# 6. Hafta – Işık Matrisi, Hoparlör ve Hub Düğmeleri

## Haftanın Amacı:

Bu haftanın amacı öğrencilere ışık matrisini, hoparlörü ve hub düğmelerini kullanmayı öğretmektir. Öğrencilerin ışık matrisini kullanarak hareketli yazı yazdırmaları, pikselleri ayrı ayrı kullanmaları, ışık matrisine şekil çizdirmeleri, hoparlörden farklı notalarda bip sesi çıkartmaları ve hub düğmelerini kullanmaları beklenmektedir. Bunun yanında öğrenciler temel düzeyde değişken, döngü ve liste komutlarını kullanacaktır.

## Haftanın Kazanımları:

* Öğrenciler ışık matrisini ve ilgili komutları tanır.
* Öğrenciler hoparlörü ve ilgili komutları tanır.
* Öğrenciler hub düğmelerini ve ilgili komutları kullanır.
* Öğrenciler değişkenleri, döngüleri ve listeleri basit problemlerin çözümünde kullanır.
* Öğrenciler şimdiye kadar öğrendiği komutları bir araya getirerek problemler çözebilir.

## Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanları.

## GÖZLE VE UYGULA

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Hub Hakkında*** |
| *Lego Spike Prime’da robotun programlanabilir kısmına Hub (hab diye okunur) ismi verilmiştir. Motorlar ve sensörler Hub üzerinde bulunan 6 farklı port üzerinden bağlanabilir. Hub’da bulunan altı portun tamamı hem girdi hem de çıktı işlemleri için kullanılabilir. Hub üzerinde 5x5 ışık (LED) matris, üç buton, bir durum ışığı, bir hoparlör, USB ve Bluetooth bağlantı noktaları ve altı eksenli hareket sensörü bulunmaktadır.*    *Hub üzerinde Python programlarının çalıştırılması için MicroPython işletim sistemi yüklüdür. Hub üzerindeki cihazların kullanılabilmesi için öncelikle bir PrimeHub nesnesi oluşturularak aşağıdaki komutta görüldüğü gibi Hub’ın tanıtılması gerekmektedir. Bu komut zaten Python programları için önceden yazılmış olarak gelmektedir. Yeniden yazılmasına gerek yoktur.*  *hub = PrimeHub()* |

### Gözle: Işık Matrisinde Yazı Yazdırmak

Bu etkinlikte amaç Hub üzerinde bulunan ışık matrisinde sağdan sola doğru hareket eder şekilde yazılar yazdırmaktır. Bu iş için ışık matrisinin *light\_matrix.write* metodu kullanılır. Işık matrisi ile ekranda her defasında bir karakter bulunur ve karakterler soldan sağa kayarak kelimeler gösterilir. Bu metot metin ve sayı cinsinden parametre alabilir. Ekrana bir metin, örneğin TÜBİTAK yazdırmak için aşağıdaki şekilde kod yazılmalıdır.

hub.light\_matrix.write("TUBITAK")

### Gözle: Değişkenlere Giriş

Işık matrisi ile sayısal değerler de ekrana yazdırılabilir. Python programlama dilinde çift veya tek tırnak içerisinde yazılan veriye metin cinsinden veri denilir. Bu veri tipi Python’da *str* (string’in kısaltması) şeklinde ifade edilir. Buna örnek olarak “Prime” verilebilir. Python’da sayısal veri tipleri de bulunmaktadır. Tam sayı cinsinden olan veriye *int*, ondalıklı veriye ise *float* isimleri verilmiştir. Int veya float cinsinden bir veriyi *write* metodu ile ışık matrisinde yazdırmak mümkündür. Örneğin aşağıdaki ifade ile tam sayı olan 5 ışık matrisinde yazdırılır.

hub.light\_matrix(5)

Python’da bazen program boyunca veya programın bir kısmında çeşitli verilere ihtiyaç duyulur. Bu verilerin değişme ihtimali bulunur. Python’da bu tip verilerin saklandığı programlama yapısına değişken ismi verilir. Günlük atılan adım sayısı, hava sıcaklığı ve yenen yemek miktarı değişkenlere günlük yaşamdan örnek olarak verilebilir. Programlamada değişkenler verimli bir şekilde kullanmak için isimlendirilirler. Örneğin aşağıdaki kod ile *str* cinsinden metin isimli bir değişken oluşturulmuştur ve bu değişkene değer olarak “TUBITAK” metni atanmıştır.

