

5. ÜNİTE: BT.5.5. PROBLEM ÇÖZME VE PROGRAMLAMA



Problem Nedir?

Problem, çözülmesi gereken sorun ya da aşılması gereken engel anlamına gelir. Günlük hayatta sık sık problemlerle karşılaşırız.

Karşılaştığımız bir problemi çözmek için ne yaparsınız?

Bir Problemin Çözümü İçin...

- Problemi anlama
- Bir plan yapma
- Planı uygulama
- Çözümü değerlendirme

Problemi anlama: Bir problemi doğru bir şekilde çözebilmenin ilk aşaması problemi anlamakla başlar, problemi anlamak aşağıdaki sorulara yanıt verilebildiği zaman gerçekleşmiş olur.

- Bu problem ile ilgili elimizdeki temel bilgiler nelerdir?
- Çözüme ulaşabilmek için bu temel bilgilerin dışında ihtiyaç duyduğlarımız varsa nelerdir?
- Problemde kimler, nasıl etkilenmektedir?
- Problemin şeması oluşturulabilir mi? Anlaşılabilirliği arttırmak için problemi görselleştirebilir miyiz?
- Problem nasıl ortaya çıkmış? Bildiğimiz ve daha önce çözüm ürettiğimiz benzer problemler oldu mu?

Bir plan yapma: Her problemin çözümüne farklı çözüm yollarından ulaşılabilir, bunu için doğru çözüm planını tercih ederek uygulamak gerekmektedir. Bir problem için aşağıdaki çözüm planları uygulanabilir:

- Deneme yanılma yoluyla çözümü test etme, olası adımları ve tahmini sonuçları bulma
- Problemi daha basit parçalara ayırarak ilerleme
- Sondan başa doğru ilerleyerek farklı bir bakış açısıyla çözüme ulaşmaya çalışmak
- Olası çözümleri içeren bir görsel harita ya da zihin haritası çıkarmak (Çünkü bazılarımız çizerek çalışmayı sever)
- Beyin fırtınası yapmak

Planı uygulama: Bir planın uygulama sonucu başarıyla sonuçlanabileceği gibi başarısızlıkla da sonuçlanabilir, başarısızlıkla sonuçlanan denemelerde öğrenme deneyiminin bir parçasıdır.

PROBLEM ÇÖZME STRATEJİLERİ

Hayatımızda birçok problem ile karşılaşabiliriz; birçoğunun tek bir çözümü olacağından kolay çözülebilirken bazıları ise daha karmaşık olabilir ve problem çözme stratejilerini kullanmak gerekebilir.

Örneğin matematik derslerinde problemler bir metin içerisinde yer alıyor ve buradan çözüme ulaşmanız gerekiyor. Böyle problem türlerinde hangi çözüm stratejilerini kullandığımızı bir hatırlayalım:

- Öncelikle soruyu dikkatli bir şekilde okumalısınız.
- Problemde önemli olan yerlerin altını çizerek önemli bilgiyi ayırt etmelisiniz.
- Eğer ihtiyacınız olursa resim çizebilirsiniz; çözüm konusunda size fikir verebilir.
- Tekrarlayan öğeler, örüntüler olabilir; bunları bulmalısınız.
- Daha basit bir problemi çözmeye çalışabilirsiniz. Bu süreç size çözümde yardımcı olabilir.

- Sondan başa doğru gidebilirsiniz; sorunun sonundan başlayarak geriye dönük çalışmak problem çözümünde yardımcı olabilir.
- Bulduğunuz bir çözüm için deneme yapmanız size yardımcı olabilir.
- Son olarak çözümü test etmeniz ve işe yarayıp yaramadığını görmemiz gerekebilir. Çözüme ulaşılmadığı durumda farklı stratejiler ile çözüm aramak gerekebilir.
- Başka çözüm stratejileri de ifade edilebilir; deneme yanılma, canlandırma yapma, model oluşturma, problemi küçük parçalar halinde ele alma, tahmin etme, problemi sadeleştirme, neden sonuç ilişkisi kurma, hesaplama yapma, algoritma oluşturma vb.
- Bu konudaki farkındalığı arttırmak için öğrencilere problem çözmek için hangi stratejileri kullandıkları sorulabilir.

Problem Çözme

Günlük yaşamda karşılaştığımız problemleri bilerek veya farkında olmadan adım adım çözmeye çalışırız. Örneğin yazı yazarken kaleminizin ucu kırıldığında şu adımları takip ederek bu sorunu çözersiniz.

1. Kalemtraşını çıkar.
2. Kalemi al.
3. Çöp kovasının yanına git.
4. Kalemin ucunu aç.
5. Sırana geri dön.
6. Yazmaya devam et.

VER ELİNİ VERİ

Günlük işlerimizin tamamını yaparken adeta bilgi yağmuru ile karşı karşıya kalırız. Beynimiz ise bu bilgi yağmurundan işine yarayacağını düşündüklerini alır yorumlar ve biz de kararlarımızı bu yorumlara göre veririz. Hatta zihnimiz daha sonra kullanmak için bazı bilgileri saklar. Örneğin sizce, evden okula yürüyerek gelen bir öğrenci ne tür bilgilerle karşılaşmaktadır.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Bilgisayarlar da tıpkı beynimiz gibi çeşitli kararlar vermek veya işlem yapabilmek için bilgi girişine ihtiyaç duyarlar. İşte bilgisayarların sonuca ulaşabilmek için algıladığı, işlediği, sonuç ürettiği veya daha sonra kullanmak üzere depoladığı her şeye “**veri**” denir.

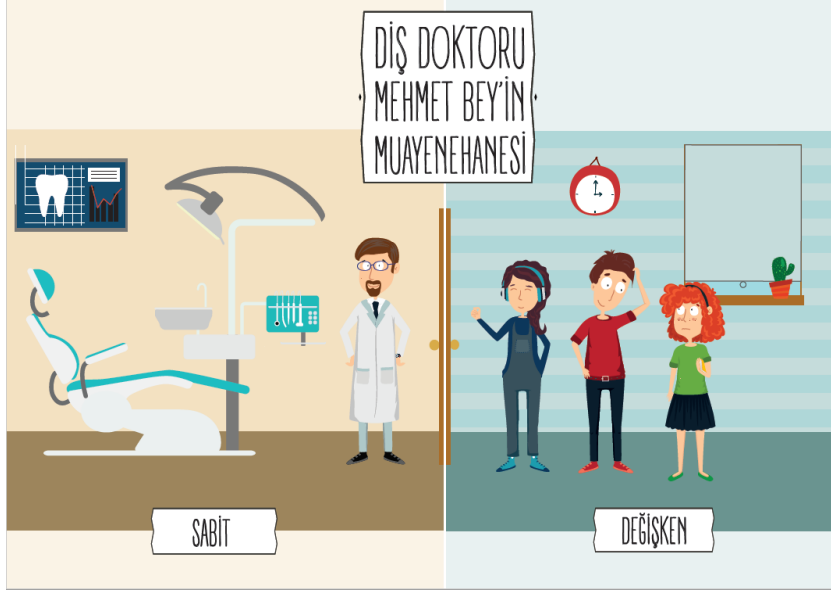
Veri, dünyanın gerçekleridir aslında. Kendimizi örnek alacak olursak; boyunuz 1.50 cm uzunluğunda olabilir, kahverengi saçlı ve mavi gözlü olabilirsiniz. Bunların hepsi birer “veri” dir. Birçok yönden, veriler dünyanın bir tanımı olarak düşünülebilir. Bu verileri duyularımızla algılarız ve beyin bu veriyi işleyebilir. İnsanlar, dünyayı tanımak ve anlamlandırmak için verileri kullanırlar.

Bilgi, öğrendiklerimizi duyularımızın ötesinde genişletmemizi sağlar. Bilgiler arasında verileri yakalayabiliriz.

Bunu basit bir örnek ile anlatacak olursak;
Eğer sizin bir fotoğrafınızı çekersem, fotoğraf benim için bir bilgi olmuştur.
Neye benzediğiniz ise verilerdir.



Fotoğrafınızı dosyaya koyabilirim, çantamda taşıyabilirim, çıktısını alabilirim, e-posta ile başkaları ile paylaşabilirim. Ancak, sizi fiziksel olarak çantamda gezdiremem, sizin yansımanızı taşıyorum. İşte fark buradadır. Fotoğrafi kaybedersem veya yırtılırsa, bu sizin görünümünüzü değiştirmez.



Sabit	Değişken



Peki Ya Bilgisayarlar?

Bilgisayarlar da problemleri tıpkı bizler gibi çözmeye çalışır. Kullanıcı tarafından kendisine verilen komutları **adım adım** uygulayarak problemin çözümüne ulaşır.

Kullandığımız yazılımların tamamı «kod» adı verilen bilgisayarın anlayacağı dilde yazılmış özel komutlardan oluşur. Bu kodlarda programların yapacakları işlemler adım adım belirtilir. Bilgisayar bu kodları sırasıyla adım adım uygular ve sonuca ulaşır. Bu kodlar bilgisayar yazılımcıları tarafından yazılır.

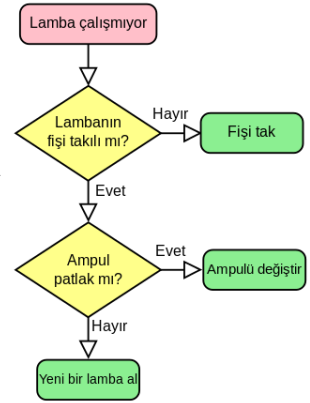
```
1 #include <stdio.h>
2 #include <conio.h>
3 //www.bilgisayarkavramlari.com
4 int main(){
5     int sayi;
6     printf("Bir sayi giriniz");
7     scanf("%d",&sayi);
8     for(int i = 2;sayi>1;i++){
9         while(sayi%i==0){
10             printf("%d ",i);
11             sayi = sayi / i;
12         }
13     }
14     getch();
15 }
```

****Şimdi öğretmenimizden “Ağustos Böceği ile Karınca: Bir Algoritma Masalı” hikayesini dinleyelim.**

Algoritma

Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur.

Algoritma basamaklarının bir başlangıcı ve sonu bulunur. Her adımda yapılacak işlem açıkça belirtilir.



Örnek Algoritma

Şimdi basit bir problemin çözümünü gösteren bir algoritma hazırlayalım.

Ayran yapıp bardağa dolduralım.

- Adım 1: Başla
- Adım 2: Yoğurdu kaba koy.
- Adım 3: Su ekle.
- Adım 4: Çırp.
- Adım 5: Tuz koy.
- Adım 6: Bardağa doldur.
- Adım 7: Bitir.



Örnek Algoritma – 2

Arabayı çalıştırıp yola çıkaralım.

Adım 1: Başla

Adım 2: Sürücü koltuğuna geç.

