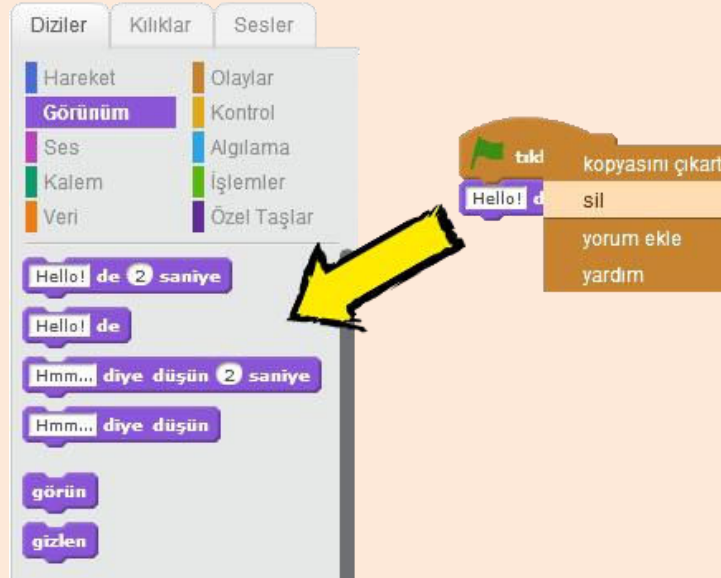
Sesler



Programlarımıza 1 kütüphanedeki hazır sesleri ekleyebildiğimiz gibi 2 kendi kaydettiğimiz sesleri ya da 3 bilgisayarda kayıtlı bulunan ses dosyalarını ekleyebiliriz.



Programımıza bir sesi eklemek için öncelikle o sesi **Sesler** menüsünden eklemelisiniz. Ekledikten sonra **sesini çal** bloğu ile o sesi kullanabiliriz.



Kullanmadığımız kod bloklarını silmek için iki yöntemimiz mevcut. Birinci yöntem, kod bloğuna sağ tıklayıp silmek. İkincisi ise kod bloğunu sürükleyip diziler menüsündeki kod bloklarının olduğu yere bırakmak.

HAREKET ve GÖRÜNÜM

Hareket Blokları

Scratch'da bir sahne düzeni mevcut. Biz yazdığımız programlarla sahne üzerindeki karakteri kod blokları ile kontrol ederiz. Karakteri kontrol etmek için kullanılan kod blokları ise kendi içerisinde gruplara ayrılmıştır. Bu gruplar içerisinde en üstte bulunan ve mavi renge sahip olan bloklar hareket bloklarıdır. Hareket bloklarını, karakteri sahne üzerinde hareket ettirmek için kullanırız.



İlk olarak karakterimizi hareket ettirelim. Hareket bloklarından olan **10 adım git** git bloğu ile karakterimizin ileriye doğru hareket ettiririz. Burada yeşil bayrak tıkladığında karakter 10 adım ilerleyecektir.



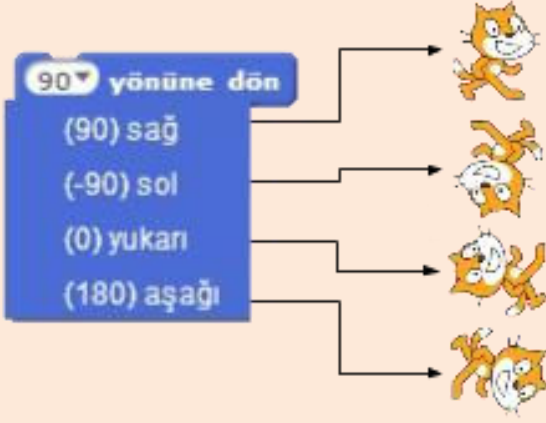
Eğer karakterimizin sahne üzerinde sürekli olarak hareket etmesini istiyorsak. Bir tekrarlama bloğu kullanmamız gerekmektedir. Kontrol menüsü altındaki **sürekli tekrarla** bloğu ile karakterimizin sahne üzerinde sürekli hareket etmesini sağlayabiliriz.



Karakterimiz sahne üzerinde hareket ederken sahne sonuna geldiğinde ilerleyemeyecektir. Bu sorunu çözmek için bir **kenara geldiysen sek** bloğu ile sahne sonuna geldiğinde geri dönmesini sağlarız.



Karakterimizin sahne sonuna gelip geri döndüğünde yönü değişeceği için ters dönecektir. Ters dönmesini engellemek içinse **kuklanın şekli** kod bloğunu kullanacağız.

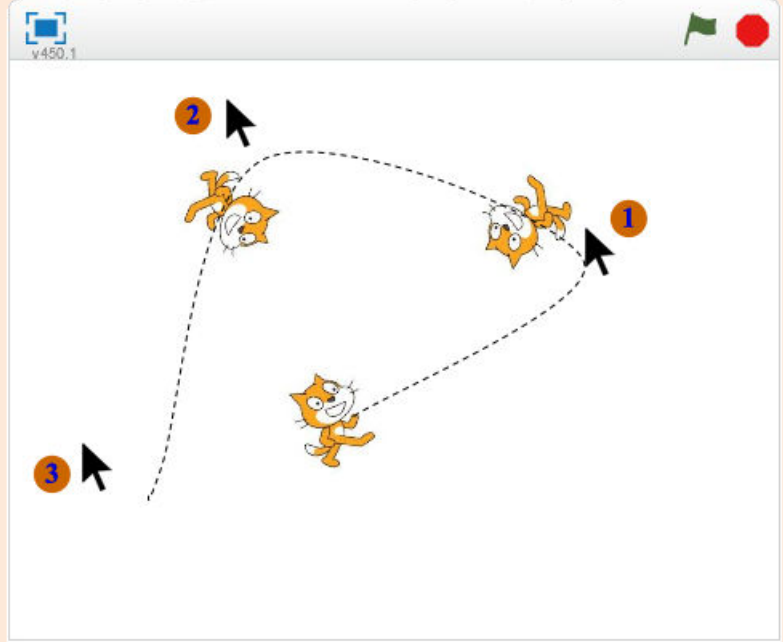


Karakterimizin sahne üzerinde yönleri vardır. Bu yönleri ayarlayarak karakterimizin hareketini kontrol ederiz. Her yönün bir açı karşılığı vardır. Karakterimize **90 yönüne dön** diyebildiğimiz gibi ara değerlerde verebiliriz. Örneğin; **45 yönüne dön** dediğimizde 0 ile 90 arasında yani yukarı ve sağ arasında bir yöne döner.

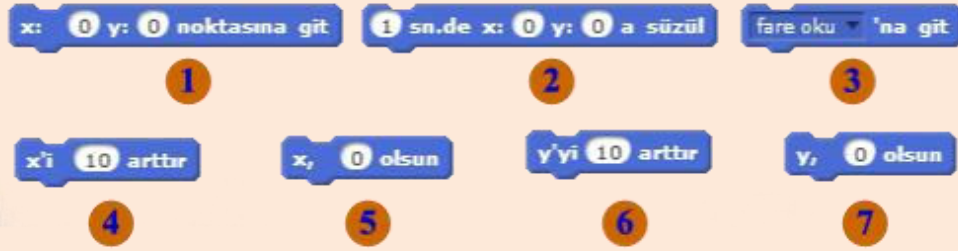


Karakterimizin küçük resmindeki info(bilgi) işaretine tıklarsak, karakter ile ilgili bilgileri görebiliriz.

- 1 Karakterimizin o anki X ve Y koordinatı
- 2 Karakterimizin o anki yönü
- 3 Her yöne dönebilsin
- 4 Sadece sağa sola dönebilsin
- 5 Hiçbir yöne dönmesin



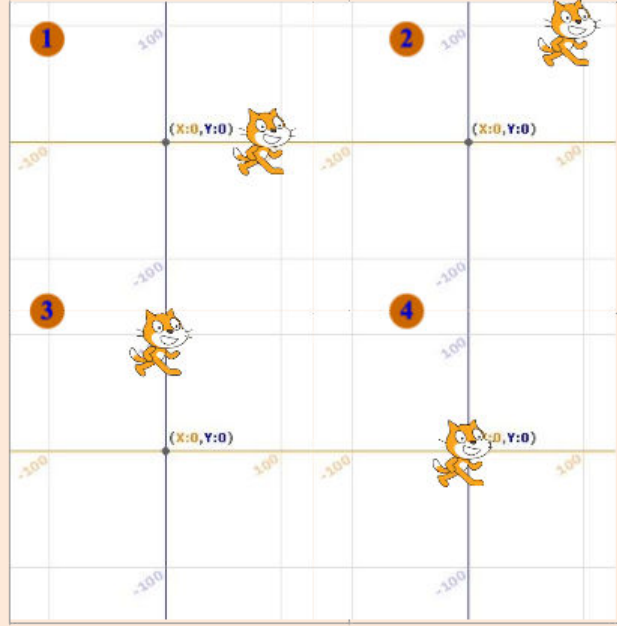
İstedığımız takdirde karakterimizi belli bir yöne değil de fare okuna doğru döndürerek karakterimizi hareket ettirebiliriz.



Sahne üzerinde bulunan her noktanın bir koordinatı vardır. Bu koordinat noktalarını kullanarak karakterimizi sahne üzerinde istediğimiz herhangi bir noktaya taşıyabiliriz.

1 Karakterimizi sahne üzerinde istediğimiz X ve Y konumuna taşıyabiliriz. 2 Bir animasyon ya da çizgi film tasarlıyorsak karakterimizin belirttiğimiz konuma yavaşça gitmesi gerekebilir. Bu durumda karakterimizin belirlediğimiz konuma belli bir saniyede gitmesini sağlayabiliriz. 3 Karakterimizin belli bir konuma değil de fare okunun bulunduğu konuma gitmesini sağlayabiliriz. 4 Koordinat düzleminde X konumunun artması karakteri ileriye götürür. Kod bloğundaki "10" rakamını negatif yaparsak "-10" bu sefer X konumu azalacağı için karakter geriye doğru gidecektir. 5 Karakterimizin belirlediğimiz X konumuna gitmesini sağlayabiliriz. 6 X konumu için yaptığımız işlemler Y konumu içinde geçerlidir. Y konumu artan karakter yukarıya doğru hareket edecektir. Kod bloğundaki "10" rakamını negatif yaparsak "-10" bu sefer Y konumu azalacağı için karakter aşağıya doğru hareket eder. 7 Karakterimizin Y konumunu istediğimiz konum olarak ayarlayabiliriz.

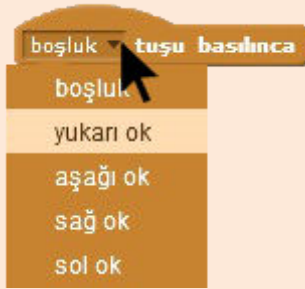
```
bkıldanmca
x: 100 y: 0 noktasına git
1 saniye bekle
x: 100 y: 100 noktasına git
1 saniye bekle
x: 0 y: 100 noktasına git
1 saniye bekle
x: 0 y: 0 noktasına git
```



Yön Tuşlarıyla Karakter Kontrolü

Programımız genel olarak yeşil bayrağa tıklayınca başlar; ancak bazı kod bloklarının başlaması için başka olayların gerçekleşmesi gerekebilir. Örneğin; fareye tıklamak, boşluk tuşuna basmak gibi. Bu örneğimizde kod bloklarımız yön tuşlarına basıldığında çalışacak.

Önce **Olaylar** menüsünden **boşluk tuşuna basılınca** kod bloğunu alalım.



Boşluk tuşunu başka tuşlarla değiştirebiliriz. Seçtiğimiz tuş basılınca kod bloğunun altına yerleştirdiğimiz kod blokları çalışacaktır.

```
yukarı ok tuşu basılınca
y'yi 10 arttır
```

```
aşağı ok tuşu basılınca
y'yi -10 arttır
```

```
sağ ok tuşu basılınca
x'i 10 arttır
```

```
sol ok tuşu basılınca
x'i -10 arttır
```

Yukarıdaki kodları, kod yazma alanına yazalım. Bu kodlar karakterimizin sahne üzerindeki hareketini kontrol edecektir. Yukarı ok tuşu basıldığında karakterin Y konumu 10 artacak, karakter yukarıya doğru hareket edecektir. Aşağı ok tuşu basıldığında Y konumu -10 artacak yani 10 azalacaktır. Y konumuna benzer şekilde sağ ok tuşu karakterin X konumunu arttırırken, sol ok tuşu karakterin X konumunu azaltacaktır. Bu şekilde karakterimiz sahne üzerinde ileri, geri,yukarı,aşağı yönlerinde hareket edecektir.

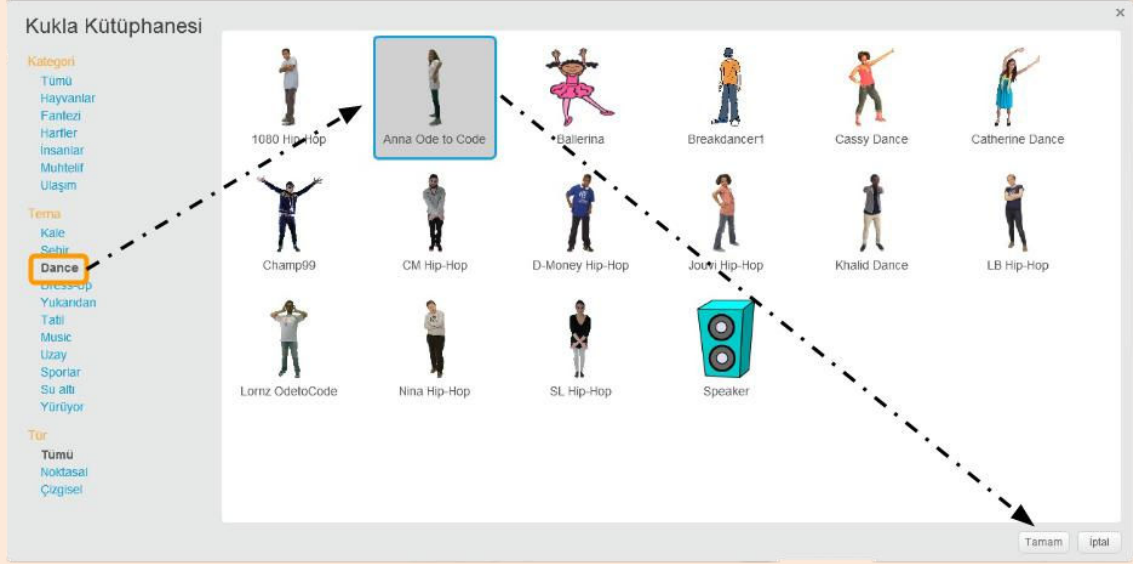
Görünüm Blokları

Karakterimizin sahne üzerinde nasıl görüneceğini, görünüm blokları ile ayarlarız. Görünüm blokları ile karakterimizi konuşturabilir, rengini ya da kılığını değiştirebilir, ekranda görünmesini veya gizlenmesini sağlayabiliriz.



Dans Partisi

Programımız için **kuklayı kütüphaneden seç** butonuna tıklayalım. Dance bölümünden bir karakter ekleyelim.

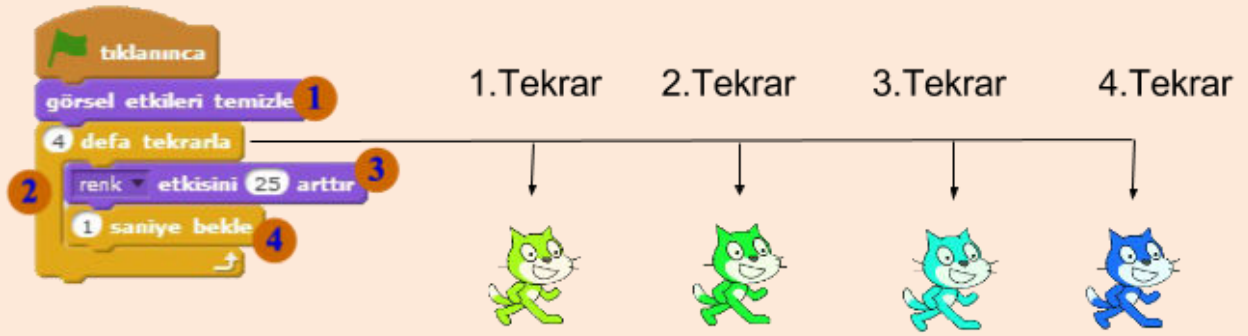


boşluk tuşu basılınca
sonraki kılık

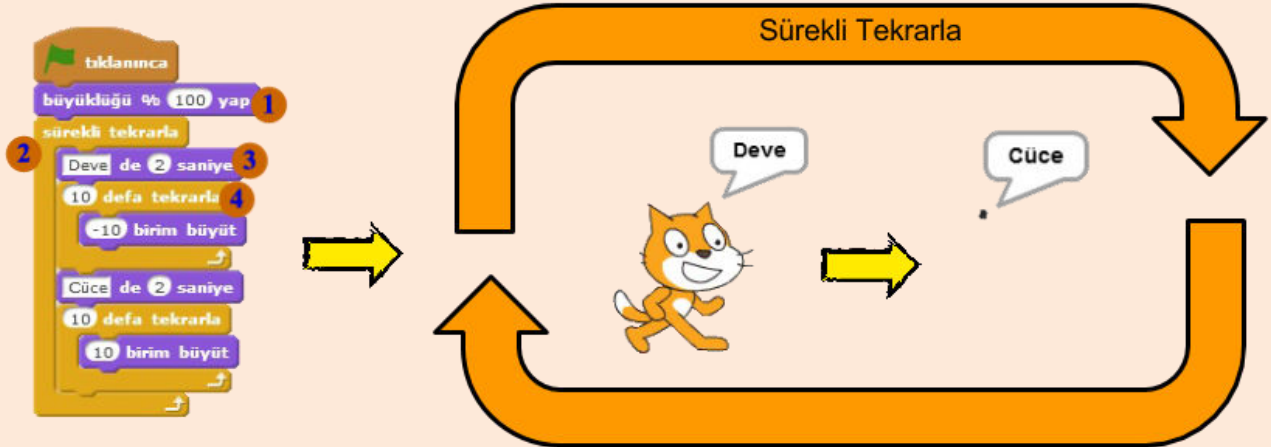
Yan taraftaki kod bloğu, karakteri ekleyince otomatik olarak eklenir. Karakterimizin 12 adet kılığı mevcut. Eklediğimiz **sonraki kılık** kod bloğu ile kılıklar arasında geçiş yaparak dans etme animasyonu yaparız. Boşluk tuşuna her bastığımızda karakterimizin kılığı değişecektir.

renk etkisini 25 artır
renk
balık gözü
fırında
benekleştire
mozaik
parlaklık
hayalet

Görünüm bloklarını kullanarak karakterimizin rengini, parlaklığını ya da ekranda görünme etkisini(hayalet) ayarlayabiliriz.



Programımızda ① **görsel etkileri temizle** bloğu ile daha önce değiştirilmiş görsel etkileri sıfırlarız. Uyguladığımız bütün görsel etkiler silinir. ② **Kontrol menüsü** altında bulunan **10 defa tekrarla** kod bloğu ile istediğimiz sayıda döngü oluşturabiliriz. Biz programımızda 4 defa tekrarla diyerek dört adet döngü oluşturduk. Döngü içerisinde her adımda ③ **renk etkisini 25 arttır** kod bloğu çalışır. Her döngüde karakterimizin rengi değişir. ④ Kontrol menüsü altında bulunan **1 saniye bekle** kod bloğunu programımızın çok hızlı çalışmaması için kullanırız. Bilgisayarlar çok hızlı çalışan elektronik aletlerdir ve bizim gözümüz çok hızlı olan değişimleri seçemez. Bu yüzden araya bir saniye bekleme koyarak programı yavaşlatırız.

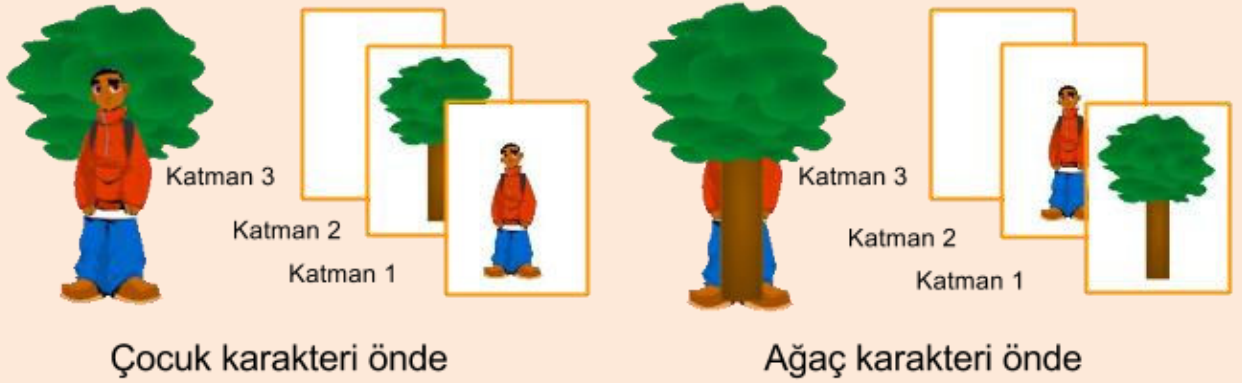


Karakterimizin büyüklüğünü ayarlamak istediğimizde **büyüklüğü %... yap** ve **10 birim büyüt** kod bloklarını kullanırız.

Programımız yeşil bayrağa basılınca başlar. ① İlk adımda **büyüklüğü %100 yap** kod bloğuyla büyüklüğü orijinal büyüklüğe getiririz. ② İkinci adımda sürekli tekrarla döngüsü çalışır. Sürekli tekrarla döngüsü içerisinde bulunan kod blokları program çalıştığı sürece sürekli tekrarlanır. ③ Sürekli tekrarla bloğu çalışmaya başladıktan sonra ilk olarak en üstte bulunan **Deve de 2 saniye** kod bloğu çalışır ve

karakterimiz ekrana Deve yazısını yazar. 4 Daha sonra bir alttaki kod bloğuna geçer ve **10 defa tekrar** kod bloğu çalışmaya başlar. On defa - **10 birim büyüt** kod bloğu çalışır. Bu kod bloğunu karakterimizin büyüklüğünü arttırmak ya da azaltmak için kullanırız. Burada -10 birim büyüt dediğimiz için her adımda karakterimizin büyüklüğü 10 birim azalacaktır ve karakterimiz küçülecektir. Daha sonra bir alt satıra geçilerek **Cüce de 2 saniye** kod bloğu ile ekrana Cüce yazdırılacaktır. Sürekli tekrarlar bloğumuzun içindeki son kod bloğu ile on defa, on birim büyüt kod bloğu çalışacak ve karakterimiz büyüyerek eski haline dönecektir. Bu durum program çalıştığı sürece sürekli olarak tekrarlanacaktır.

Katmanlar



Sahne üzerinde bir katman yapısı vardır. En üst katmanda bulunan karakter en önde gözükürken onun altındaki katmanda bulunan karakter ikinci sırada gözükür.



Üste çık komutu ile karakteri en üst katmana çıkarırken **1 katman alta in** komutu ile bir alt katmana indiririz.

Uçan Yarasa



Bat2-a Kılığı

Bat2-b Kılığı

Şimdi **kuklayı kütüphaneden seç** butonu ile yeni bir karakter ekleyelim. Hayvanlar menüsü altında bulunan Bat2 karakterini ekleyelim.



Bat2 karakteri için yazılacak kodlar

Paralel Programlama

Paralel programlama, iki kod bloğunun aynı anda çalıştırılmasıdır. Scratch'da aynı anda çalışmasını istediğimiz kod bloklarını ayrı yeşil bayrak bloklarının altına yerleştiririz. Bu bloklar yeşil bayrağa tıklandığında aynı anda çalışmaya başlar.

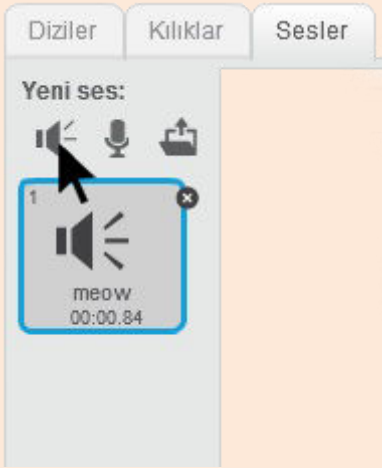
Yeşil bayrağa tıklandığında **kuklanın şekli sağa-sola dönebilsin** ve **büyüklüğü %70 yap** kod blokları aynı anda çalışmaya başlar. Daha sonra iki sürekli tekrarla bloğu da birlikte çalışmaya başlar. Karakterimiz aynı anda 10 adım giderken sonraki kılığa geçer ve uçma etkisi verilir. Bu işlemler paralel olarak yürütülür.

SES ve KALEM

Ses Blokları

Oyunlarda karakter zıpladığında, bir yere çarptığında ya da puan kazanıldığında sesle bildirim verilir. Scratch'da da sesli bildirimleri vermek için kod blokları bulunur. Scratch kütüphanesinde bulunan hazır sesleri kullanabileceğimiz gibi kendi bilgisayarımızdan da ses ekleyebiliriz. Bilgisayarda kayıtlı bir çok ses dosya uzantısı mevcuttur; ancak Scratch sadece .wav ve .mp3 uzantılı ses dosyalarını kabul eder. Bir sesi projemizde kullanabilmemiz için önce sesler menüsünden o sesi eklememiz gerekmektedir.

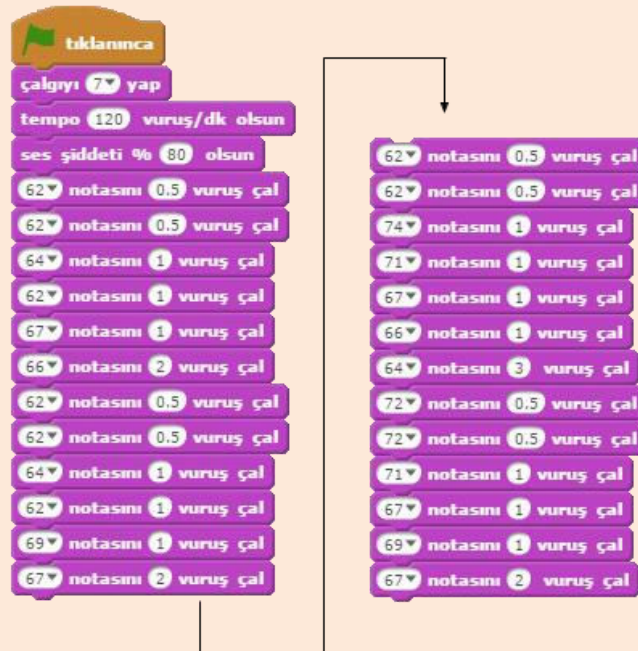
Bir arka plan müziği oluşturalım.



Sesler menüsünden sesi **kütüphaneden seç** butonuna tıklayalım. Müzik döngüleri kısmından **xylo1** sesini seçelim. Sürekli tekrarlar diyerek müziğin bittiğinde tekrar çalmasını sağlayalım.



İyi ki Doğdu Şarkısı Notaları



Scratch'da her notayı çalabilmemiz için kod blokları mevcut. İstersek çalgı türünü değiştirebilmemiz, tempoyu ayarlayabilmemiz ya da ses şiddetini değiştirebilmemiz için de kod blokları mevcuttur.

Kalem Blokları

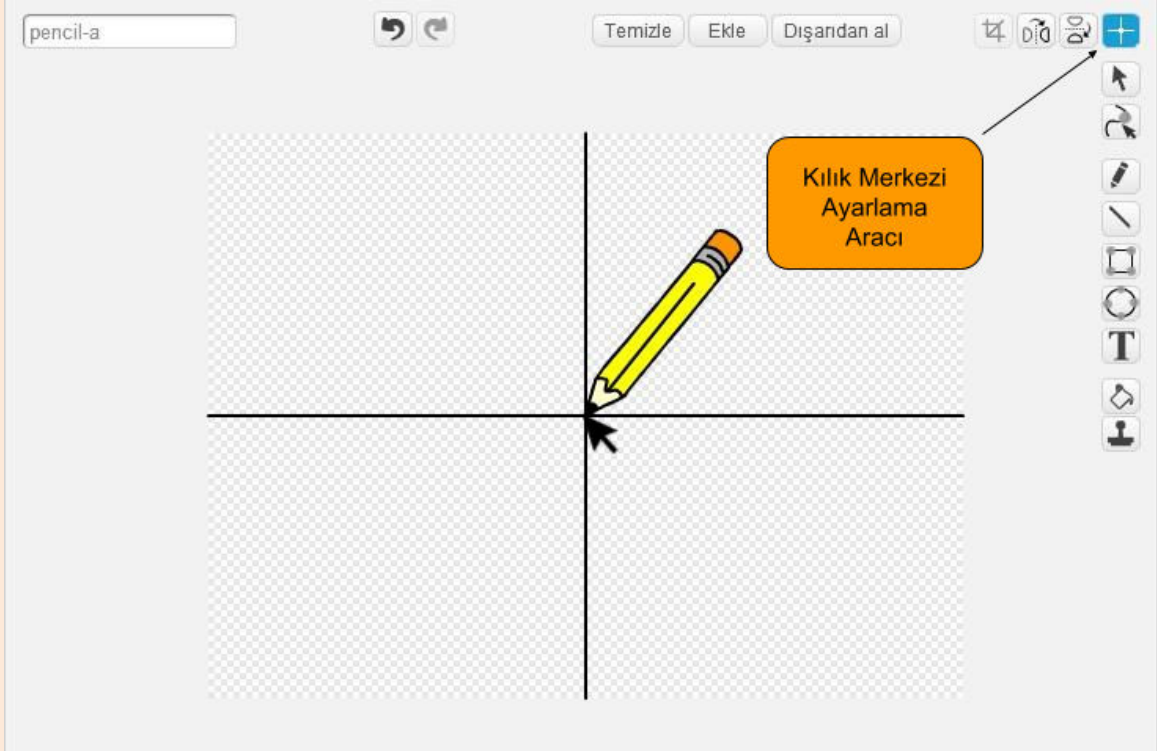
Scratch'da çizim yapabilmemiz için kalem blokları mevcut. Şimdi hep beraber bu blokları kullanarak bazı geometrik şekilleri çizelim.

Kuklayı kütüphaneden seç butonunu tıklayalım ve **muhtelif** menüsü altındaki **pencil** karakterini ekleyelim.

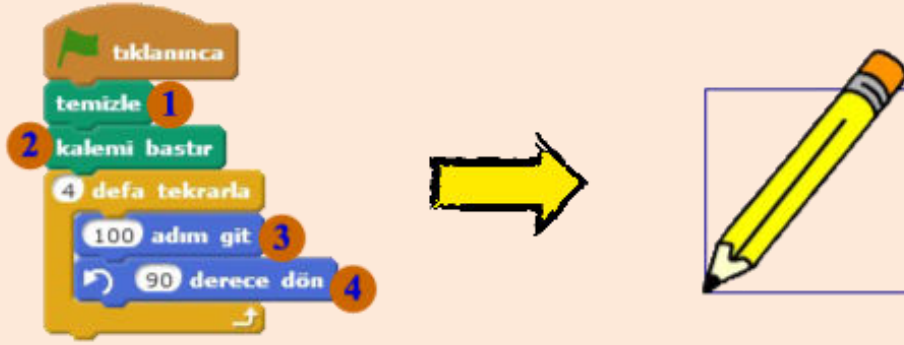


Pencil karakterinin kılık merkezi tam orta noktası olduğu için çizimlerimizi yaparken kalemin ucundan değil de ortasından çiziyormuş gibi görünecektir. Bu yüzden **resim düzenleme aracını** kullanarak kılık merkezini ayarlayalım.

Pencil Karakteri



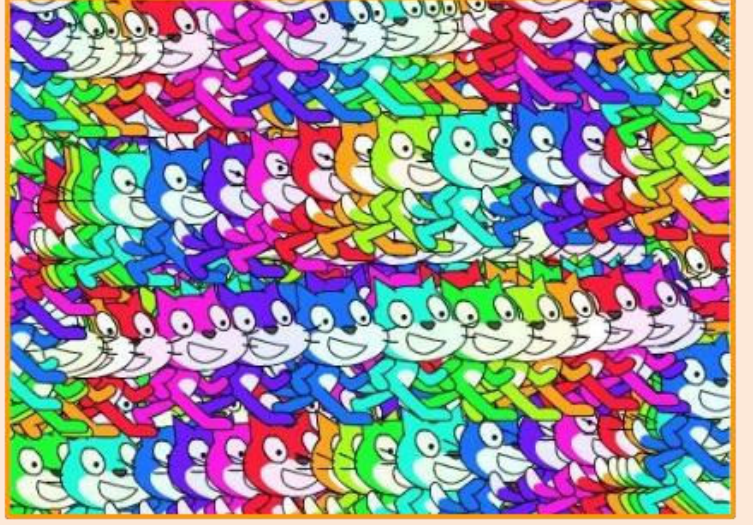
Karakterimizin kılık merkezini tam ucu yaptığımız için karakter çizimleri kalemin ucu ile yapacaktır.