metin="TUBITAK"

Bu değişkenin içerisindeki değer değiştirilmek isteniyorsa ona yeni bir değer atanmalıdır. Örneğin aşağıdaki kod ile metin isimli değişkene “Kodlama” metni atanmıştır.

metin="Kodlama"

Aşağıdaki kod ile her defasında aynı komutla *hub.light\_matrix.write(metin)* ışık matrisinde metin yazdırılsa da ekrana önce TUBITAK ardından Kodlama yazdırılacaktır. Çünkü metin isimli değişkenin değeri değiştirilmiştir.

metin="TUBITAK"

hub.light\_matrix.write(metin)

metin="Kodlama"

hub.light\_matrix.write(metin)

Değişkenler sayısal değerler de alabilir. Aşağıda görüldüğü üzere *sinif\_mevcudu* değişkenine tam sayı (*int*) değeri olan 20 atanmıştır. Değişkenlere 10.5 gibi ondalık (*float*) değerler de atanabilir.

sinif\_mevcudu=20

### Gözle ve Uygula: 0’dan 5’e kadar sayıyorum

Rehber öğretmen öğrencilerden ışık matrisinde sırayla 0, 1, 2, 3, 4 ve 5 sayılarını gösteren kodu yazmalarını ister. Öğrenciler büyük ihtimalle aşağıdakine benzer bir kod yazacaktır.

hub.light\_matrix.write(0)

hub.light\_matrix.write(1)

hub.light\_matrix.write(2)

hub.light\_matrix.write(3)

hub.light\_matrix.write(4)

hub.light\_matrix.write(5)

Bu kod istenen görevi yerine getirmez. Çalıştırıldığında aralarda bekleme olmadığı için, hızlıca ilerler ve ışık matrisinde sadece 5 yazıldığı görülür. Rehber öğretmen öğrencilere problemin üzerinde uğraşması için yeteri kadar süre verir. Her bir yaz (*write*) komutundan sonra bekleme süresi konulması gerektiğini söyler. Bunu Python ile iki şekilde yapılabileceğini söyleyerek aşağıdaki konuyu anlatır.

Python’da saniye cinsinden belirli bir süre duraklama/bekleme yapılabilmesi için aşağıdaki komut kullanılabilir. Saniye cinsinden beklenecek süre ondalık değerler alabilir. Örneğin 0.5 değeri verilirse yasım saniye bekleme yapılır.

wait\_for\_seconds(saniye cinsinden beklenecek süre)

Bu metot dışında Python’da bulunan *time* modülü de bu iş için kullanılabilir. Time modülünün kullanılabilmesi için öncelikle yazdığımız programa dahil edilmesi gerekir. Bunun için “import time” komutu programın üst kısmına diğer *import* satırlarının hemen altına (*from math import \** İfadesinin altındaki satır) yazılmalıdır. Bundan sonra aşağıdaki komut ile istenen süre kadar işlemler duraklatılabilir.

time.sleep (saniye cinsinden beklenecek süre)

Rehber öğretmen sonuç olarak aşağıdaki kodu yazar ve öğrencilerin de aynı kodu yazıp çalıştırmalarını sağlar. Bu kod bloğunda *time.sleep* terine *wait\_for\_seconds* fonksiyonu da kullanılabilir. İkisi de aynı işi görecektir.

import time

hub = PrimeHub()

hub.light\_matrix.write(0)

time.sleep(0.5)

hub.light\_matrix.write(1)

time.sleep(0.5)

hub.light\_matrix.write(2)

time.sleep(0.5)

hub.light\_matrix.write(3)

time.sleep(0.5)

hub.light\_matrix.write(4)

time.sleep(0.5)

hub.light\_matrix.write(5)

time.sleep(0.5)

### Gözle ve Uygula: 0’dan 5’e kadar sayıyorum

Bir önceki örneğimiz tekrar incelendiğinde altı defa aynı görev tekrarlanmaktadır: ekrana n sayısını yaz ve 0.5 saniye bekle. Programcılar bu tip tekrarlı durumlarda aynı kodu 6 farklı satırda yazmak yerine döngüleri kullanırlar. Döngüler isminden de anlaşılacağı üzere belirli bir sayıda tekrarlanması istenen veya belirli bir koşula bağlı olacak şekilde tekrar etmesi istenen komutlar için kullanılır. Python’da iki çeşit döngü bulunur. Bunlardan ilki *for* döngüsüdür. Aşağıdaki kod ışık matrisine 0 ile 5 arasındaki sayıları 0.5 saniye ara ile yazar.

for n in range(6):

    hub.light\_matrix.write(n)

    wait\_for\_seconds(0.5)

Rehber öğretmen bu kodu ekrana yansıtır ve öğrencilerden bu kod üzerinde çeşitli değişiklikler yaparak kodun kullanım kurallarını olabildiğince kendilerinin keşfetmelerini sağlar. Öğrencilere yeteri kadar süre verdikten sonra aşağıda verilen bilgileri kullanarak konuyu açıklar.