Adım 3: Emniyet kemerini tak.

Adım 4: Aynaları kontrol et.

Adım 5: Anahtarı tak.

Adım 6: Kontakı çevir.

Adım 7: El frenini indir.

Adım 8: Vitese geç.

Adım 9: Gaza bas.

Adım 10: Bitir.

Neden Algoritma Kullanıyoruz?

Sizce kodlamaya başlamadan önce niçin algoritma hazırlıyoruz?

Gerekli tüm bilgi ve birikime sahipsiniz ve sizden bir
Yapacağınız ilk iş ne olurdu?



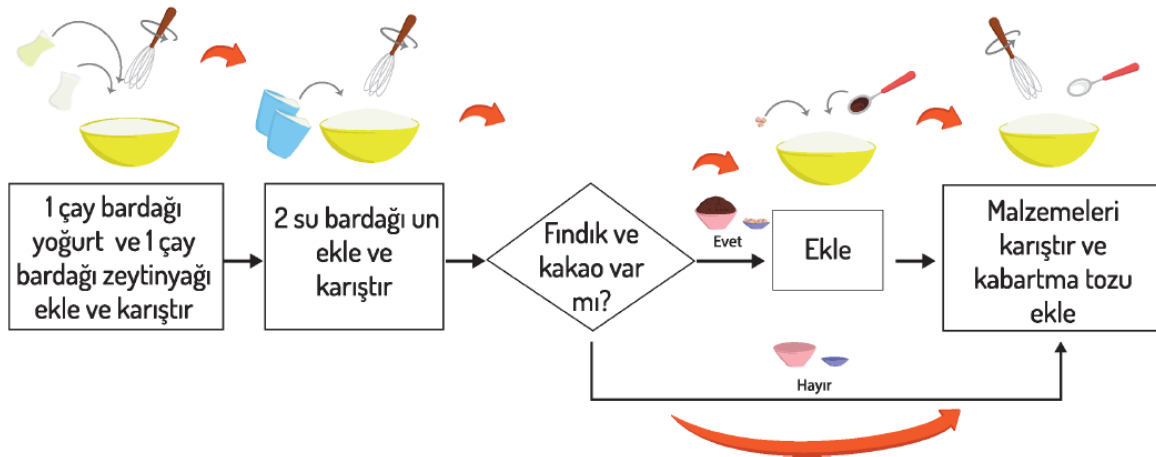
Neden Algoritma Kullanıyoruz?

Oluşturacağımız yazılımın kusursuz olması için öncelikle her adımını
gösteren planını, yani algoritmasını hazırlamalıyız.



Akış Şeması

Bir sürecin adımlarını görsel ya da sembolik olarak gösterir. Farklı Hareketler için farklı semboller kullanılır.



Bir problemin çözümüne yönelik oluşturduğumuz algoritmaları adım adım tanımak ve programlarken hangi işlemleri yapacağımızı anlamak amacıyla akış şemalarından yararlanırız.



Akış şemaları çeşitli şekillerden oluşur.

Bu şekillerin bir çoğu matematik dersinde kullandığımız geometrik şekillere çok benzer.

Akış Şeması Sembolleri

Elips: Akışı Başlatır ve bitirir. başlangıcında ve bitişinde bu sembolü kullanırız.



Dikdörtgen: Dikdörtgen, Algoritmaların akışında yapılacak olan eylemi / işlemi belirtmek için kullanılır.



Paralelkenar:

Akış şemasında dışarıdan bir bilgi/veri girişi yapılırken paralelkenar kullanılır. Örneğin klavye ile girmemiz gereken bir veri varsa bu sembolü kullanırız.



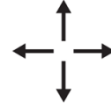
Eşkenar Dörtgen: bir duruma karar verme merkezidir. Birkaç farklı seçenek varsa eşkenar dörtgen kullanılarak seçenekler yazılır.



Dalgalı Dörtgen: Algoritmanın sonucunu (ekranda) bir çıktı olarak göstermek istediğimizde dalgalı dörtgen sembolünü kullanırız.



Oklar: Akış yönünü gösterir ve sembolleri birbirine bağlar.



Altıgen: Tekrar eden komutları belirtir.



Akış Şeması Örnekleri

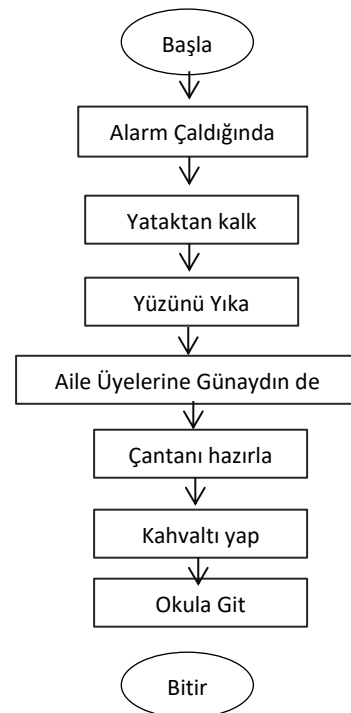
Sabah Rutini

Algoritma Alarm Çaldığında,

1. Yataktan kal
2. Yüzünü Yıka
3. Aile üyelerine günaydın de
4. Giysilerini giy
5. Çantayı hazırla
6. Kahvaltı yap
7. Okula git



Akış Şeması



Çıkartma İşlemi



Çıkartma İşlemi

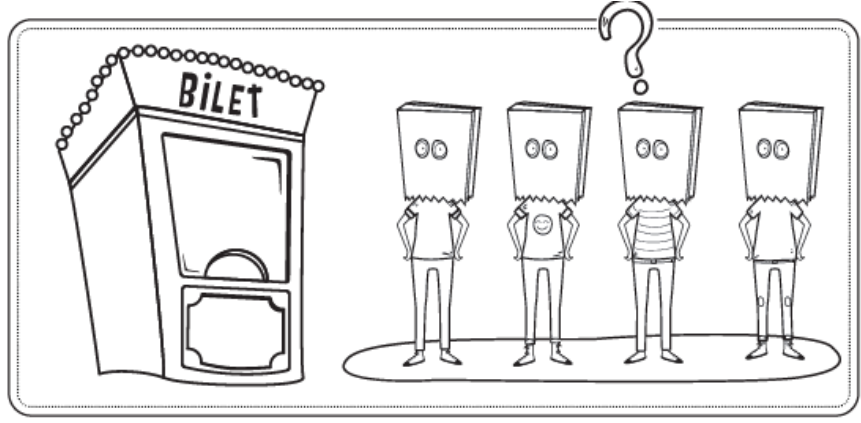


TORTOP'UN EĞER-YOKSA YAŞAMI

Tortop'un hava durumuna bağlı hareketleri akış şeması	Tortop'un ılık ve bulutlu bir günde yaptıkları akış şeması
<p style="text-align: center;">Başla</p>	<p style="text-align: center;">Başla</p>

EYVAH AKIŞ ŞEMALARI KARIŞMIŞ

BİLET KUYRUĞU



Dört arkadaş: Aylin, Efe, Nazan ve Hakan bir filme bilet almak için sıraya girdiler. Bilet kuyruğu ile ilgili bazı bilgiler aşağıda verilmiştir:

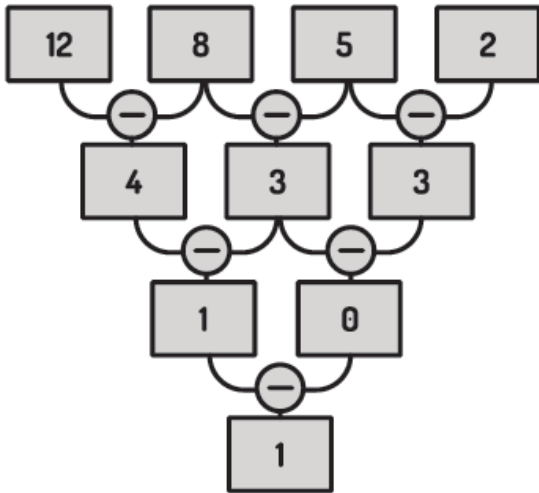
- 1) Hakan sıranın başında değil.
- 2) Nazan Efe'nin hemen önünde.
- 3) Aylin sırada Hakan'ın arkasında.

Soru; Yukarıdaki bilgiler göz önünde bulundurulduğunda, sırada üçüncü olan kimdir?

- A. Efe B. Nazan C. Aylin D. Hakan

PİRAMİT SATIRLARI

İşlem makinesi ilk satırdaki 4 rakamı girdi olarak almaktadır. Her satırda makine sayılar arasındaki farkı hesaplamaktadır. Aşağıdaki resimde örnek bir işlem verilmiştir.



Soru: Aşağıdaki girdi rakamlarından hangisi son satırda sonucun "0" (sıfır) olmasını sağlar?

- A. 13 9 7 6
- B. 16 9 4 1
- C. 13 8 4 2
- D. 5 5 5 1



DOĞRULUK



Bilge kunduz sadece Pazartesi, Çarşamba ve Cuma günleri doğruları söylerken diğer günlerde her zaman yanlış bilgi vermektedir. Bugün için "Yarın, doğruları söyleyeceğim" demiştir.

Buna göre bugün günlerden hangi gündür?

A) Salı

B) Cuma

C) Cumartesi

D) Pazar



ALIŞVERİŞ

Çınar, arkadaşı Bulut'u ziyaret eder. Arkadaşına hediye olarak bir parça kunduz peyniri almıştır. Bulut hediye için ücretini öğrenmesini diye fişin bir kısmını koparmıştır. Ama yine de Bulut ücreti öğrenebilmektedir. Fişin şekilde gösterilen hangi kısımları ile peynire yapılan ödeme miktarı hesaplanabilir ?

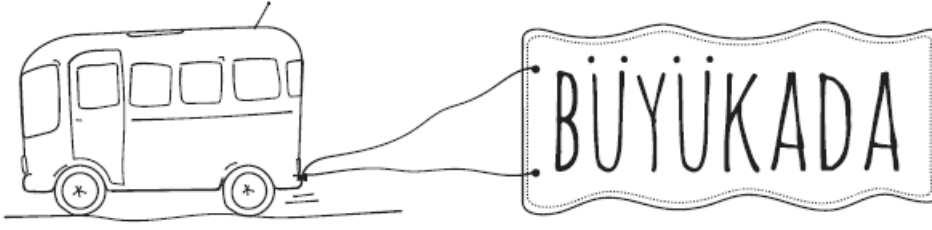
A) A

B) A ve B

C) C

D) C ve D





Büyükada'da yaşayan Bilge Kunduz ulaşım için otobüs kullanmaktadır. Ağaç, Orman ve Akarsu bölgelerine giden otobüslerin numaraları aşağıda verilmiştir. Otobüs numaraları bölge adında yer alan karakterlere göre belirlenmektedir.