1 **Temizle** bloğuyla daha önce kalemle çizilmiş şekilleri siliyoruz. 2 Kalemle çizim yapabilmemiz için kalemin basılı olması gerekmektedir. Kalem basılıyken karenin dört kenarı olduğu için dört kez 3 **100 adım git** 4 **90 derece dön** komutlarını uygulatıyoruz. Burada 100 adım karenin uzunluğunu belirler. 90 derece dönmemizin nedeni ise karenin ilk kenarını çizdikten sonra diğer kenarını çizebilmemiz için karenin dış açısı kadar dönmemiz gerektiği içindir. Karenin bir iç açısı 90 olduğu için dış açısı da 90 olur.

| | | |
|--|--|--|
| | | |
| | | |

Renkli Kediler

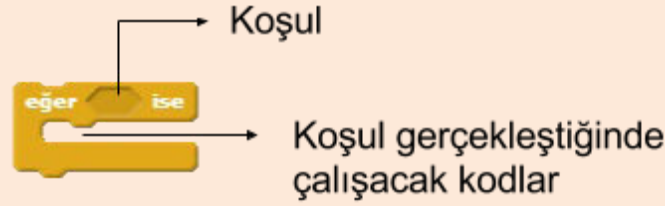


❶ **Fare okuna git** kod bloğu çalıştığında karakter fare okunun olduğu koordinata gider. Kod bloğumuz sürekli bloğunun içinde olduğu için karakter sürekli fare okunun olduğu konuma gider. ❷ **İz bırak** kod bloğuyla da karakterin sahneye bir kopyasını bırakırız. ❸ **Renk etkisini 25 arttır** kod bloğu ile de karakterimizin rengini değiştiririz ve sahneye karakterimizin izini bırakırken rengarenk bir görüntü oluştururuz.

KONTROL ve ALGILAMA

Eğer ise Kontrol Bloğu

Bizim günlük hayatta bazen karar almamız gerektiği gibi bilgisayarlarında bazen karar almaları gerekebilir. Bilgisayara karar aldırırken **eğer ise** kontrol bloğunu kullanırız. Karar alırken bir koşulumuz vardır bir de koşul gerçekleştiğinde yapılacak işler. Örneğin; boşluk tuşuna basıldığında zıpla, engel geçildiğinde puan kazan gibi.



Eğer ise Kod Bloğu ile Kullanılabilecek Bazı Algılama Blokları



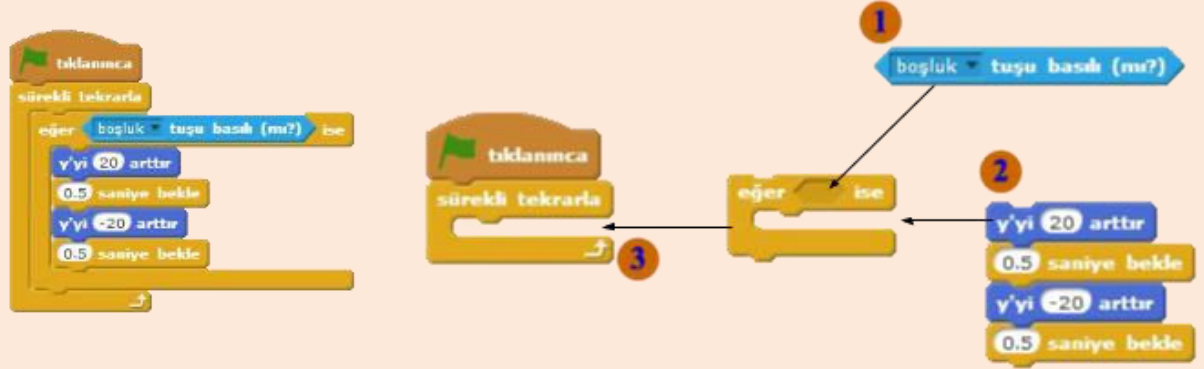
Algılama blokları karakterin sahne üzerindeki durumunu, konumunu ya da farenin koordinatları gibi durumları programlarımızda kullanmamıza yarayan bloklardır. Kontrol bloklarını genellikle algılama blokları ile beraber kullanacağız.

- 1 Fare okunun karaktere değip değmediğini kontrol eder. Eğer fare oku karaktere değiyorsa koşulumuz gerçekleşmiş olur ve **eğer ise** bloğunun içindeki kodlar çalışmaya başlar. Fare oku yazan yeri değiştirerek diğer karakterlere değip değmediğini de kontrol edebiliriz.
- 2 Karakterin belirtilen renge değip değmediğini kontrol eder. Bloğun üzerindeki rengin üzerine tıkladıktan sonra seçmek istediğimiz renge tıklarsak renk değişir.
- 3 Karakterin üzerindeki bir rengin başka bir renge değip değmediğini

kontrol eder. ⁴ Klavyede boşluk tuşuna basılıp basılmadığını kontrol eder. Boşluk tuşunu yazan yeri değiştirebilirsiniz. ⁵ Farenin sol tuşuna basılıp basılmadığını kontrol eder.

Boşluk Tuşuna Basıldığında Zıplayan Kedi

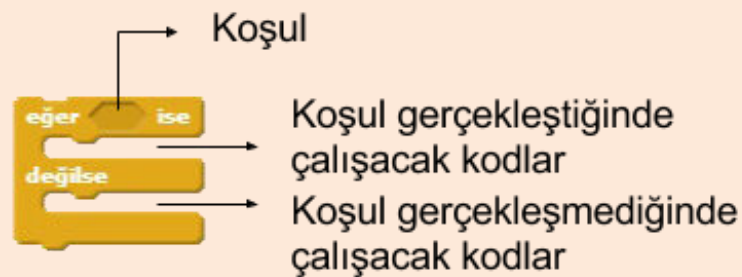
Kedi karakterimiz için aşağıdaki kodları yazalım.



¹ Burada kodlarımızın çalışması için belirlediğimiz koşul **boşluk tuşuna basılı (mı?)**. Kodlarımız ne zaman boşluk tuşuna basılırsa o zaman çalışacaktır. ² Koşulumuz gerçekleştiğinde ise karakterimizin Y konumunu arttırıyoruz 0.5 saniye bekleyip tekrar azaltıyoruz ve basit bir zıplama efekti yapıyoruz. ³ Programımızın bir kez çalışıp durmaması için sürekli bloğunun içine koyuyoruz ki sürekli olarak boşluk tuşuna basılıp basılmadığını kontrol etsin. Eğer kontrol bloğumuzu sürekli bloğunun içine koymazsak, program sadece bir kez çalışıp duracaktır.

Eğer ise Değilse Kontrol Bloğu

Eğer ise kontrol bloğunda koşulumuz gerçekleştiğinde yapılacak işler vardı. Koşulumuz gerçekleşmediğinde yapılacak herhangi bir görev yoktu. **Eğer ise değilse** kontrol bloğunda ise hem koşul gerçekleştiğinde yapılacak işler hem de koşul gerçekleşmediğinde yapılacak işler vardır.



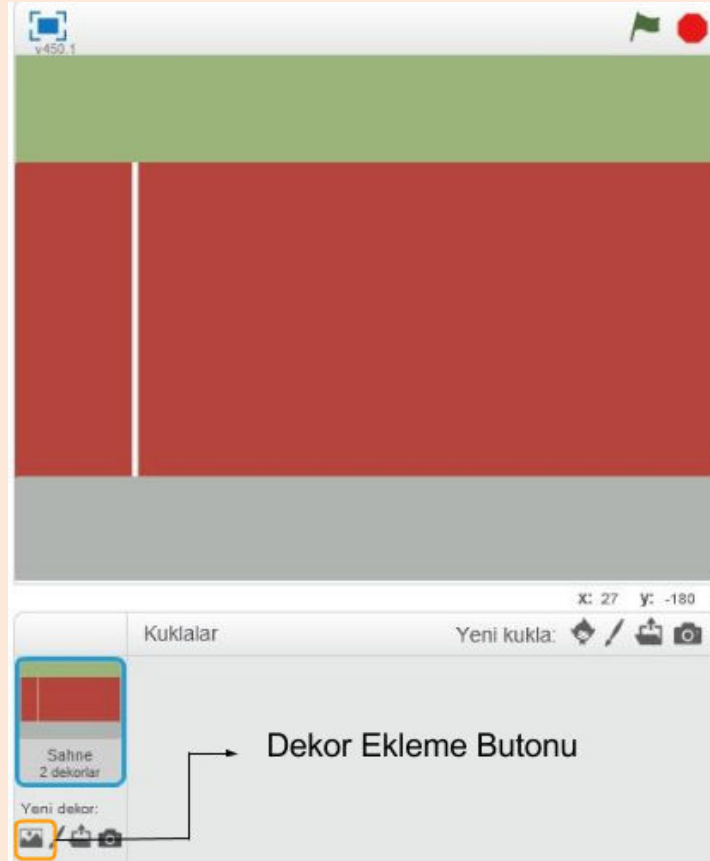
Fare Basıldığında Gizlenen Basılmadığında Görünen Kedi



Kedi karakterimiz için yan tarafta bulunan kodları yazalım. Program yeşil bayrağa tıklanınca başlar. Burada koşulumuz fareye basılı olup olmaması. Eğer fareye basılı ise karakterimiz sahnede gizlenirken, eğer fare basılı değilse karakterimiz sahnede gözükür.

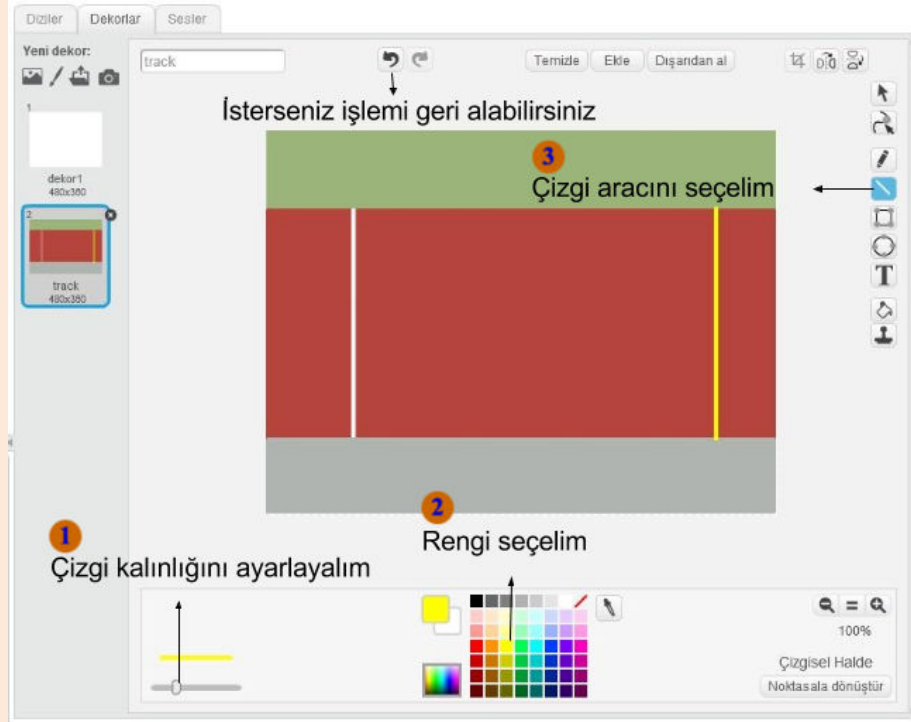
Koşu Yarışması

Şimdi mini bir koşu yarışması yapalım. Bunun için öncelikli olarak bir arka plan seçelim.



Dekor kütüphanesinden sporlar teması altında bulunan track temasını ekleyelim.

Oyunumuzda bir bitiş çizgisine ihtiyacımız var. Bunun için dekor seçiliyken dekorlar sekmesine tıklayarak resim düzenleme aracını açalım.

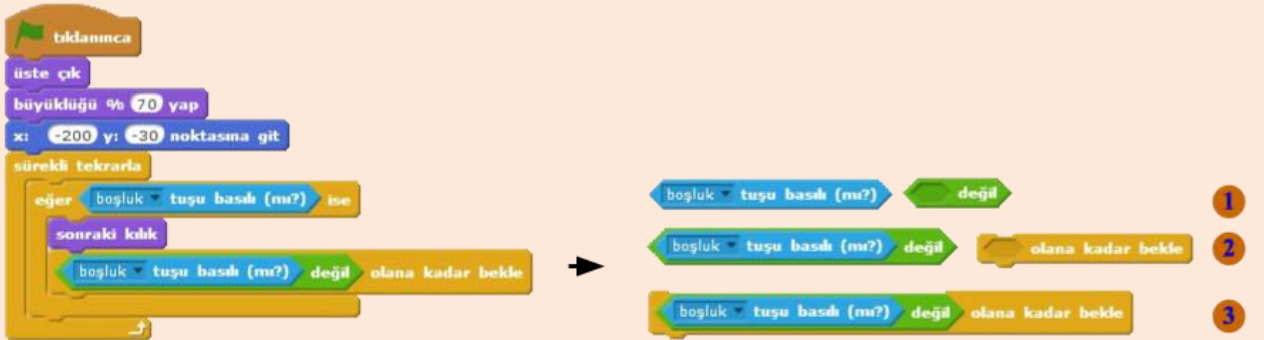


Daha sonra **kuklayı kütüphaneden seç** butonuna tıklayalım. Yürüyor teması altında bulunan Jamie Walking ve Dog2 karakterlerini ekleyelim.



Jamie Walking Karakteri için Yazılacak Kodlar

Karakterimiz için kod yazabilmek için kuklalar kısmından Jamie karakterine tıklayınız. Daha sonra diziler sekmesi açıkken kodlarımızı yazalım.



1 İşlemler menüsü altında bulunan değil bloğunu alalım.

2 Algılama menüsü altında bulunan boşluk tuşuna basılı mı bloğunu değil bloğunun boş kısmına yerleştirelim.

3 Birleştirdiğimiz blokları olana kadar tekrarla bloğunun boş kısmına yerleştirelim.

Jamie karakteri için yazacağımız ilk kod bloğunda, katman olarak üste çıkmasını sağladık, X,Y konumunu ve büyüklüğünü ayarladık. Daha sonra bir eğer ise bloğuyla boşluk tuşuna basılmasını kontrol ettik. **Eğer boşluk tuşuna basılı** ise kılık değiştirmesini sağladık ve yürüme efektini verdik.

Olana Kadar Bekle Bloğu

Bazen yazdığımız programlarda bir sonra adıma geçmek için bir olayın gerçekleşmesini beklememiz gerekebilir. Bu durumda olana kadar bekle bloğunu kullanırız. **Olana kadar bekle** bloğunda koşul gerçekleşinceye kadar program o adımda bekler. Koşul gerçekleştiğinde bir sonraki adıma geçilir.

Koşu oyunumuzda da boşluk tuşuna her bastığımızda kılığın bir defa değişmesini istediğimizden **sonraki kılık** bloğundan sonra bir **olana kadar bekle** bloğu kullanıyoruz. Koşul olarak bu sefer **boşluk tuşuna basılı mı? değil** durumunu kontrol ediyoruz. İşlemler menüsü altında bulunan **değil** bloğu içine koyduğumuz **boşluk tuşuna basılı mı?** bloğu ile bu sefer tam ters bir durumu boşluk tuşuna basılı olmama durumunu kontrol ediyoruz. Böylece boşluk tuşuna basılı durumda karakter sonraki kılığa geçiyor ama altındaki olana kadar bekle bloğu ile boşluk tuşu basılı olmama durumuna kadar bekliyor. Her seferinde boşluk tuşuna basıldığında sadece bir kez kılık değiştirmiş oluyor.



Diğer kod bloğumuzda ise aynı yöntemle bu sefer her boşluk tuşuna basıldığında sadece bir kez 7 adım gitmesini sağladık. Program boşluk tuşuna basıldığında 7 adım gider ve boşluk tuşundan elimizi çeken kadar bekler. Bu şekilde her boşluk tuşuna basıldığında sadece bir kez 7 adım gider.



Jamie için yazacağımız son kod bloğunda ise bitiş çizgisinin rengine değip değmeme durumunu kontrol ediyoruz. Eğer karakterimiz bitiş çizgisinin rengine değmişse ekrana başardım yazdırıyoruz.

Renk olarak bitiş çizgisinin rengini seçmek için; önce bloktaki renk kısmına tıkladıktan sonra bitiş çizgisine tıklıyoruz. Renk olarak bitiş çizgisinin rengini belirlemiş oluyoruz.

Bitiş çizgisine vardıkdan sonra önce başardım yazısı çıkar, 2 saniye sonra da **hepsi durdur** bloğu ile tüm program durur.

Dog2 Karakteri için Yazılacak Kodlar



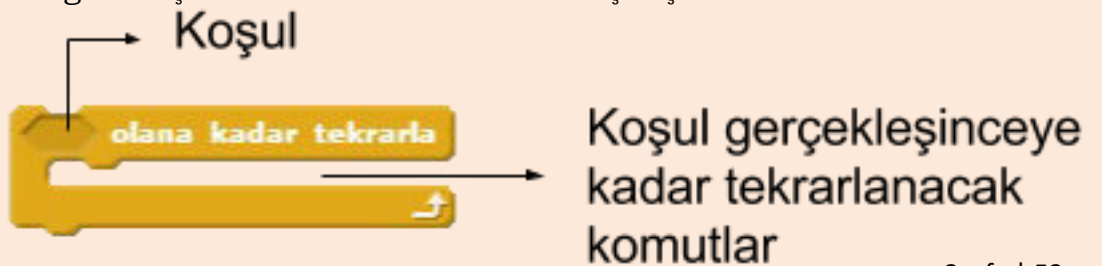
Karakterimiz için kod yazabilmek için kuklalar kısmından Dog2 karakterine tıklayınız. Daha sonra diziler sekmesi açıkken kodlarımızı yazalım.



1 İlk kısımda karakterimizin katmanını ayarlıyoruz. Önce üste çık sonra 1 katman alta in diyerek katman olarak Jamie karakterimizin altında olmasını sağlıyoruz. Daha sonra karakterimizin X ve Y konumunu ayarladıktan sonra bir olana kadar bekle blođuyla bitiş çizgisinin rengine deđene kadar sonraki kılıđı geçmesini sağlıyoruz. Bitiş çizgisine varıldıđında olana kadar tekrarla döngüsü biter ve bir sonraki kod blođu çalışır ve Dog2 karakteri ekrana Hav Hav yazar.

Olana Kadar Tekrarla Blođu

Olana kadar tekrarla blođunda koşulumuz gerçekleşene kadar içerisindeki kodları çalıştıran bir döngü oluşturulur. Ne zaman koşul gerçekleşir o zaman döngüden çıkılır ve bir sonraki komut çalışır.



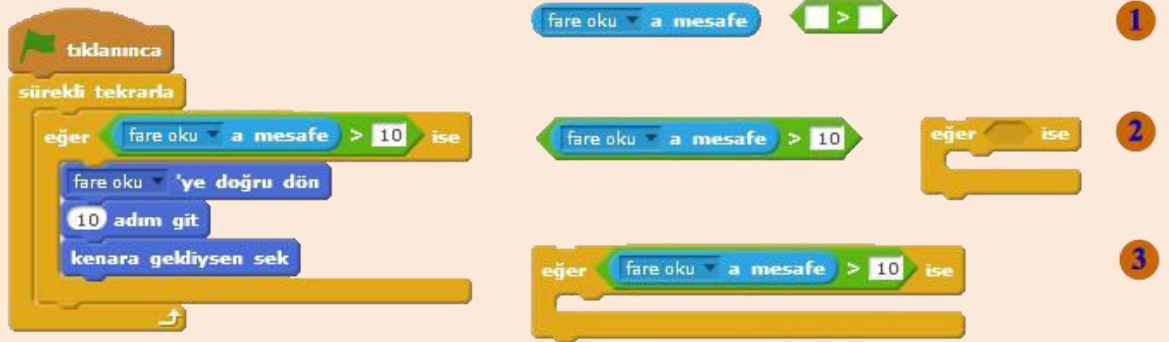
2 İkinci kısımda karakterimizin ileriye doğru hareketini sağlıyoruz. Yine bir **olana kadar tekrarla** bloğu ile Dog2 karakterinin bitiş çizgisine varana kadar ilerlemesini sağlıyoruz.

Fare ile Hareket Kontrolü

Algılama blokları içerisinde farenin X ve Y konumunu ya da karaktere olan uzaklığını tespit etmemize yarayan kod blokları mevcuttur. Şimdi blokları kullanarak karakterimizi kontrol edelim.

- 1 fare oku a mesafe
 - 2 farenin x'i
 - 3 farenin y'si
- 1 Fare okuna olan mesafeyi hafızasında tutan kod bloğu. Fare okunu değiştirip eklediğimiz başka bir karaktere olan mesafeyi de ölçebiliriz.
- 2 Farenin o anki X koordinatını veren kod bloğu

3 Farenin o anki Y koordinatını veren kod bloğu



1 İşlemler menüsü altında bulunan **büyükür** işlem bloğunu ve algılama menüsü altında bulunan **fare okuna mesafe** bloğunu alalım. **Fare okuna mesafe** bloğunu **büyükür** işlem bloğunun sol tarafına yerleştirelim.

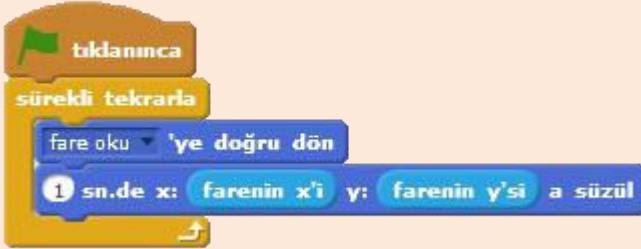
2 Birleştirdiğimiz blokları eğer ise bloğunun boş kısmına yerleştirelim.

3 Birleştirdiğimiz bloklar fare okuna olana mesafeyi ölçecektir. Fare okuna olan mesafe 10'dan büyükse koşul doğru olacak ve eğer ise bloğumuzun içindeki kodlar çalışmaya başlayacaktır.

Bu programımızda bir eğer ise kod bloğuyla karakterimizin fare okuna olan mesafesini ölçeceğiz. Eğer mesafe 10'dan büyükse koşulumuz gerçekleşecek ve kodlarımız çalışmaya başlayacak.

Koşul gerçekleştiğinde karakterimiz sürekli olarak fare okuna dönecek, 10 adım gidecek ve kenara geldiye sekecektir. Fare oku ile olan mesafe 10'dan küçük değilse kod bloklarımız çalışmayacak ve karakterimiz duracaktır.

Bu programımızda ise karakterimizin bir taraftan sürekli olarak fare okuna dönmesini sağlarken diğer taraftan X ve Y konumunu farenin X ve Y konumu olarak ayarladık. Karakterimiz sürekli olarak farenin X ve Y konumuna belirtilen saniyede süzülecektir.



İkiz Yaratma

Scratch'da oyunlarımızı yaparken bazen bazı karakterlerden birden fazla kullanmamız gerekebilir. Bu durumda karakterimizin ikizini yaratarak o ikizi oyunumuzda kullanırız.

1 kendim in ikizini yarat

2 ikiz olarak başladığımda

3 bu ikizi sil

1 Karakterimizin ikizini yaratmak için **kendimin ikizini yarat** kod bloğunu kullanırız. Bu kod bloğu çalıştığında karakterimizin bir ikizi oluşur.

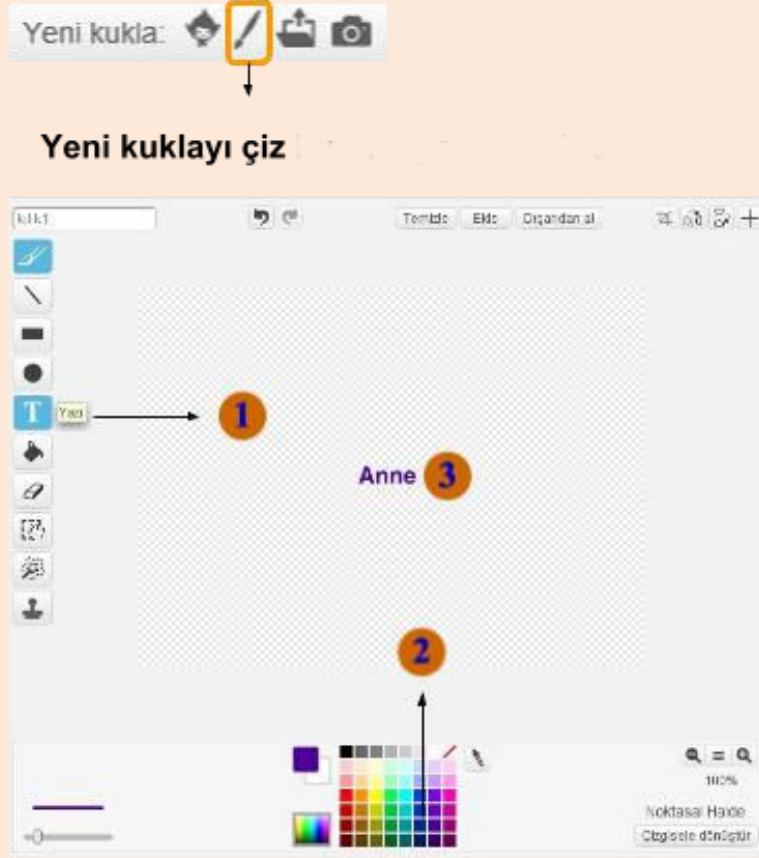
2 İkiz oluşturulduktan sonra ne yapacağını **ikiz olarak başladığımda** kod bloğunun altına yazıyoruz.

3 Yarattığımız ikizin işi bittiğinde silmemiz gerekir. İkizi silmek için **bu ikizi sil** bloğunu kullanırız.

Canım Annem Animasyonu

Hepimiz annelerimizi çok severiz. Gelin isterseniz hep beraber annemiz için bir animasyon yapalım.

Yeni bir karakter çizelim ve anne yazalım.



Yeni kuklayı çiz butonuna tıkladıktan sonra resim düzenleme editörü açılacaktır. Yazı yazma aracını ve rengi seçtikten sonra tam orta noktaya bir anne yazısı yazalım.

Böylelikle yeni bir karakter oluşturmuş olduk.

Anne Karakteri için Yazılacak Kodlar



Oluşturduğumuz anne karakteri seçiliyken yan tarafta bulunan kodları yazalım. X ve Y konumunu, büyüklüğü ayarlayalım ve üste çık diyerek katman olarak en üstte olmasını sağlayalım.

Daha sonra **kuklayı kütüphaneden seç** butonunu kullanarak **muhtelif** kısmından **Heart** karakterini ekleyelim.

Heart Karakteri için Yazılacak Kodlar

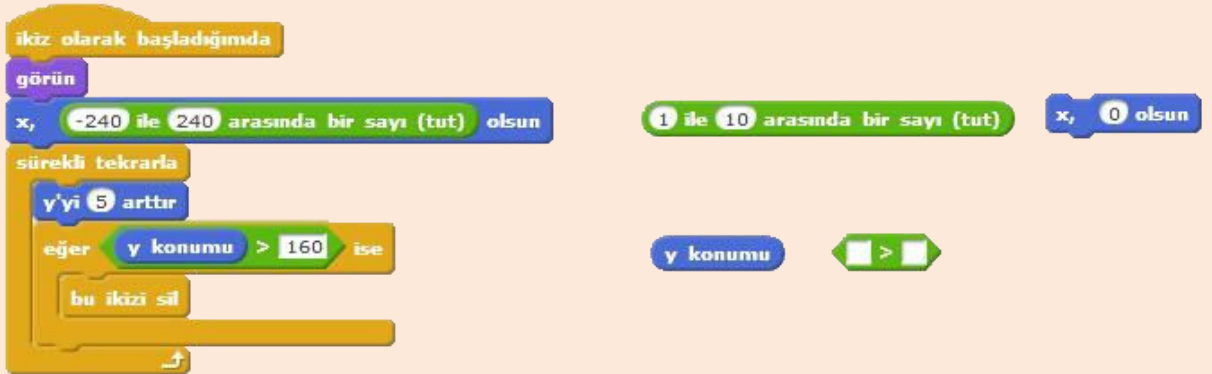


Karakterimiz için kod yazabilmek için kuklalar kısmından Heart karakterine tıklayınız. Daha sonra diziler sekmesi açıkken kodlarımızı yazalım.



Karakterimizin kopyasını çıkarmaya başlamadan önce, karakteri gizliyoruz. Karakterimizin ikizlerini kullanacağımız için asıl karakterin gözükmemesine ihtiyacımız yok. Daha sonra karakterlerimizin ikizlerini yarattıktan sonra yukarıya doğru yükselme efekti vereceğimiz için Y konumunu -200 yapıyoruz.

Sürekli tekrarla bloğu ile karakterin sürekli ikizini yaratıyoruz her seferinde yaratılan ikizin farklı renk olması için **renk etkisini 25 arttır** komutu ile rengini değiştiriyoruz. Karakterin ikizlerini yaratırken programın çok hızlı olmaması için **0.5 saniye bekle** bloğu ile programı yavaşlatıyoruz.



İkiz olarak başladığımda bloğunun altına ikiz yaratıldıktan sonra o ikizin ne yapması gerektiğini yazarız. Burada öncelikle ikizimizin görünmesini sağlıyoruz çünkü; asıl karakterimizi gizlemiştik. Daha sonra işlemler menüsü altında bulunan **1 ile 10 arasında bir sayı tut** bloğuyla X konumunu ayarlarız. X konumu -240 ve 240 arasında değerler aldığı için **-240 ile 240 arasında bir sayı tut** bloğuyla X konumunun rastgele bir değer almasını sağlıyoruz.

Sürekli tekrarlar bloğuyla da Y konumunu sürekli arttırarak yukarıya doğru yükselme efekti veriyoruz. Sürekli tekrarlar bloğunun içinde kullandığımız eğer ise bloğuyla da Y konumu 160 değerini geçtiğinde **bu ikizi sil** bloğuyla ikizi siliyoruz. Y konumunu hafızasında tutan **Y konumu** bloğunu hareket blokları altında bulabilirsiniz. **İşlemler** menüsü altında bulunan **büyüktür** bloğuyla karşılaştırma yapıyoruz. Karakterin Y konumu 160'dan büyükse koşulumuz gerçekleşiyor ve ikiz siliniyor.



Sürekli kalp karakterinin ikizi yaratılıyor. Yaratılan ikizin rengi her seferinde farklı oluyor ve yavaşça yukarıya doğru süzülüyor. Böylelikle çok güzel bir animasyon yapmış oluyoruz. İstersek animasyonumuza bilgisayarımızda kayıtlı olan bir ses dosyasını ya da Scratch kütüphanesinde bulunan ses dosyalarından birini ekleyerek animasyonumuzu daha da güzelleştirebiliriz.

İŞLEMLER ve VERİ

İşlemler Blokları

Programlamada matematiksel işlemler önemli bir yere sahiptir. Bazen bir karakterin büyüklüğü ya da bir özelliğinin diğer özelliğine oranı gibi hesaplama gerektiren işlemler yapmamız gerekebilir. Bu gibi durumlar için Scratch'da işlemler bloklarını kullanacağız.

Aritmetik İşlem Blokları

İşlem bloklarını başka blokların içerisinde kullanacağız. Sürükle bırak yöntemiyle işlem bloğumuzu istediğimiz bloğun içine sürükleyip bırakacağız.



Hello de bloğunda hello yazan yere sürükle bırak yöntemiyle işlem bloğumuzu taşıyoruz. Sürükleyip bıraktığımız yerde artık işlem bloğumuz çalışmaya başlıyor. İşlem blokları ile toplama, çıkarma, bölme, çarpma gibi aritmetik işlemleri yapabiliyoruz.

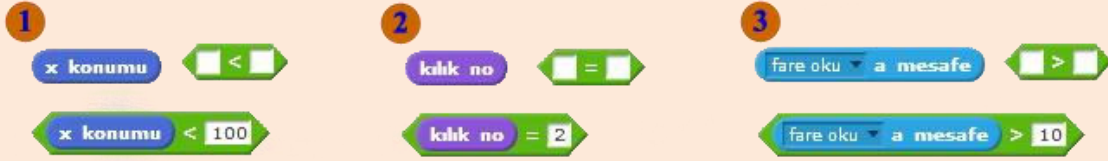


Hepimizin oynadığı bir oyun vardı. Aklından bir sayı tut oyunu. Oyunda aklımızdan bir sayı tutardık ve onu tahmin etmeye çalışırdık.

Bilgisayarlar akıllarından bir sayı tutamazlar ama rastgele bir sayı üretebilirler. **1 ile 10 arasında bir sayı tut** bloğu ile bilgisayar belirttiğimiz aralıkta rastgele sayılar üretir. Burada da X konumumuzun rastgele değişmesi için 1 ile 10 arasında bir sayı tut bloğunu kullandık. X konumu -240 ile 240 arasında bir değer alacağı için değerlerimizi -240 ve 240 olarak belirledik. Böylelikle karakterimiz bu kod bloğu çalıştığında -240 ile 240 arasında rastgele bir X konumuna gidecektir.