*Range* metodu ile belirli bir aralıkta sayı üretilmesi sağlanır. Örneğin, range(6) ifadesi ile 0-5 arasındaki tam sayılar üretilir. 0-6 arasındaki tam sayılar için range(7) ifadesi kullanılmalıdır. Görüldüğü gibi *range* metodu ile sayı üretme işlemi sıfırdan başlar ve metoda parametre olarak verilen sayının bir eksiğine kadar devam eder. Böylece istenen büyüklükte sayı üretilmiş olur. Aşağıdaki ifade ile ekrana üç kere TUBITAK yazılacaktır.

for n in range(3):

    hub.light\_matrix.write("TUBITAK")

Burada dikkat edilmesi gereken önemli noktalar bulunmaktadır. Döngünün içerisinde bulunan ve döngü gövdesi olarak adlandırılan, tekrar etmesi istenen, komutları belirtmek için Python’da öncelikle *for* satırının sonuna iki nokta üst üste işareti ifadesi konulmuştur. Bu ifade döngü gövdesinin başlangıcını belirtir. Döngü gövdesindeki komutların belirlenmesi için ise *Tab* tuşu ile bir kere girinti yapılmalıdır. Yukarıdaki kodda görüldüğü üzere ışık matrisine TUBITAK yazılması için kullanılan komut *Tab* ile girintili olacak şekilde yapılmıştır. Çoğu zaman programcının girinti yapmasına gerek yoktur. Girinti işlemini otomatik olarak Python programlama ortamları yapmaktadır. Döngü n=0 için başlar döngü gövdesindeki komutlar, yani ışık matrisine TUBITAK yazılması, gerçekleştirilir. Ardından n=1 için döngü gövdesindeki komutlar gerçekleştirilir. Son olarak n=2 için döngü gövdesindeki komutlar gerçekleştirilerek döngüden çıkılır. Buradaki n bir değişkendir range(3) komutu ile sıralı olarak oluşturulan değerler her defasında n’ye atanır. Burada n yerine farklı bir isim de verilebilir. Örneğin aşağıdaki komut çalıştırıldığında ekrana sırasıyla 0, 1 ve 2 basılır.

for i in range(3):

    print(i)

*Range* komutu ile 0’dan n’ye kadar ardışık sayılar üretilebildiği gibi 0’dan n’ye kadar belirli adım miktarında atlayarak sayılar da sayı üretilebilir. *Range* fonksiyonunun genel kullanımı aşağıdaki gibidir.

range(başlangıç, bitiş, adım)

Örneğin aşağıdaki kod çalıştırıldığında ışık matrisinde 0, 2, 4, 6, 8 sayıları görünecektir. Başlangıçtan itibaren teker teker değil adım sayısı olan ikişer ikişer ilerlenmiştir. Üçer üçer ilerlenmesi isteniyorsa adım parametresi yerine 3 girilmelidir. Burada başlangıcın dahil olup bitişin dahil olmadığını yeniden belirtmekte fayda var. Aşağıdaki kod ile 10 değerinin de yazılması isteniyorsa bitiş parametresi yerine 11 yazılmalıdır.

for n in range(0,10,2):

    hub.light\_matrix.write(n)

    wait\_for\_seconds(0.5)

### Uygula: Geri Sayım Yaptırıyorum

Rehber öğretmen öğrencilerden 5’ten 1’e kadar geri sayan ve bu sayıları ışık matrisinde gösteren programı yazmalarını ister. Bu iş için aşağıdaki kod kullanılabilir.

for n in range(5,0,-1):

    hub.light\_matrix.write(n)

    wait\_for\_seconds(0.5)