Ağaç 9397
Orman 12496
Akarsu 989205

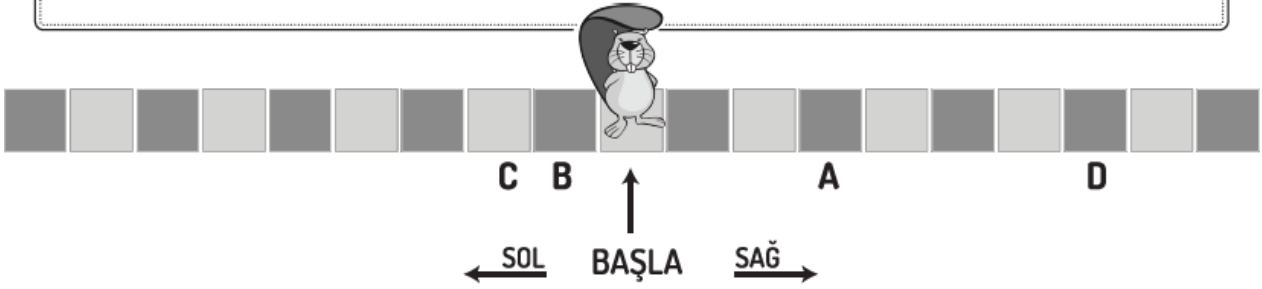
"Çamur" bölgesine gidebilmek için Bilge Kunduz'un hangi numaralı otobüse binmesi gerekir?

- A. 79452 B. 78524 C. 60983 D. 69431

ZIP ZIP
BİLGE
KUNDUZ

Bilge Kunduz farklı şekillerde hareket ediyor. Hareket etmeye tam ortadan başlayacak. Sırayla sağa ve sola olmak üzere toplamda beş hamle yapacak: Önce sağa, sonra sola, sonra sağa, sonra sola ve sonunda tekrar sağa...

Bilge Kunduz her seferinde 1, 2, 3, 4 veya 5 kare ilerleyebilir. İlerleyeceği kare sayısını yalnızca bir kez kullanabilir. Örneğin, 2 kare sağa, 1 kare sola, 5 kare sağa, 4 kare sola ve 3 kare sağa gidebilir. Sonunda başladığı kareden $2 - 1 + 5 - 4 + 3 = 5$ kare sağa ilerlemiş olacaktır.



Soru; Belirtilen hareket serisinden sonra Bilge Kunduz'un erişebileceği dört kare A, B, C ve D harfleriyle işaretlenmiştir. Bunlardan biri yanlışdır. Buna göre Bilge Kunduz hangi kareye ilerleyemez?

- A. A konumu B. B konumu C. C konumu D. D konumu

Studio.code.org

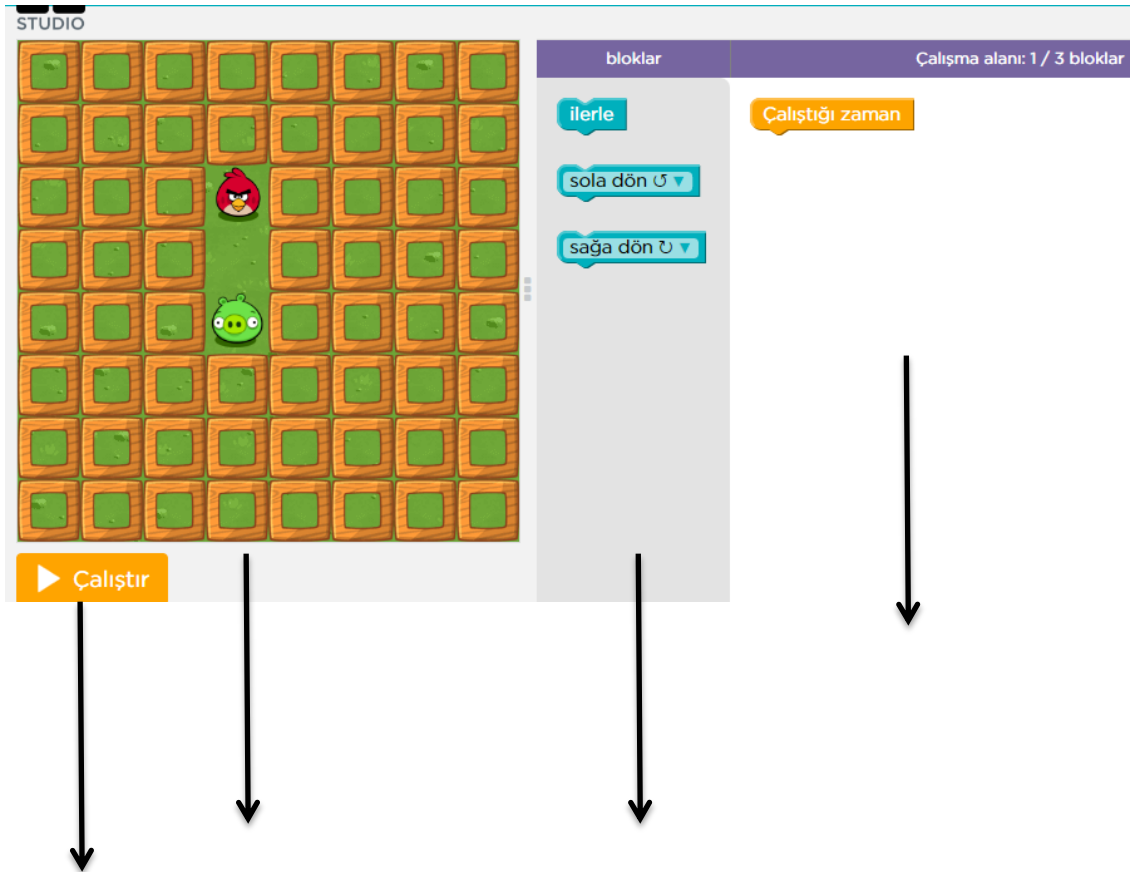
Kitabın bu bölümünde yukarıdaki sitede dersimizi işleyeceğiz. Site size oyun gibi gelebilir hatta bu dersten sonra işleyeceklerimiz her ders size oyun gibi gelebilir. İlerleyen derslerde kendi oyunlarımızı tasarlamak için bir adım atıyoruz çocuklar. Bu sitede sizinle beraber sizin çok iyi bildiğiniz bazı oyun karakterleri üzerinden problemler çözeceğiz. Örneğin aşağıdaki resimde örnek bir problem var. Gördüğünüz gibi karakterlerimiz Angry oyunundan alınmış.

Gelin problemi öğretmenimizden dinleyelim ve beraber çözelim.

Siteye öğretmeninizin site vereceği adresten kendi isminize tıklayarak yine öğretmeninizin vereceği gizli kodunuz ile giriş yapacaksınız. Gizli kod ile giriş yapmadığınız takdirde yaptığınız çalışmalar kayıt altına alınmayacak ve öğretmeniniz çalışmalarınızı takip edemeyecektir. Adresi ve gizli kodunuzu aşağıdaki bölüme not almayı unutmayın.

Adres : studio.code.org/sections/.....

Gizli Kod :



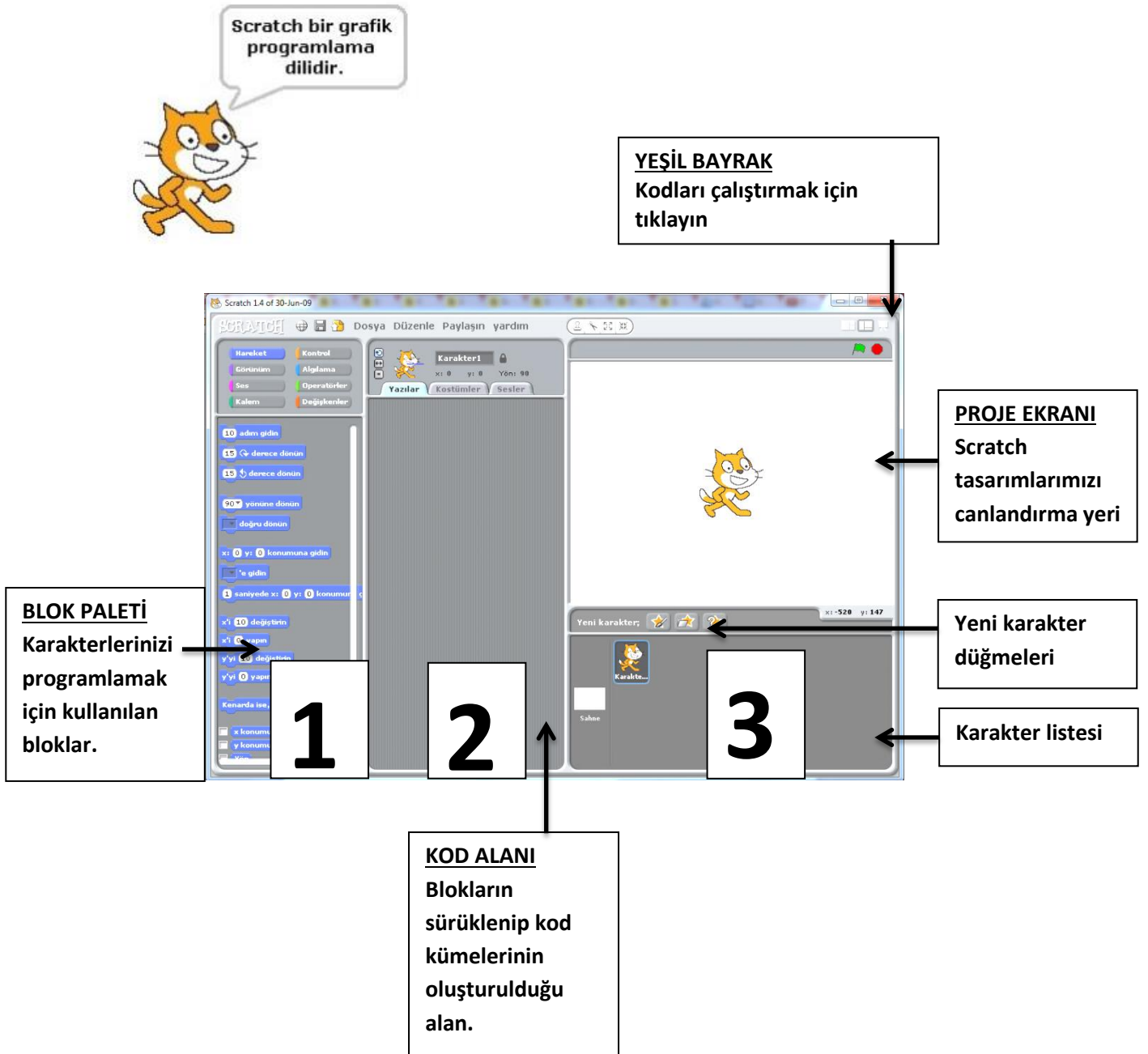
1. Hafta

Scratch bir grafik programlama dilidir.