Mantıksal Karşılaştırma Blokları

Programımızın belli bölümlerinde bazı karşılaştırma işlemleri yapacağız. Örneğin; fare konumu belli bir değerden büyük mü? Karakterimizin Y konumu istediğimiz değeri geçmiş mi? Bu durumları ölçmek ve karşılaştırmak için karşılaştırma bloklarını kullanacağız. Karşılaştırma bloklarını daha çok kontrol blokları ile kullanacağız.

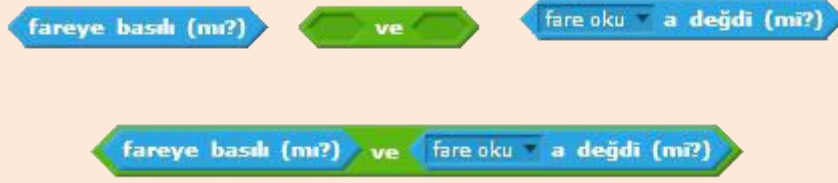


1 Küçüklük durumunu karşılaştırmak için kullanılır. Sol tarafı sağ tarafla karşılaştırır ve eğer sol taraf sağ taraftan küçükse koşul sağlanmış olur. Eğer sol taraf sağ taraftan küçük değilse koşul sağlanmaz. Burada hareket menüsünde bulunan karakterin X konumu bloğunun değeri 100'den küçükse koşul sağlanmış olur. 100'e eşit ya da 100'den büyükse koşul sağlanmaz.

2 Eşitlik durumunu kontrol etmek için kullanılır. Sol taraf sağ tarafla karşılaştırılır değerlerin birbirine eşit olmasına bakılır. Eğer değerler birbirine eşitse koşul sağlanmış olur. Burada görünüm menüsünde bulunan kılık no karakterin kılığının numarasını hafızasında tutar. Kılık no bloğunun hafızasında tuttuğu sayı 2'ye eşitse koşul sağlanır. 2'den küçük ya da büyükse koşul sağlanmamış olur.

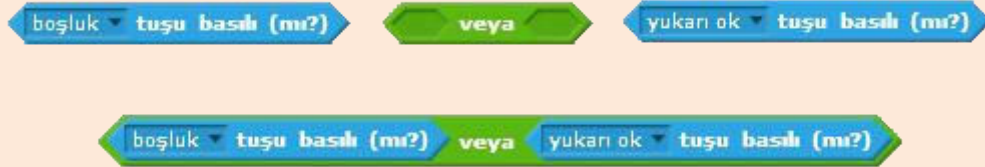
3 Büyüklük durumunu kontrol etmek için kullanılır. Sol taraf sağ tarafla karşılaştırılır, sol taraftaki değer sağ taraftaki değerden büyük olması gerekir. Eğer sol taraftaki değer sağ taraftaki değerden büyükse koşul sağlanmış olur. Eğer sol taraftaki değer sağ taraftaki değere eşit ya da büyükse koşul sağlanmaz. Burada algılama menüsü altında bulunan fare okuna mesafe bloğu karakterin fare okuna olan mesafesini hafızasında tutar. Eğer karakterin fare okuna mesafesi 10'dan büyükse koşul sağlanır.

Ve Bloğu



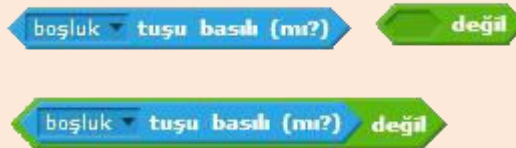
Programlarımızı yazarken belli kodların çalışması için tek koşul yetmeyebilir. Bazen iki koşulun aynı anda gerçekleşmesi gerekebilir. Örneğin; Facebook internet sitesine girerken hem kullanıcı adı hem de şifrenizin doğru olması gerekir. Bu durumda koşul olarak kullanıcı adı ve şifre doğru mu? Koşulunu uygularız. Aynı anda iki koşulu birleştirmek için **ve** bloğunu kullanırız. Burada da koşulumuz farenin basılı olması **ve** fare okunun karaktere değmesi. Koşulun sağlanması için fare okunun karaktere değerken fareye tıklanması gerekmektedir. Sadece fareye tıklamak tek başına yeterli olmayacaktır.

Veya Bloğu



İki koşulu aynı anda uygulamak istediğimizde **ve** bloğunu kullanıyoruz. İki koşuldan en az biri çalıştığında koşulumuzun gerçekleşmesini içinse **veya** bloğunu kullanıyoruz. Burada boşluk tuşuna basılı iken, yukarı ok tuşu basılı olsa da olmasa da koşulumuz gerçekleşmiş olur. Yani sadece boşluk tuşu ya da sadece yukarı ok tuşu basılı olsa bile koşulumuz gerçekleşmiş olur.

Değil Bloğu



Programlarımızda bazen koşulumuzun tam tersi bir durumu kontrol etmemiz gerekebilir. Doğru bir ifadenin tersi yanlıştır, yanlış bir ifadenin tersi ise doğru. Buradaki örneğimizde koşul boşluk tuşuna basılı olması. Bu koşulun değili ise boşluk tuşuna basılı olmamasıdır. Koşulumuzun

çalışması için koşulun doğru olması gerekir. Burada koşulun çalışması için boşluk tuşuna basılı olmaması gerekir.

Birleştir Bloğu



Birleştir bloğu ile iki bloğu ya da cümleyi birleştirebiliriz. Birleştir bloğunun sol tarafı ile sağ tarafı birleştirilerek işlem yapılır. Burada sabit olarak gelen bloktaki sol tarafta yazan hello yazısı ile sağ tarafta bulunan world yazısı birleştirilerek işleme alınır. Ekran yazdır komutunu kullandığımız için ekrana "hello world" yazdırılır.



Bu örneğimizde sol tarafa "Farenin X konumu:" yazısını ekledik. Sağ tarafa ise **farenin x'i** bloğunu ekledik. Bu kod bloğu çalıştığında bu iki kısım birleştirilerek ekrana farenin X konumu yazdırılır.



Bazı programlarımızda birleştirilecek kısım sayısı 2'den fazla olabilir. Bu durumda sağ tarafa bir **birleştir** bloğu daha ekleriz. Bu şekilde birleştirilecek 3 kısım elde ettik. Sol tarafa algılama menüsü altındaki **şimdiki saat** bloğunu, ortaya ":" , sağ tarafa ise yine algılama bloğu altındaki **şimdiki** bloğunu aldık ve saat yerine dakikayı seçtik. Bu şekilde ekrana o anki saat ve dakikayı yazdırmış olduk.



1 Kelimenin uzunluđunu tutan kod blođu. Burada “world” kelimesinin uzunluđu 5 olduđu için ekrana 5 yazar. 2 Kelimenin istenilen harfini seçmemize yarayan kod blođu. Burada 1 yazdığımız için 1. Harf olan “w” harfini ekrana yazar. 3 Sol tarafta bulunan sayının sađ tarafta bulunan sayıya bölümünden kalan sayıyı verir. Burada 6’nın 2’ye bölümünden kalan 0 olduđu için ekrana 0 yazar. 4 Verilen sayıyı yukarıya ya da aşağıya yuvarlayan kod blođu. Burada sayı 2.6 olduđu için yukarıya yuvarlanır ve ekrana 3 yazılır. 5 İstenilen sayının karekökünü, mutlak deđerini vb. deđerleri bulan kod blođu. Burada 9 sayısının karekökünü buldurduğumuz için ekrana 3 yazılır.

Sor ve Bekle Bloęu

Programlarımızda kullanıcının dışarıdan bilgi girmesi gerektiğinde sor ve bekle bloęunu kullanırız. Mesela bir oyun tasarladığınızı düşünelim. Bu oyunda bir skor tablosu oluşturmak istiyorsunuz. Bu durumda oyununuzu oynayan kişinin adını programa girmesi gerekir. Kullanıcının adını programa girebilmesi için sor ve bekle bloęunu kullanacağız. Sor ve bekle bloęu çalıştıktan sonra ekranda bir veri giriş kısmı çıkar bu ekrana yazılanlar yanıt bloęunun hafızasına alınır. Yanıt bloęunu da kullanarak işlemlerimizi yaparız.



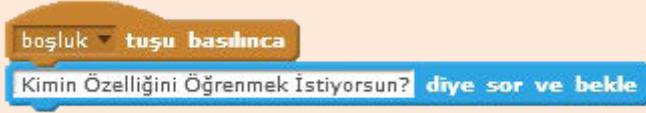
Boşluk tuşuna bastığımızda **adın ne? diye sor ve bekle** bloęu çalışır ve ekrana yazı yazabileceğimiz bir pencere gelir.



Pencereye yazdıklarımız yanıt bloęunun hafızasında tutulur ve merhaba ile yanıt bloęunu birleştirdiğimiz için ekrana Merhaba ve yanıt yazdığınız yazı birlikte yazılır.

Sınıf Arkadaşlarım Uygulaması

Hep birlikte sınıftaki arkadaşlarımızın özelliklerini söyleyen bir program yapalım. Klavyeden sınıf arkadaşımızın ismi girildiğinde onun özelliğini ekrana yazdıralım.



Boşluk tuşuna basılınca altına bir sor ve bekle bloğu ekleyelim. Her boşluk tuşuna basıldığında yazdığımız soruyu soracak ve bizim sınıf arkadaşlarımızdan birini ismini yazmamızı bekleyecek.



Yanıt girildikten sonra kontrol kısmı başlayacak. Her isim için bir eğer ise bloğu kullanacağız. Koşul olarak yanıtın arkadaşımızın ismine eşit olup olmadığını belirleyeceğiz. Burada isim olarak Bora seçildiği için eğer girilen isim Bora ise ekrana “Zeka küpü ama kendini bilmiş” yazar. Ekrana bu yazının çıkması için koşulun gerçekleşmesi gerekir yani yanıtın Bora olması gerekir.

Diğer arkadaşlarımız içinde kontrol bloğu oluşturmak için Eğer ise bloğuna sağ tıklayalım ve kopyasını çıkart diyelim.



Bloğa sağ tıkladıktan sonra kopyasını çıkart dediğimizde bloğun bir kopyası oluşur. O kopyada isim kısmını ve ekrana yazılacakları değiştirdikten sonra diğer eğer ise bloğunun altına ekleyelim.



Biz örneğimizde 4 adet öğrenci için kontrol bloğu oluşturduk. Siz sınıftaki arkadaşlarınızın sayısı kadar oluşturabilirsiniz.

Büyük küçük harf ayrımı yoktur yalnız I harfi istisna. Scratch'ı hazırlayanlar İngiliz alfabesini kullanırlar. İngilizcede de küçük ı bulunmaz. İngilizcede büyük I küçük i vardır. O yüzden Işık ismini kullanırken bir veya bloğu ile hem küçük harf hem de büyük harf ile başlangıç olasılığını dikkate aldık.

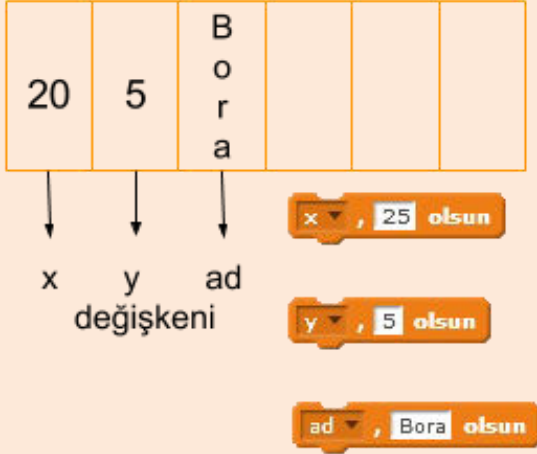
Değişkenler

Oyun oynarken bir engeli geçtiğinizde ya da verilen görevleri yerine getirdiğinizde puan kazanırsınız. Peki bu kazandığınız puanları bilgisayar hafızasında nasıl tutuyor hiç düşündünüz mü?

Bilgisayarlar hafızalarında bir sayıyı ya da yazıyı tutmak için değişkenleri kullanır. Hafızada tutulmak istenen değer oluşturulan bir değişkene aktarılır ve bu değer o değişkende saklanarak hafızada tutulmuş olur.

Değişkenleri bilgisayar hafızasındaki odalara benzetebiliriz. Bir değişken oluşturduğumuzda bilgisayar hafızasındaki bir oda bu değişkene verilir. Bu değişkene aktardığımız değerler o odaya kaydedilir ve kullanmak istediğimizde odada kayıtlı olan değeri kullanırız.

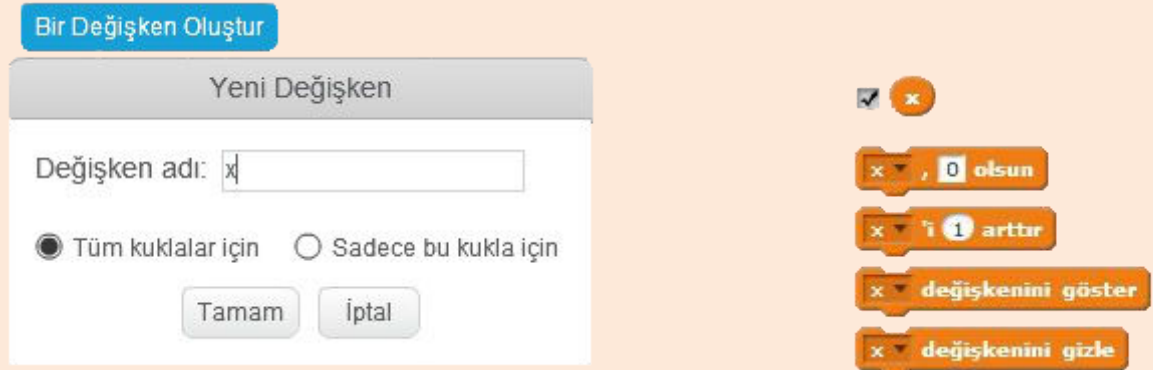
Bilgisayar Hafızası



Her değişken oluşturduğumuzda bilgisayar hafızasında da o değişkene ait bir oda oluşturulur ve değişkene aktarılan değer bu odaya kaydedilir. Değişkenin değeri değişirse odadaki değerde değiştirilir. Değişkenin değerini kullanmak istediğimizde odadaki değere bakılır ve o değer çağırılarak kullanılır. Bu yöntemle oyundaki puanımız, silahımızda kalan mermi gibi değerler bilgisayar hafızasında tutulur.

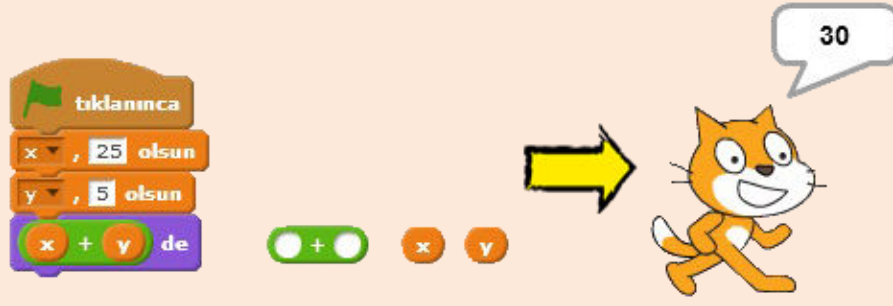
Değişken Oluşturma

Scratch'da değişken oluşturmak ve kullanmak çok basittir. Veri menüsünden bir değişken oluştur dedikten sonra gelen pencereden değişkenin ismini yazarız ve tamam dediğimizde değişken oluşturulmuş olur. Oluşan değişkeni kullanabilmemiz için onunla birlikte bazı kod blokları da oluşur.



Oluşan bloklardan X olsun ile değişkene değer aktarabiliriz, X'i 1 arttır ile değişkenin değerini arttırıp azaltabiliriz, X değişkenini gizle ve göster ile ekranda görünmesini ya da gizlenmesini sağlayabiliriz. Eğer X'den başka değişkenimiz varsa X yazan yeri değiştirerek bu işlemler diğer değişkenlere de uygulayabiliriz.

Sadece X yazan küçük yuvarlak blokta da X değişkeninin değeri tutulur. X değişkeninin değerini kullanmak istediğimizde bu kod bloğunu kullanırız.



X ve Y adında iki deęişken oluřturalım. X deęişkenine 25 deęerini Y deęişkenine 5 deęerini aktaralım. Daha sonra bir topla bloęuyla bu X ve Y deęişkenlerinin toplamını ekrana yazdıralım. X deęişkeni 25 deęerini Y deęişkeni de 5 deęerini tařıdığı için ekrana 30 yazılır.

Tıklama Oyunu

Karakterimizin ekranda saklambaç oynar gibi gizlendięi ve bizim onu fareyle yakalamaya çalıştıęımız bir oyun yapalım.

- 1 **Kuklalar** kısmından sahneyi seęelim. Sahnenin arka planını **dekorlar** kısmından **kova aracı** ile tamamen yeřil yapalım.
- 2 **Kuklayı kütüphaneden seę** butonu ile **muhtelif** kategorisinden **Tree** ve **Trees** aęaçlarından ikiřer tane ekleyelim.
- 3 Ekledięimiz Tree ve Trees karakterlerinin sahnedeki konumlarını, büyüklüklerini ve katmanlarını ayarlayalım. Ařaęıdaki resim sadece örnektir, siz kendinize göre deęiřtirebilirsiniz.

Resimdeki kodları ilgili ağaç karakterine yazalım. Ağaç karakterlerini tek tek seçerek kodlarımızı yazalım



Oyunda karakterimiz rastgele bu ağaçların olduğu konuma gidecek ve biz karaktere tıklamaya çalışacağız. Bunu sağlayabilmek için bir değişken oluşturacağız ve bu değişkene rastgele değerler aktaracağız. Bu değerlere göre de karakterimizin rastgele bir ağaca gitmesini sağlayacağız.

4 **Kedi karakterimizi seçelim.** Kedi karakteri için puan ve tahmin adında iki değişken oluşturalım. Tahmin değişkenini karakterimizi rastgele bir ağacın konumuna göndermek için, puan değişkenini ise karakterimize tıkladığında kazanılan puanı bilgisayar hafızasında tutmak için kullanacağız.



Oluşturduğumuz tahmin değişkeni için. Tahmin değişkeni olsun bloğunu alıyoruz. Boş kısmına sürükleyip bırak yöntemiyle bir adet 1 ile 10 arasında sayı tut bloğunu yerleştiriyoruz. 1 ile 10 yerine 1 ile 4 arasında bir sayı tut diyoruz. Böylece tahmin değişkeni 1 ile 4 arasında rastgele bir değer almış oluyor.

5 Tahmin değişkeninin aldığı değere göre eğer ise blokları ile hangi ağaca gideceğini belirleyelim.



Tahmin değişkenimiz 1,2,3,4 değerlerinden birini rastgele alacaktır. Biz de eğer ise bloklarıyla tahmin değişkeni 1 değeri alırsa ne yapacak, 2 değerini alırsa ne yapacak yazacağız.

Dört adet eğer ise bloğunu alıyoruz. Tahmin değişkeninin değeri 1'e eşitse Tree'na git bloğu ile Tree karakterinin olduğu konuma gitmesini sağlıyoruz. Tree'na git bloğu hareket menüsü altındaki fare oku'na git bloğudur. Fare oku yazan yeri değiştirerek Tree yaptık. Büyüklüğünü de %30 yaptık ki sanki uzaktaymış gibi algılanmasını sağladık.

Aynı işlemleri tahmin değişkeninin 2,3 ve 4 gelme ihtimali için de yaptık.

Not: Bir karakter için kod yazmadan önce kuklalar kısmından karakteri seçiniz. Daha sonra kod yazma kısmından o karakter için yazacağınız kodları yazınız.

6 Karakterimiz her seferinde bir ağaç karakterinin olduğu konuma gidecektir. Bir 10 adım git bloğunun içine -50 ile 50 arasında sayı tut bloğu koyarak ağacın sağında ya da solunda belirmesini sağlıyoruz.



1 saniye bekle bloğuyla bu işlemleri her yaptığında 1 saniye beklemesini sağlıyoruz. Eğer 1 saniye bekle bloğunu kullanmazsak, program çok hızlı çalışacaktır ve bize kedi karakterine tıklayabilecek zaman kalmayacaktır.

7 Karakterimizin sürekli olarak rastgele konumlara giderek saklanmasını sağladık. Şimdi ise karakterimize tıkladığında puan kazanalım.



Olaylar menüsü altında bulunan **bu karakter tıklanınca** bloğunun altına **puan'ı 1 arttır** ve **pop sesini çal** bloklarını ekliyoruz.

Not: Pop sesini çalabilmek için öncelikle seslerden pop sesini ekleyerek, sesini çal bloğunda seçmeniz gerekmektedir.

Kedi Karakteri için Yazılacak Tüm Kodlar



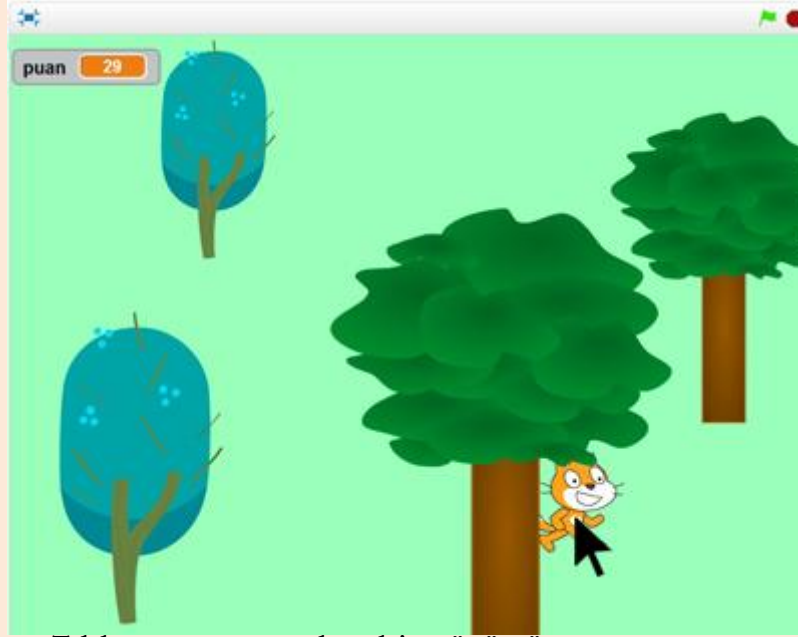
```
bu kukla tıklanınca
puan 'i 1 arttır
pop 'sesini çal
```

Puan 0 olsun kod bloğuyla program her başladığında puanımızın sıfırlanmasını sağladık.

Puan değişkenini göster ile puanımızın sahnede gözükmesini sağlarken, tahmin değişkenini gizle ile de tahmin değişkeninin değerlerinin sahnede gözükmemesini sağladık.

Programımızı bir sürekli bloğu içine aldık ki bir defa tekrarlanıp bitmesin. Sürekli olarak devam etsin.

```
tıklanınca
puan , 0 olsun
puan değişkenini göster
tahmin değişkenini gizle
sürekli tekrarla
tahmin , 1 ile 4 arasında bir sayı (tut) olsun
eğer tahmin = 1 ise
Tree 'na git
büyüklüğü % 30 yap
eğer tahmin = 2 ise
Tree2 'na git
büyüklüğü % 60 yap
eğer tahmin = 3 ise
Trees 'na git
büyüklüğü % 50 yap
eğer tahmin = 4 ise
Trees2 'na git
büyüklüğü % 20 yap
-50 ile 50 arasında bir sayı (tut) adım git
1 saniye bekle
```



Tıklama oyunundan bir görünüm

Oyununuzu test ederken, karaktere tıklarken ağaçların yer değiştirmemesi ya da katman yapısının bozulmaması için tam ekrandayken test edin.

Listeler

Bilgisayardaki bilgileri düzenli bir şekilde saklamak ya da kolay ulaşmak için onları bilgisayar hafızasında tutarken listeleri kullanırız. Listeler ile bilgiler daha düzenli olur. Biz de günlük hayatta birçok konuda listeleri kullanırız. Örneğin; telefonumuzdaki telefon numaralarını belli bir düzende telefon rehberinde saklarız. Telefon rehberi listelere çok benzer.

Bir Liste Oluştur

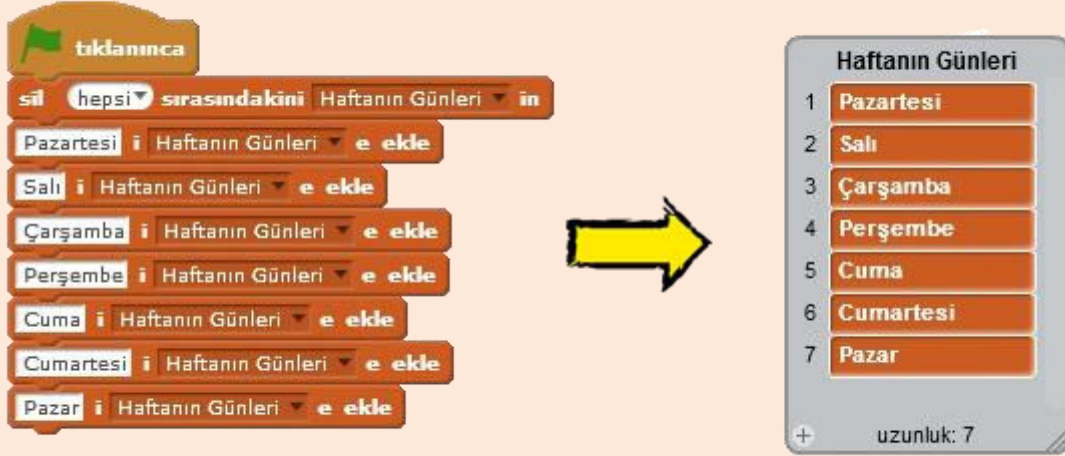
Yeni Liste

Liste adı:

Tüm kuklalar için Sadece bu kukla için

Veri menüsü altından bir liste oluştur dediğimizde yeni bir liste oluşur. Bu liste ile birlikte listeyi kontrol etmemize yarayan kod blokları da oluşur.

Bir liste oluştur butonuna tıklayalım ve liste adı kısmına Haftanın günleri yazıp tamam diyelim. Listemizi oluşturduğumuzda bu liste ile ilgili işlemleri yapabilmek için kod bloklarımızda oluşur.

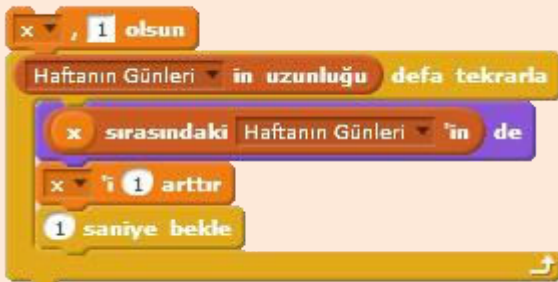


Veri menüsü altında bulunan liste blokları ile listemizdeki tüm elemanları ya da seçtiğimiz sıradaki elamanı silebiliriz. Listeye yeni bilgi ekleyebiliriz.

Not: Sahne üzerindeki liste penceresini sağ alt köşeden çekerek büyütebilirsiniz. Listenin sol alt kısmında bulunan + işaretine basarak listeye eleman ekleyebilirsiniz.

Bu programımızda önce listede kayıtlı olan tüm elemanları siliyoruz. Daha sonra sırasıyla listeye Pazartesi, Salı, Çarşamba, Perşembe, Cuma, Cumartesi, Pazar günlerini ekliyoruz. Listemize eklediğimiz her günün listede bir sırası oluyor. Pazartesi gününü ilk eklediğimiz için Pazartesi 1. sırada Salı günü ise 2. sıradadır. Listenin 1. sıradaki elemanını çağırdığımızda Pazartesi 5. sıradaki elemanını çağırdığımızda Cuma gelir.

Şimdi listemizdeki günleri sırasıyla ekrana yazdıralım. Bunun için X adında bir değişken oluşturalım. Bir döngü ile de bu değişkeni her seferinde 1 arttıralım. Değişkenin ilk değeri 1 olsun ve X sırasındaki liste elemanını ekrana yazdıralım. X değişkeninin ilk değeri 1 olduğu için ekrana Pazartesi yazdırılır. Daha sonra döngü içinde her seferinde değişken 1 artacağı için bir sonraki eleman yazdırılır. Döngünün 2. adımında X değişkeni bu sefer 2 değerini alacağı için 2. sıradaki eleman olan Salı ekrana yazdırılır.



Döngümüzü tekrarlatırken sayı yerine listenin uzunluğu defa tekrarla dedik. Sürükleyip bırak yöntemiyle **Haftanın Günleri'in uzunluğu** bloğunu döngüde 10 yazan yere yerleştirelim.

x sırasındaki Haftanın Günleri 'in de

x 1 sırasındaki Haftanın Günleri 'in

1 sırasındaki Haftanın Günleri'in kod bloğu ile 1. sıradaki liste elemanını çağırırız. Burada 1 yerine X değişkenini koyduğumuz için X değişkeninin değeri kaçsa o sıradaki eleman çağırılır.

Döngü ilk başladığında X'in değeri 1 olduğu için ekrana 1. sırada olan Pazartesi yazılır. Daha sonraki adımda X değişkenin değeri 1 artar ve X değişkenin değeri 2 olur, 1 saniye bekledikten sonra döngü başa döner. Tekrar çalışan blokta bu sefer X değişkeni 2 olduğu için ekrana 2. sıradaki liste elemanı yani Salı yazdırılır. Bir sonraki adımda X değişkeninin değeri tekrar 1 artar ve 3 olur. Döngüyü liste eleman sayısı kadar tekrarlattığımız için bu işlemler 7 kez tekrarlanır. Önce 1. sıradaki daha sonra 2. sıradaki derken 7. Sıradaki eleman ekrana yazdırılır ve haftanın bütün günleri ekrana yazdırılmış olur.

Yazılacak Tüm Kodlar

```

tıldanmca
sil hepsi sırasındaki Haftanın Günleri in
Pazartesi i Haftanın Günleri e ekle
Salı i Haftanın Günleri e ekle
Çarşamba i Haftanın Günleri e ekle
Perşembe i Haftanın Günleri e ekle
Cuma i Haftanın Günleri e ekle
Cumartesi i Haftanın Günleri e ekle
Pazar i Haftanın Günleri e ekle
x 1 olsun
Haftanın Günleri in uzunluğu defa tekrarla
x sırasındaki Haftanın Günleri in de
x i 1 arttır
1 saniye bekle

```

Puan Tablolu Tıklama Yarışması

Kim fareye daha hızlı tıklıyor görelim. Bir tıklama oyunu yapalım ve listeler ile herkesin puanını kaydedelim. Bakalım kim 1. olacak.

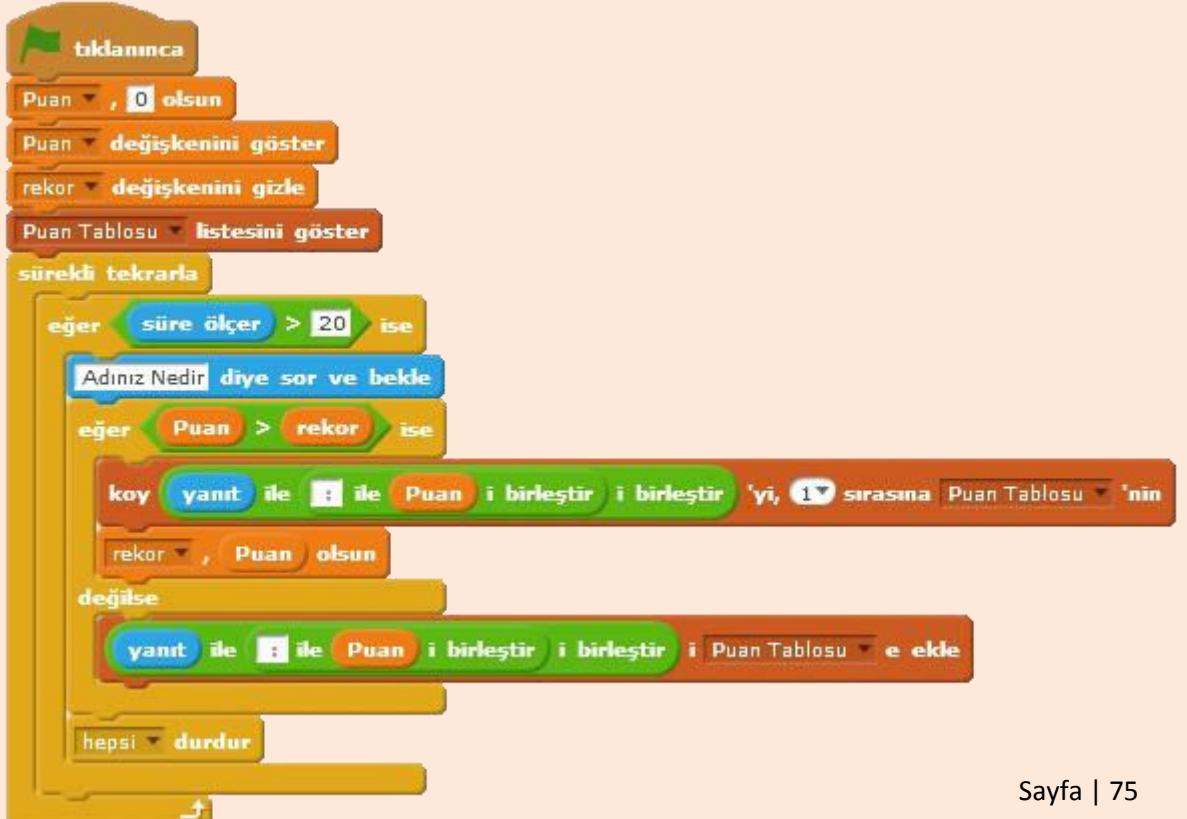
Kodlarımızı yazmaya başlayalım.