### Gözle: Piksel Kontrolü ile Işık Matrisi

Işık matrisi 5x5 boyutundadır. Yani 5 kolunu ve 5 satırı olmak üzere toplamda 25 tane hücresi bulunur. Buradaki hücrelerden her biri piksel (pixel) olarak adlandırılır. Bu durum aşağıda temsili olarak gösterilmiştir. Satırlar x eksenini sütunlar ise y esenini göstermektedir. Bu iki değer kullanılarak istenen piksele ulaşılabilir. Örneğin birinci satırın en sonundaki piksel için (0,4) değeri kullanılır.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| 0 |  |  |  |  |  |
|  |  |  |  |  |  |
| 1 |  |  |  |  |  |
|  |  |  |  |  |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |
| 3 |  |  |  |  |  |
|  |  |  |  |  |  |
| 4 |  |  |  |  |  |

Python ile ışık matrisinde bulunan piksellerin her birinin yanıp söndürülmesi ve parlaklığının ayarlanması mümkündür. Bunun için kullanılan metot aşağıda gösterilmiştir.

set\_pixel(x, y, parlaklık)

Buradaki x değeri ulaşılmak istenen pikselin hangi satırda olduğunu, y değeri ise ulaşılmak istenen pikselin hangi sütunda olduğunu belirtmektedir. Bu yüzden her ikisi de 0-4 arasında değer alır. Parlaklık değeri ise 0-100 arasında değişebilir. 100 azami ölçüde parlaklık anlamına gelirken 0 hiç ışık verilmemesini temsil eder. Parlaklık değeri girilmediğinde ön tanımlı değer olarak 100 kabul edilir.

Işık matrisindeki bütün piksellerin bir anda söndürülmesi için *off* metodu bulunmaktadır. Aşağıdaki komut kullanılarak ışık matrisindeki bütün pikseller söndürülebilir.

hub.light\_matrix.off()

Rehber öğretmen ışık matrisindeki piksellerin nasıl yakıldığını anlattıktan sonra, öğrencilere aşağıdaki kodu gösterir ve açıklar. Kodu açıklarken iç içe döngüler konusuna değinir. Öğrencilerin iç içe döngüler konusunu anlaması önemlidir. Rehber öğretmen konu anlatımına gerekli süreyi ayırır. Bu kod ile bütün pikseller yakılmış olur. Rehber öğretmen bu kodu öğrencilere açıklayarak anlatır.

for x in range(5):

    for y in range(5):

        hub.light\_matrix.set\_pixel(x,y)

### Uygula: Geri Sayım Öncesi Yanıp Sönme

Rehber öğretmen öğrencilerden 5’ten 1’e kadar geri sayım yapan programın geliştirilmesini ister. Geri sayıma başlanmadan önce bir defa bütün piksellerin yanıp sönmesi istenir. Bu görev için aşağıdaki kod kullanabilir.

for x in range(5):

    for y in range(5):

        hub.light\_matrix.set\_pixel(x,y)

wait\_for\_seconds(0.5)

hub.light\_matrix.off()

wait\_for\_seconds(0.5)

for i in range(5,0,-1):

    hub.light\_matrix.write(i)

    wait\_for\_seconds(0.5)

Rehber öğretmen öğrencilere kodu tamamlaması için yeteri kadar süre verir. Ardından bu kodu biraz daha geliştirmeleri istenir. Sayma öncesinde bütün pikseller bir kere değil üç kere yanıp sönmelidir. Bütün etkinlik bittikten sonra öğrencilere bütün pikselleri yakıp söndürme işlemini nasıl yaptıkları sorulur. Eğer bu iş için döngü kullanmamışlarsa tekrarlanan komutların programlamada döngüler vasıtası ile yapıldığını hatırlatır ve döngü içeren komutu yazar.

### Gözle ve Uygula: Ekrana Resim Çiziyorum

Işık matrisinin her bir pikseli ayrı ayrı kullanılarak gülen yüz gibi şekiller yapılabileceği gibi daha önceden tanımlanan şekillerin ışık matrisinde gösterilmesi de sağlanabilir. Bu iş için *show\_image* metodu kullanılır. Örneğin, daha önce de gösterildiği gibi, aşağıdaki komut kullanılarak ışık matrisinde gülen yüz gösterilir.

hub.light\_matrix.show\_image('HAPPY')

Işık matrisi için gülen yüz dışında birçok şekil tanımlanmıştır. Örneğin “ARROW\_N” kuzeyi gösteren ok işareti, “ARROW\_S” güneyi gösteren ok işareti, “AROW\_E” doğuyu gösteren ok işareti ve “ARROW\_W” batıyı gösteren ok işareti için kullanılabilir. Bunun dışında da tanımlanmış şekiller bulunmaktadır. Bu şekillerin isimlerine Bilgi Bankasından ulaşılabilir.