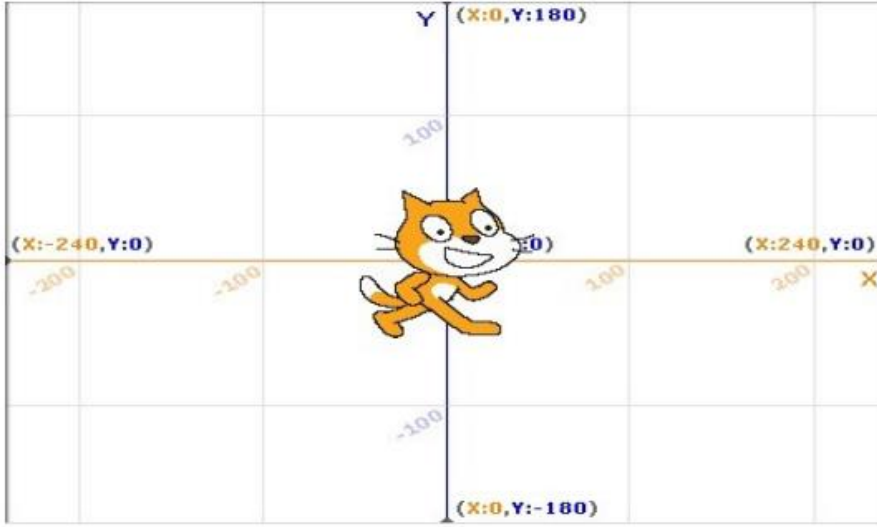
Scratch ile resim, ses, müzik gibi çeşitli medya araçlarını bir araya getirebilir, kendi animasyonlarımızı, bilgisayar oyunlarımızı tasarlayabilir ya da interaktif hikayeler anlatabiliriz.

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

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

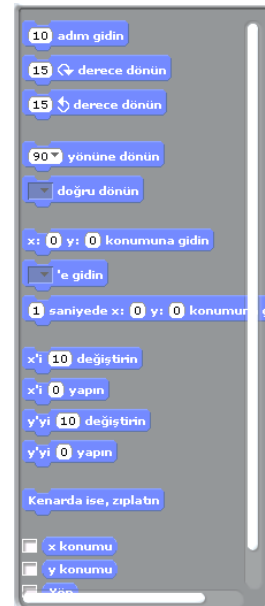


Scratch sahnesi aslında bir koordinat düzlemidir. Bütün animasyonlarımız ve bilgisayar oyunlarımız bu koordinat düzlemi üzerinde tasarlanır. Scratch karakteri olan kedi program açıldığı zaman (0,0) noktasındadır. Ekranın üstünde karakterlerimizin x ve y değerlerini görebiliriz.



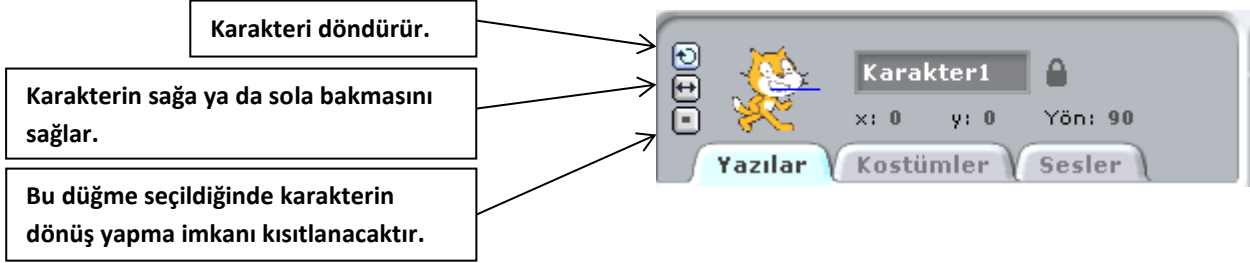
1. Kod Bölümü

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



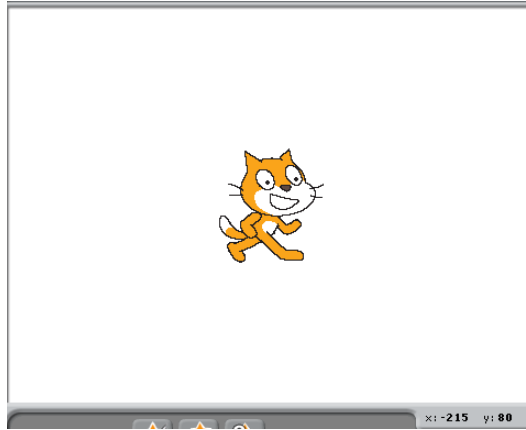
2. Kodlama Paneli

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



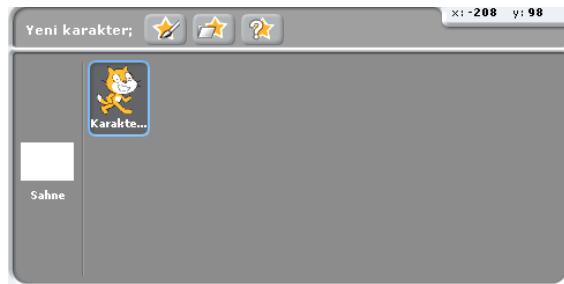
3. Sahne ve Karakterler

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

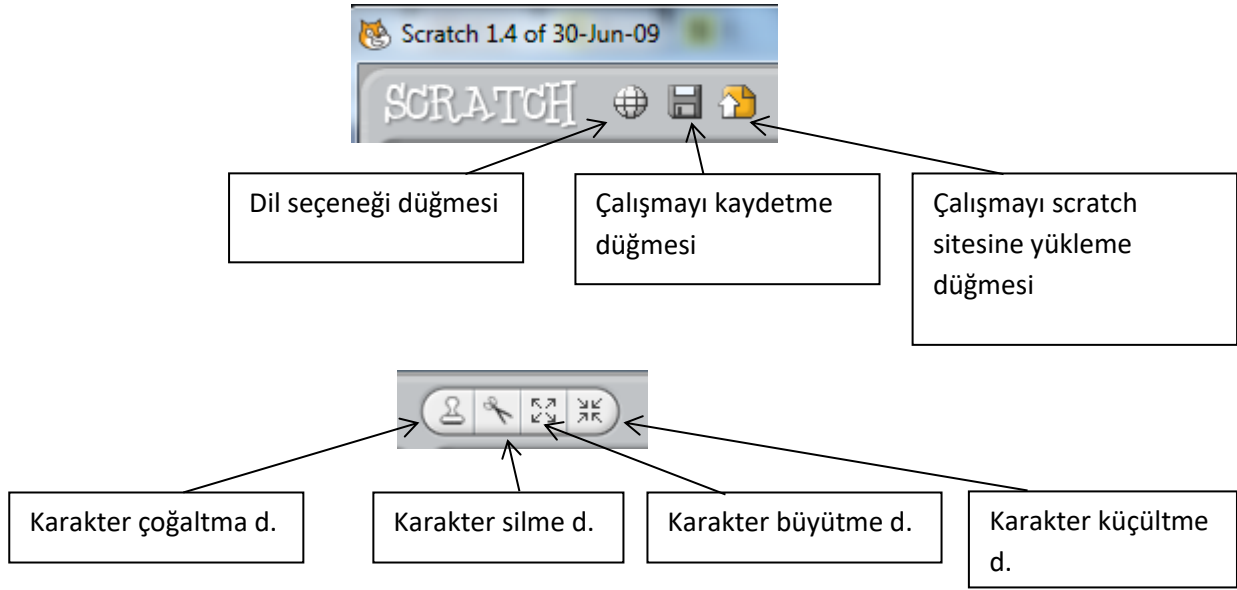


Sahnenin sağ üst köşesindeki yeşil bayrak programı çalıştırmanızı sağlar. Kod bloğunda yazmış olduğunuz komutlar çalışmaya başlar. Yine sahnenin sağ üst köşesinde bulunan kırmızı yuvarlak ise çalıştırmış olduğunuz bir programı durdurmanızı sağlar.

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



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



1. Papağan Uygulaması

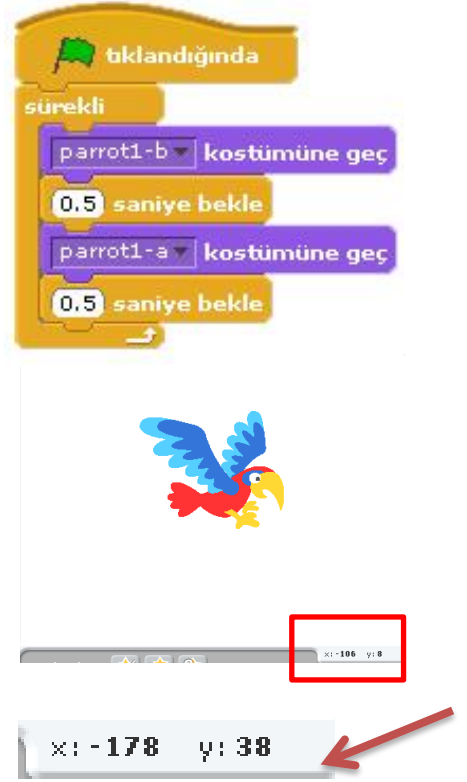
1. Yeni karakter ekle butonundan Animals klasöründen papağan “papağan1-a ” karakterini sahnemize ekliyoruz.
2. Sahnemizde olan kedi karakterini sağ tıklayıp sahneden siliyoruz.
3. Papağan karakterimize uçuş görünümünü vermek için papağanımızın 2. bir resmi olan “papağan1-b” yi papağanımız seçili iken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



5. Papağanımızın sürekli uçuşmasını istediğimiz için süreklilik içeren bir hareketlilik için kontrol kod bloğundan “sürekli” kodunu aşağıdaki gibi ekliyoruz.
6. Resimde de görüldüğü üzere “**sürekli**” kodunun içi boş, sürekli ne olacağını bu kısma ekleyeceğiz. Hatırlarsanız 2 papağan resmimiz vardı kanat çırpın. Bu iki görünüm arasında geçiş yapacağımız için “**Görünüm**” kod bloğundan “..... **kostümünü seçin**” kodunu sürekli kodunun içine alıyoruz. Boşluk kısmında gördüğümüz gibi papağan1-a yazıyor. İlk görünümümüz bu sonraki görünümün papağan1-b olmasını istiyoruz. Bunun için son eklediğimiz kodundan bir tane daha altına ekliyoruz.