Bir adet puan adında bir adette rekor adında iki deęişken oluřturalım.

İlk yazacađımız kod karaktere tıklanınca puan kazanmayla ilgili. Yalnız puan kazanmanın 20 saniye sonra bitmesi için bir kod kullandık. **Algılama** menüsü altında bulunan **süre ölçer** bloęu yeřil bayraęa basıldıktan sonra geçen süreyi saniye cinsinden hafızasında tutar. **Süre ölçer** bloęunu kullanarak süreyi ölçüyoruz ve eđer süre 20 saniyenin altındayken karakter tıklanırsa puan deęişkenini 1 arttırıyoruz. Süre 20 saniyeyi ařmıřsa artık puan kazanamıyoruz.

Puan tablosu adında bir liste oluřturalım ve kodlarımızı yazmaya devam edelim. Bu kısımda 20 saniyelik süre bittięinde puanımızı listeye yazdıran kodları yazalım. Eđer puanımız diđer puanlardan daha yüksekse adımızı listenin ilk sırasına yazdıralım.



Kodlar biraz karışık görünmüş olabilir. Her şey ilk bakışta karışık gözükür. Şimdi adım adım programımızın çalışmasını inceleyeceğiz. Görüldüğü kadar karışık olmadığını göreceksiniz.



1 Programımızın yapacağı kontrolleri sürekli yapması için bir sürekli bloğunun içine alıyoruz.

2 İlk kontrolümüz süre ölçerin 20'den büyük olması. Süre ölçer 20'den büyük olmadıkça koşul sağlanmaz ve kodlarımız çalışmaya başlamaz.

3 Süre ölçer 20'den büyük olduğunda yani koşul sağlandığında ilk çalışacak blok **sor ve bekle** bloğudur. Bu adımda program **Adınız Nedir?** diye sorar ve bekler.

4 Adımımızı girdikten sonra ilk çalışacak kod bloğu **eğer ise değilse** kod bloğudur. Eğer ise değilse bloğumuzda koşul kazandığımız puanın rekor puanından büyük olup olmamasıdır. Rekor değişkeni ilk başta bir değer almamıştır. İlk kazanılan puan her zaman rekordan büyük olacak ve 5 ve 6 nolu bloklar çalışacak. Ekranı puan 1. olarak yazılacak ve yeni rekor bu puan olacaktır. İlk puan kazanıldıktan tekrar **yeşil bayrağa** bastığımızda yeni oyun için program hazır olacaktır. Program tekrar 4. basamağa geldiğinde bu sefer kazanılan puanı rekorla karşılaştıracak eğer bu yeni puan da rekordan büyükse 5 ve 6 nolu bloklar çalışacak ve yeni rekor bu puan olacaktır. Eğer puan rekordan büyük değilse bu sefer değilse kısmı çalışacak ve puan sadece tabloya eklenecektir.

5 Puanımız rekordan büyükse bu kod bloğu ile adımımızı ve puanımızı ekrana yazdırırız. Birleştir bloklarını kullanarak klavyeden girilen yanıtı ve puanımızı birleştiriyoruz. Bu birleştir bloklarını kullanarak da puanımızı listenin 1. sırasına yazdırıyoruz.



6 Eğer puanımız rekordan büyükse çalışacak diğer kod bloğu rekor puan olsunur. **Rekor 0 olsun** bloğunun 0 olan kısmına puan değişiminin değerini tutan bloğu sürükleyip bırak yöntemiyle taşıyoruz. Bu blok çalıştığında, eski rekordan büyük olduğu için sizin puanınız yeni rekor oluyor.



7 Eğer puanımız rekordan büyük değilse ekle bloğu ile puanımız listeye ekleniyor. Burada puanımız rekordan büyük olmadığı için listenin sonuna eklenir. 5 nolu blokta olduğu gibi birleştir bloklarıyla adımız ve puanımız birleştirilerek listeye eklenir.

8 **Eğer ise değilse** bloğu çalıştıktan sonra çalışacak olan son kod **hepsini durdur** kod bloğudur. Bu kod bloğu çalıştığında bütün program durur. Yeniden tıklama oyununu başlatmak için yeşil bayrağa tıklamamız gerekmektedir.

Son kodlarımızı yazalım.



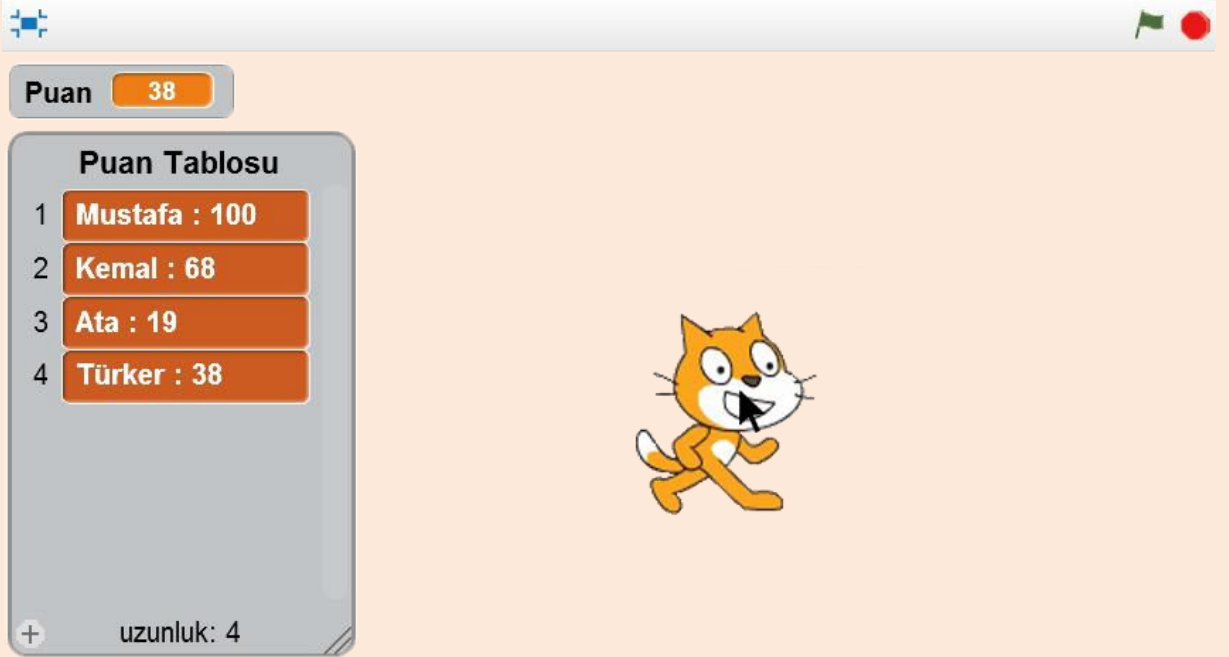
Son kod bloğumuzda ise boşluk tuşuna basıldığında rekoru sıfırlıyoruz ve liste elemanlarının hepsini siliyoruz. Artık listemiz yeni rekorlara hazır.

Yazılacak Tüm Kodlar



```
when clicked
  Puan, 0 olsun
  Puan değişkenini göster
  rekor değişkenini gizle
  Puan Tablosu listesini göster

  sürekli tekrarla
    eğer süre ölçer > 20 ise
      Adınız Nedir diye sor ve bekle
      eğer Puan > rekor ise
        koy yanıt ile 1 ile Puan i birleştir i birleştir 'yi, 1 sırasına Puan Tablosu 'nin
        rekor, Puan olsun
      değilse
        yanıt ile 1 ile Puan i birleştir i birleştir i Puan Tablosu e ekde
    hepsi durdur
```

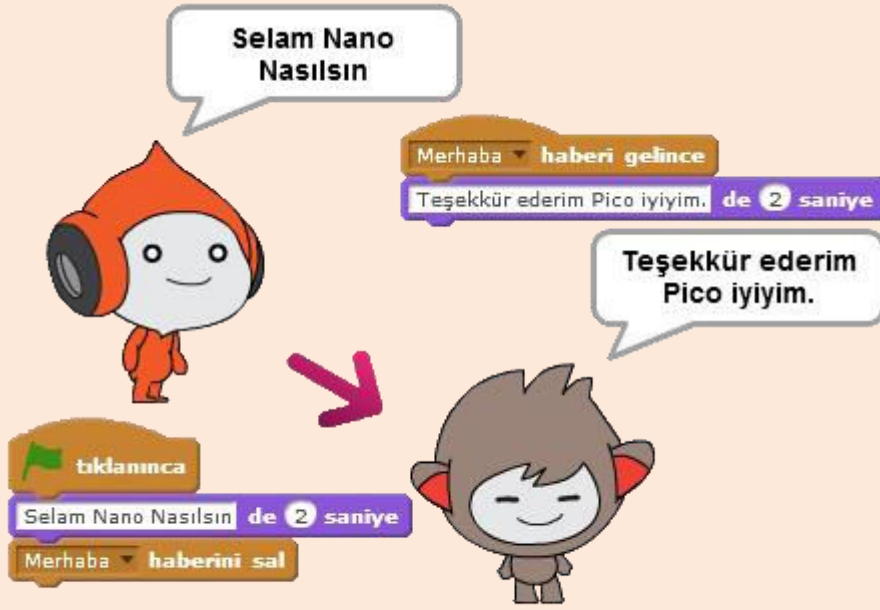


Puan tablolı tıklama yarışması ekran görüntüsü

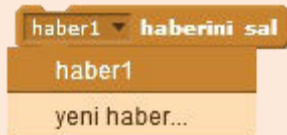
MESAJLAŞMA ve ÖZEL TAŞLAR

Mesajlaşma Blokları

Scratch'da oyunlarımızı yazarken birden fazla karakter kullandığımız durumlarda karakterlerin birbirleri ile haberleşmeleri gerekmektedir. Karakterler arasında haberleşmek için mesajlaşma blokları kullanılır. Bir durum gerçekleştiğinde bir karakter diğerine mesaj yollar ve diğer karakter mesajı aldığı anda önceden belirttiğimiz işleri yapar.



Mesaj göndermek istediğimiz yerde olaylar menüsü altındaki **haberini sal** bloğunu kullanıyoruz. **Haberini gelince** bloğu ile de o haber gelince yapılacak işlemleri yaptırıyoruz.



Haber oluşturmak istersek yeni haber diyerek haber oluşturabiliriz.



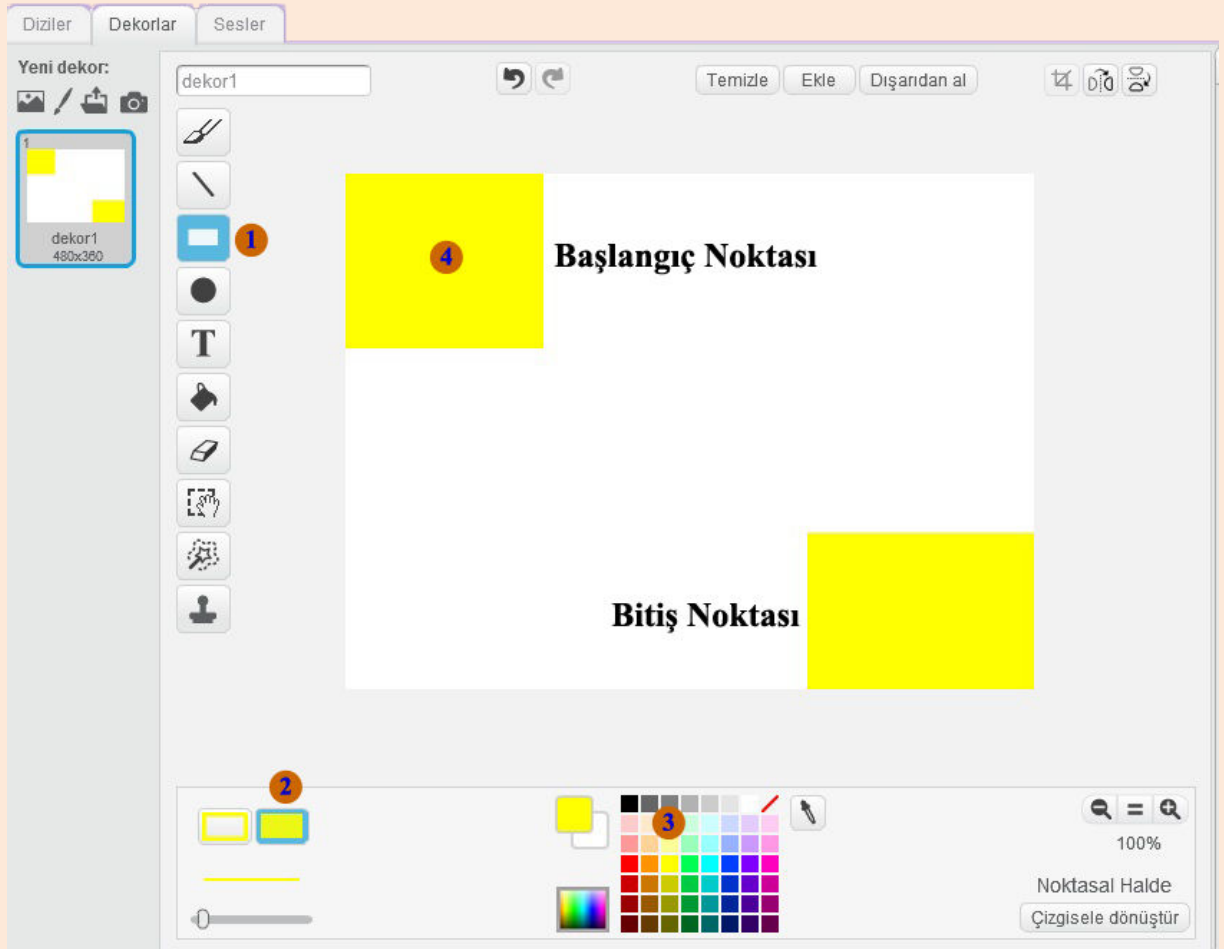
Haberini sal bloğunu kullandığımızda hiç bekleme yapmadan haberi gönderir ve bir alttaki blok çalışmaya başlar. **Haberini sal ve bekle** bloğunu kullandığımızda ise haber salınır, haberi alan karakterdeki

görevlerin bitmesi beklenir ve görevler bittikten sonra bir alttaki blok çalışmaya başlar.

Bölmümlü Labirent Oyunu

Mesajlaşma bloklarını kullanarak bölüm atlayabileceğimiz bir labirent oyunu tasarlayalım. Bunun için iki adet dekor çizelim. Dekor sayısı isteğe bağlı olarak arttırılabilir.

Kullanılacak Kuklalar

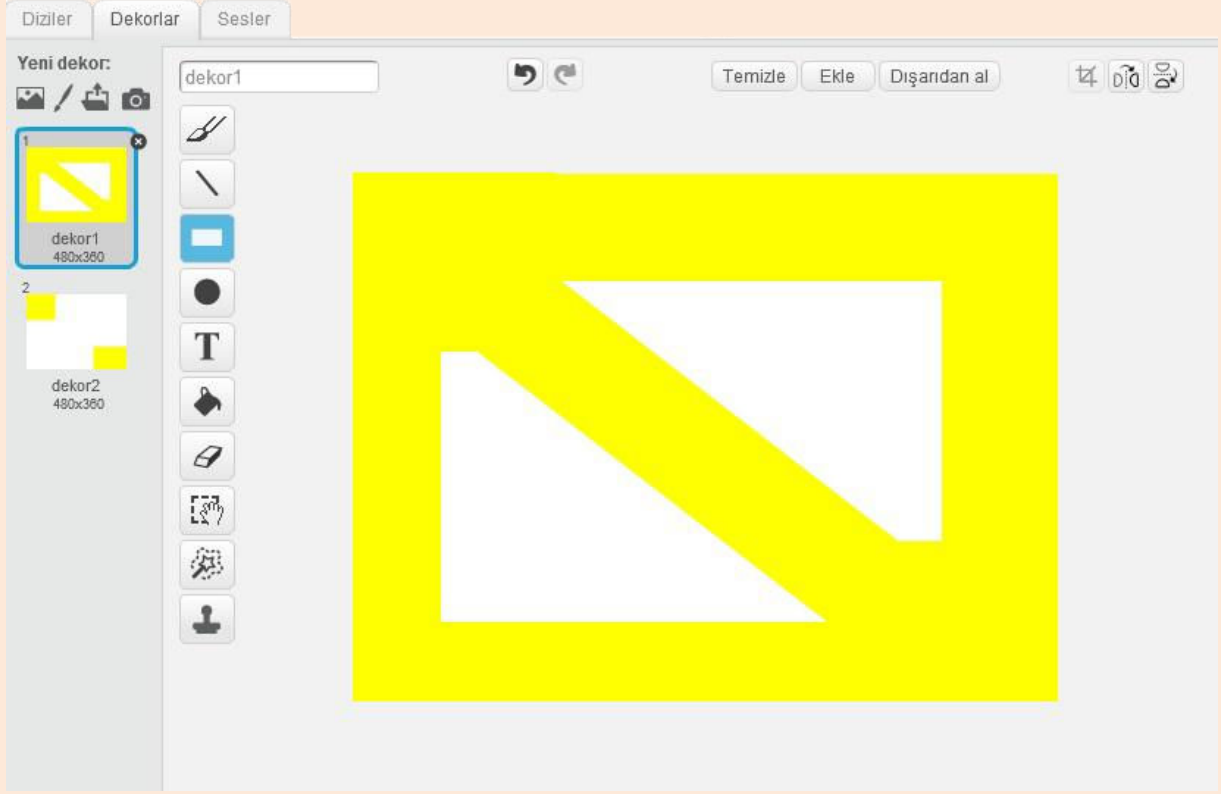


Kuklalar kısmından sahneyi seçip dekorlar kısmına gelelim. Şimdi bölümlerimizi çizeceğiz. Karakterimiz çizdiğimiz yolda ilerleyecek.

- 1 Dikdörtgen çizme aracını seçelim.
- 2 Dikdörtgenlerimizin içini boyalı yapma aracını seçelim.

3 Rengimizi seçelim.

4 Biri sol üstte diğeri sağ altta iki adet kare çizelim. Sol üstteki kare karakterimizin başlangıç noktası, sağ alttaki de bitiş noktası olacak. Bu iki kareyi bir birine bağlayarak bölümlerimizi oluşturacağız.



Karelerimizi çizdikten sonra kaç bölüm oluşturmak istiyorsanız dekora sağ tıklayarak o kadar kopyasını oluşturun.

Aralara istediğiniz gibi dikdörtgenler çizerek bölümlerinizi oluşturun. Bölümleri istediğiniz sayıda ve istediğiniz gibi oluşturabilirsiniz.

Bölümlerimizi oluşturduktan sonra karakterlerimizi ekleyelim ve programlayalım.

Kütüphanemizden yürüyor kısmındaki **Pico walking**, muhtelif kısmındaki **Ball**, fantezi kısmındaki **Dragon** karakterlerini ekleyelim. **Dragon** karakterinden iki tane ekleyelim.

Pico walking karakteri için yazacağımız kodları yazmaya başlayalım.

Pico Walking Karakteri için Yazılacak Kodlar



Karakterimizi labirentte hareket ettirmek için yön tuşlarını kullanacağız. Karakterimiz sarı bölgede hareket edecek ve sarı bölgenin dışına çıkamayacak. Bunun için iç içe iki eğer ise bloğu kullanacağız. Birinci eğer ise bloğunda yukarı ok tuşuna basılıp basılmadığını kontrol edeceğiz ikinci adımda ise yukarı

ok tuşu basılıyken beyaz renge değip değmediğini. Yukarı ok tuşuna bastığımızda karakterimiz Y'yi 10 arttıracak ve yukarı doğru hareket edecek. Yukarı doğru hareket ederken beyaz renge değip değmediğini kontrol edecek. Eğer beyaz renge değiyorsa yaptığı işin tam tersini yapacak ve Y'yi 10 azaltacak. Böylece sarı bölgenin dışına çıkamayacak.

Not: Renk bloğunda beyaz rengi seçebilmek için renk bloğundaki renk kısmına tıklayarak sahne üzerindeki beyaz bir kısma tıklayınız.

Yukarı ok için yazdığımız kodlara benzer kodları aşağı,sağ ve sol ok tuşları için de yazacağız.

Önce karakterimizin X ve Y konumunu ayarlıyoruz. Daha sonra karakterimizin sarı labirentte hareketini kontrol edeceğimiz kodları yazıyoruz.

Yukarı ok tuşundan farklı olarak aşağı ok tuşuna basıldığında karakterin Y'yi azaltırken beyaz renge değdiğinde Y'yi arttırıyoruz. Bu sayede aşağıya ya da yukarıya giderken sarı bölgeden çıkmasını engelliyoruz.



Benzer işlemleri sağ ve sol yön tuşları için tekrarlıyoruz. Sağ ok tuşu basıldığında X'i 10 arttırırken beyaz renge değdiğinde X'i 10 azaltıyoruz.

Sol ok tuşuna basıldığında ise X'i 10 azaltırken beyaz renge değdiğinde X'i 10 arttırıyoruz.

Eğer ise bloklarını bir sürekli tekrarla bloğu içine koyarak kodlarımızın sürekli çalışmasını sağlıyoruz.

Not: Eğer ise bloklarını doğru yazdığınıza, X ve Y bloklarını karıştırmadığınıza ve -10, 10 kısımlarını doğru yazdığınıza emin olunuz.



Hareket ile ilgili bloklarımızı oluşturduktan sonra karakterimizin sahne üzerinde yürüme efekti alabilmesi için gerekli kodları yazalım ve büyüklüğünü ayarlayalım.

Bir eğer ise kod bloğu içine **sonraki kılık** ve **0.1 saniye bekle** kod bloklarını koyalım.

Koşul olarak ise **boşluk tuşuna basılı mı?** kod bloğunda boşluk yazan yeri değiştirerek **herhangi bir tuşa basılı mı?** yapalım.



Pico walking karakterinden başka Ball ve Dragon karakterlerini ekleyeceğiz. Ball karakteri bitiş noktasında olacak. Pico walking

karakterimiz Ball karakterine deđdiđinde yeni blm haberini salacađız. Yeni blm haberini alan karakterler de verdiđimiz grevleri yapacaklar.

Not: **Fare oku'na deđdi mi?** kod blođunda fare okunu deđiřtirip Ball yaptığınızda blok, Ball karakterine deđip deđmediđini kontrol eder.

Dragon karakterleri ise ortalıkta gezecek ve biz onlara deđmemeye alıřacađız eđer Dragon veya Dragonun kopyasını ıkararak yapacađımız Dragon2 karakterine deđersek X=-200 Y=140 noktasına yani bařlangı noktasına geri gideceđiz.

```
 yeni blm haberi gelince
 x: -200 y: 140 noktasına git
```

Yazacađımız son kod blođunda ise yeni blm haberi geldiđinde bařlangı noktasının koordinatlarına git diyeceđiz.

Pico Walking Karakteri iin Yazılacak Tm kodlar

```
 tıktanınca
 x: -200 y: 140 noktasına git
 srekli tekrarla
 eđer yukarı ok tuřu basıldı mı? ise
   y'yi 10 arttır
 eđer rengine deđdi mi? ise
   y'yi -10 arttır
 eđer ařađı ok tuřu basıldı mı? ise
   y'yi -10 arttır
 eđer rengine deđdi mi? ise
   y'yi 10 arttır
 eđer sađ ok tuřu basıldı mı? ise
   x'i 10 arttır
 eđer rengine deđdi mi? ise
   x'i -10 arttır
 eđer sol ok tuřu basıldı mı? ise
   x'i -10 arttır
 eđer rengine deđdi mi? ise
   x'i 10 arttır

 tıktanınca
 byklđ % 30 yap
 srekli tekrarla
 eđer herhangi tuřu basıldı mı? ise
   sonraki kk
   0.1 saniye bekle

 tıktanınca
 srekli tekrarla
 eđer Ball a deđdi mi? ise
   yeni blm haberini sal
 eđer Dragon a deđdi mi? veya Dragon2 a deđdi mi? ise
   x: -200 y: 140 noktasına git

 yeni blm haberi gelince
 x: -200 y: 140 noktasına git
```



Ball Karakteri için Yazılacak Kodlar



Karakterimizin X ve Y konumunu bitiş noktası olarak ayarlıyoruz. Pico karakterimiz bu Ball karakterine ulaşmaya çalışacak. Daha sonra başlangıç kılığını ayarlıyoruz.

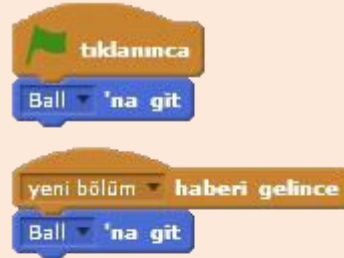


Pico karakterimiz Ball karakterine geldiğinde yeni bölüm haberi salınıyordu. Ball karakterimiz için de yeni bölüm haberi gelince sonraki kılağa geç dedik.

Dragon Karakterleri için Yazılacak Kodlar

Sıra geldi Dragon karakterimizin kodlarını yazmaya. Etrafta rastgele gezecek olan bu ejderhalar için önce Ball karakterine git dedik. Yeni bölüm haberi geldiğinde de Ball karakterinin olduğu konuma gidecek.

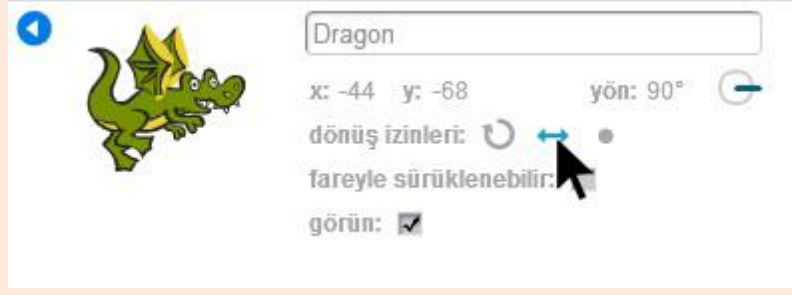
Daha sonra sarı bölgede rastgele hareket etmesi için önce 45 yönüne dön dedik ve büyüklüğünü ayarladık.



Sürekli olarak 3 adım gitmesini ve beyaz renge değip değmemesini kontrol etmesini sağladık. Eğer beyaz bölgeye değerse bu sefer yaptığı işin tam tersini yaparak -3 adım gidecek ve böylece sarı bölgede kalacak.

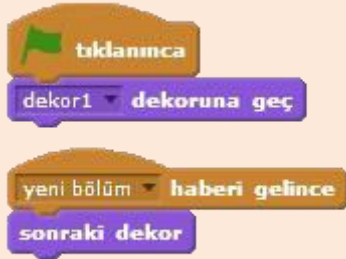
Beyaz renge değdiğinde -180 ile 180 arasında rastgele bir yöne dön diyerek de sarı bölgede rastgele bir yere gitmesini sağladık.

Dragon karakteri bu yazdığımız kodlar sayesinde sürekli olarak 3 adım ilerleyecek. Eğer beyaz renge değeri yani sarı bölgenin dışına çıkarsa -3 adım geri gelecek ve rastgele bir yöne dönerek tekrar ilerlemeye devam edecek.



Dragon karakterlerimizin dönüş iznini sadece sağa sola dönebilsin yapalım.

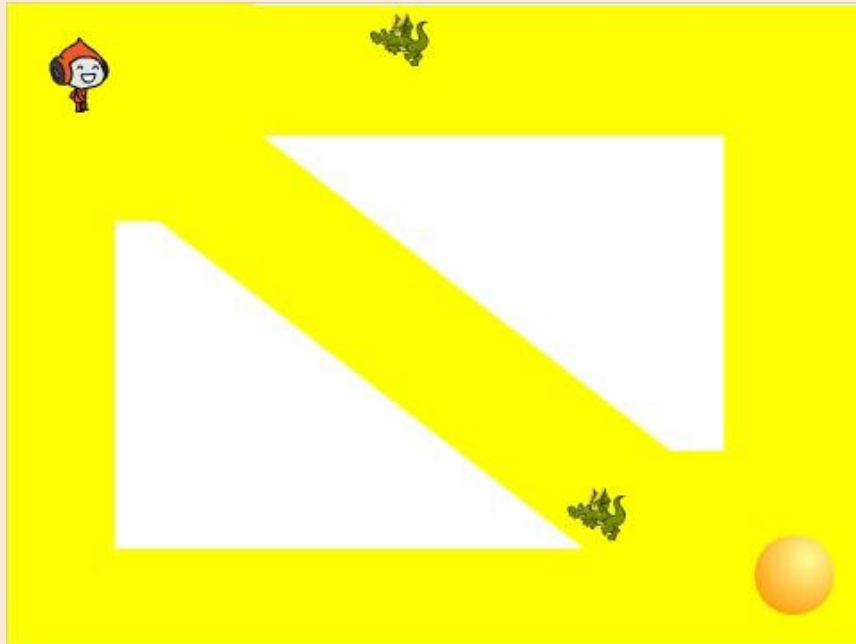
Sahne için Yazılacak Kodlar



Son yazacağımız kodlar sahne için. Kuklalar bölümünden sahneyi seçtikten sonra yandaki kodları sahne için yazalım.

Yeşil bayrak tıklandığında önce ilk dekora geçmesini sağlıyoruz. Böylece her yeşil bayrağa bastığımızda seçtiğimiz dekora geçecektir.

Daha sonra yeni bölüm haberi geldiğinde sonraki dekor diyerek çizdiğimiz diğer dekorlara geçmesini ve bölüm atlamasını sağlıyoruz.



Bölümlü labirent oyunundan bir görünüm

Özel Taşlar

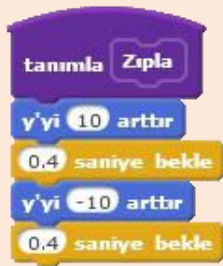
Scratch'da yapacağımız işler için kendi özel bloklarımızı oluşturabiliriz. Oluşturduğumuz blok çalıştığında bizim tanımladığımız işleri yapar. Bu sayede fazla kod yazmaktan kurtuluruz. Büyük ve karışık projelerde özel taşlar çok büyük kolaylıklar sağladığı için sıkça kullanılır.



Özel taşlar menüsünde **Bir Taş Oluştur** butonuna tıklayarak yeni bir blok oluşturabiliriz. Bir taş oluştur butonuna tıkladıktan sonra gelen pencereden özel taşımızın ismini belirledikten sonra tamam tuşuna bastığımızda bloğumuz oluşur.



Bloğumuz oluştuğunda kod yazma alanında da tanımla kısmı oluşur. Tanımla kısmının altına bloğumuz çalıştığında yapılacak görevleri ekleriz.



Oluşturduğumuz **Zıpla** kod bloğu çalıştığında tanımla kısmını altına yazdığımız kod blokları çalışır. Bu sayede aynı kod bloklarını tekrar tekrar yazmak yerine **Zıpla** bloğunu kullanırız ve fazla kod yazma zahmetinden kurtuluruz.

Tanımla kısmının altına basit bir zıplama efekti yapmak için kodlarımızı ekledik. Bu kodlarla karakterimizin Y konumunu önce arttırıyoruz bir süre bekledikten sonra tekrar azaltıyoruz.

Zıpla kod bloğu çalıştığında tanımla kısmının altındaki kod blokları çalıştırılır ve her boşluk tuşuna bastığımızda karakterimiz zıplar.

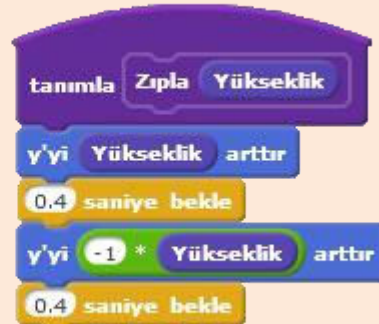
Bloğumuza Sayısal Değer Ekleyelim

Oluşturduğumuz blokları sayısal ya da sözel değerler ekleyip tanımla kısmına bu değerleri gönderip kullanabiliriz. Bloğumuza zıplama yaptırırken Y konumunu ne kadar arttırmak istediğimizi yazabileceğimiz bir kısım oluşturalım.



Yeni taş oluşturup ismini yazalım. Seçeneklere tıkladıktan sonra gelen kısımdan rakamsal girdi ekleye tıklayalım. Bloğumuzun sağ tarafına rakamsal girdi eklenecektir. Number1 yazan yere zıplama yüksekliğini gireceğimiz için **yükseklik** yazalım.

Rakamsal girdi eklediğimiz için tanımla kısmında gireceğimiz rakamsal değeri kullanmak için bir blok oluşur. Bu bloğu sürükleyip bırak yöntemiyle kullanmak istediğimiz blokların içine yerleştirebiliriz.



1 X ve Yazı isminde iki değişken oluşturalım. Tanımla kısmının altına X 1 olsun, Yazı değişkeni ise 0 yazan yeri silerek boş olsun diyelim.