Rehber öğretmen değişik şekillerin ışık matrisinde gösterilmesini sağlar ve kodu öğrencilere anlatır. Bunun ardından, rehber öğretmen öğrencilerden Bilgi Bankasına girerek ışık matrisi altındaki *show\_image()* metodu altında bulunan resim/şekil değerleri için kullanılabilecek olası değerleri keşfetmelerini ister. Onlara bu iş için yeterli süreyi verir.

### Uygula: Ok Animasyonu

Rehber öğretmen öğrencilerden sırayla kuzey, doğu, güney ve batı yönlerini gösteren bir animasyon yapmalarını ister. Bu görev için aşağıdaki kod kullanılabilir.

hub.light\_matrix.show\_image("ARROW\_N")

wait\_for\_seconds(0.5)

hub.light\_matrix.show\_image("ARROW\_E")

wait\_for\_seconds(0.5)

hub.light\_matrix.show\_image("ARROW\_S")

wait\_for\_seconds(0.5)

hub.light\_matrix.show\_image("ARROW\_W")

wait\_for\_seconds(0.5)

### Gözle ve Uygula: Sürekli Tekrarlanan Görevler (While Döngüsü)

Python programlama dilinde ikinci döngü çeşidi *while* tipi döngüdür. Bu döngü tipinde işlemler koşula bağlı olarak tekrar edilir. *While* döngüsünün genel kullanımı daha sonra anlatılacaktır. Burada *while* döngüsünün özel bir durumu gösterilecektir. Bu özel durum sonsuz döngüdür. Hub açık olduğu müddetçe yapılması istenen komutlar sonsuz döngü içerisinde yaptırılır. Başka bir şekilde isimlendirmek gerekirse sonsuz döngü sürekli tekrarlanacak işlemleri gerçekleştirir. Blok tabanlı programlama araçlarında buna sürekli tekrarla tipi döngü denilir. Yukarıda yapılan ok animasyonu bir kereye mahsustur. Bu animasyonun sürekli olması için aşağıdaki kod kullanılabilir.

while True:

    hub.light\_matrix.show\_image("ARROW\_N")

    wait\_for\_seconds(0.5)

    hub.light\_matrix.show\_image("ARROW\_E")

    wait\_for\_seconds(0.5)

    hub.light\_matrix.show\_image("ARROW\_S")

    wait\_for\_seconds(0.5)

    hub.light\_matrix.show\_image("ARROW\_W")

    wait\_for\_seconds(0.5)

While döngüsünde de döngü gövdesinin başlangıcı için iki nokta üst üste ve döngü gövdesindeki komutları belirtmek için bir *Tab* girinti kullanılır. While tipi döngülerde *while* ifadesinden sonra koşul yazılır ve bu koşul doğru olduğu müddetçe döngü gövdesindeki işlemler yapılır. Burada *while*’dan sonra *True (Doğru)* yazıldığı için sürekli doğru olan bir koşul olduğu düşünülerek gövdede bulunan komutlar sürekli çalıştırılacaktır.

Rehber öğretmen sürekli tekrarlanan ok animasyonunu kullanarak öğrencilere *while* döngüsünü anlatır. Ardından öğrencilerden sürekli tekrarlanan başka bir animasyonu yapmalarını ister.

### Gözle ve Uygula: Listeler

Bu derste daha önce anlatıldığı gibi değişkenler içerisinde bir defasında sadece bir değer tutulabilir. Python programlama dilinde birden farklı veriyi bir arada tutmak için liste ismi verilen veri yapıları kullanılır. Listeler içerisinde her türlü veri aynı anda tutulabilir. N elemanlı bir liste tanımlamak için aşağıdaki komut kullanılabilir. Görüldüğü üzere listelerde indeksleme 0’dan başlamaktadır. Rehber öğretmen listeler konusunu öğrencilere anlatmalıdır. Bunun için append ve clear gibi temel liste metotlarından da bahsetmelidir. Bu metotlar ilerideki etkinliklerde kullanılacaktır.

liste\_ismi=[eleman0, eleman2…, elemanN-1]