7. Şimdi sahnemizi yeşil bayrak butonuna basarak çalıştırıp baktığımızda papağanımızın çok hızlı olarak kanat çırdığını göreceğiz. Bunu belli bir süre aralığında yapabilmesi için papağanın iki görünümünden sonrası için süre ekleyeceğimiz bunun için kontrol kod bloğundan “**1 saniye bekleyin**” kodunu resimdeki gibi ekliyoruz. Dilersek beyaz alanlarda yazan 1 sayısını tıklayarak değiştirebiliriz.
8. Yeşil bayrak butonuna basıp sahnemizi izlediğimizde görüyoruz ki papağanımız durduğu yerde kanat çırdırabiliz. Sahnemizde papağanımızın uçuşa nereden başlamasını istiyorsak oraya taşıyoruz. Bunun için papağanımız seçili iken kontrol bloğundan yeni bir “**yeşil bayrak tıkladığında**” komutu ekliyoruz. Papağanımızı sahnede hareket ettirirken sahnede bulunduğu yeri belirten x ve y koordinatları değişmektedir. Papağanımız uçarak nereye gitmesini istiyorsak o konumun x ve y koordinatlarını bilmeliyiz. Bu koordinatlar faremiz sahnede hareket ederken sahnenin sağ alt köşesinde gösterilmektedir.



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



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

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

Haftanın Özeti

Bu çalışma için kullandığımız kod blokları: Kontrol, Hareket ve Görünüm kod bloklarını kullandık.

Kontrol Kod Bloğundan kullandığımız komutlar;

- “Yeşil bayrak tıkladığında” : Programınızı başlatınca yapılmasını istediğiniz olayları bu kodun altında tanımlarız.
- “1 saniye bekleyin” : Bir sonraki koddan önce beklenilecek süreyi belirleyen komuttur.
- “sürekli” : Bu kod içerisine yerleştirdiğimiz komutları sonsuz bir döngüde tekrar tekrar çalıştıracaktır.

Hareket Kod Bloğundan kullandığımız komutlar;

- “x: y: Konumuna gidin” : Bu komut genellikle karakterlerimizin sahnedeki başlangıç yerlerini belirlemek için kullanılır.
- “1 saniyede x: y: Konumuna gidin” : Belirlediğimiz süre boyunca, karakterimizin bulunduğu noktadan belirttiğimiz koordinatlara gitmesini sağlayan koddur.

Görünüm Kod Bloğundan kullandığımız komutlar;

- “..... kostümüne geçin” : Karakterimizin tanımlanmış diğer kostümler arasında geçiş yapmasını sağlar.

ÖDEV: Bu hafta hazırlamış olduğumuz sahneye bir insan karakteri koyarak bu karakteri belli bir süre içerisinde yürütün.

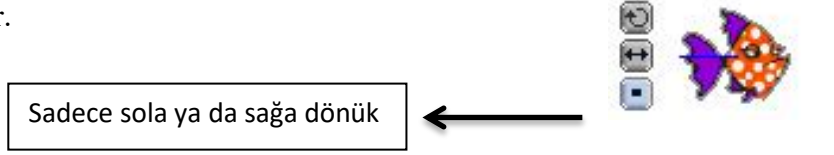
2. Akvaryum Uygulaması

1. Akvaryum uygulaması için öncelikle sahnemize 3 tane balık ekliyoruz. Balıkların boyutlarını ve sahnede bulunacakları konumları isteğimize göre düzenliyoruz.
2. Balıklarımızdan biri seçili iken kontrol kod bloğundan “yeşil bayrak tıkladığında” kodunu alıyoruz. Balıkların sürekli tekrar eden bir şekilde hareket etmesini istediğimiz için altına yine kontrol kod bloğundan sürekli kodunu ekliyoruz.
3. Sürekli kodunun içine hareket kod bloğundan “10 adım gidin” kodunu ekliyoruz. Bu kod balığın hareket etmesini sağlayacaktır. 10 sayısı tıkladığında değiştirilebilir.



4. Çalışmamızı bu haliyle çalıştırdığımızda balığımız sahne kenarına kadar gidip duracaktır. Balığımızın sahne için sağa sola doğru hareket etmesini istediğimiz için sürekli kodunun içine hareket kod bloğundan “kenarda ise zıplattın” kodunu ekleriz. Bu kod balığımızın ters yöne hareket etmesini sağlayacaktır.

5. Çalışmayı şimdi çalıştırdığımızda balığımız kenar geldiğinde ters yöne doğru hareket ettiğini göreceğiz yalnız balığımız başını çevirmeden hareket etmektedir. Balığımızın yönünü de hareket yönüne doğru dönmesi için karakter özelliklerinin olduğu yerde “sadece sola ya da sağa dönük” butonunun aktif ederse balığımız dönecektir.



6. Bütün balıklarımızın hareket etmesini istediğimiz için bu kod kümesini diğer balıklar için kopyalamamız gerekmektedir. Yeşil bayrak tıkladığında kod başlığında sağ tıklayıp kopyala ile kopyalama işlemini yaptıktan sonra bu kod kümesini diğer balıklardan birinin üzerine sürükleyerek kopyalamış oluruz. Ya da her balık için aynı kod kümesini tekrardan oluşturmak gerekmektedir.

7. Kod kümesini kopyaladıktan sonra her balığın yüzme hızı aynı olmadığından “10 adım gidin” kodundaki sayıyı her balık için farklı yapalım.

8. Balıklarımız artık hareket halinde olacaktır. Sahnenin arka planını akvaryum görünümünü yapıp sahneye başka balıklarda ekleyebilirsiniz.

9. Akvaryumumuza ses efekti ekleyelim şimdide. Bunun için sahnemiz seçili iken kontrol kod bloğundan “yeşil bayrak tıkladığında” kodunu alıyoruz. Ve sesin sahne içinde sürekli çalmasını istediğimizden altına yine kontrol kod bloğundan “sürekli” kodunu ekliyoruz.



10. Sürekli kodunun içine **ses kod bloğundan** sesimizin sürekli çalması için “**pop sesini bitene kadar çal**” kodunu ekliyoruz. Dikkat ederseniz pop yazısının yanında bir ok işareti vardır bu ok işaretinden sesi değiştirebiliriz. Bu sesi değiştirmek için öncelikle kodlarımızı eklediğimiz yazı bölümünde sesler sekmesine tıklıyoruz.

11. Sesler sekmesinde içeri aktar butonuyla efektlerden “bubbles” sesini ekliyoruz ve kodda bu pop yerine

bubbles sesini seçiyoruz.



Sahnemize bir tane ağız kapalı bir köpek balığı ekleyelim. Köpek balığının kodları farklı olacağından kodlarını biz yazacağız. “yeşil bayrak tıkladığında” komutunu köpek balığımız seçili iken alıyoruz. Altında sürekli kodunu ekliyoruz. sürekli kodunun içine bu sefer hareket kod bloğundan “... **doğru dönün**” kodunu ekliyoruz.

12. “... doğru dönün” kodunda boş olan bölüme tıkladığımızda açılan seçeneklerden “Mouse işaretçisi”ni seçiyoruz. Yapmaya çalıştığımız şey şu, bu kodla köpek balığımız faremizin yönüne dönmesini sağlar. Faremizi takip etmesi içinde hareket kod bloğundan “10 adım gidin” kodunu sürekli kodunun içine ekleyerek köpek balığımızı hareket ettirmiş oluyoruz.



Haftanın Özeti

Bu çalışma için kullandığımız kod blokları: Kontrol, Hareket ve Ses kod bloklarını kullandık.

Kontrol Kod Bloğundan kullandığımız komutlar;

- “Yeşil bayrak tıkladığında” : Programımızı başlatınca yapılmasını istediğiniz olayları bu kodun altında tanımlarız.
- “sürekli” : Bu kod içerisine yerleştirdiğimiz komutları sonsuz bir döngüde tekrar tekrar çalıştıracaktır.

Hareket Kod Bloğundan kullandığımız komutlar;

- “10 adım gidin” : Karakterimizi koordinat sisteminde x ekseninde istenilen birim değeri kadar ilerletir.
- “Kenarda ise zıplatın” : Karakterimizin sahne sınırları içerisinde sağa ve sola doğru hareketini sağlar.
- “.... doğru dönün” : Karakterimizin sahnedeki diğer karakterlerden herhangi birine ya da mouse işaretçisine doğru dönmesini sağlar.

Ses Kod Bloğundan kullandığımız komutlar;

- “..... sesini bitene kadar çalın” : Seçilen bir ses dosyası içeriğini, sonuna kadar çalmak için kullanılan komuttur.

ÖDEV: Arkaplanı akvaryum olan bir sahneye dalgıç karakteri ekleyelim ve bu dalgıç fare hareketiyle sahnede yüzsün. Sahnemiz de bir de köpek balığımız olsun. Köpek balığımızda akvaryum içinde kendi kendine sağa sola gidip hareket etsin sürekli. (dalgıç karakteri scratch programının içindeki karakterlerde vardır.)

NOT: Yukardaki ödevin yerine bu haftaya kadar öğrendiğimiz kodları kullanarak kendiniz farklı bir projede hazırlayabilirsiniz.

3. Yön Tuşları Uygulaması

1. Sahnemizde kedi karakterimiz seçili iken kontrol kod bloğundan “boşluk tuşuna basıldığında” kodunu alıyoruz. Bu kodun boşluk kısmına tıklayıp “yukarı ok” olarak değiştiriyoruz. Bu yukarı klavyede yön tuşu olan yukarı okuna basıldığında anlamına gelmektedir.

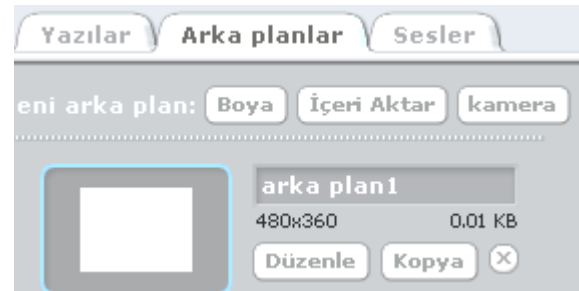


2. Bu kodun altına yukarı yön tuşuna basıldığında ne olmasını istediğimizi belirten kodları ekleyeceğiz. Karakterimize hareket vereceğimiz için hareket kod bloğundan **“90 yönüne dönün”** kodunu ekliyoruz ve beyaz bölümde yazan 90 sayısını tıklayarak yukarıyı ifade eden (0) ı seçiyoruz karakterimiz böylece yukarıya döner.
3. Yukarıya dönen karakterimizi hareket ettirmek için yine hareket kod bloğundan **“10 adım gidin”** kodunu altına ekliyoruz. Aynı kod kümesinden 4 tane olacak şekilde çoğaltıyoruz. Her bir kontrol kümesi yön tuşlarından(aşağı, sağa, sola) kullanılacak şekilde ayarlamalar yapıyor.
4. Şimdi kedimize yürüyormuş gibi bir görünüm vermek için kedimizin kostümlerini kullanacağız. Hatırlarsak karakter özelliklerinde kostümler bölümü vardı burada karakterimize ait iki tane yürüyüş resmi var. Şimdi kod bölümümüze kontrol kod bloğundan **“yeşil bayrak tıkladığında”** kodunu alıyoruz, bunun altına yine kontrol kod bloğundan **“sürekli”** kodunu ekliyoruz. Sürekli kodunun içine kostümler arası geçiş için görünüm kod bloğundan **“kostüm2 kostümüne geçin”** kodunu iki defa ekliyoruz ve birinin kostüm2 bölümünü kostüm1 olarak değiştiriyoruz. Kostümler arası geçişi görebilmemiz içinde aralarına kontrol kod bloğundan **“1 saniye bekleyin”** kodunu ekliyoruz.
5. Artık kedimiz yön tuşları ile adım atarak yürüyecektir.