2 Ekranaya yazımızı harf harf yazacağız. Önce 1. harf daha sonra 1. harf ve 2. harfin birleşimi daha sonra 1,2 ve 3. harflerin birleşimi ekranda gözükecek. Yani önce “S” daha sonra “Sc” daha sonra “Scr”, “Scra”, “Scrat”, “Scratac”, “Scratch” diye sırayla ekranda gözükecek. Bunu sağlamak için bir döngü oluşturuyoruz. Bu döngünün ekrana yazıları istediğimiz gibi yazabilmesi için yazının uzunluğu kadar tekrarlanması gerekir. Bu yüzden işlemler menüsündeki **world in uzunluğu** bloğunu kullanacağız. World yazısının yerine tanımla kısmındaki **Yazı** bloğunu yerleştirerek tekrarlama sayısı olarak bu bloğu kullanacağız. Döngümüz özel bloğumuzdaki yazı yazılan alana yazdığımız yazının uzunluğu kadar tekrarlanacak.

3 Bu adımda boş olan yazı değişkenine tek tek yazının harflerini ekleyeceğiz. Yazı değişkeni önce bloğumuza yazdığımız yazının ilk harfi olacak daha sonra ilk ve ikinci harfi ve daha sonra ilk üç harfi olacak. Bu şekilde adım adım harfleri yazı değişkenine aktaracağız. Her adımda da yazı değişkenini ekrana yazdıracağız.



Her adımda bloğumuza yazdığımız yazının x. harfini yazı değişkeni ile birleştireceğiz. X ilk seferde 1 olacağı için ve Yazı değişkeni boş olacağı için Yazı değişkeni ilk adımda “S” olur. Döngü bir kere dönüp tekrar geldiğinde X değişkeni 2 olur bu sefer bloğa yazdığımız yazının 2. harfi yazı değişkenine aktarılır ve yazı değişkeni “Sc” olur bu şekilde adım adım bütün harfleri yazı değişkenine aktarırız.

4 Bloğumuza yazdığımız yazıyı her adımda harf harf yazı değişkenine aktaracağımız için döngünün 1. adımında 1. harfi , 2. adımında 2. harfi ekleyebilmek için işlemler menüsünde 1 yazan yere X değişkeninin değerini tutan bloğu koyuyoruz. Her adımda X değişkenini 1 arttıracığımız için döngü her döndüğünde yazının bir sonraki harfi yazı değişkenine aktarılır.

5 Bloęa yazdıęımız yazıyı harf harf yazı deęiřkenine aktarıırken her adımda da ekrana yazdırıyoruz.

6 Döngünün her adımına 0.1 saniye bekleme koyarak daktilo yazısı animasyonun hızını ayarlıyoruz. Hızı deęiřtirmek için 0.1 yazan yeri deęiřtirebilirsiniz.

3

Scratch Online Editör

Scratch'da yaptığınız projelere internet olan her yerden ulaşabilmeniz ve arkadaşlarınızla paylaşabilmeniz için bir internet sitesi mevcut.

Bu bölümde, Scratch internet sitesini inceleyeceğiz ve Scratch online editörü kullanacağız.

SCRATCH ONLINE EDİTÖR

Scratch çalışmalarınızı internet aracılığıyla yapabilmemiz ve arkadaşlarınızla paylaşabilmemiz için bir internet sitesi mevcut. Yaptığınız çalışmalara internet olan her yerden <https://scratch.mit.edu> adresini kullanarak ulaşabilirsiniz. Kendi projelerini paylaşabileceğiniz gibi diğer insanların yaptıkları projeleri de inceleyebilirsiniz.

Scratch internet sitesinde 150 değişik ülkeden 40 dilde üretilmiş 17 milyonun üzerinde proje bulunmaktadır. Sizde <https://scratch.mit.edu> internet sitesini kullanarak projelerinizi üretebilir ve arkadaşlarınızla paylaşabilirsiniz. Bu şekilde ortak çalışmalar yürüterek çalışmalarınızı daha da ileriye götürebilirsiniz.



Scratch internet sitesinden bir görünüm

- 1 Oluştur dediğimizde yeni çalışmalar üretebileceğiniz online editör açılır. Online editör ile internet olan her yerden çalışmalarınızı oluşturabilir, düzenleyebilir ve paylaşabilirsiniz.
- 2 Keşfet ile yapılmış olan projeleri ve içindeki kodları inceleyebilirsiniz.
- 3 Tartış kısmından aklınıza takılan soruları diğer Scratch kullanıcılarına sorabilirsiniz. Açılmış konulardan gelişmeleri takip edebilirsiniz. Konu açmak ya da cevap yazmak için siteye üye olmalısınız. Türkçe tartışmaları <https://scratch.mit.edu/discuss/25> adresinden takip edebilirsiniz.
- 4 Scratch'a katıl diyerek gerekli bilgileri doldurduktan sonra Scratch internet sitesine üye olabilirsiniz.
- 5 Eğer üye olduysanız giriş yap diyerek hesabınıza giriş yapabilirsiniz.

Scratch İnternet Sitesine Üye Olmak

Scratch internet sitesine üye olmadan da projelerinizi geliştirebilirsiniz ancak; projelerinizi kaydetmek ve paylaşmak için siteye üye olmalısınız.

Scratch'a Katıl dedikten sonra gerekli bilgileri girelim ve internet sitesine üye olalım.

Scratch'a Katıl

Scratch'da bir hesap açmak hem çok kolay hem de bedava.

Bir Scratch Kullanıcı Adı Belirle Gerçek adımı kullanma

Bir Şifre seç

Şifre Sağlaması



1 2 3 4

Bir kullanıcı adı ve şifre seçelim. Kullanıcı adınızda ş,ç,ğ,ı,ö,ü gibi Türkçeye has harfleri kullanmayınız. Bilgileri girdikten sonra sonraki butonuna tıklayınız.

Scratch'a Katıl

Bu sorulara verdiğin cevaplar gizli tutulacak.

Neden bu bilgileri istiyoruz ?

Doğum Ayı ve Yılı - Ay - - Yıl -

Cinsiyet Erkek Kadın

Ülke - Ülke -



1 2 3 4

Doğum ayı ve yılı cinsiyet ve ülke bilgilerini girip sonraki butonuna tıklayalım

Scratch'a Katıl

E-posta adresini gir. Hesabını onaylaman için sana bir e-posta göndereceğiz.

E-posta adresi

E-posta adresini doğrula

Bir e-posta adresi belirtmen gerekiyor



1 2 3 4

Sonraki

E-posta kısmına gireceğin e-posta adresi çok önemli. Bu kısma e-postanızı doğru yazınız çünkü e-postamızı doğrulamamız gerekecek. Üye olduktan sonra e-posta adresinize bir doğrulama e-postası gönderilecektir. E-posta adresinize girip bu doğrulama e-postasını onaylamanız gerekiyor. Eğer e-postanız yoksa ya da e-posta ile ilgili problemlerinizi varsa annenizin, babanız ya da bir yakınınızın e-posta adresini girebilirsiniz.

Scratch'a Katıl


Scratch'e Hoşgeldin!

Kayıt oldun! Artık keşfetmeye ve projeler oluşturmaya başlayabilirsin.

Paylaşmak ve yorum yapmak istiyorsan, **E-posta adresin** 'de sana gönderdiğimiz linke tıkla..

Yanlış e-posta mı? E-posta adresini [Hesap Ayarları](#)ndan değiştirebilirsin.

Having problems? [Please give us feedback](#)



1 2 3 4

Tamam Başlayalım!

Tebrikler üyelik işlemlerini tamamladınız. Tamam başlayalım butonuna tıklayarak projeler üretmeye başlayabilirsiniz.



Hesabınıza giriş yaptıktan sonra kullanıcı adınıza tıkladığınızda açılan menüden **profil** menüsü ile profilinizi düzenleyebilirsiniz. **Kendiminkiler** menüsü ile kendi çalışmalarınızı görüntüleyebilir, **hesap ayarları** menüsü ile de şifre ve e-postanızı değiştirebilirsiniz.

Mesaj simgesi gelen mesajları, klasör simgesi ise sizin çalışmalarınızı görüntülemek içindir.

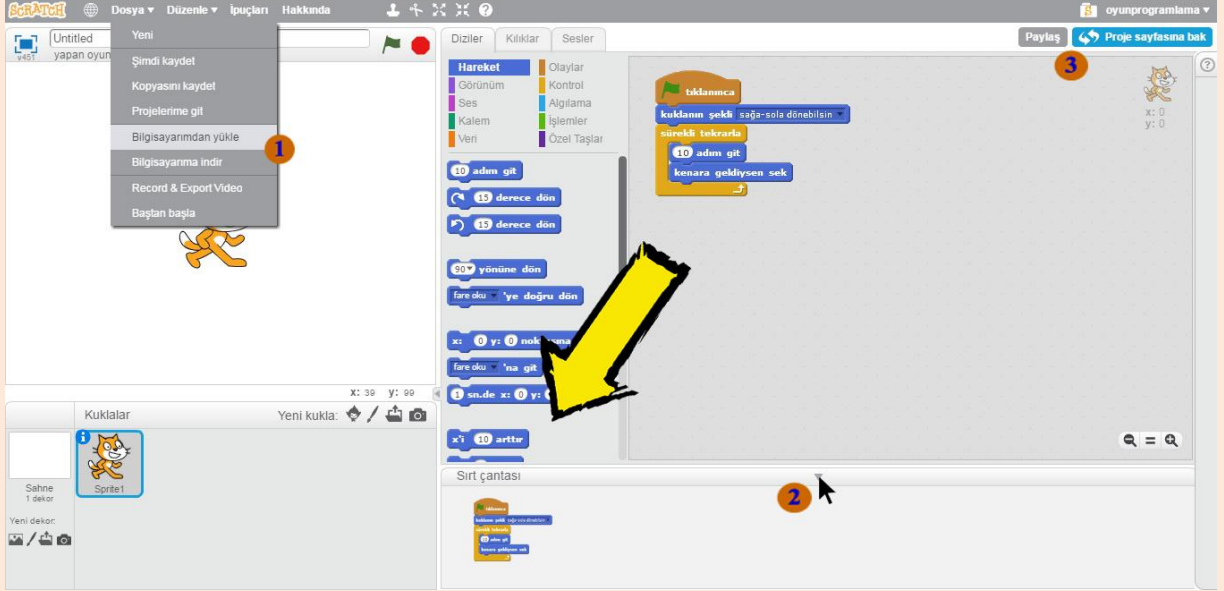


Profilinizi açtığınızda diğer insanların sizin profilinize girdiklerinde ne göreceklerini düzenleyebilirsiniz.

Resim kısmına tıklayarak profil resminizi değiştirebilirsiniz ya da hakkımda kısmını düzenleyerek kendinizi tanıtan bilgiler ekleyebilirsiniz.

Scratch Online Editör

Oluştur menüsüne tıkladığımızda editörümüz açılır. Online editör ile offline editör arasında bazı ek özellikler haricinde hiçbir fark yoktur.



1 En önemli farklılardan biri yaptığımız çalışmaları bilgisayarımıza indirmemiz ve bilgisayarımızda kayıtlı olan dosyaları online editöre yükleyebilmemizdir. **Bilgisayarımdan yükle** menüsü ile bilgisayarımızda kayıtlı olan Scratch çalışmasını online editöre yükleriz. **Bilgisayarıma indir** menüsü ile de online editör ile yaptığımız çalışmayı bilgisayarımıza indiririz.

2 Online editörün en sevdiğim özelliklerinden biri de **sırt çantası** özelliğidir. Başka karakterlerde ya da başka çalışmalarda kullanmak istediğiniz kodları ya da karakterleri sürükleyip bırak yöntemiyle sırt çantası bölümüne sürükleyip bırakırız. Bu sırt çantasına kaydolun kodları ya da karakterleri daha sonra istediğimiz projelerde kullanabiliriz.

3 Online editörün en güçlü özelliklerinden biri de yaptığımız çalışmaları paylaşabilmemizdir. **Paylaş** butonuna basarak projemizi paylaşabiliriz. Burada dikkat etmemiz gereken nokta; paylaşım yapabilmemiz için e-postamızı onaylamamız gerekmektedir.

Projenizi paylaştığınızda diğer insanların sizin projenizi gördüğü bir proje sayfası vardır. Bu proje sayfasında çalışmanız ile ilgili bilgiler bulunur. **Proje sayfasına bak** butonu ile bu proje sayfasına bakabilir ve bilgileri güncelleyebiliriz.

Projeleri Keşfetmek

Keşfet butonuna tıkladığımızda gelen ya da ana sayfada bulunan projelerden birine tıkladığımızda projenin sayfası gelecektir. Buradan yeşil bayrağa basarak projeyi inceleyebilirsiniz ya da açıklamaları okuyarak proje hakkında bilgi edinebilirsiniz.

Fen bilimleri kuvvet
yazan teknogencilik

31 dizi
17 kukla

İçine bak

Yönergeler

Arabaya kuvvet uygulamak için 5N,10N,20N luk kuvvetlerden birini seçin ve uygulamak istediğiniz yönde arabaya değdirin.

Eğer araba dengede değilse, yani bir kuvvet etkisi altındaysa oynat butonuna tıkladığınızda kuvvet etkisi altında hareket edecektir.

Oynat: Kuvvetin etkisini görmek için
Sıfırla: Değerleri sıfırlamak için

Notlar ve Takdirler

Fen dersi projesi

simulations animations fen

© Paylaşma tarihi: 10 Nis 2015 Son değişim: 10 Nis 2015

101 151 Stüdyolar Göm Bunu bildir 580 1

Scratch internet sitesinde yapılan tüm projeler açık kaynaklıdır. Yazılan tüm kodları görebilir ve değiştirebiliriz. **İçine bak** butonuna tıklayarak proje için yazılmış tüm kodları görebiliriz.



İçine bak butonu ile yazılan kodları inceleyebilir ve istediğiniz kodlarda değişiklik yapabilirsiniz. **Katkı yap** butonu ile de projenin bir kopyasını çıkarıp kendi projelerimize ekleyebiliriz. Bu şekilde başka projeleri inceleyebilir ve nasıl yapıldıkları hakkında bilgi edinebilirsiniz.

Yıldıza tıklayarak projeyi gözdelerinize ekleyebilirsiniz. Kalp işaretine tıklayarak da projeyi beğenebilirsiniz.

Scratcher Olmak

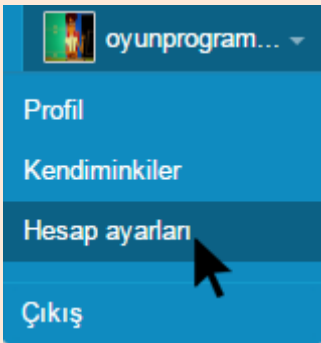
Scratch internet sitesine üye olduğunuzda site size bir rütbe verir. Yeni üyeler new Scratcher (yeni Scratch'çı) olur. Bazı koşulları yerine getirirseniz Scratcher olursunuz. Scratchdaki bazı özellikleri kullanmak için Scratcher olmanız gerekir.



İsminize tıkladığınızda açılan menüden profil kısmına tıklarsanız rütbenizi görebilirsiniz.



Scratcher olmak için öncelikli olarak e-posta hesabınız doğrulamalısınız. Scratch internet sitesine kayıt olurken verdiğiniz e-posta hesabına bir e-posta gider. Hesabınıza girerek o maili onaylamalısınız.



Eğer e-posta gelmemişse ya da problem olmuşsa hesap ayarları menüsünden e-posta kısmına tıklayarak tekrar gönder diyebilirsiniz. E-posta adresine doğrulama için bir e-posta tekrar gönderilir.

Eğer e-postanızı yanlış girdiyse ya da unuttuysanız.

Type new Email:

Type new Email kısmından yeni bir e-posta adresi girin. Eğer e-posta adresi ile bir sıkıntı yaşıyorsanız, annenizin, babanızın ya da bir yakınınızın e-posta adresini yazabilirsiniz.

Not: Eđer e-postanızı onaylamazsanız, paylaşım yapamaz ve yorum yazamazsınız!

E-postanızı onayladıktan sonra bazı şartları yerine getirmelisiniz.

- En az 2 proje paylaşmalısınız.
- Birkaç projeye yorum yazmalısınız.
- Bazı projeleri beğenmeli ve favorilere eklemelisiniz.
- Scratch'a kayıt olalı en az 2 hafta olmalı.

Koşulları yerine getirdikten sonra profilinize girdiğinizde [become a scratcher \(Scrather ol\)](#) yazısı çıkacaktır. Yazıya tıklayın ve yazılanları onaylayın.

Scratch İnternet Sitesini Kullanırken Dikkat Etmeniz Gerekenler

Saygılı Olun

Yorum yaparken karşınızdakilere karşı nazik ve kibar olun. Eleştirileriniz yapıcı olsun. Karşınızdakinin 8 veya 10 yaşında bir öğrenci olabileceğini ve ilk defa programlama öğreniyor olabileceğini unutmayın!

Spam Yapmayın

Gereksiz yorumlar yapmayın ve ağız dalaşlarına girmeyin. Bu gibi durumlarda Scratch ekibi üyeliğinizi iptal edebilir.

Proje Sahibini Belirtin

Bir projeye katkı yapıp remixlediğinizde projenin sahibine teşekkür etmeyi unutmayın!

Kişisel Bilgilerinizi Paylaşmayın

Kendi güvenliğiniz için tanımadığınız insanlarla özel bilgilerinizi paylaşmayınız.

Dürüst Olun

Diğer kullanıcıları taklit etmeyin. Kendi projelerinizi kullanın.

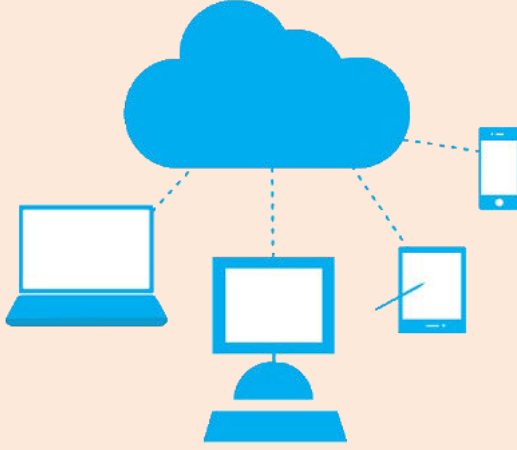
Siber Zorbalık

Siber zorbalık, bir çocuğun veya ergenin başka bir çocuk veya ergen tarafından internet, interaktif, dijital ve mobil teknolojiler kullanılarak tehdit edilmesi, aşağılanması, utandırılması, rahatsız edilmesidir.

Siz karşınızdakilere karşı saygılı olun. Eđer siber zorbalıęa maruz kalırsanız. Bu durumu Scratch ekibine ve օęretmeninize ya da bir bŸyŸnŸze bildirin.

Bulut Deęiřken

Bilgisayarlarda bilgilerimiz normalde harddisklerde saklanır. Bu bilgilere



sadece o bilgisayardan eriřebilirsiniz. Bulut ise bilgilerin internet ortamında yedeklenmesidir. Bilgilerinizi internet ortamına yedeklersiniz ve internet olan her yerden bilgilerinize eriřebilirsiniz.

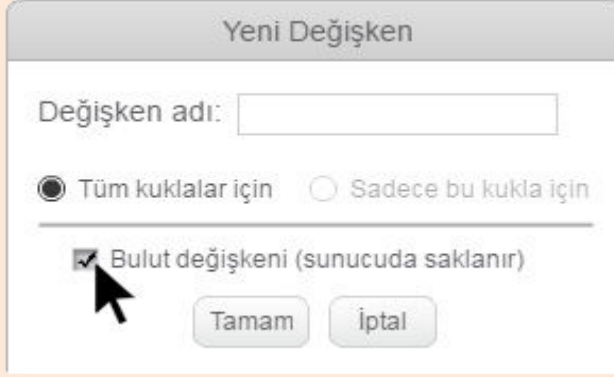
Deęiřkenleri bilgisayar hafızasındaki odalara benzetmiřtik. Deęiřkene aktardığımız deęer bu odaya koyuluyordu. Bu řekilde bilgisayar hafızasında saklamak istediğimiz bilgileri saklıyorduk.

Scratch'da bilgisayar hafızasında tutmak istediğimiz bilgileri internette bir odaya koyan deęiřkenlere ise bulut deęiřken denir. Bulut deęiřkene kaydettiğimiz bilgi internette olduęu iin dięer kullanıcıların bilgilerini de kaydedebiliyoruz. Oyunumuza giren bir kullanıcının kullanıcı adını ve puanını bŸylelikle hafızaya alabiliyoruz.

Not: Bulut deęiřken oluřturabilmek ya da kullanabilmek iin Scratcher olmalısınız.

Bulut Değişken Oluşturmak

Oluştur menüsüne tıkladıktan sonra açılan online editörden **veri** menüsü altında **bir değişken oluştur** butonuna tıklıyoruz.



Gelen menüden değişkenin adını girdikten sonra bulut değişkeni yazan yeri işaretliyoruz ve tamam diyoruz.

Oluşan bulut değişkene aktardığımız bilgiler artık Scratch'ın internet bilgisayarına kaydoluyor ve bu şekilde herkesin bilgilerini bu değişkende saklayabiliyoruz.



Oluşan bulut değişimin normal değişkenden görünüş olarak tek farkı yan tarafında bulut simgesi olmasıdır.

Not: Bulut değişkende şimdilik sadece sayısal değerler saklanabilmekte. Yazıları bulut değişkene kaydedemiyoruz.

Online Anket

Online editörü kullanarak şimdi bir anket oluşturalım. Sorumuz “sebze yemeklerini seviyor musunuz?” olsun. Kullanıcılar sebze yemeklerini seviyorlarsa “evet” sevmiyorlarsa “hayır” yazsınlar. Sebze yemeğini sevenleri ve sevmeyenleri bulut değişkenlere aktaralım. Böylece kaç kişi sebze yemeğini seviyor kaç kişi sevmiyor öğrenelim. Herhangi bir karakter için aşağıdaki kodları yazalım.



Sebze Yemeğini Sevenler ve **Sebze Yemeğini Sevmeyenler** diye iki adet **bulut değişken** oluşturalım.

Yeşil bayrak tıklanınca bloğunun altına **What's your name? Diye sor ve**

bekle bloğunu alalım. Soru kısmına “Sebze yemeğini seviyor musunuz? evet/hayır” yazalım.

Onun altına bir eğer ise değilse bloğu alalım. Koşul olarak girilen yanıtın evet olmasını yazalım. Eğer girilen yanıt evetse **Sebze Yemeğini Sevenler** değişkenini 1 arttıralım. Değilse kısmını boş bırakalım. Daha sonra bir eğer ise değilse bloğu daha alalım. Bu blokta koşul olarak yanıtın hayır olmasını seçelim. Eğer yanıt hayırsa **Sebze Yemeğini Sevmeyenler** değişkenini 1 arttıralım. Değilse kısmına da geçersiz yanıt girdiniz yazdıralım.

Altındaki eğer ise değilse bloğunu sürükleyip bırakarak yukarıdaki eğer ise değilse bloğunun değilse kısmına bırakalım.



Program yeşil bayrağa tıklanınca başlar. Sebze yemeğini seviyor musunuz? diye sorar ve cevabın girilmesini bekler. Cevap girildikten sonra, eğer ise değilse kontrol yapısı çalışır. İlk kısımda girilen yanıtın evet olup olmadığı kontrol edilir. Yanıt evetse **Sebze Yemeğini Sevenler** değişkeni 1 artar ve

program sonlanır. Yanıt evet değilse bu sefer ilk eğer ise değilse bloğunun değilse kısmına geçilir. Burada da diğer eğer ise değilse bloğuyla yanıtın hayır olup olmaması kontrol edilir. Eğer yanıt hayırsa **Sebze Yemeğini Sevmeyenler** değişkeni 1 artar ve program sonlanır. Geriye kalan son ihtimal, yanıtın ne evet ne de hayır olması. Evet ya da hayır dışında yanıt girildiğinde ise program ekrana “Geçersiz yanıt girdiniz.” yazar ve sonlanır.

Programı bitirdikten sonra **paylaş** butonuna tıklayalım. Paylaş butonuna bastığımızda proje paylaşılacak ve projenin sayfası gelecektir.



Proje sayfası görünümü

Proje sayfasında yönergeler kısmından program hakkında bilgileri giriniz. Bu sayfa diğer insanların sizin projenizi inceleyecekleri sayfadır.

Dünya Rekoru Yapma

Offline editörde yaptığımız gibi basit bir tıklama oyunu yapalım. Bu sefer oyunumuza dünya rekoru ekleyelim. Eğer oyunu oynayan kişi rekor puanı geçerse yeni rekor onun puanı olsun.

Puan adında normal bir değişken, **dünya rekoru** adında da bir bulut değişken oluşturalım.

Herhangi bir karakter için aşağıdaki kodları yazalım.



İlk kodda kukla tıkladığında süre 30'un altındaysa puan değişkenini 1 arttırıyoruz. Bu sayede karakter her tıkladığında 1 puan kazanıyoruz.

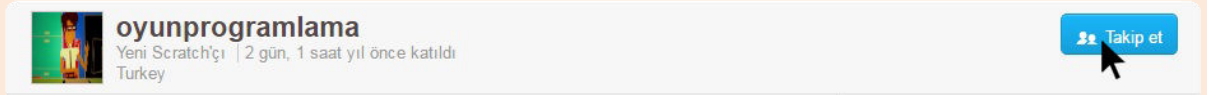
Yeşil bayrak tıkladığında ise öncelikle puan değişkenini sıfırlıyoruz. Daha sonraki adımda süre ölçerin 30'dan büyük olup olmamasını kontrol ediyoruz. Süre ölçer 30'dan büyük olduğunda ilk eğer ise bloğunun içindeki kodlar çalışmaya başlar. Ekranı "Süreniz Bitti!" yazılır ve diğer eğer ise bloğu çalışmaya başlar. Bu eğer ise

kısımında ise puan deęişkenin dünya rekoru deęişkeninden büyük olup olmamasını kontrol ediyoruz. Puan deęişkeni dünya rekorundan büyükse yeni dünya rekoru puan deęişkeninin deęeri olur.

Not: Burada dünya rekoru deęişkenini bulut deęişken olarak oluşturmazsak kullanıcıların puanını saklayamayız. Proje sayfası yeniden yüklendiğinde deęişken sıfırlanır.

Arkadaşlarınızı Takip Etme

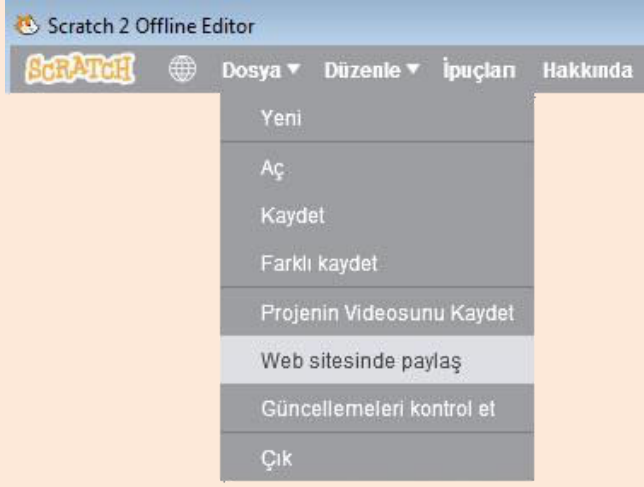
<https://scratch.mit.edu/users/KullanıcıAdı> adresinde kullanıcı adı yazan yeri arkadaşınızın kullanıcı adı ile deęiştirip web tarayıcı programınızın adres çubuğuna yazıp ve enter tuşuna basalım.



Gelen profil penceresinden takip et butonuna tıkladığınızda artık arkadaşınızı takip etmiş olursunuz. Arkadaşınızla ilgili haberler sizin ana sayfanızda neler olup bitiyor kısmında gözükür.

Not: Kitapta kullanılan tüm projelere <https://scratch.mit.edu/users/oyunprogramlama/> adresinden ulaşabilirsiniz. Projelerin içine bakarak kodlarını inceleyebilir ya da projeleri deęiştirip remixleyebilirsiniz.

Offline Editörde Yaptığımız Çalışmayı Online Editöre Aktarma



Offline editörde hazırladığımız çalışmaları Scratch internet sitesindeki hesabımıza yüklemek için offline editörün dosya menüsünden **web sitesinde paylaş** butonuna tıklıyoruz.

Gelen menüden projenin adını, Scratch kullanıcı adınızı ve parolanızı yazdıktan sonra tamam dediğinizde projeniz internet hesabınıza yüklenir.

The image shows a dialog box titled 'Scratch Web Sitesinde Paylaş'. It contains three input fields: 'Proje adı:', 'Scratch'daki adın:', and 'Parola:'. Below the fields are two buttons: 'Tamam' and 'İptal'.

4

Scratch ile Oyun Programlama

Bu bölümde Scratch'da bir oyun nasıl programlanır, örnekleriyle inceleyeceğiz.

SCRATCH ile OYUN PROGRAMLAMA

Şimdi Scratch'da kendi oyunlarımızı yapacağız. Yapacağımız oyunlar çok gelişmiş oyunlar olmayacak ancak algoritma mantığını ilerletmemizin yanında kendi oyunumuzu oynamanın keyfine varacağız.

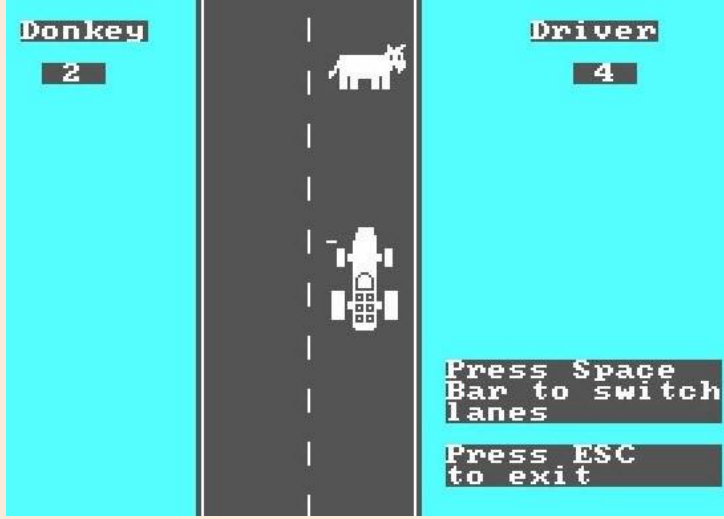
Bir oyun tasarlamadan önce o oyun için kullanılacak karakterleri ve oyunun amacını belirlememiz, kafamızda şablonunu çıkarmamız gerekmektedir. Oyunun amacını ve hangi karakterleri kullanacağımızı kararlaştırdıktan sonra bu amaca ulaşmak için hangi kodları yazmamız gerektiğini belirleriz. Örneğin; oyunda bir karakterden çok sayıda gerekiyorsa ikiz yaratma bloklarını kullanırız. Tek tek karakterlerin rollerini belirledikten sonra kendi aralarında olan ilişkileri de hesaba katarak kodlarımızı yazmaya başlarız.

Oyunu yazmaya ilk başladığımızda kafamızdaki kurguyu oyuna yansıtma noktasında bazı zorluklarla karşılaşabiliriz. Yazdığımız kodlarda bazı hatalar bulunabilir. Bu sizi sakın yıldırmasın! Bu işte ilerlemek istiyorsanız kesinlikle hata yapmalısınız. Hata yapmadan doğru algoritmaları çıkaramazsınız. Bir konuda hata yaptıktan sonra o hatanın kaynağını bulup düzelttikçe daha az hata ile algoritmalar oluşturup kodlarınızı yazabilirsiniz.

Bugün oynadığımız oyunlar, Scratch'da yapacağımız oyunlar ile kıyaslanamayacak kadar gelişmiş düzeyde olabilir ancak unutmamamız gereken nokta; oyunlar her zaman bu kadar gelişmiş değillerdi. İlk bilgisayar oyunları bugünkü oyunlara göre çok basit düzeydeydi. Bilgisayar teknolojisinin bugüne göre çok geride olması ve programlama dillerinin henüz çok gelişmemiş olması sebebiyle o günkü oyunlar bugün için bize çok basit gelebilir ancak o günkü şartlarla değerlendirildiğinde, aslında efsane oyunlar oldukları unutulmamalıdır.

İlk Bilgisayar Oyunu

Bilgisayarlar için ilk oyun 1981 yılında Microsoft'un kurucusu Bill Gates tarafından geliştirilmiştir.



Sizin Scratch ile çok kolay bir şekilde yapabileceğiniz bu oyun, yapıldığı dönemde türünün tek örneği idi. Oyunun amacı kontrol ettiğiniz yarış arabası ile önünüze çıkan eşeklere çarpmamaktı.

Gözünüze çok basit gelmiş olabilir. Şimdi gözünüzü kapatın ve o dönemde yaşadığınızı hayal edin.

Sizce o zaman için bu oyun o kadar basit bir oyun mudur?

Not: Dilerseniz Google Play'da ya da App Store'da oyunu "Donkey Racing" adıyla aratıp akıllı telefonunuza yükleyebilir ve oyununun akıllı telefonlar için yapılmış versiyonunu oynayabilirsiniz.