Sürekli tekrarlanan görevler etkinliğinde dört farklı yönü gösteren ok kullanılmıştır. Bu ok’ların ismini tutmak için liste kullanılabilir.

oklar=["ARROW\_N","ARROW\_E","ARROW\_S","ARROW\_W"]

Bu kod bloğu ile kod isimleri aynı liste içerisinde tutulmaktadır ve bu listeye oklar ismi verilmiştir. Oklar isimli listesi *for* tipi döngü ile kullanılarak daha gelişmiş kodlar yazılabilir. Örneğin sürekli tekrarlanan görevler etkinliğinde okların gösterilmesi için ayrı ayrı dört *show\_image* metodu kullanılmıştır. Bunun yerine *liste ve for* döngüsü birlikte kullanılabilir.

oklar=["ARROW\_N","ARROW\_E","ARROW\_S","ARROW\_W"]

while True:

    for ok in oklar:

        hub.light\_matrix.show\_image(ok)

        wait\_for\_seconds(0.5)

Rehber öğretmen yukarıdaki kodu öğrencilere açıklayarak anlatır. Ardından bir önceki etkinlikte kendilerinin geliştirdiği animasyonu liste kullanarak yeniden yapmalarını ister.

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Ses Kayıt Hakkında*** |
| *MIDI nota numarası 44 ile 123 arasında değer alabilir. Bu değerler notalara karşılık gelmektedir. Bu değerlerin hangi notaya karşılık geldiğini bulmak için Sözcük Blokları programlama ortamına geçilir. Bunun için Yeni Proje butonuna tıklanır. Aşağıdaki resimde gösterildiği gibi SÖZCÜK BLOKLARI seçilerek OLUŞTUR butonuna tıklanır.*    *Açılan pencere sözcük blokları tabanlı programlama ortamıdır. Önceki derslerde anlatıldığı üzere istenirse robot buradan blok tabanlı olarak programlanabilir. Açılan pencereden aşağıdaki resimde gösterildiği gibi SES palet başlığı tıklanır.*    *Bu başlık altından aşağıdaki resimde gösterildiği gibi bip sesini çalmaya başla bloğundaki sayısal değer üzerine tıklanır. Buradan nota numarası parametresine ulaşılabilir.*    *Burada 60-Do, 62-Re, 64-Mi, 65-Fa, 67-Sol, 69-La, 71-La notasına karşılık gelmektedir. 72 yeniden Do notasına karşılık gelir ve 72’den sağa doğru gidildikçe yeniden Re, Mi, Fa, Sol, La ve Si notaları elde edilir. 60’ın sol tarafında ise Si’den Do’ya gidilecek şekilde aynı notanın farklı frekansları mevcuttur.* |

### Gözle: Robotum Ses Veriyor

Hub üzerinde 12 Bit 16KHz (mono) ses kalitesine sahip bir hoparlör bulunmaktadır. Bu hoparlör ile farklı frekanslarda veya notalarda bip sesi çalmak mümkündür. Bip sesi çalmak için iki metot bulunmaktadır. Bunlardan ilki *start\_beep* metodudur. Kullanımı aşağıdaki gibidir.

start\_beep(MIDI nota numarası)

Rehber öğretmen aşağıdaki kodu çalıştır ve *start\_beep* metodu ile bip sesinin stop veya başka bir bip sesi komutu çalıştırılana kadar bip sesinin çalınmaya devam ettiğini anlatır.

hub.speaker.start\_beep(60)

*stop()* komutu ile *start\_beep* metodu ile başlanan bip sesi çıkarma işlemi sonlandırılabilir. Aşağıdaki komut ile Do notasında bir saniye boyunca bip sesi çalınır. Rehber öğretmen öğrencilere kodu açıklayarak anlatır.

hub.speaker.start\_beep(60)

wait\_for\_seconds(1)

hub.speaker.stop()

Hoparlörden bip sesi çıkartmak için kullanılabilecek ikinci metot *beep* metodudur. Kullanımı aşağıda gösterilmiştir.

beep(MIDI nota numarası, saniye cinsinden süre)

Bu metot ile bip sesi istenen nota ile istenen sürede çalınabilir. Örneğin aşağıdaki komut ile Do notasında 1 saniye boyunca bip sesi çalınır.

hub.speaker.beep(60,1)

Hoparlörden çıkan sesin seviyesi *set\_volume* metodu ile ayarlanabilir. Bu metodun kullanımı aşağıda verilmiştir. Ses seviyesi 0-100 arasında değişiklik gösterebilir.