4. Labirent Uygulaması

1. Yön tuşları uygulaması labirent uygulaması için ön hazırlıktı. Öncelikle kendimize yeni karakter çizin butonundan ufak bir yuvarlak karakter oluşturuyoruz. Karakterimize piyon adını verip sahnemizde oyunumuzu başlatacağımız yere koyuyoruz.



2. Sahnemizin arka planında oyunumuz için engeller oluşturacağız bunun için, sahnemiz seçili iken arka plan özelliklerinden **“Boya”** butonuna basarak arka plana engeller çiziyoruz. Bu engellerin hepsini aynı renkte yapalım. Sahnemizde birde çıkış kapımız olacak çıkış kapısını farklı bir renkte seçelim.

3. Sahnemizi hazır hale getirdikten sonra tekrar piyonumuzu seçip, komutlarımızı yazmaya başlıyoruz. Oyunun otomatik başlaması için kontrol kod bloğundan **“yeşil bayrak tıkladığında”** kodunu alıyoruz. Bunun altına kontrol kod bloğundan **“Eğer..... ise sürekli”** kodunu bir önceki kodun altına ekliyoruz.



4. Şimdide piyonumuzun sahnemizdeki engelleri algılayabilmesi için en son eklemiş olduğumuz **“Eğer İse sürekli”** kodunun boş kısmına algılama



bloğundan “**rengine yaklaşıyor mu?**” kodunu resimdeki gibi ekliyoruz.

5. Son eklediğimiz kodun içindeki renk bölümüne tıkladığımızda imlecimiz damlalık şeklini alacaktır. İmlecimiz damlalık şeklindeyken sahnemizde engellerden birine tıklıyoruz. Böylece kodumuzun içindeki renk engellerimizin rengine dönüşmüş oldu.



6. Sonraki adımda piyonumuz engellerimize çarpınca ne olacağını gösteren kodları eklemek olacak. Bunun için “ise sürekli” kodu içindeki boşluğa hareket bloğundan “**15 derece dönün**” ve “**10 adım gidin**” kodlarını peş peşe ekliyoruz. Bu kodlar ile piyonumuz engellere çarptığında engelin üzerinden geçmeden engelden 10 adım ters yönde uzaklaşacaktır.

7. Şimdide piyonumuz çıkış kapısına geldiği zaman oluşacak durumun kodlarını oluşturacağız. Piyonumuz seçili iken kontrol bloğundan “yeşil bayrak tıkladığında” kodunu alıyoruz. Bu kodun altına yine kontrol bloğundan “..... **olana kadar bekleyin**” kodunu ekliyoruz.



8. “..... **olana kadar bekleyin**” kodunda boş olan yere algılama bloğundan “**rengine değişiyor mu?**” kodunu ekliyoruz ve rengi çıkış kapısının rengi yapıyoruz. Oyunumuz bittiği zaman çıkış kapısına piyonumuz gelince “oyunu kazandın” diye bize mesaj vermesini istiyoruz. Bu mesaj için kodumuzun devamına görünüm bloğundan “söyleyin Merhaba” kodunu alıyoruz.

Yeşil bayrağa tıklayıp çalışmayı başlattığımızda oyunumuzun çalışıp çalışmadığı hakkında öğretmenimize geri dönüt verin eğer oyunumuz çalışmıyorsa nerde hata yaptığımızı ne yapmamız gerektiği konusunda öğretmenimizle fikir alışverişinde bulunalım! Oluşan sorunu beraber çözelim!

Haftanın Özeti

Bu çalışma için kullandığımız kod blokları: Kontrol, Hareket ve Görünüm, Algılama kod bloklarını kullandık.

Kontrol Kod Bloğundan kullandığımız komutlar;

- “**Yeşil bayrak tıkladığında**” : Programınızı başlatınca yapılmasını istediğiniz olayları bu kodun altında tanımlarız.
- “**sürekli**” : Bu kod içerisine yerleştirdiğimiz komutları sonsuz bir döngüde tekrar tekrar çalıştıracaktır.
- “**..... tuşuna basıldığında**”: Bu kod, karakterin veya bir olayın başlaması için gereken komutun istenilen tuşa basılması gerektiğini işaret eder.
- “**Eğer..... ise sürekli**” : Bir koşula bağlı olarak kontrol grubuna ait komutların sürekli olarak çalıştırılmasını sağlar.

- “..... olana kadar bekleyin”: bu kodun altına yerleştirdiğimiz tüm kodların çalışması için bir koşulun gerçekleşmesini bekler.

Hareket Kod Bloğundan kullandığımız komutlar;

- “10 adım gidin” :Karakterimizi koordinat sisteminde x ekseninde istenilen birim değeri kadar ilerletir.
- “90 yönüne dönün” : Karakterimize istenilen yöne dönüş yaptırmayı sağlayan koddur.
- “15 derece dönün” : Karaktere ok yönünde dönüş yaptırmayı sağlar.

Görünüm Kod Bloğundan kullandığımız komutlar;

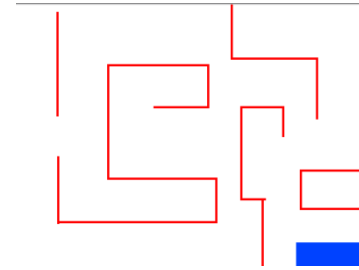
- “..... kostümüne geçin” : Karakterimizin tanımlanmış diğer kostümler arasında geçiş yapmasını sağlar.

Algılama Kod Bloğundan kullandığımız komutlar;

- “rengine değişiyor mu?” : Karakterimizi sahnedeki renklere duyarlı hale getiren komuttur.

5. Pacman Oyunu

1. Öncelikle sahnemizdeki kedi karakterimizi sahnemizden silelim. Ardından kendi pacman karakterimizi yeni karakter oluştur butonundan oluşturalım, pacman oyunumuzda karakterimizi canavarları yiyeceği için pacman karakterimizi bir ağız kapalı bir ağız açık 2 tane kostüm oluşturalım.
2. Oyunumuz için gerekli labirent çizgilerini çizelim yandaki şekilde görüldüğü gibi çizelim.
3. Scratch in kendi içindeki fantastik karakterler bölümünden sahnemize 3 tane farklı karakter ekleyelim ve sahnemizde istediğimiz yerlere bu karakterleri yerleştirelim.
4. Canavarlarımız sahnede sürekli sağa sola hareket halinde
5. sahnede dolaşacaklar. Bunun için gerekli komutlarımızı yazalım.
6. Canavarlarımızdan biri seçili iken “yeşil bayrak tıkladığında” komutunu alıyoruz. Bu komutun altına sürekli aynı hareketi yapacaklarından dolayı “sürekli” döngüsü ekleyelim. Sürekli döngüsünün içine yürüme hareketi için “10 adım gidin” komutunu ekleyelim.



7. Canavarlarımızın kırmızı duvar çizgilerine kadar gidip tekrar geri dönerek yürümesi için kontrol kod bloğundan “eğer İse sürekli” komutunu sürekli döngüsünün içine ekleyelim. Bu komut içindeki boş kalan alana algılama bloğundan “.... Rengine



değişiyor mu?” komutunu ekleyerek renk kısmını sahnemizdeki duvar rengi yapalım. Geçen haftadan hatırlayacağınız üzerine basit pacman oyunumuzda duvara çarpan karakterimiz geri dönüyordu. Aynı komutları burada da kullanalım. Kırmızı zemine çarpan karakterimiz geri dönsün. Eğer komutunun içine **“... derece dönün”** ve **“10 adım gidin”** komutlarını ekleyelim.



8. Canavarlar pacman karakteri yaklaştığında sahneden kaybolması gerekiyor. Bunun için canavarımızın seçili iken yeşil bayrak tıkladığında komutunu yeniden alalım. Oyunumuzu her başlattığımızda canavarımızı sahneye getirmesi için görünüm kod bloğundan **“Göster”** komutumuzu ekleyelim. Ardından pacman karakterimiz yaklaşıp olana kadar beklemesi için kontrol kod bloğundan **“... olana kadar bekleyin”** komutunu alıyoruz. Bu komutun boş kalan kısmına algılama bloğundan **“... değişiyor mu?”** komutunu resimdeki gibi koyalım. Pacman karakteri canavara yaklaştığı zaman canavarın sahneden kaybolması için görünüm kod bloğundan **“Gizle”** komutunu ekleyelim.



ÖDEV: En son yazdığımız kod bloğundan “Göster” komutunu neden kullandık, kullanmasaydık ne olurdu? Açıklayınız.

9. Bir canavar karakteri için oluşturduğumuz kod bloklarını diğer canavar karakterleri için kopyalayalım.
10. Ardından pacman karakterimizi seçili iken karakterimizi sahnede yön tuşları ile hareket etmesini sağlayan komutlarımızı ve engellere geldiğinde engellerin üzerinden geçmemesi için gereken komutlarımızı oluşturalım.(bu komutlar geçen hafta işlenmişti.)
11. Bugüne kadar oynadığımız oyunlardan bildiğimiz üzere oyunlarda oyun puanını gösteren skor vardır. Biz puanımızı pacman karakterimiz canavar karakterimize yaklaşıp onları yediği zaman kazanacağız. Bunun için canavarlarımızdan birinin kod bölümünü açalım. Buraya oyun içeriğinde değişecek bir değişken olarak skor ekleyeceğiz. Bunun için “değişkenler kod bloğu”nu açalım ve değişken oluşturun butonuna tıklayıp açılan bölüme skor yazalım. Bunu yazdığımızda göreceğiz ki değişkenler bloğunda skor değişkenine ait komutlar eklenmiş. Ve sahnemize skor adından bir görüntü eklemiştir. Değişkenler bloğundan “skoru 1 kadar değiştirin” komutunu resimde görülen komutların altına ekleyelim. Bu komutu her canavar için gerekli yere ekleyelim.