Elma Yakalama Oyunu

İlk geliştireceğimiz oyun, elmaların yukarıdan aşağıya doğru düştüğü ve bizim onları bir kase yardımıyla yakalamaya çalıştığımız bir oyun. Oyunumuzda kullanacağımız elma karakterinin ikizlerini yaratarak aşağıya doğru düşüyormuş gibi görünmesini sağlayacağız. Eklediğimiz kase karakterini de fare ile kontrol ederek düşen elmaları yakalamaya çalışacağız.

Kedi karakterine sağ tıklayıp silelim ve muhtelif kısımdan Apple ve Bowl karakterlerini ekleyelim.



İlk olarak Bowl karakterimizin kontrolünü sağlayalım. Bowl karakteri oyunda düşen elmaları yakalamak için sağa ve sola doğru hareket edebilmelidir. Bu hareketi yön tuşları ile yapabileceğimiz gibi fare ile de yapabiliriz. Yön tuşlarını kullanarak sağ ok tuşuna basıldığında X konumunu arttırarak, sol ok tuşuna basıldığında ise X konumunu azaltarak bu hareketi sağlayabiliriz. Diğer yöntemde ise farenin konumunu kullanarak fare hareketi ile karakterimizi sağa ve sola doğru hareket ettirebiliriz. Biz oyunda fare ile kontrol seçeneğini kullanacağız.



Algılama menüsü altındaki **farenin x'si** kod bloğunu hareket menüsü altındaki **X olsun** kod bloğunda 0 yazan yere yerleştiriyoruz. Bu sayede farenin X konumuna göre karakterimizin X konumunu ayarlayarak karakterimizi sağa ve sola hareket ettirebiliyoruz. Bu kod bloğunu da bir sürekli bloğunun içine koyarak sürekli olarak

tekrarlanmasını sağlıyoruz.

Not: Karışıklığı önlemek için her kod bloğu resminin sağ tarafında hangi karaktere ait olduğu belirtmek için karakterin küçük bir resmi bulunmaktadır.

Karakterimizin sağa ve sola hareketini ayarladıktan sonra sahne üzerindeki konumunu ayarlayalım.

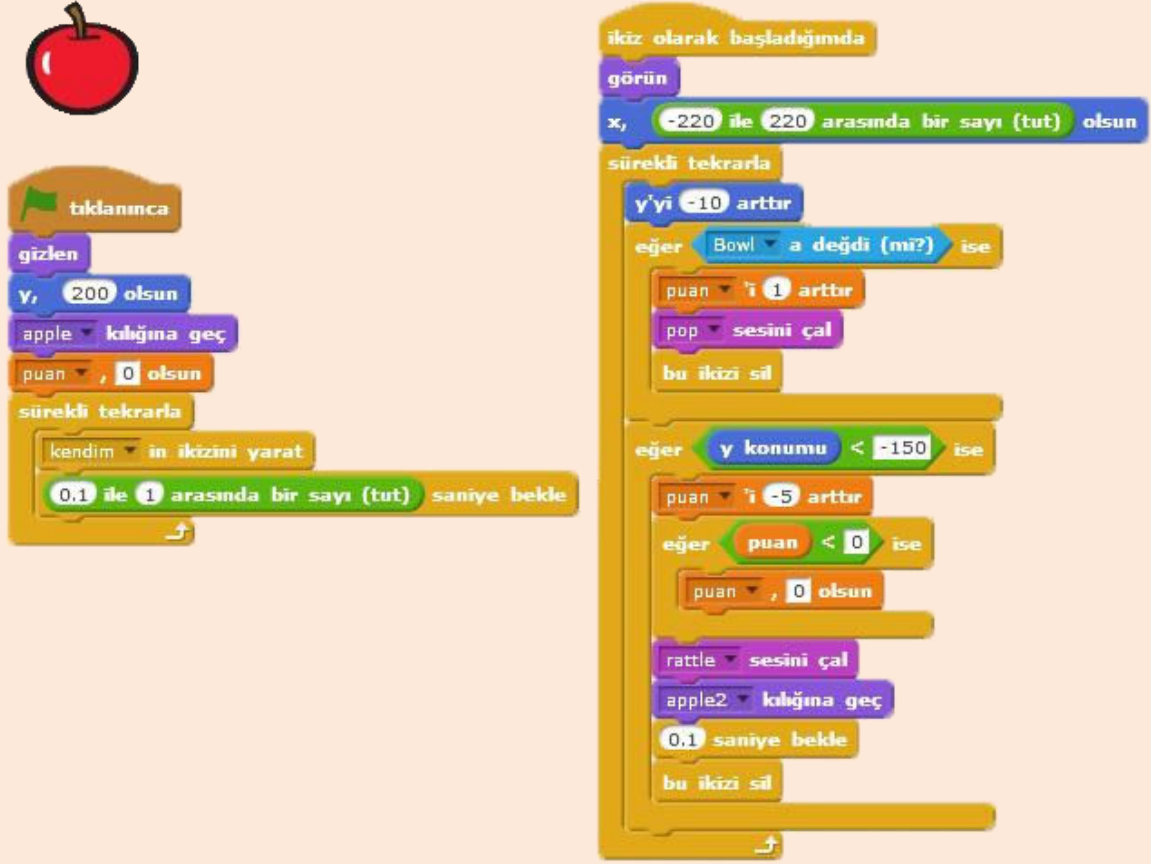


X konumunu fare ile ayarlayacağımız için sadece Y konumunu ayarlıyoruz. Bowl karakterimizin sahnenin en alt kısmında olması için Y konumunu -150 yapıyoruz.

Not: Programda yazdığınız kodların tam olarak çalışıp çalışmadığını yeşil bayrağa basarak kontrol edebilirsiniz. Testiniz bittikten sonra durdur butonuna (kırmızı yuvarlak buton) basarak kodlarınızı yazmaya devam edebilirsiniz.

Bowl karakterimiz için yazacağımız kodlarımızı bitirdik. Şimdi Apple karakterimiz için yazacağımız kodlara geçelim. Apple karakterimizden birden fazla gerektiği için karakterimizin ikizini yaratacağız. Sahne üzerinde bu ikizler aşağıya doğru düşerken Bowl karakteri ile onları yakalamaya çalışacağız.

Apple Karakteri için Yazılacak Tüm Kodlar



Önemli Not: Apple karakteri için yazılacak tüm kodlar yukarıdaki resimdedir. Kodları yukarıdaki resme göre yazınız. Aşağıdaki kodları tekrar yazmayınız. Aşağıda bu kodların yazım aşaması anlatılmaktadır.



Apple karakterimizin ikizlerini yaratıp onları kullanacağımız için esas karakterimizin gözükmemesine gerek yok o yüzden karakteri gizliyoruz. Daha sonra yukarıdan aşağıya düşeceği için Y konumunu 200 olarak ayarlıyoruz.



Bir sonraki adımda ikizlerimizi oluşturmaya başlıyoruz.

Kendimin ikizini yarat bloğuyla sürekli olarak ikiz yaratıyoruz. İkiz yaratma işleminin çok hızlı olamaması

için de araya bekleme koyuyoruz. Apple karakterinin hep aynı sıklıkta gelmesini engellemek için de **0.1 ile 1 arasında sayı tut** diyerek değişik saniyelerde beklemesini sağlıyoruz. Program her adımda 0.1 ile 1 arasında bir değer üreterek o kadar saniye bekledikten sonra ikiz yaratmaya devam ediyor.

Önemli Not: Eğer yazdığınız kodlar düzgün çalışmıyorsa; kodları yanlış yazmadığınıza ve kod bloklarını yanlış yerlere yerleştirmedinize emin olun.



Karakterimizin ikizlerini yarattıktan sonra kullanacağımız blok **ikiz olarak başladığımda** bloğu. Daha önce asıl karakterimizi sahne üzerinde gizlemiştik şimdi ise ikiz olarak başladığında görün diyerek ikizin görünmesini sağlıyoruz. Yapacağımız bir diğer iş de sahne üzerindeki X konumunu ayarlamak. Her zaman aynı X değerini almaması için -220 ile 220 arasında bir sayı üreterek X konumunun rastgele belirlenmesini sağlıyoruz.



İkizlerimizin aşağıya doğru düşmesini sağlamak için sürekli bloğu içinde Y konumunu -10 arttırıyoruz. Bu şekilde ikimiz sürekli olarak aşağıya doğru ilerliyor. Aşağıya doğru ilerlerken iki ihtimalimiz var. Birincisi kase ile elmayı yakalamamız. İkincisi ise

elmanın yakalanmadan düşmeye devam etmesi. Bu iki durumda da ikizimizin görevi bittiği için ikizimizi siliyoruz.

Algılama menüsündeki **fare okuna değdi mi?** kod bloğunda fare yerine Bowl karakterini seçiyoruz. Bowl karakterine değdiğinde ikimiz siliniyor.

Bir diğer ihtimalde de elmayı yakalayamadığımızda elma düşmeye devam ediyor. Elma karakteri belli bir süre sora düşerek sahne dışına çıkacaktır. Sahne dışına çıkmadan Y konumu -150'den de daha aşağıdaysa ikizi siliyoruz.



Sıra oyunumuza puan eklemeye geldi. Apple karakteri Bowl karakterine değdiğinde elmayı yakalamışız demektir. Elmayı yakaladığımızda da puan kazanmak için oluşturduğumuz puan değişkeni için eklenen puan'ı 1 arttır kod bloğunu eğer ise bloğunun içine koyuyoruz. Apple karakteri Bowl karakterine değdiğinde ikiz silinmeden önce puanımız 1 artıyor.



Elma karakterini yakalayamadığımızda elma karakteri düşmeye devam ediyordu ancak belli bir konumdan sonra elma karakterini siliyorduk. Burada da elma karakterini yakalayamadığımızda yani elma karakteri düşmeye devam ettiğinde ceza olarak puanımızı -5 arttırıyoruz yani 5 azaltıyoruz.



```
if (y konumu < -150) ise
  puan 'i -5 arttır
  eğer puan < 0 ise
    puan , 0 olsun
  bu ikizi sil
```



Elmayı yakalayamadığımızda ceza olarak -5 puan aldığımızda eğer hiç puanımız yoksa puanımız eksi değerlere düşecektir. Bu durumu önlemek için de eğer puan değişkeninin değeri 0'dan küçükse puan 0 olsun diyerek puan değişkenimizin eksi değerlere düşmesini engelliyoruz.



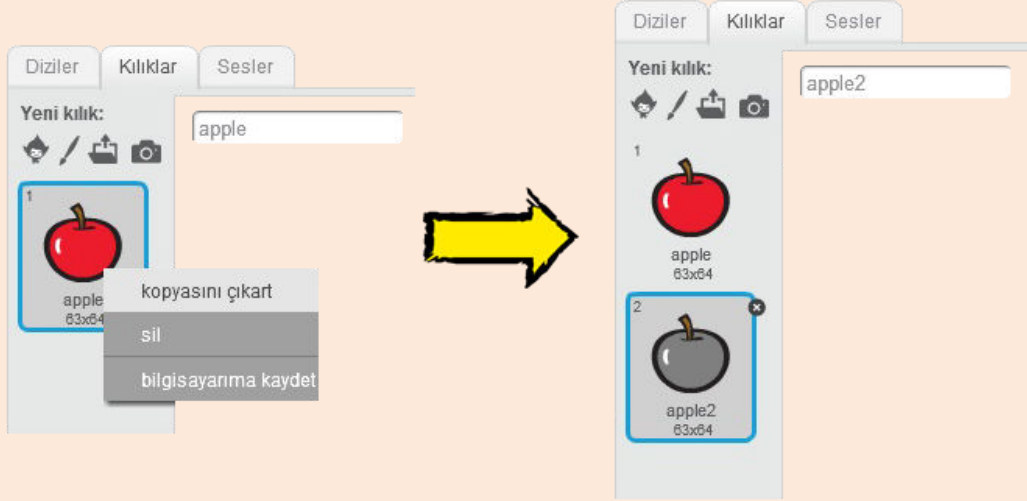
```
if (Bowl a değdi (mi?)) ise
  puan 'i 1 arttır
  pop sesini çal
  bu ikizi sil
```



```
if (y konumu < -150) ise
  puan 'i -5 arttır
  eğer puan < 0 ise
    puan , 0 olsun
  rattle sesini çal
  bu ikizi sil
```



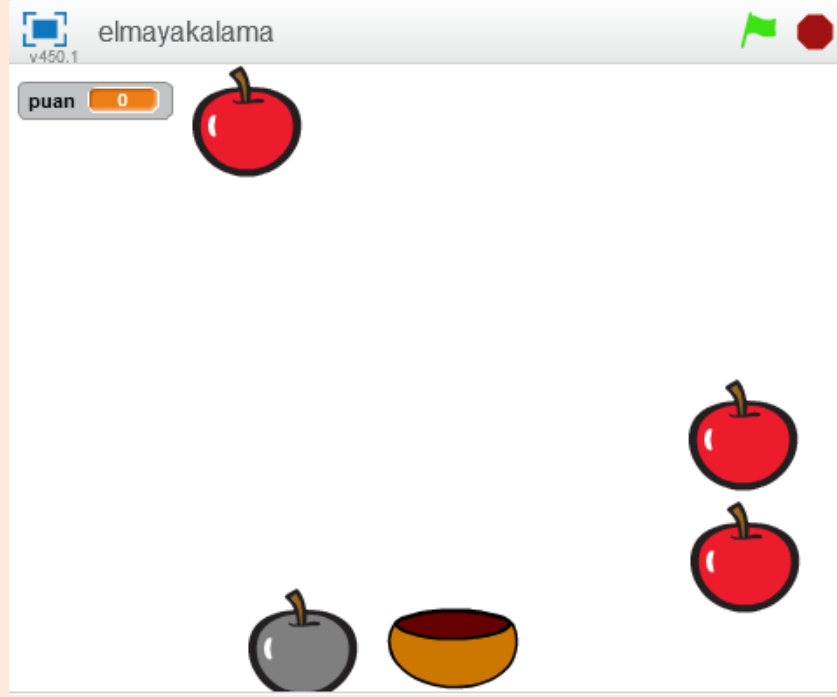
Puan ayarlarını yaptıktan sonra sıra projemize sesli bildirimler eklemeye geldi. Sesler kısmından sesi kütüphaneden ekle butonuna tıklayarak etkiler bölümünden pop ve rattle seslerini ekleyelim. Daha sonra ses bloklarından sesini çal bloğunu eğer ise yapılarına resimdeki gibi ekleyelim. Elmayı yakaladığımızda pop sesi, kaçırdığımızda ise rattle sesi çalsın.



Elmayı yakalayamadığımızda elma karakterinin sanki yere düştüğünde çürüyormuş gibi gözükmesini sağlamak için kılıklar kısmından elma karakterinin kılığının üzerinde sağ tıklayarak bir kopyasını çıkaralım. Resim düzenleme editöründe renklerden gri rengi seçtikten sonra kova simgesini seçerek Apple2 kılığını gri renge boyayalım.



Elmayı yakalayamadığımızda elmaya çürüme efekti verebilmek için oluşturduğumuz Apple2 kılığına geç diyoruz. Apple2 kılığına geçmek için görünüm menüsü altında bulunan **kılığına geç** bloğunu alarak Apple yazan yeri değiştirerek Apple2 olarak ayarlıyoruz. İkizi silmeden önce de 0.1 saniye bekleme koyarak programın hızını ayarlıyoruz. Araya bekleme koymadığımız takdirde Apple2 kılığını göremeden ikiz silinir.



Elma yakalama oyunu ekran görüntüsü

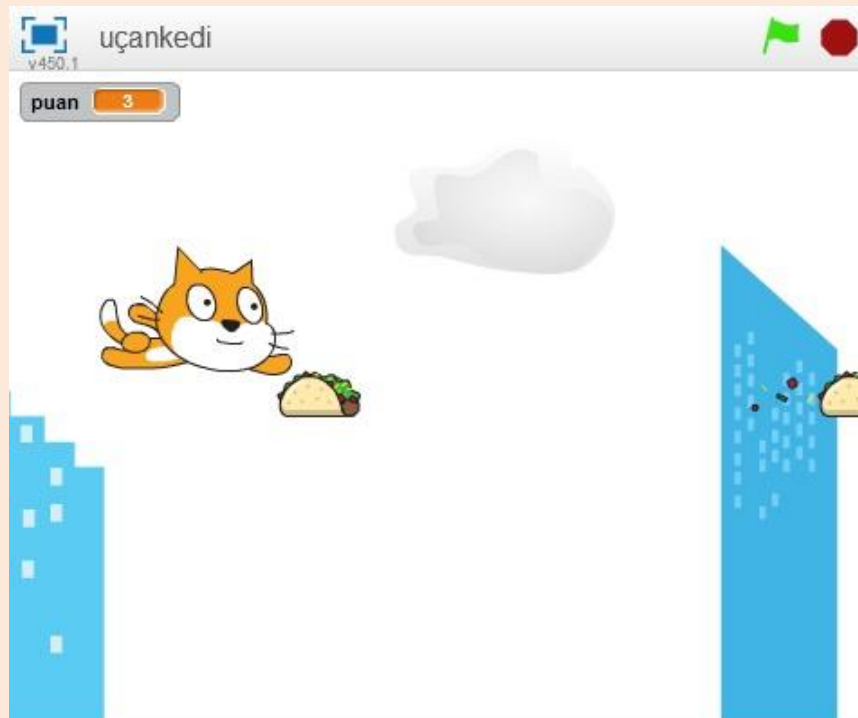
UÇAN KEDİ

Sıradaki oyunumuzda bir uçan kedi tasarlayacağız. Kedi karakterimize uçma efekti vereceğiz ve gelen Taco karakterlerinden kaçınmasını sağlayacağız.



Sahne uçuş efekti verirken aslında kedi karakterimiz hareket etmeyecek. Sahne üzerinde, eklediğimiz bina ve bulutları hareket ettirerek kedi karakterini sanki uçuyormuş gibi göstereceğiz.

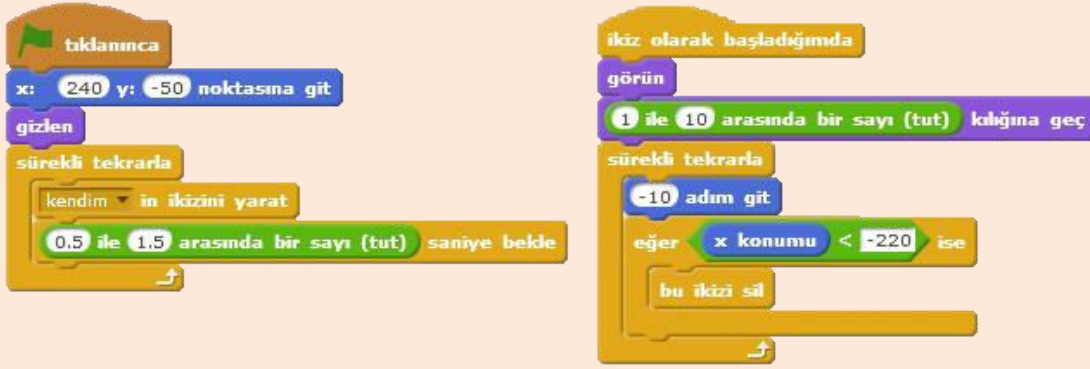
Kedi karakterimiz üzerine doğru gelen Taco karakterlerinden kaçmaya çalışacak, eğer kedi karakterlerimiz Taco karakterine değerse oyun bitecek. Taco karakterinden her kaçındığında ise 1 puan kazanacağız.



Uçma efekti verebilmek için eklediğimiz bina ve bulut karakterlerinin ikizlerini yaratarak sahne üzerinde -10 adım ilerleterek kedi karakterimize doğru ilerlemesini sağlayacağız. Bu şekilde bizim gözümüz kedi karakterini sanki ileriye doğru gidiyormuş gibi göreceğiz.

Bunun için öncelikli olarak kukla ekleme bölümü **Yukarıdan** kısmından **Buildings** karakterini ekleyelim.

Buildings Karakteri için Yazılacak Kodlar



Yeşil bayrak tıklandığında Buildings karakterimiz sahnenin sağından soluna doğru ilerleyecek. Bunu sağlama için öncelikli olarak sahne üzerindeki X ve Y konumunu ayarlıyoruz. Sağdan başladığı için X konumunu 240 olarak ayarlıyoruz. Daha sonra esas karakteri değil de ikizleri kullanacağımız için gizlen diyoruz.

Bir sürekli bloğunun içinde de ikizini yaratıyoruz. Belirlediğimiz sayılar arasında bekleme koyarak sahne üzerinde ikizin yaratılma sıklığını ayarlıyoruz.

İkiz olarak başladığında ise görünmesini sağlıyoruz. Buildings karakterimizin 10 adet kılığı var. Hep aynı bina gelmemesi için 1 ile 10 arasında bir kılığa geç diyerek rastgele bir binanın gelmesini sağlıyoruz.



Kılığa geç bloğunda bulding-a yazan yere 1 ile 10 arasında bir sayı tut bloğunu sürükleyip bırakıyoruz.

Sürekli tekrarlar bloğu ile de sürekli -10 adım git diyerek sola doğru ilerlemesini sağlıyoruz. Sola doğru ilerletirken de eğer X konumu sahne dışına çıkmaya yakınsa yani -220'den küçükse bu ikizi siliyoruz.

Binalar hazır, şimdi bulut karakterlerimizi programlayalım.

Kukla ekleme bölümü **Yukarıdan** kısmından bu sefer **Cloud** karakterini ekleyelim.

Cloud Karakteri için Yazılacak Kodlar



Buldings karakteri için yazdığımız kodlara benzer kodları Cloud karakteri için de yazıyoruz. Yeşil bayrak tıklandığında öncelikli olarak karakterimizin gizlenmesini sağlıyoruz ve X,Y koordinatlarını ayarlıyoruz. Daha sonra karakterlerimizin ikizlerini yaratıyoruz. Burada bekleme koyarken saniye aralığını 1 ile 2.5 belirliyoruz çünkü; bulut karakterimiz biraz daha az aralıklarla ekranda gözüksün.

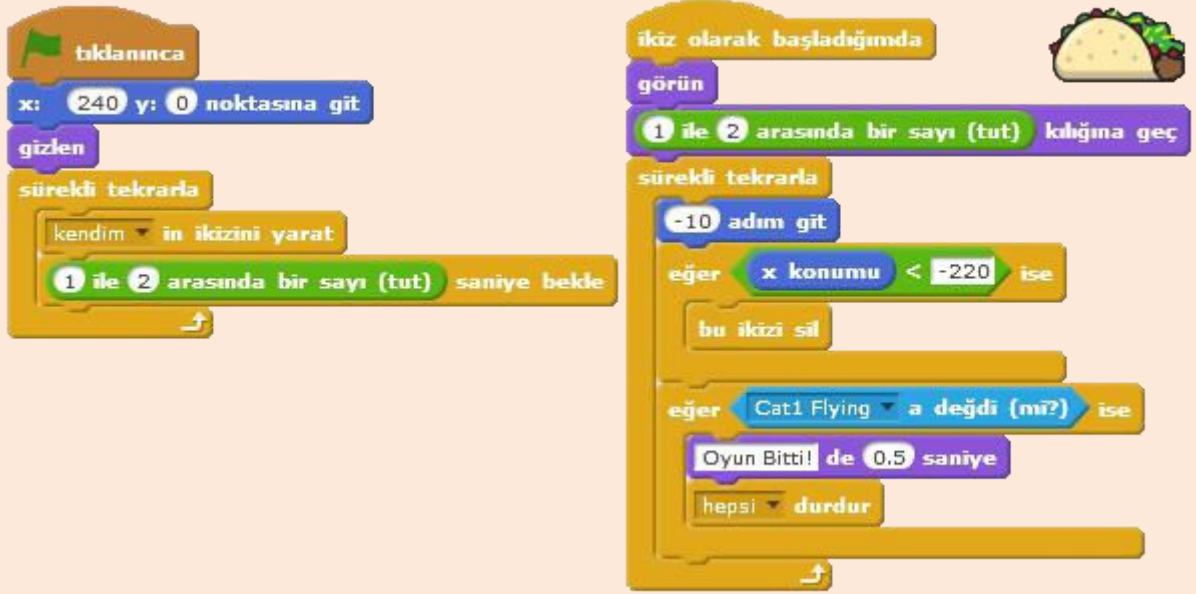
İkiz olarak başladığında ise ikizimizin görünmesini sağlıyoruz. Her zaman aynı yükseklikte olmaması için de Y konumunu -30 ile 30 arasında bir sayı arttırıyoruz. Daha sonra sürekli bloğunun içerisinde **-10 adım git** diyerek geriye doğru hareketini sağlıyoruz. İkiz olan bulutumuzun ekranda giderek silikleşmesi içinde **hayalet etkisini 2 arttır** diyoruz. Buldings karakterimiz için yazdığımız kodlardaki gibi belli bir konumdan sonra ikizimizi siliyoruz. Bir eğer ise bloğunun içinde ikizimizin X konumunu kontrol ediyoruz. Eğer X konumu -220'nin altındaysa ikizi siliyoruz.

Buldings ve Cloud karakterlerimizin ikizleri sürekli olarak ekranın sağından soluna doğru ilerler. Bu ilerleme, bizde sanki kedi karakterimiz ileriye doğru gidiyormuş gibi bir izlenim uyandırır. Aslında kedi karakterimiz olduğu yerde sabit durmaktadır.

Buldings ve Cloud karakterlerini ekledikten ve kodlarını yazdıktan sonra sıra engel olarak ekleyeceğimiz Taco karakterlerinde.

Kukla ekleme bölümü **Yukarıdan** kısmından **Taco** karakterimizi ekleyelim.

Taco Karakteri için Yazılacak Kodlar



Taco karakterimizi engel olarak ekleyeceğiz. Buildings ve Cloud karakterimiz gibi Taco karakteri de ekrandan sağdan sola doğru ilerleyecek ancak kedi karakterimiz Taco karakterinden kaçmaya çalışacak. Her kaçındığında da kedi karakterimiz ile 1 puan kazanacağız. Eğer Taco karakterine değerseniz de oyun bitecek.

Yeşil bayrak tıkladığında Taco karakterimizin X ve Y konumunu ayarlıyoruz ve esas karakterimizi sahne üzerinde gizliyoruz. Daha sonra sürekli olarak karakterimizin ikizini yaratıyoruz. Bekleme süresi olarak da 1 ile 2 arasında bir sayı tut diyerek, 1 ya da 2 saniye bekletiyoruz. Bu şekilde ikizimiz ya 1 saniye arayla ya da 2 saniye arayla oluşacak.

İkiz olarak başladığımızda ise önce görünmesini sağlıyoruz. Taco karakterimizin 2 adet kılığı var. Bu kılıklardan rastgele biri gelmesi için 1 ile 2 arasında bir sayı tut kılığına geç diyoruz. Üretilen sayıya göre karakterimiz ya kılık no 1 olan kılığa ya da kılık no 2 olan kılığa geçiyor. Sürekli bloğuyla da -10 adım git diyerek sürekli olarak diğer karakterler gibi karakterimizin sağ taraftan sol tarafa doğru ilerlemesini sağlıyoruz. X konumu -220'nin altına düşmüşse ikizimizi siliyoruz. Diğer karakterlere yazdığımız kodlardan farklı olarak da birazdan ekleyeceğimiz **Cat1 Flying**

karakterine deęip deęmemesini kontrol ediyoruz. Eęer Cat1 Flying karakterine deęerse ekrana Oyun Bitti! Yazdırıyoruz ve bütn blokları durduruyoruz.

Şimdi kukla ekleme bölümünü **Yukarıdan** kısmından **Cat1 Flying** karakterini ekleyelim.

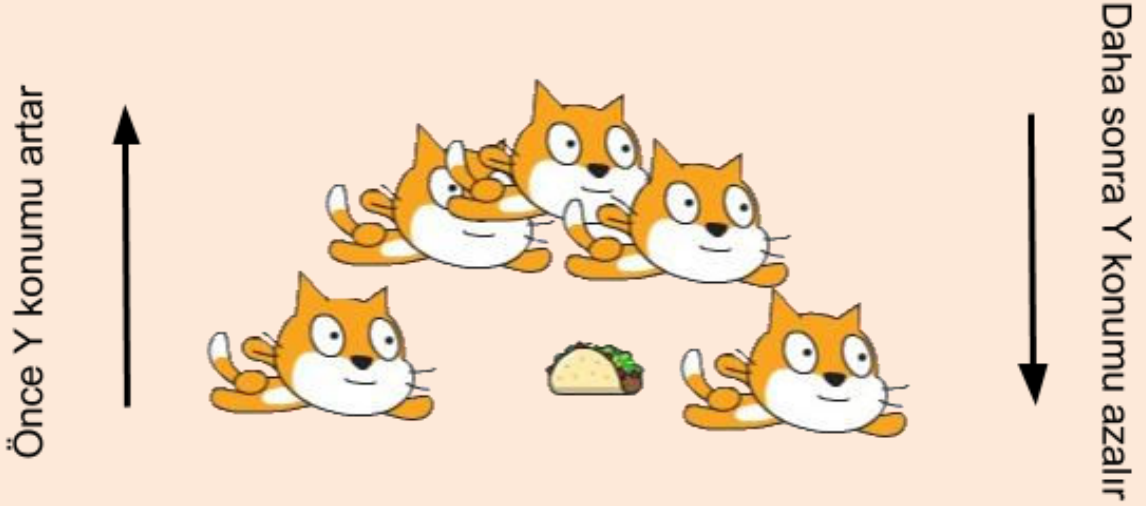
Cat1 Flying Karakteri için Yazılacak Kodlar



Kedi karakterimizin üzerine gelen Taco karakterlerinden kaçmak için kedi karakterinin Y konumunu kademeli olarak arttıracamız ve daha sonra kademeli olarak azaltacağız.

Elinize bir top alın yukarıya doğru fırlatın. Top önce hızlı bir şekilde yukarıya doğru çıkmaya başlar ama hızı giderek yavaşlar ve en sonunda durur. Bundan sonraysa aşağıya doğru önce yavaş ilerlerken daha sonra hızı artarak ilerler. Biz de karakterimizde buna benzer bir şekilde Y konumunu arttırıp, azaltacağız.

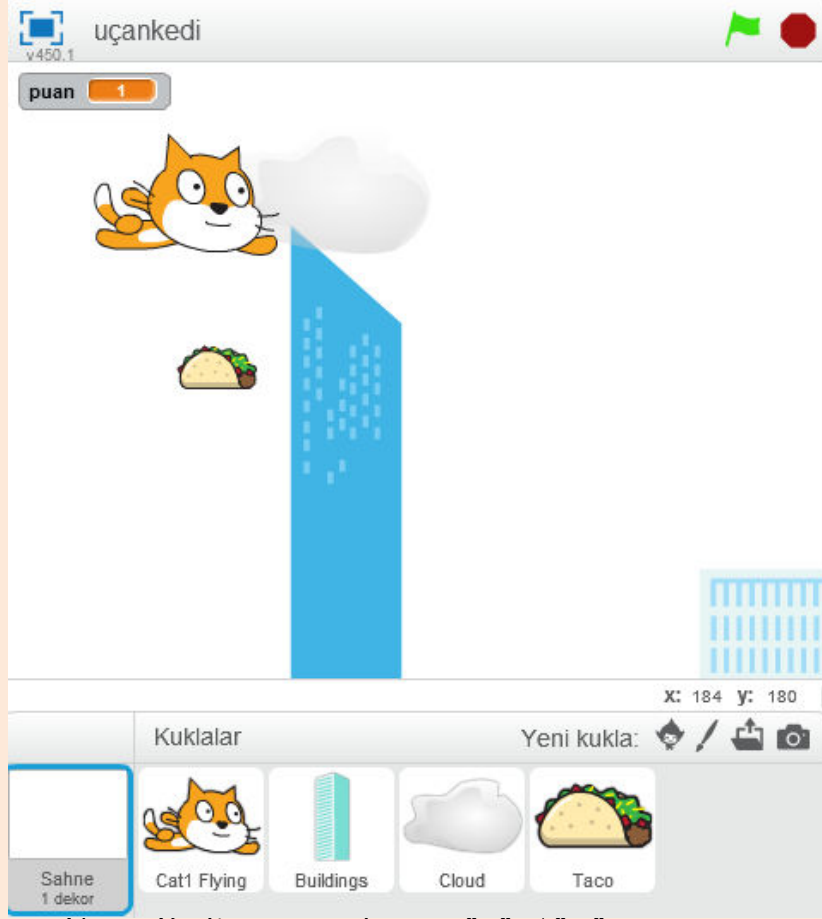
İlk Yeşil bayrak bloğumuzun altına yazdığımız kodlarla karakterimize, katman olarak üste çıkması için üste çık diyoruz. Daha sonra oluşturduğumuz zıpla ve puan değişkenlerinin başlangıçta sıfır olmasını sağlıyoruz, X ve Y konumunu ayarlıyoruz. Sürekli bloğunda ise karakterimizin Y konumunu zıpla değişkeninin değeri kadar değiştiriyoruz.



İlk başta zıpla değişkeni 0 olduğu için Y konumu sabit kalır.