set\_volume(ses seviyesi)

Aşağıdaki kod ile “Mutlu Yıllar Sana” isimli parçanın bir kısmı çalınmaktadır. Rehber öğretmen bu kodu açıklayarak yazar, ardından çalıştırır ve öğrencilerin dinlemesini sağlar.

hub.speaker.set\_volume(80)

hub.speaker.beep(62,0.25)

hub.speaker.beep(62,0.25)

hub.speaker.beep(64,0.5)

hub.speaker.beep(62,0.5)

hub.speaker.beep(67,0.5)

hub.speaker.beep(66,1.5)

hub.speaker.beep(62,0.25)

hub.speaker.beep(62,0.25)

hub.speaker.beep(64,0.5)

hub.speaker.beep(62,0.5)

hub.speaker.beep(69,0.5)

hub.speaker.beep(67,1.5)

### Uygula: Şarkı Çalıyorum

Bu etkinlikteki amaç öğrencilerin basit bir şarkıyı Hub’ın hoparlöründen çaldırmasıdır. Öğrenciler İnternet’ten kısa bir şarkının notasını bulabilir veya notasını bildiği bir parçayı kullanabilir. Rehber öğretmen öğrencileri zor şarkılar seçmeme konusunda ve hızlı olma konusunda yönlendirmelidir.

### Uygula: Kare Şeklinde Hareket Ederken Ses Çıkarıp Yön Gösteren Robot

Bu etkinlikte robot kare şeklindeki bir yol üzerinde hareket edecektir. Hareket ederken gittiği yönü ok işareti ile gösterecek ve köşelere geldiğinde bip sesi çalacaktır. Bu görev için aşağıdaki kod kullanılabilir. Rehber öğretmen öğrencilerin bu görev için nasıl bir kod yazdığını gözlemler. Kodu yazarken döngü kullanmayan öğrenci gruplarını döngü kullanmaları konusunda yönlendirir.

surus\_tabani=MotorPair("C","D")

for i in range(4):

    hub.light\_matrix.show\_image("ARROW\_N")

    surus\_tabani.move(20,"cm",0)

    hub.speaker.start\_beep(60)

    hub.light\_matrix.show\_image("ARROW\_E")

    wait\_for\_seconds(0.5)

    hub.speaker.stop()

    surus\_tabani.move(180,"degrees",100)

### Gözle: Hub’ın Sol ve Sağ Düğmeleri

Python kodu ile hub üzerinde bulunan sol ve sağ düğmelerin basılıp bırakılma durumu algılanabilir ve bu durumlarda istenen işlemler gerçekleştirilebilir. Düğmelerden birine basılana kadar beklemek için aşağıdaki olay kullanılabilir.

wait\_until\_pressed()

Düğmelerden biri bırakılana kadar beklemek için aşağıdaki olay kullanılabilir. Bu ifadeleri sol düğme ile kullanmak için *left\_button*, sağ düğme için kullanmak için *right\_button* ifadesi kullanılır.

wait\_until\_released()

Aşağıdaki kod çalıştırıldığında Hub üzerinde bulunan sol düğmeye her basıldığında robot 90 derece sola dönecektir. Rehber öğretmen kodu öğrencilere açıklayarak anlatır.

surus\_tabani=MotorPair("C","D")

hub = PrimeHub()

while True:

    hub.left\_button.wait\_until\_pressed()

    surus\_tabani.move(180,"degrees",-100)

Bazı durumlarda bekleme yapmadan sağ veya sol butona basılıp basılmadığını öğrenmek faydalı olabilir. Bu durumlarda is\_pressed metodu kullanabilir. Düğmeye basılı ise True (doğru), basılı değilse False (yanlış) değerlerini üretecektir.

### Uygula: Sağ Düğmeye Tıklanınca Çalışan Robot

“Kare Şeklinde Hareket Ederken Ses Çıkarıp Yön Gösteren Robot” etkinliğinde sürüş tabanı program robota yüklenir yüklenmez çalışmıştır. Burada öğrencilerden bu programı geliştirmeleri istenmektedir. Program yüklenir yüklenmez çalışmayacak bunun yerine Hub’ın sağ düğmesine basıldığında çalışacaktır.