Kullanılacak Komutlar

Bu çalışma için kullandığımız kod blokları: Kontrol, Hareket ve Görünüm, Algılama kod bloklarını kullanacağız.

Kontrol Kod Bloğundan kullandığımız komutlar;

- “..... tuşuna basıldığında” : Bu kod, karakterin veya bir olayın başlaması için gereken komutun istenilen tuşa basılması gerektiğini işaret eder.
- “..... ise sürekli” : Bir koşula bağlı olarak kontrol grubuna ait komutların sürekli olarak çalıştırılmasını sağlar.
- “..... olana kadar bekleyin” : bu kodun altına yerleştirdiğimiz tüm kodların çalışması için bir koşulun gerçekleşmesini bekler.

Hareket Kod Bloğundan kullandığımız komutlar:

- “15 derece dönün” : Karaktere ok yönünde dönüş yaptırmayı sağlar.

Görünüm Kod Bloğundan kullandığımız komutlar:

- “Merhaba diye konuş” : İstenilen durumda karakterimizi konuşma baloncuğu ile konuşturma yapıyor.
- “Göster” : Karakterin sahnede görünmesini sağlayan komuttur.
- “Gizle” : Karakteri sahnede gizleyen komuttur.

Algılama Kod Bloğundan kullandığımız komutlar:

- “..... yaklaşıyor mu?” : Karakterimizi sahnedeki diğer karakterlere duyarlı hale getiren komuttur.
- “... rengine yaklaşıyor mu?” : Karakterimizi sahnedeki renklere duyarlı hale getiren komuttur.

ÖDEV: Benzer bir labirent tasarlayalım ve bu sefer labirentte 5 tane canavarımız olsun ve pacman karakterimizi engel olan duvarlara her çarptığında oyunun başlangıç noktasına geri dönsün.

6. Rasgele Sayılar Uygulaması

1. Sahnemizde bir tane karakterimiz olsun. Bu karaktere belli sayılar arasında rasgele bir sayı söylemesi için kod oluşturalım. Bunun için karakterimiz seçili iken “yeşil bayrak tıkladığında” komutunu alıyoruz. Bu kodun altına görünüm bloğundan “söyle merhaba süre 2sn” komutunu ekleyelim. Bu komut ile 2sn boyunca merhaba dedirtiyoruz kendimize.
2. Hatırlarsak en son eklediğimiz komuttaki beyaz alanda yazan Merhaba yazısını biz değiştirebiliyorduk. Bu beyaz alana **operatörler** bloğundan “1 ile 10 arasında rastgele seçin” komutunu sürükleyerek yanda görüldüğü gibi ekleyelim.
3. Projemizi çalıştırdığımızda göreceğiz ki karakterimiz 2sn süre ile belirlediğimiz sayılar arasında rasgele bir sayı söylüyor.
4. Bu sayıların 2 sn bir değişmesi için kodumuzda nasıl bir değişiklik yapmamız gerektiğini tartışalım ve boşluğa yazalım.



.....
.....

5. Şimdide kedimizi sahnemizde hareket ettirelim hareket ederken de sahnede iz bırakmasını sağlayalım. Bunun için ikinci bir yeşil bayrak tıkladığında komutu ekliyoruz. Ardından projeyi her çalıştırdığımızda sahnemizin temizlenmesi için **kalem** bloğundan **Kalemle çizilenleri temizle** komutunu ekliyoruz.



6. Sonrasında karakterimizin hareketlerde süreklilik olması için sürekli komutu ekliyoruz.

7. Sürekli komutunun içine hareket bloğundan 10 adım gidin komutunu ekleyelim. Karakterimiz 10 adım ile ilerleme yapacaktır. Biz bu adımın rasgele bir sayı olmasını istiyoruz. O zaman operatörler bloğundan “1 ile 10 arası rasgele üret” komutunu önceki kodun beyaz alanına resimdeki gibi ekleyelim.



8. Karakterimizi hareket ettirdiğimizde göreceğiz ki karakter sahnenin kenarına kadar gidiyor ve orada kalıyor. Peki bu karakterin sahnenin kenarına gittiğinde geri dönüp sahnenin diğer yönüne doğru yürümesini istiyoruz bunun için hangi komutu eklememiz gerekiyor tartışalım ve aşağıdaki boşluğa yazalım.

.....
.....

9. Gerekli olan komutu ekledikten sonra karakterimiz sahnemizde bir sağa bir sola sürekli hareket etmekte ve her 2sn de bir rasgele bir sayı söylemektedir.

10. Karakterimize sahnede iz bırakma komutları verelim. Bunun için kalem bloğundan “**Damgala**” komutunu ekleyelim ve çalıştıralım.



11. Sahnede bıraktığı iz farklı renklerde olsun istiyorsak son bir komut ekleyelim. Görünüm bloğundan “**renk efektini 25 ile değiştirin**” komutunu ekleyelim.

Haftanın Özeti

Bu çalışma için kullandığımız yeni kodlarımız: Operatörler, Kalem, Görünüm kod bloklarını kullanacağız.

Operatörler Kod Bloğundan kullandığımız yeni komutlar;

- “1 ile 10 arasında rasgele üret” : Belli aralıktaki sayı kümesi içinden rasgele bir sayı seçilmesini sağlar.

Kalem Kod Bloğundan kullandığımız komutlar;

- “**damgala**” : Bu komut karakterimizin sahnede resminin çıkmasını sağlamaktadır.
- “**Kalemle Çizilenleri temizle**” :Kalem koduyla yaptığımız tüm işlerin sahnemizden temizlemek için kullanılır.

Görünüm Kod Bloğundan kullandığımız komutlar;

- “**renk efektini 25 ile değiştirin**” : Karakterimizin renk değerlerini belirlediğimiz sayı kadar değiştirir.

7. Sahne Değiştirme ve Soru Cevap Uygulaması

1. Bu uygulamada sahne geçişleri yapacağımız için öncelik kendimize 2 tane sahne oluşturalım. Bunun için sahneyi seçip arkaplan bölümünden var olan sahnemize düzenle diyerek sahnemizin rengini değiştirelim ve yapacağımız soru cevap uygulamasına hazırlık olarak sahnemize “İngilizce renkler çalışması lütfen soruların cevaplarını küçük harflerle yazalım” mesajını yazalım. Ve son olarak sahnemizin adını başlangıç olarak düzenleyelim.
2. Ardından boya butonuyla 2. Bir sahne oluşturalım bu sahnemiz boş olsun isterseniz sahneye sadece renk verebilirsiniz. Sahnemizin adını sorular olarak düzenleyelim.
3. Gelelim sahne geçişi için gerekli komutları oluşturmaya. Bu komutları sahnemizin yazılar bölümünde oluşturacağız. Çalışma başladığında sahneler arasında otomatik geçiş olmasını istiyoruz. Bunun için kontrol bloğundan yeşil bayrak tıkladığında komutunu alıyoruz ve ardından çalışmanın başında hep başlangıç sahnesi ile başlamasını istediğimiz için görünüm bloğundan **“arka planı başlangıç ile değiştirin”** komutunu ekliyoruz. Her iki sahne arasında belli bir süre beklemesi içinde kontrol bloğundan **“1 sn bekleyin”** komutunu alıp süreyi kendimize göre ayarlayalım. Ve ardından tekrar **“arka planı başlangıç ile değiştirin”** komutunu ekliyoruz bu sefer başlangıç olan sahneyi soru cevap olarak değiştirelim.



4. Sahnemize soruları soracak bir karakter ekleyelim. Bu karakterin 2. Sahnede karşımıza çıkmasını istiyoruz yani çalışmanın başlangıcında karakter sahnede gizli durumunda olacak bunun için karakterimize yeşil bayrak tıkladığında komutunu alıyoruz ve altına görünüm bloğundan **“Gizleyin”** komutunu ekliyoruz ve bu gizlilik süresini beliyoruz ve kontrol bloğundan **“1 sn bekleyin”** komutunu ekleyip süreyi belirliyoruz. Karakterimizin 2. sahneden tekrar görünür olması için de son olarak **“Göster”** komutunu ekleyelim.



5. Gelelim soru- cevap bölümünü oluşturmaya. Karakterimizin bize sorular sormasını ve bizimde klavyeden yazarak bu sorulara cevap vermeyi istiyoruz. Verdiğimiz doğru ya da yanlış cevaba göre de karakter bize geri dönüt verecek. Bunu için karakterimize yeni bir yeşil bayrak tıkladığında komutu alalım ve buna öncelikle çalışma 2. Sahneye geçtiğinde soruyu sorsun diye bekleme süresi ekleyelim. Bekleme süresinin ardından karakterle beraber sorumuzda sahnemize gelsin. Bunun için algılama bloğundan “Adı ne? Sor ve bekle” komutunu ekleyelim ve soru kısmını kendimiz istediğimiz soru olarak düzenleyelim.



6. Sorumuzu sorduk çalışmayı başlattığımızda göreceksiniz ki karakterimiz soruyu soruyor ve bizim cevabı yazmamız için sahnenin altında bir bölüm açılıyor. Şimdi yazacağımız cevabın doğru ya da yanlış olduğunu kontrol eden komutları ekleyelim. Burada cevabımız doğru ise bize “harikasin doğru cevabı buldun” mesajını, yanlış cevapta ise, “yanlış cevap” mesajını versin istiyoruz. Burada bir şart olduğu için şart durumlarında kullanılan komut olan Eğer ise komutunu kullanacağız bunun için kontrol bloğunun altından “eğer ise, başka” komutunu alalım. Burada eğer ile ise kelimeleri arasında bir koşul koyacağız. Bunun için operatörler bloğundan “...=...” komutunu alalım. Bu komut bizim cevabımız ile gerçek cevabı karşılaştıracaktır. Şöyleki bu = işaretinin soluna algılama bloğundan “yanıt” komutunu ekleyelim. Yanıt komutu bizim klavyeden yazacağımız yanıttır. = işaretinin sağ kısmına da doğru cevabı yazalım(resimdeki gibi). Eğer yanıtımız doğru ise bize yukarıda belirttiğimiz “harikasin doğru cevabı buldun” mesajını verecek bunun için görünüm bloğundan “2. Sn ye boyunca merhaba diye konuş” komutunu ilk boşluğa ekleyelim. Eğer yanıtımız yanlış ise “yanlış cevap istersen tekrar dene” için aynı komutu ikinci boşluğa ekleyelim.



7. Bunu birden fazla soru için nasıl arttırabileceğimizi öğretmenimizle tartışalım.

Haftanın Özeti

Bu çalışma için kullandığımız yeni kodlarımız: Operatörler, Kalem, Görünüm kod bloklarını kullanacağız.