İkinci yeşil bayrak tıklanınca bloğuyla da boşluk tuşuna basılıp basılmadığını kontrol ederiz. Boşluk tuşuna basıldığında water drop sesini çaldırdıktan sonra zıpla değişkeninin değerini 14 yaparız. Sürekli Y'yi zıpla arttır dediğimiz için karakterimizin Y konumu önce 14 artar. Daha sonra 28 defa tekrarlar bloğuyla zıpla değişkenini -1 arttırırız yani azaltırız. Bir sonraki adımda zıpla değişkeninin değeri 1 azalır ve 13 olur bu sefer karakterimizin Y konumu 13 artar. Bir sonraki adımda zıpla değişkeninin değeri yine 1 azalacağı için karakterimizin Y konumu 12 artar. Bu şekilde karakterimizin Y konumu sırayla 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 artar zıpla değişkeni 0 olduğunda ise karakterimizin Y konumunun artması durur. Bu adımdan sonra zıpla değişkeni eksi değerler almaya başlar. Önce karakterimizin Y konumu -1 artar yani 1 azalır ve karakterimiz düşmeye başlar. Yükseldiği gibi düşmeye başlayan karakterin Y konumu sırayla -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11, -12, -13, -14 artar yani azalır. Bu şekilde kademeli olarak yükseldiği gibi yine kademeli olarak düşer.

Tekrarlar bloğunun görevi bittikten sonra zıpla değişkeninin değerini 0 yaparız. Y konumunun değerinin değişmesi durur. Puan değişkenini de 1 arttırarak puan kazanırız ve Y konumunu her ihtimale karşı eski konum değeri olan 0 yaparız. Karakter bu şekilde Taco karakterine değmeden ilerlemesini sürdürebilir.

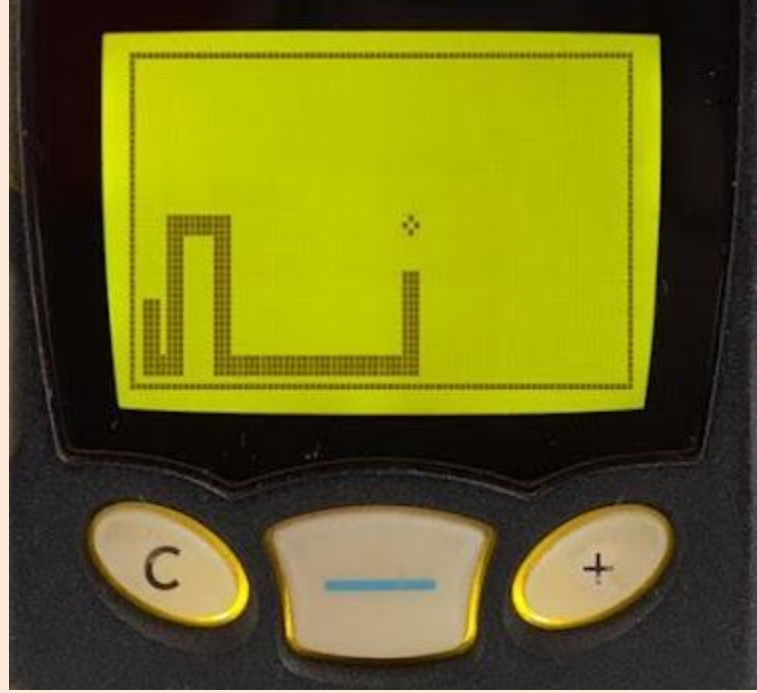


Uçan Kedi oyunu ekran görüntüsü

Uçan Kedi oyunu ekran görüntüsü

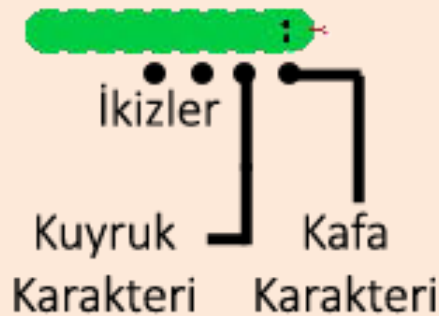
YILAN OYUNU

Dünyanın ilk mobil oyunu, yılan oyunudur. Cep telefonlarının ilk çıktığı yıllarda teknoloji henüz bu kadar ilerlememişti ve telefonlar akıllı telefon olarak anılmıyordu. Siyah beyaz ekrana sahip olan cep telefonlarında eğlenmek için bazı oyunlar bulunurdu ve bu oyunların ilki yılan oyunuydu.

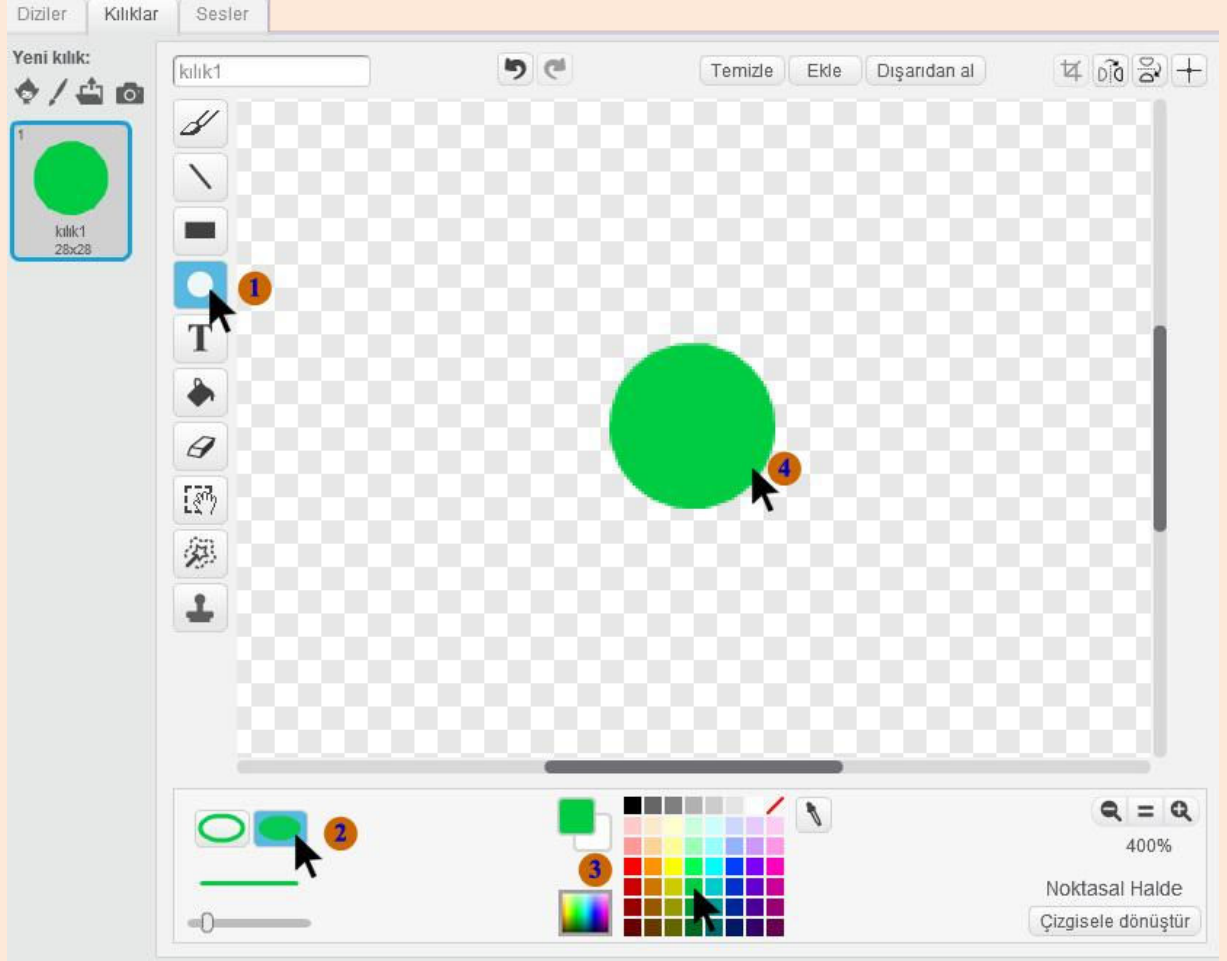


Eski telefonlardan bir görünüm

Biz de Scratch'da bu nostaljik oyunun bir benzerini yapacağız. Bir yılan oluşturacağız ve yılan ekranda rastgele konumlarda beliren fareyi yemeye çalışacak. Fareyi yedikçe yılanımız büyüyecek. Bu oyunda ikiz yaratma bloklarını kullanacağız. Bir karakter çizerek yılanın kafasını oluşturacağız, bir de bu karakterin kopyasını çıkarıp kuyruk kısmını oluşturacağız. Kuyruk kısmının ikizini yaratarak kafa karakterinin uzantısı gibi göstereceğiz.

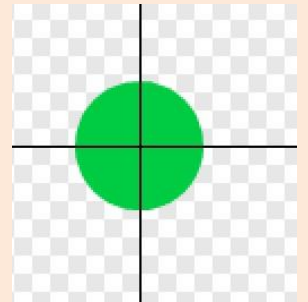


Oyunda kullanacağımız kafa karakteri için bir karakter çizelim. **Yeni Kuklayı Çiz** butonuna basalım.



Yakınlaştırmayı %400 yapalım ve resimdeki büyüklük kadar bir daire çizelim. Bu daire yılanımızın kafası olacak.

Dairemizin kılık merkezini tam ortası olarak ayarlayalım.

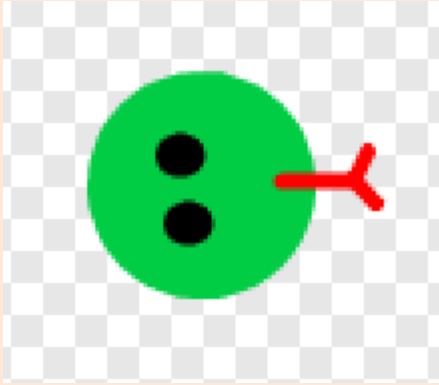


- info
- kopyasını çıkart
- sil
- bilgisayarıma kaydet
- görün

Bir daire kafa karakteri, bir daire de kuyruk karakteri olacağı için çizdiğimiz karaktere sağ tıklayıp kopyasını çıkart diyoruz.

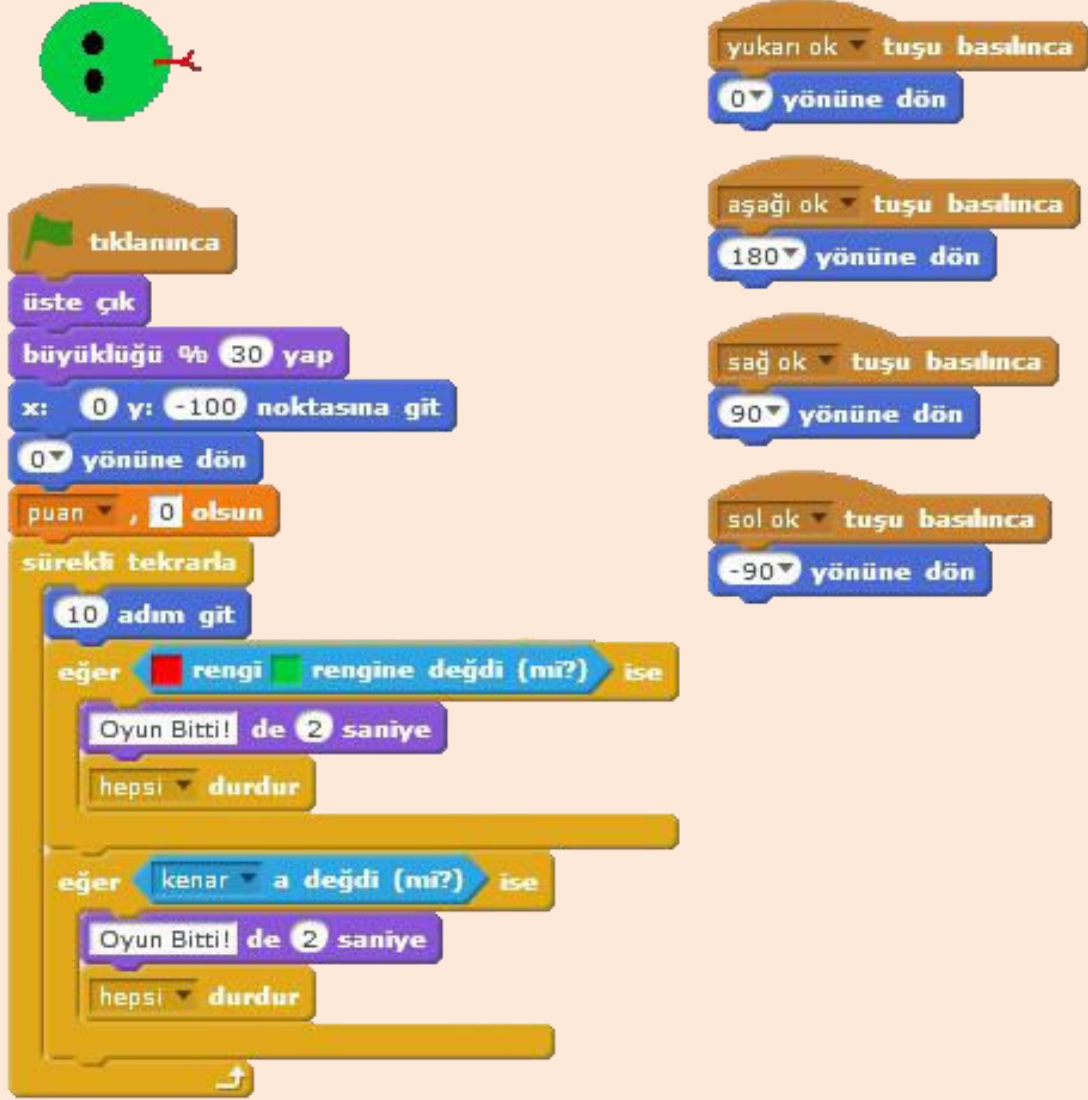


İlk çizdiğimiz karakterin küçük resmindeki **İ** kısmına tıklıyoruz ve gelen menüden **Kukla1** yazan yeri **Kafa** olarak deęiřtiriyoruz. Artık karakterimizin ismi **Kafa**.



Adını **Kafa** olarak deęiřtirdiğimiz karakterin **kılıklar** kısmına geliyoruz ve dairenin üstüne iki siyah yuvarlak göz ve kırmızı bir dil çiziyoruz.

Kafa Karakteri için Yazılacak Kodlar



Not: Kafa karakteri için yazılacak tüm kodlar yukarıdaki resimdedir. Aşağıda bu kodların açıklaması yapılacaktır.



Scratch sahnesi üzerinde yönlerimiz vardı. Karakterimiz 0 yönüne döndüğünde yukarı yönüne döner. Yukarı yönü 0, aşağı yönü 180, sağ yönü 90, sol yönü -90'dır.

Yılanımızı hareket ettirirken 10 adım git diyeceğiz. Karakterimiz hangi yöne dönmüşse o yöne doğru 10 adım giderek ilerler. Yön tuşlarıyla da hangi yöne doğru ilerleyeceğini belirleriz.



10 adım ilerlemesini sağlıyoruz. Karakterimiz yön tuşları ile belirlediğimiz yöne sürekli 10 olarak 10 adım gider.

Kafa karakterimizin önüne kırmızı bir dil yapmıştık. Yılanımızın gövdesi ise yeşil renktedir. Yılanımızın kendine çarpmasını algılamak için bu kırmızı dilin gövdeye değmesini algılamamız gerekmektedir. Bunun için de eğer ise bloğumuzda koşul olarak kırmızı rengin yeşil renge değmesini belirliyoruz. Kırmızı dil yeşil gövdeye değdiğinde ekrana oyun bitti yazdırıyoruz ve bütün blokları durduruyoruz.

Yılan karakterimizin kenara değmesi de istemediğimiz bir durum. Bunu engellemek için bir eğer ise bloğu ile kenara değdiyse yine ekrana oyun bitti yazdırıyoruz ve bütün blokları durduruyoruz.

Yeşil bayrak tıkladığında ise karakterimize **üste çık** komutu ile katman olarak en üste çıkmasını söylüyoruz.

Büyüklüğü ayarladıktan sonra başlangıç X ve Y konumunu ayarlıyoruz ve 0(yukarı) yönüne dönmesini sağlıyoruz.

Puan adında bir değişken oluşturuyoruz ve başlangıçta puan değişkenini 0 yapıyoruz.

Daha sonra bir sürekli bloğuyla

Kuyruk Karakteri için Yazılacak Kodlar



Kafa karakterimiz hareketi yönlendirirken bir kuyruk karakteriyle de onun gövdesini oluşturuyoruz.

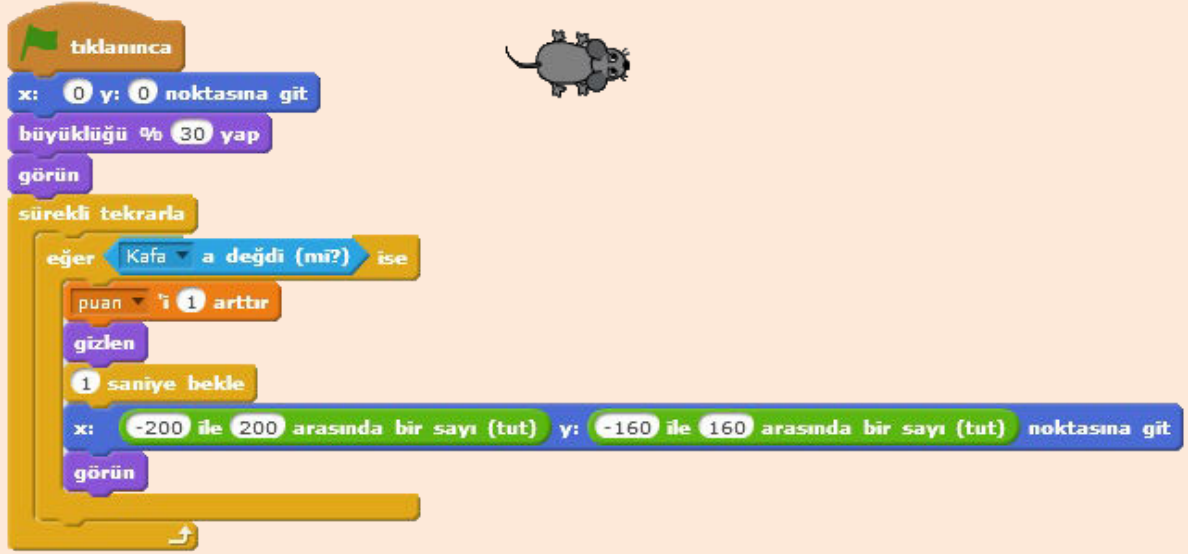
Yeşil bayrak tıklanığında önce büyüklüğü ayarlıyoruz. Daha sonra kuyruk karakterinin sürekli olarak kafa karakterine gitmesini sağlıyoruz ve sürekli olarak kuyruk karakterinin ikizini yaratıyoruz.

İkiz olarak başladığında ise puan/5 saniye bekleyerek ikizi siliyoruz. Puan arttıkça ikizin silinme süresi artacak ve yılanımızın kuyruğu uzayacaktır. Puan değişkeni, kafa değişkeni ekrandaki fareye dokunduğunda artacaktır.

Sürekli olarak kuyruk karakterinin ikizini yaratıyoruz ve kafa karakterine gitmesini sağlıyoruz. İkiz olarak başladığında ise puan değişkeninin değeri oranında bir süre sonra ikizi siliyoruz. Kafa karakteri ilerledikçe arkasında kuyruk karakterinin ikizlerini bırakıyor. Bu şekilde gövdeyi oluşturuyoruz. Bu oluşturduğumuz gövdenin de uzunluğunu puan değişkeni belirliyor. Puan değişkeni arttıkça bu ikizlerin silinme süresi artıyor ve kuyruk uzamış görünür.

Şimdi Kukla ekleme bölümü **Hayvanlar** kısmından **Mouse1** karakterini ekleyelim.

Mouse1 Karakteri için Yazılacak Kodlar

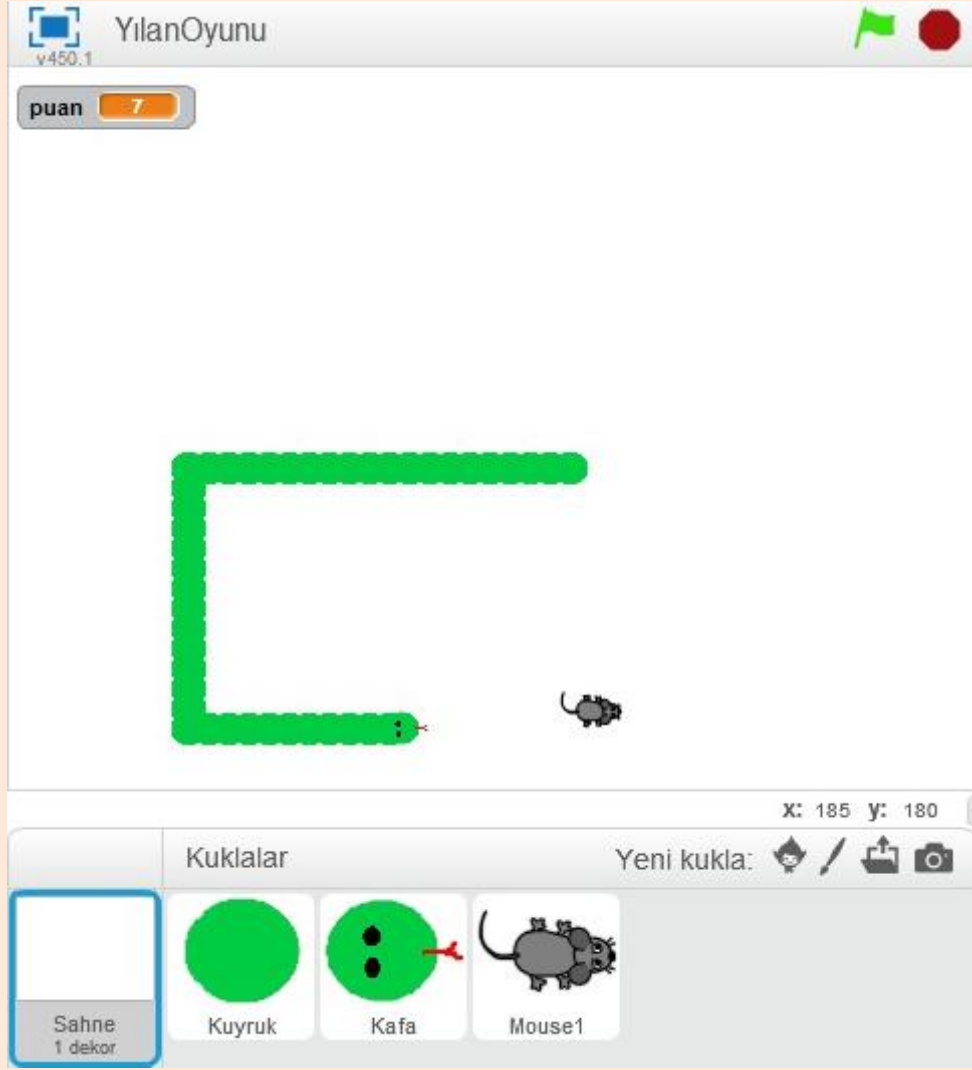


Yeşil bayrak tıklandığında öncelikli olarak fare karakterinin X ve Y konumunu, büyüklüğünü ayarlıyoruz ve sahne üzerinde görünmesini sağlıyoruz. Daha sonra bir sürekli bloğu içinde bir eğer ise bloğuyla kafa karakterine değip değmediğini kontrol ediyoruz. Eğer kafa karakteri fare karakterimize değerse puan değişkenini 1 arttırıyoruz ve sahne üzerinde gizlenmesini sağlıyoruz. Araya bir saniye bekleme koyduktan sonra belirlediğimiz sayılar arasında rastgele bir X ve Y noktasına gitmesini ve sahne üzerinde tekrar görünmesini sağlıyoruz.

Yön tuşları ile hareketini kontrol ettiğimiz yılan ile sahne üzerindeki fare karakterine dokunmaya çalışıyoruz. Yılan karakteri ile fare karakterine dokunduğumuzda sanki yılan fareyi yemiş gibi gözüküyor ve puanımız 1 artıyor. Puanımız arttığı için kafa karakterinin arkasında oluşan ikizlerin silinme süresi de artıyor ve kuyruk daha uzun gözüküyor.

İpucu: Değişkenlerin sol tarafında bulunan tık işareti ile değişkenin sahne üzerinde gözüküp gizlenmesini ayarlayabilirsiniz.





Yılan oyunu ekran görüntüsü

Giydirme Oyunu

Sıradaki tasarlayacağımız oyun karaktere elbise giydirmeye alakalı olacak. Karakter olarak eklediğimiz elbiselerle manken karakterimizi giydireceğiz. Bu oyunda değişkenleri sıkça kullanacağız. Her karaktere bir değişken atayacağız. Elbise değiştikçe değişkende değişecek. Örneğin; şapka adında bir değişken oluşturacağız.

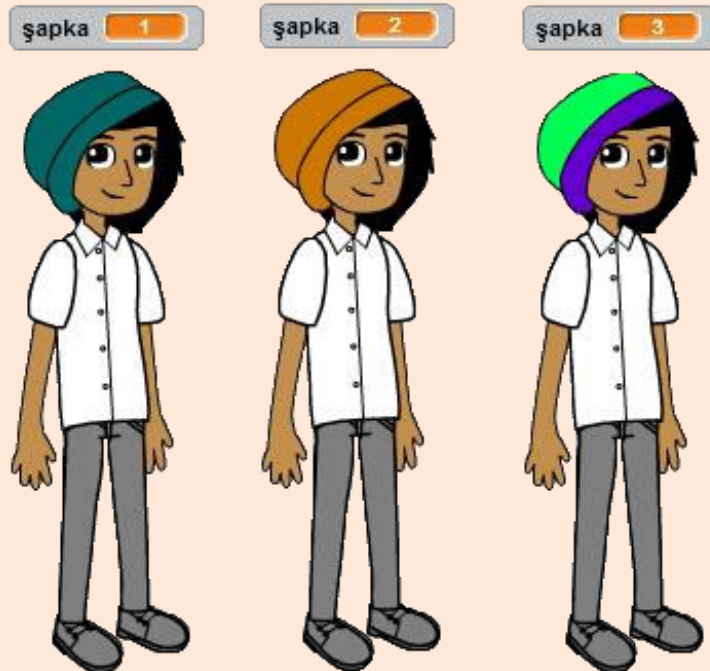
şapka , 1 olsun

Şapka değişkeninin değerine göre de mankenin kafasındaki şapka değişecek. Farz edelim ki elimizde 3 adet şapka karakteri olsun.



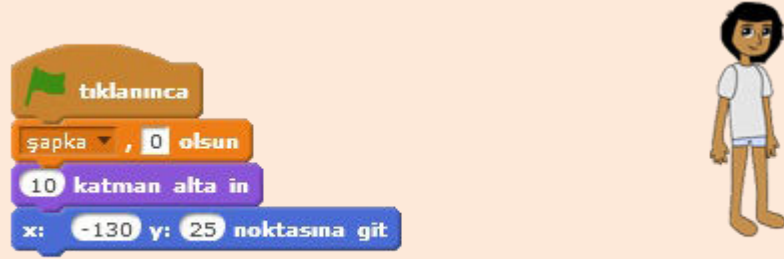
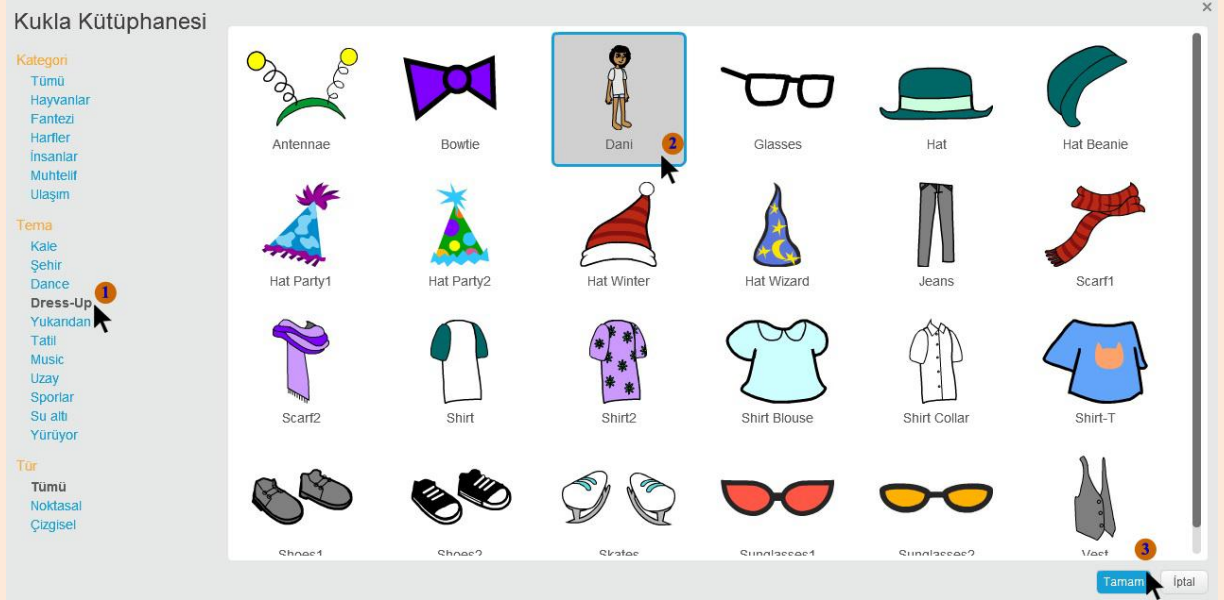
Şapka değişkeninin değeri 1 olduğunda ilk sıradaki yeşil şapka mankenin üzerinde olacak. Şapka değişkeninin değeri 2 olduğunda turuncu, 3 olduğunda ise yeşil-mor şapka mankenin üzerinde olacak.

Her elbisenin bir değişken numarası olacak, istersek biz klavyeden de bu değişken numaralarını girerek kıyafetleri değiştirebileceğiz. Mankeni giydirmemiz bittikten sonra da kod ver diyeceğiz. Oyun bize kıyafetlerin değişkenlerinin numaralarının olduğu bir kod verecek. Biz bu kodu girdiğimizde de bizim seçtiğimiz kıyafetler mankenin üzerine otomatik olarak gelecek. Girdiğimiz kodun şapka bölümü 1 ise yeşil, 2 ise turuncu, 3 ise yeşil-mor şapka mankenin üzerinde olacak.



Şimdi tasarladığımız oyunun kodlarını yazalım.

Öncelikli olarak manken olması için **Dress-Up** kısmından **Dani** karakterini ekleyelim.



Dani karakterimizi ekledikten sonra yukarıdaki resimde bulunan kodları yazalım. Şapka isminde bir değişken oluşturalım. Şapka değişkeni oyun başladığında 0 olsun. Dani karakterinin katman olarak kıyafetlerden aşağıda olması için 10 katman alta in diyelim. Oyun üzerindeki X ve Y konumlarını ayarlayalım.

Mankenimizi ekledikten sonra mankene giydireceğimiz kıyafetlerin karakterlerini ekleyelim. **Kuklalar** kısmından **Dress-Up**'a gelelim ve şapkalardan biri olan **Hat Beanie** karakterini ekleyelim. Bu karakteri çoğaltıp rengini değiştirerek 3 adet şapka elde edeceğiz ve her birini programlayacağız.

Eklediğimiz **Hot Beanie** karakteri için aşağıdaki kodları yazalım.



Karaktere tıklandığında mankenin üzerine gitmesi için şapka değişkenini 1 yapıyoruz. Biz karaktere tıkladığımızda şapka değişkeni 1 olacak ve karakter, mankenin saçlarının olduğu koordinatlara gidecek.

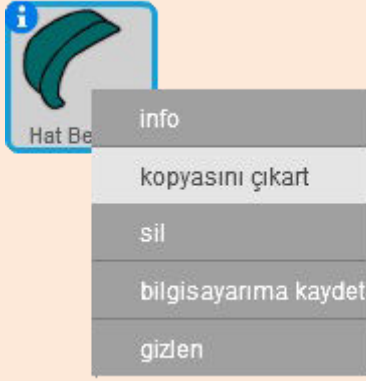
Bir sürekli bloğunun içine eğer ise değilse bloğu koyuyoruz. Koşul olarak şapka değişkeninin 1'e eşit olmasını belirliyoruz. Eğer koşul gerçekleşmişse yani karaktere tıklanmışsa, karakterimiz katman olarak en üste çıkar ve mankenin saçlarının olduğu koordinata gider.

Değilse kısmında ise mankenin üzerinde değilse durması gereken konuma gider.