## TASARLA

Tasarla: Dans Eden Robot

Bu etkinlikte robotun dans edecek şekilde programlanması istenmektedir. Rehber öğretmen öğrencilere dans içerisinde hareket, müzik ve ışık matrisinin şekillerinin arka arkaya gelecek şekilde kullanılması gerektiğini belirtir. Öğrenciler robot kodunun nasıl olması gerektiği hakkında düşünür ve grup olarak olası çözümler hakkında tartışırlar. Tasarlama sürecinde öğrencilerden aşağıda örneklenen tanımlama ve fikir üretme adımlarına benzer bir süreci gerçekleştirmeleri beklenir. Rehber öğretmen burada sorumluluğu öğrencilere bırakarak öğretici değil yönlendirici bir yöntem izlemelidir.

Tanımlama: Bu adımda dans işleminin neler gerektirdiği belirlenmelidir. Öğrenciler bu adımda bildikleri veya İnternetten araştırıp buldukları bir dans stilini kullanabilirler. Dans ederken yapılacak hareketler, çıkarılacak ses ve ışık matrisinde gösterilecek ifade ortaya konabilir. Öğrenciler tüm bunları maddeler hâlinde detaylı olarak yazabilir.

* Ses çalar,
* Sese bağlı olarak dans eder (bir dans stili kullanılabilir),
* Işık matrisinde ses ve harekete bağlı olarak uygun ifadelerin belirmesi,
* Dans hareketinin tekrarı sağlanır.

Fikir üretme: Bu aşamada öğrencilerden problem çözümüne yönelik daha somut öneriler ortaya koyması beklenmektedir.

* Ses çalmak için hub hoparlörü kullanılabilir. Notalar yardımı ile istenen melodiler çalınabilir. Buna ek olarak bilgisayar da müzik çalmak için kullanabilir,
* Zeybek figürleri robotun programlanması için kullanılabilir. Motor komutları ile ilgili figürler oluşturulabilir,
* Gülen yüz, kalp ve ok resimleri dans hareketine bağlı olarak ışık matrisinde gösterilebilir,
* Dans hareketi, müzik ve ışık matrisi görüntüleri uyumlu olmalıdır,
* Bu görevler sonsuz döngü vasıtası ile sürekli hale getirilebilir.

## ÜRET

Öğrenciler bu aşamada tasarlama adımında ortaya koydukları bilgiler doğrultusunda programlarını yazmaya başlarlar. Öğrenciler problem çözümünde aktif rol üstlenmelidir. Rehber öğretmen öğretici rolünde değildir. Doğrudan yanıtlar vermeyip yönlendirici olması yeterlidir. Öğrenciler problem çözerken yinelemeli çalışabilirler. Önce basit bir şekilde dans eden robot kodu geliştirler ve denerler. Ardından yazdıkları kodu biraz daha geliştirip gerçeğe benzer dans bileşenlerini oluştururlar. Son olarak robotun tam olarak dans etmesini sağlarlar. Bu etkinlikte kullanılacak dans bileşenleri öğrencilerin yaratıcılığına kalmıştır. Öğrenciler çok farklı dans bileşenleri kullanacağı gibi geleneksek halk danslarına yönelik program da oluşturabilirler. Rehber öğretmen öğrencileri kendi istedikleri yönde ilerlemesi için cesaretlendirmelidir.

## DEĞERLENDİR

Günün sonunda öğrenciler ile halka oluşturulur ve ilk olarak aşağıdaki sorular tahtaya yansıtılarak tartışılır:

1. Aşağıdaki programda yanlış yazılan satırları bulup düzeltiniz.

|  |
| --- |
| hub = Primehub()  for i in rang(4):      hub.speaker.beep(60,1,0) |

1. Aşağıdaki komut çalıştırıldığında kaç adet *bip* sesi duyulur?

|  |
| --- |
| hub = Primehub()  for i in range(4):      hub.speaker.start\_beep(60) |

Daha sonra rehber öğretmen aşağıdaki maddeleri kullanarak öğrencilerin sınıfça tartışmasını sağlar.

* Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
  + Döngüleri kullanmada zorlandınız mı?
  + Listeleri kullanmada zorlandınız mı?
  + Komutları akılda tutmakta veya doğru yazmakta zorlandınız mı?
  + Şarkı çaldırırken notaları bulmakta zorlandınız mı?
* Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
* Grup arkadaşınızla fikir ayrılığına düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
* Grup arkadaşınızdan ne öğrendiniz?
* Bir sonraki derse daha verimli bir öğrenme için neleri yapıp neleri yapmamanız gerekir?