Kontrol Kod Bloğundan kullandığımız yeni komutlar;

- “Eğer ise, başka” :Belli koşullarda yerine getirilmesi gereken görevlerin tanımlanmasını sağlayan komuttur. Koşullu olarak çalıştırmak istediğimiz komutu iki ayrı parça halinde tanımlar.(örnek: Eğer cevap **doğru ise** sağa dönsün, **yanlış ise** sola dönsün)

Operatörler Kod Bloğundan kullandığımız yeni komutlar;

- “1 ile 10 arasında rasgele seçin ” : Belli aralıktaki sayı kümesi içinden rasgele bir sayı seçilmesini sağlar.

Algılama Kod Bloğundan Kullandığımız yeni komutlar;

- “..... Sor ve bekle” :Soru sorarak cevaplar alabileceğimiz ortam oluşturur.
- “yanıt” :”..... sor ve bekle” komutuyla beraber çalışır. “..... sor ve bekle” komutundan sonra vereceğimiz cevap yanıt olarak algılanır.

Operatörler Kod Bloğundan Kullandığımız yeni komutlar;

- “....=....” :Eşitlik durumunu karşılaştırır.

ÖDEV: Buna 2 soru daha ekleyerek toplam 3 soruluk bir çalışma hazırlayalım. Sahneye bir değişken ekleyelim bu bizim puanımız olsun sorulardan doğru cevap aldıkça puanımız artsın. Sorular matematik, Türkçe, Sosyal Bilgiler ya da İngilizce derslerinde işlediğiniz konularla ilgili olursa daha güzel olur. Seçeceğiniz derse yönelik bir sahnede oluşturmayı unutmayın. ☺

8. Soru Cevap Uygulaması

Bu hafta geçen hafta yaptığımız soru cevap uygulamasını biraz daha farklı komutlarla ve soru sayısını arttırarak uygulayacağız.

1. Öncelikle geçen hafta öğrendiğimiz şekilde karaktere bir soru sorduralım ve bu soruya verdiğimiz cevabın doğru ya da yanlış olduğunu kontrol ettiren komutları aşağıdaki gibi oluşturalım.



2. Soruya doğru cevap verilene kadar aynı soruyu sormaya devam etsin istiyoruz. Bunun için yukarda oluşturduğumuz komutlarımıza kontrol kod bloğunda “ olana kadar **tekrarlayın**” komutunu ekleyeceğiz. Bu komutu sorduğumuz soruyu ve sorunun cevabını kontrol eden komutları içine alacak şekilde ekleyelim ve komut içindeki boş bölüme, komut içindeki bölümün ne zamana kadar tekrarlanmasını istediğimiz şartı yazalım. **Sartımız;** doğru cevabı alana kadar, sorunun tekrarlanmasıydı. O zaman “yanıt =” Komutlarını aşağıdaki gibi ekleyelim.



3. Soruya doğru yanıt verildiği takdirde 2. Soruyu yayınlamasını istiyoruz. Bunun için soruya doğru yanıt verdiğimizde “tebrikler doğru cevap” sözünü söylesin komutunun hemen altına kontrol bloğundan “..... duyurusu yap” komutunu ekleyelim. Bu komutu ekleyince başındaki



boşluğa tıklayıp 2. Soru yazalım. Bu kısma ne yazdığımız çok da önemli değil. İstedığınızı yazabilirsiniz ama ne yazdığınızı unutmamak için anlamlı bir şey yazmanız daha iyi olur.

4. Bu yayınla komutu neyi yayınlayacak ama, şimdi 2. Soruyu yapalım. Birinci sorunun komutlarında doğru cevaptan sonra yayınlanacak 2. Soru için farklı bir komut ile başlayacağız. İlk komutlarda kullandığımız “..... **duyurusu yap**” komutu, yine kontrol bloğundan “..... **aldığımda**” komutu ile birbirini tetikleyerek çalışmaktadır. Bu yüzden ikinci soru için başlangıç komutu olarak “..... aldığımda” komutunu kullanalım.
5. Aslında yukardaki ilk soru için oluşturduğumuz komutlardan tek fark bu yeni komut. Geri kalan komut bloklarını ilk sorudan kopyala yapıştır ile alabiliriz. Ve sonra soruyu değiştiririz.

KLAVYE PANOMUZ



Birbirinde farklı modelde ve özellikle klavyeler vardır. Senin klavyedeki tuş sayısı ve tuşların dizilimi aşağıdaki gibi **olmayabilir!**



1 Escape

O sırada yapılan işi yarıda kesmek, vazgeçmek veya geriye dönmek amacıyla kullanılır. Kullandığımız programlarda Esc tuşu genellikle iptal düğmesini temsil eder.

2 Tab (Sekme)

Belirlenen miktarda boşluk bırakılarak aynı hizada yazı yazmaya yarar. Tablolarda hücreler arası geçiş yapmayı sağlar. İnternet sayfalarında nesnelere arası geçiş yapmayı sağlar.

3 Caps Lock

Büyük harf yazma tuşudur. Klavyenin sağ üst köşesinde bulunan uyarı ışığı yanıyorsa tuş aktif haldedir ve büyük harfler yazılır.

4 Shift

Yazı yazarken ikincil karakterleri (! ' % ? : ; gibi) yazmak için kullanılır. Ayrıca Caps Lock pasifken büyük harf, aktifken küçük harf yazmayı sağlar.

5 Ctrl

Tek başına kullanılmaz. Değişik görevleri vardır. Ctrl+C: **Kopyala**
Ctrl+X: **Kes** Ctrl+V: **Yapıştır** gibi...

6 Windows Tuşu

Windows işletim sisteminde Başlat menüsünü açmayı sağlar.

7 Alt

Diğer tuşlarla beraber kullanılır. Örn: Ctrl+Alt+Del: Görev Yöneticisini açar

8 Boşluk Tuşu

Kelimeler arasında boşluk bırakır.

9 Alt Gr

Tek başına kullanılmaz. Üçüncül karakterleri (@ [] gibi) yazmayı sağlar.

10 Sağ Fare Tuşu

Farenin sağ tuşu ile aynı görevi görür.

11 Enter

Onay tuşudur. Bir işlemi onaylamak için kullanılır. Ayrıca kelime işlemci programında alt satıra geçerek yeni bir paragraf oluşturmaya yarar.

12 Back Space

Fare imlecinin önünde bulunan karakteri siler. Her başta 1 karakter siler.

13 Delete

Seçilen nesneyi silmeye yarar. Fare imlecinin sağında bulunan karakteri siler.

14 End

İmleci satırın sonuna getirir.

15 Page Down

Bir sayfa (bir ekran görüntüsü kadar) aşağı inmemizi sağlar.

16 Insert

Insert açıkken yazılan karakter sağındaki karakteri silerek onun yerine geçer.

17 Home

İmleci satırın sonuna getirerek satırın sonuna gitmeyi sağlar.

18 Page Up

Bir sayfa (bir ekran görüntüsü kadar) yukarı çıkarmızı sağlar.

19 Num Lock

Klavyenin sağ tarafında bulunan **nümerik tuşların** kullanılmasını sağlar. Uyarı ışıklarından Num Lock ışığı yanarken nümerik tuşları kullanabilirsiniz.

20 Uyarı Işıkları

Num Lock, Caps Lock ve Scroll Lock tuşlarının aktif olup olmadığını gösterir.

ESC (ESCAPE) TUŞU

FONKSİYON TUŞLARI

ÖZEL TUŞLAR

NÜMERİK TUŞLAR

YÖN TUŞLARI

UYARI IŞIKLARI

KLAVYE KISAYOLLARI

- | | |
|---|---|
| <ul style="list-style-type: none">• CTRL+C (Kopyala)• CTRL+X (Kes)• CTRL+V (Yapıştır)• CTRL+Z (Geri Al)• DELETE (Sil)• ÜSTKRKT+DELETE (Seçili öğeyi Geri Dönüşüm Kutusu'na atmadan kalıcı olarak sil)• Bir öğeyi sürüklerken CTRL (Seçili öğeyi kopyala)• Bir öğeyi sürüklerken CTRL+ÜSTKRKT (Seçili öğeye kısayol oluştur)• F2 tuşu (Seçili öğeyi yeniden adlandır)• CTRL+SAĞ OK (Ekleme noktasını sonraki sözcüğün başına götür)• CTRL+SOL OK (Ekleme noktasını önceki sözcüğün başına götür)• CTRL+AŞAĞI OK (Ekleme noktasını sonraki paragrafın başına götür)• CTRL+YUKARI OK (Ekleme noktasını önceki paragrafın başına götür)• Ok tuşlarının herhangi biriyle birlikte CTRL+ÜSTKRKT (Bir metin bloğu vurgula)• Ok tuşlarının herhangi biriyle birlikte ÜSTKRKT (Pencere veya masaüstünde birden fazla öğe seç veya bir belgede metin seç)• CTRL+A (Tümünü seç)• F3 tuşu (Bir dosya ya da klasör ara)• ALT+ENTER (Seçili öğenin özelliklerini görüntüle)• ALT+F4 (Etkin öğeyi kapat veya etkin programdan çık) | <ul style="list-style-type: none">• ALT+ENTER (Seçili nesnenin özelliklerini görüntüle)• ALT+ARA ÇUBUĞU (Etkin pencere için kısayol menüsünü aç)• CTRL+F4 (Aynı anda birden çok belge açmayı sağlayan programlardaki etkin belgeyi kapat)• ALT+SEKME (Açık öğeler arasında geçiş yap)• ALT+ESC (Öğeler arasında açılma sıralarına göre dön)• F6 tuşu (Bir penceredeki veya masaüstündeki ekran öğeleri arasında dolaş)• F4 tuşu (Bilgisayarım veya Windows Gezgini'ndeki Adres çubuğu listesini görüntüle)• ÜSTKRKT+F10 (Seçili öğe için kısayol menüsünü görüntüle)• ALT+ARA ÇUBUĞU (Etkin pencere için Sistem menüsünü görüntüle)• CTRL+ESC (Başlat menüsünü görüntüle)• ALT+Bir menü adındaki altı çizili harf (Karşılık gelen menüyü görüntüle)• Açık bir menüdeki bir komut adındaki altı çizili harf (Karşılık gelen komutu gerçekleştirir)• F10 tuşu (Etkin programda menü çubuğunu etkinleştirir)• SAĞ OK (Sağdaki sonraki menüyü aç veya bir alt menü aç)• SOL OK (Soldaki sonraki menüyü aç veya bir alt menüyü kapat)• F5 tuşu (Etkin pencereyi güncelleştir)• GERİ AL (Bilgisayarım ya da Windows Gezgini'nde bir seviye üstteki klasörü görüntüle)• ESC (Geçerli görevi iptal et) |
|---|---|