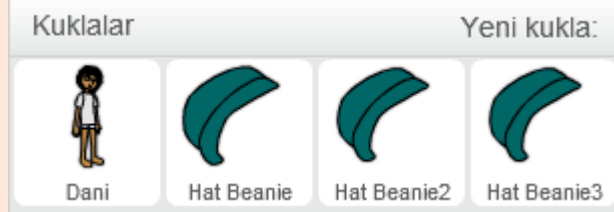


X ve Y konumlarını kendimiz belirlediğimiz için karakterlerimizin oyun içinde fare ile sürüklenme özelliğini kaldırıyoruz. Önce karakterin küçük resmine tıkladığımızda açılan İ butonuna tıklıyoruz ve gelen menüden fareyle sürüklenebilir özelliğindeki tık işaretini kaldırıyoruz.

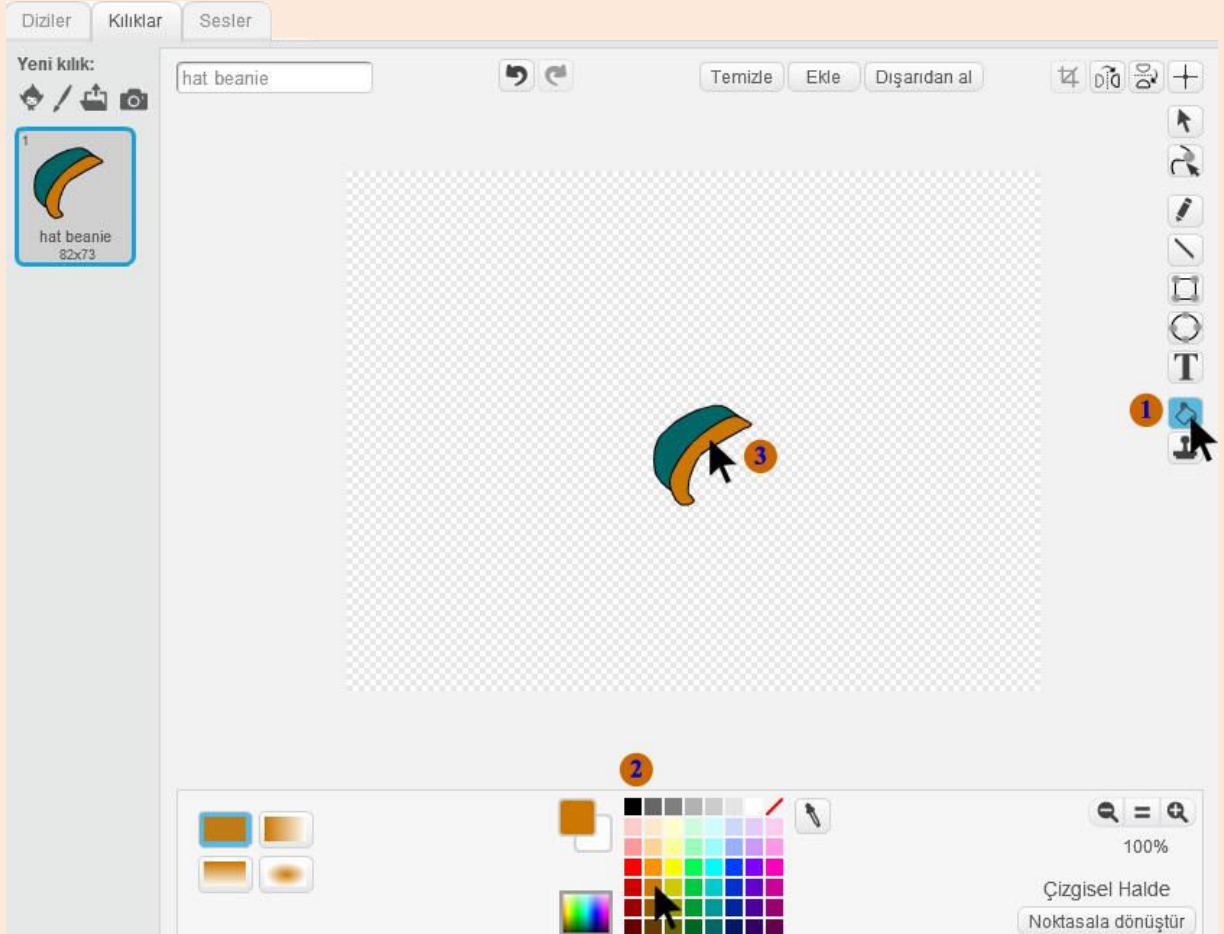
Not: Bütün karakterlerin fareyle sürüklenebilir özelliğini kaldırın.



Şapka karakterimizden üç tane gerektiği için karaktere sağ tıklayarak **kopyasını çıkart** diyelim. Bu işlemi tekrarlayarak iki adet kopyasını çıkaralım. Kopyasını çıkart dediğimizde kodlarında kopyası çıkararak diğer kopyalara eklenecektir.



Elimizde üç adet şapka karakteri oldu. Şimdi kılıklar kısmından şapkalarımızın rengini değiştirelim.



Kılıklar kısmından kova aracını seçelim ve istediğimiz renkleri kullanarak **Hat Beanie2** ve **Hat Beanie3** karakterlerini boyayalım.

Rengini deęiřtirdiđimiz **Hat Beanie2** karakteri iin sarı dikdörtgenle belirtilmiř yerleri deęiřtirelim.



X konumunu ayarlayalım ve bu sefer karaktere tıklandıđında şapka deęiřkeni 2 olsun. Şapka deęiřkeni 2 olduđunda yani Hat Beanie2 karakteri tıklandıđında mankenin üzerine Hat Beanie2 karakteri gider.

Not: Burada karaktere tıklandıđında direkt olarak mankene gitmesini sađlayabilirdik ancak dıřarıdan kod girildiđinde de manken üzerindeki kıyafetlerin otomatik deęiřebilmesi iin bu şekilde bir yapı kullandık.



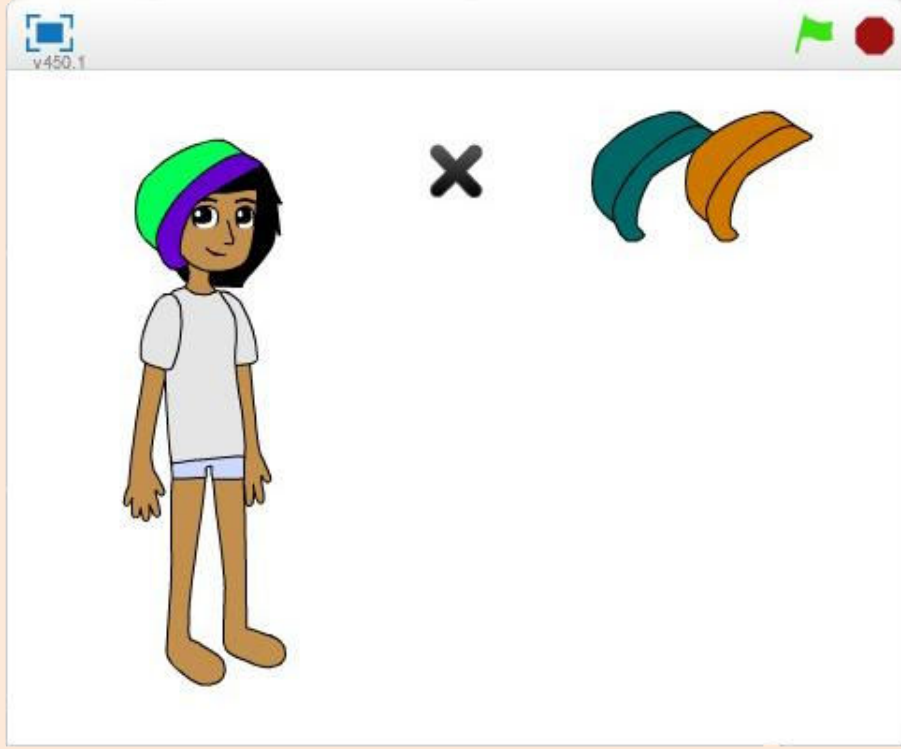
Aynı değişiklikleri **Hat Beanie3** karakteri için de yapıyoruz.

Mankenin üzerinde şapka olmasını istemiyorsak, şapka olmaması için bir karakter ekleyerek bu sefer bu karaktere tıklandığında şapka 0 olsun diyeceğiz. Şapka değişkeni 0 olduğunda hiçbir şapka karakteri mankenin üzerinde olmaz.

Kukla ekle dedikten sonra **Muhtelif** bölümünden **Button5** karakterini ekleyelim.



Butonumuzun büyüklüğünü, X ve Y konumunu ayarlayalım. Butonumuza tıklandığında da şapkaların hiçbirinin mankenin üzerinde olmaması için de bu kukla tıklanınca şapka 0 olsun diyelim.



Bu adıma kadar yaptıklarımızı kontrol edelim. Yeşil bayrağa tıkladığımızda yukarıdaki gibi bir ekran olacaktır ve hangi şapkaya tıklarsanız o şapka mankenin üzerine gidecektir. Çarpı işareti olan karaktere tıkladığınızda ise şapkaların hiçbiri mankenin üzerinden olmayacaktır.

Not: Yazdığınız kodları test ederken oyunun düzgün görüntülenmesi için tam ekranda test ediniz.

Şapka karakterlerini ekledikten ve kodlarını yazdıktan sonra sıra gözlük karakterlerini eklemekte. **Kukla** ekle kısmında **Dress-Up** bölümünden **Glasses** karakterini ekleyelim. Rengini ayarladıktan sonra aşağıdaki kodları yazalım.



Değişken olarak bir gözlük değişkeni tanımlayalım. İlk gözlük karakterimize tıkladığında gözlük değişkeni 1 olsun.

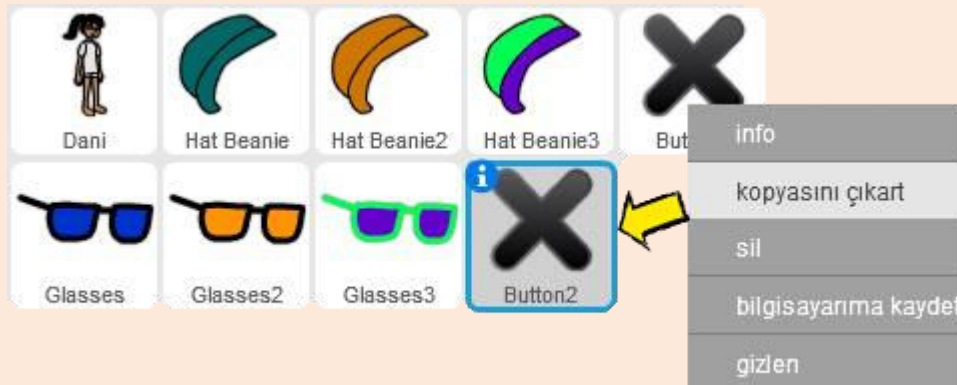
Gözlük değişkeni 1 olduğunda katman olarak üste çıksın ama şapkanın altında olması için 1 katman alta insin, X ve Y koordinatı olarak belirttiğimiz konuma gitsin diyoruz.

Değilse kısmında ise normalde sahne üzerinde duracağı koordinatlarda durmasını sağlıyoruz.

Glasses karakterimize sağ tıklayarak iki adet kopyasını çıkarıyoruz, renklerini ayarladıktan sonra kodlarda gereken yerleri değiştirerek aşağıdaki resimde bulunan kodları elde ediyoruz.



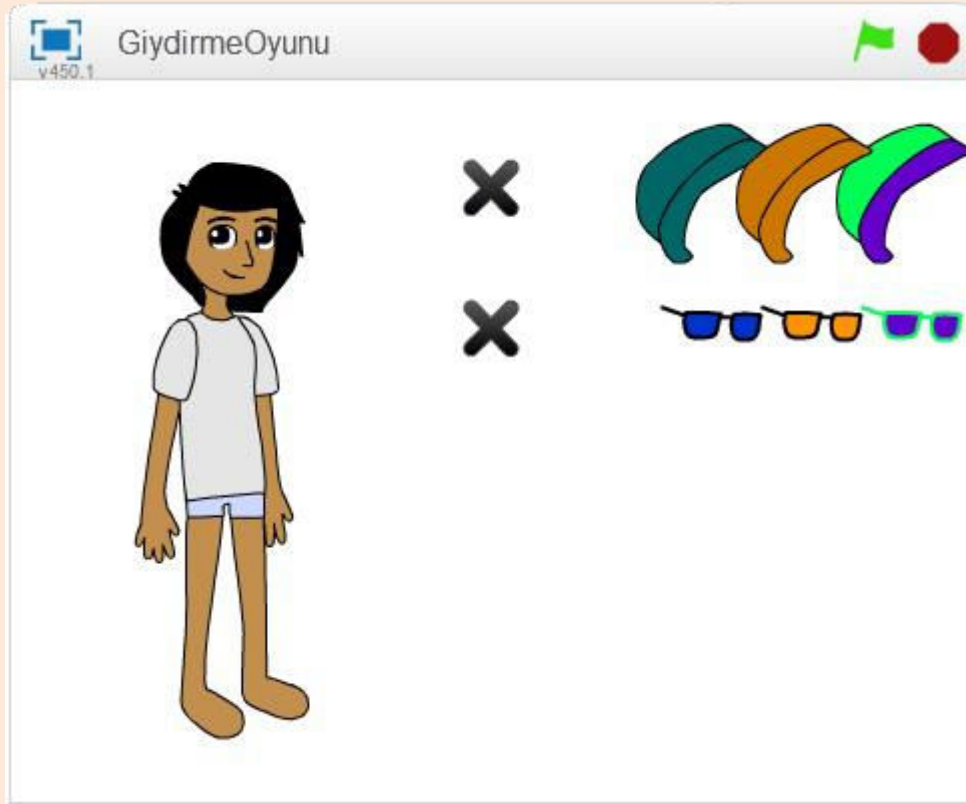
Gözlük karakterlerimizi ayarladıktan sonra çarpı butonunu ayarlayalım. Şapka karakterlerimizde olduğu gibi bir çarpı karakteri ile manken üzerinde gözlük olmamasını sağlayacağız.



Çarpı butonuna sağ tıklayalım ve kopyasını çıkart diyelim. İçindeki kodlarla birlikte bir kopyası çıkacaktır.



Oluşan kopyanın kodlarını değiştirerek yandaki resimdeki gibi ayarlayalım.



Oyunun oluşan yeni ekran görüntüsü aşağıdaki gibi olacaktır.

Şimdi de tişörtlerimizi ekleyelim. **Dress-Up** kısmından bu sefer karakter olarak **Shirt** karakterini ekleyelim.

```
bu kukla tıklanmca
tişört , 1 olsun

tıklanmca
sürekli tekrarla
eğer tişört = 1 ise
üste çık
2 katman alta in
x: -133 y: 20 noktasına git
değilse
x: 100 y: 0 noktasına git
```



Diğer karakterler için yaptığımızı benzer şekilde tişört isminde bir değişken ekleyelim.

Yandaki resimde bulunan kodları **Shirt** karakteri için ekleyelim.

Şapkaların, gözlüklerin ve tişörtlerin ayrı katmanı olduğu için **2 katman alta in** diyoruz.

Diğer tişört karakteri için Dress-Up kısmından **Shirt2** karakterini ekleyelim.

```
bu kukla tıklanmca
tişört , 2 olsun

tıklanmca
sürekli tekrarla
eğer tişört = 2 ise
üste çık
2 katman alta in
x: -133 y: 20 noktasına git
değilse
x: 150 y: 0 noktasına git
```



Eklediğimiz **Shirt2** karakteri için yan taraftaki kodları yazalım.



İpucu: Bir karakterde bulunan kod bloklarının kopyasını başka bir karakterde kullanmak için kod bloğunu karakterin küçük resminin üzerine sürükleyip bırak yöntemiyle sürükleyip bırakırsanız kod bloğunun bir kopyasını o karakter için eklersiniz.



Son tişört karakterimizi oluşturmak için **Shirt2** karakterimize sağ tıklayalım ve kopyasını çıkart diyelim.



Shirt2 karakterinin kopyasını çıkartarak oluşturduğumuz **Shirt3** karakterinin kılıklarına gelerek **Shirt2-b** kılığını seçelim ve rengini değiştirelim.

bu kukla tıkanınca
tişört , 3 olsun



Yan tarafta bulunan kodları **Shirt3** karakterimiz için yazalım.

tıkanınca
sürekli tekrarla
eğer tişört = 3 ise
üste çık
2 katman alta in
x: -133 y: 20 noktasına git
değilse
x: 200 y: 0 noktasına git

bu kukla tıkanınca
pantolon , 1 olsun



Pantolonlar için Dress-Up kısmından Jeans karakterini ekleyelim. Yandaki kodları Jeans karakteri için yazalım.

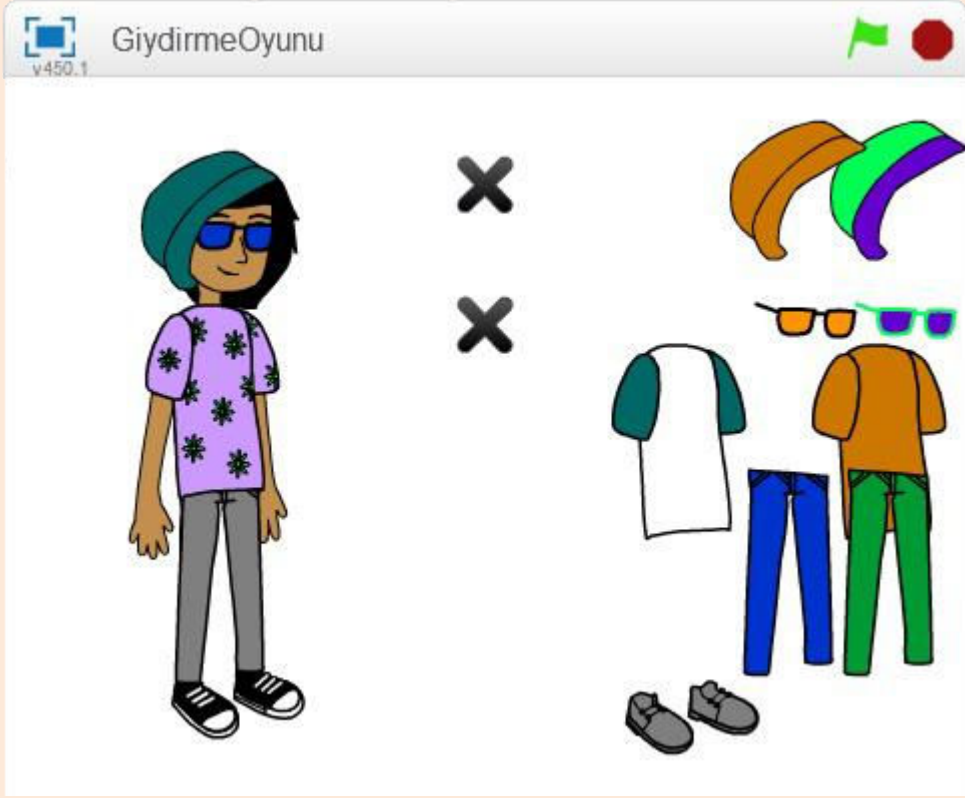
tıkanınca
sürekli tekrarla
eğer pantolon = 1 ise
üste çık
3 katman alta in
x: -134 y: -44 noktasına git
değilse
x: 100 y: -40 noktasına git

Jeans karakterinin iki adet kopyasını çıkartalım ve renk düzenlemelerini yapalım. Aşağıdaki resimde görülen şekilde kodlarımızı düzenleyelim.



Son olarak ayakkabılarımızı ekleyelim. **Dress-Up** kısmından **Shoes1** ve **Shoes2** karakterlerini ekleyelim. Ayakkabı adında bir değişken oluşturalım ve aşağıdaki resimde bulunan kodları karakterlerimiz için yazalım.



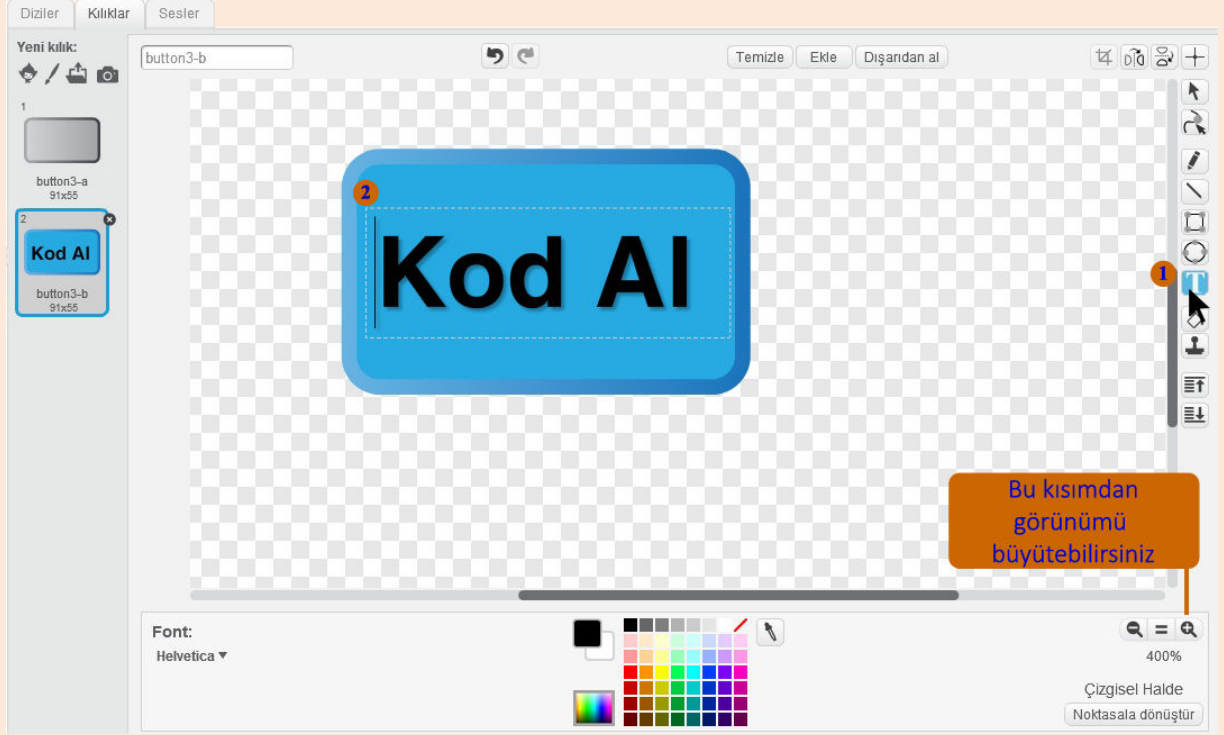


Oyunun son ekran görüntüsü

İsterseniz oyundaki sistemi koruyarak yeni kıyafetler ekleyebilirsiniz.

Oyunun ilk aşamasını bitirdik. İkinci aşamada, oyunda yaptığımız değişiklikleri daha sonrada uygulayabilmemiz için kod üreteceğiz ve bu ürettiğimiz kodu girdiğimizde o değişiklikler uygulanacak.

Kukla ekleme bölümünden **Muhtelif** kısmına gelelim ve **Button3** karakterini ekleyelim.



Button3 karakterinin kılıklarına gelelim ve **Button3-b** kılığının üzerine **Kod AI** yazalım. Bu butona tıkladığımızda bize kıyafetlerin kodunu verecek.



Veri menüsünden bir liste oluşturma tıklayalım ve Kod adında bir liste oluşturalım. Kod AI butonu için yukarıdaki kodları yazalım.

Veri menüsünde Kod adında bir liste oluşturalım. Yeşil bayrak tıkladığında Kod listesi gizlensin. Butonun büyüklüğü %60, X ve Y konumunu da belirttiğimiz değerler olsun.

Kuklaya tıkladığında yani butona tıkladığında ise öncelikli olarak Kod listesindeki bütün elemanları siliyoruz. Daha sonra kıyafetlerin değerlerini hafızada tutan değişkenlerin değerlerini birleştirerek Kod listesine ekliyoruz.

Birleştir bloklarını yapabilmek için



hello ile hello ile world i birleştir i birleştir

İki tane hello ile world birleştir bloğunu alalım. Birini diğerinin sağına sürükleyip bırakalım ve bu işlemi üç kez yapalım.



hello ile hello ile hello ile hello ile world i birleştir i birleştir i birleştir i birleştir

Daha sonra oluşturduğumuz bu blokları liste elemanlarından olan ekle bloğunun boş kısmına sürükleyip bırakıyoruz.



thing i kod ekle
hello ile hello ile hello ile hello ile world i birleştir i birleştir i birleştir i birleştir

Son olarak değişken bloklarımızı boşluğa sürükleyip bırakalım.

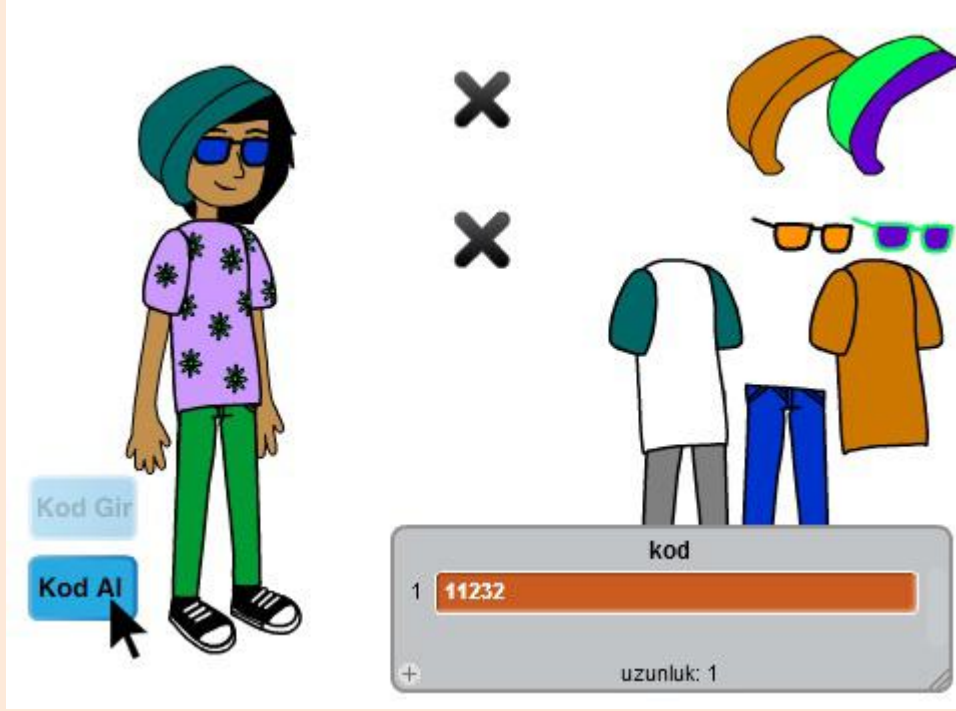


şapka ile gözlük ile tişört ile pantolon ile ayakkabı i birleştir i birleştir i birleştir i birleştir i kod ekle

Bu kod bloğumuz çalıştığında kıyafetlerimizin kodlarını birleştirecek ve biz bu kodları ekrana yazdıracaktır.

Kodları birleştirerek Kod listesine ekledikten sonra Kod listesini göster diyoruz. Belli bir süre bekledikten sonra da işi biten Kod listesini gizliyoruz.

Eğer ise değilse kısmında da butona fare oku değmiyorken hayalet etkisini 80 yaparak ekranda silik gözükmesini, fare oku ile üzerine geldiğimizde ise hayalet etkisini 0 yaparak normal gözükmesini sağlarız.



Daha önce aldığımız kıyafet kodunu girip istediğimiz değişiklikleri uygulamak için bir **Button3** karakteri daha ekleyelim. Düzenleyerek üzerine **Kod Gir** yazalım. **Kod Gir** butonu için de aşağıdaki kodları yazalım.

```

tıldanınca
büyüdüğü % 60 yap
x: -200 y: -80 noktasına git

bu kukda tıldanınca
kod listesini gizle
diye sor ve bekle

şapka , 1 . harfi yanıt in olsun
gözlük , 2 . harfi yanıt in olsun
tişört , 3 . harfi yanıt in olsun
pantolon , 4 . harfi yanıt in olsun
ayakkabı , 5 . harfi yanıt in olsun

```

```

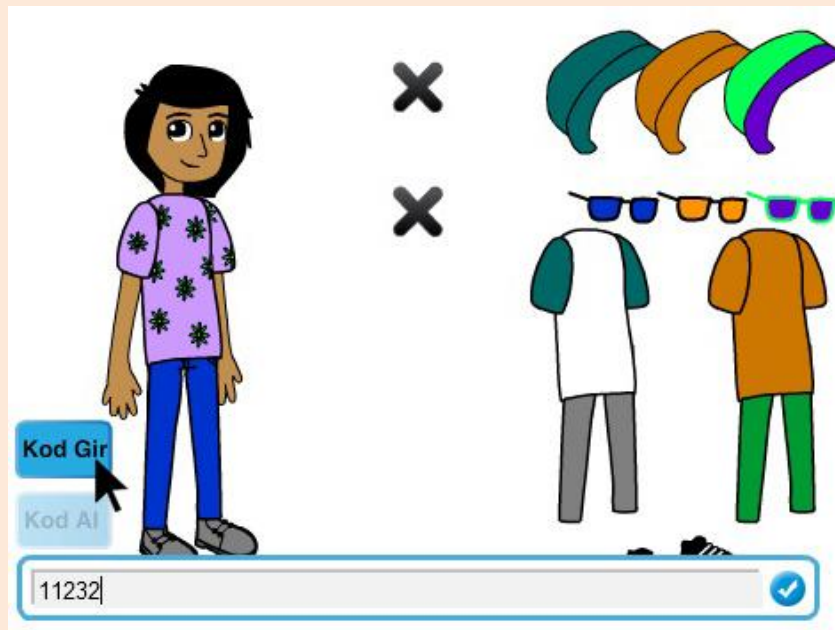
tıldanınca
sürekli tekrarla
eğer fare oku a değdi (mı?) ise
hayalet etkisi 0 olsun
değilse
hayalet etkisi 80 olsun

```

Kod Gir

Kod Al butonuna tıkladığımızda bize 5 haneli bir sayı verecektir. Örneğin; 11232 gibi. Bu rakam grubunun ilk rakamı şapkayı temsil eder. Burada ilk rakam 1 olduğu için yeşil şapka seçilecektir, ikinci rakam gözlüğü, üçüncü rakam tişörtü, dördüncü rakam pantolonu, beşinci rakam ayakkabıyı temsil ediyor.

Kod Al butonuna tıkladığımızda önce Kod listesi gizlenir ve ardından boş bir soru sorar ve yanıt girmemizi bekler. Bu açılan kısma kodumuzu gireriz. Girdiğimiz kod yanıt değişkenine aktarılır. Yanıtın ilk rakamı şapka değişkenine aktarılır. Yanıtın ikinci rakamı gözlük değişkenine aktarılır. Bu şekilde diğer değişkenlere de gerekli değerler aktarılarak kodumuzu gireriz. Girdiğimiz koda göre de gerekli kıyafet değişiklikler otomatik olarak yapılır.



Scratch'da Bir Oyunu Hacklemek

Günümüzde hack kelimesi ve hackerlık çok yanlış anlaşılmıştır. Hacker diyince hepimizin aklına şifre kıran, sistemlere sızan insanlar gelir. Bu çok yanlış bir düşüncedir. Hack kelimesi bu kadar basit bir kelime değildir ve hackerlar sadece şifre kırmazlar.

Hackerlar bir sistemi alıp kendi isteği doğrultusunda değiştiren ya da kullanan kişilerdir. Bu sistem bir bilişim sistemi olabileceği gibi herhangi bir sistem de olabilir. Örneğin; uzaktan kumandalı arabanızı alıp onu bluetooth kumandalı hale getirip akıllı telefonunuzdan kontrol ettiğinizi varsayalım. Sizin yaptığınız şey uzaktan kumandalı arabanızı hacklemektir. Ama bunu yapabilmeniz için bu sistemlerin nasıl çalıştığını bilmeniz gerekir.

Bir sistemi hacklemek için öncelikli şart; o sistemi iyi tanımdır. Bilişim sistemlerini hackleyebilmeniz için de bilgisayarların nasıl çalıştığını ve nasıl programlandıklarını öğrenmeniz gerekir. Hackerlar üst düzey programlama bilgisine sahiptirler. Eğer Hacker olmak istiyorsanız bilgisayar programlarının nasıl yapıldığını öğrenmeniz gerekir. Scratch da bu işe başlamanız için çok uygundur.

Scratch'da bir oyunu hacklemek için öncelikli olarak Scratch bloklarını iyi tanımanız ve programlama mantığını iyi bir şekilde kazanmış olmanız gerekmektedir. Daha sonra projelerin internet sayfalarından **içine bak** butonu ile kaynak kodlarını görebilir ve gönlünüzce kodlarını değiştirerek o oyunu hackleyebilirsiniz.

Gerçek hackerlar bir sistemi kırmak,yıkmak yerine o sistemi geliştirip daha iyi hale getirirler.

Aslında hack kötü bir şey değildir. Hack kültürü kırmayı değil yapmayı, inşa etmeyi temel alır. Sizde bir şeyleri kırmak yerine onu alıp daha iyi bir hale getirmeye çalışın. Örneğin; bir oyunun hatalarını bulup düzeltin ve onu daha iyi bir hale getirin.

