



Codewards

# Kodlama Saati

80 dakikalık ders

Ders Rehberi

# Kodlama Saati

---

80 dakika

**Dersin amacı:**

1. Oyun temelli ürünle tanışmak.
2. Konuya hızlı bir giriş.
3. Temel programlama kavramlarını ve gerçek hayatta nasıl uygulandıklarını öğrenmek.

---

**Terminoloji:**

1. Bilgisayar komutu
2. Nesneler
3. Algoritma
4. Optimizasyon
5. Tekrarlama
6. Döngü

---

**Bilgisayar aktivitesi:** Öğrenciler farklı konulardaki görevleri (zorlukları) çözecekler. Tamamladıkça yeni görevler otomatik olarak görünecek.

Komutlar: move (ilerle), rotate (dön), load (yükle), put (koy)

Değişkenler: right (sağ), left (sol)

Yapı: loop...end (döngü...son)

Nesneler: robot (robot), crane (vinç)

---

**Gerekli malzemeler:** Codewards sistemine erişimi olan bilgisayar veya tabletler.

---

## Ders Planı (1. Versiyon)

1. Hikayeye giriş
2. Görevleri çözmek + kavramların tek tek açıklanması.

### 1. Bölüm Giriş

Yaklaşık süre – 5 dakika

Başlangıç videosunu gösterin ve hikayeyi tanıtın.

**Öğretmen:** «Bugün çok önemli bir görevdeyiz! Su altı istasyonunun işletim sistemini yeniden kurmalı ve çalışır hale getirmeliyiz. Zorluklarımızı tamamlarken, farklı nesnelere kontrol etmek için bilgisayar komutlarını nasıl kullanacağımızı öğreneceğiz».

### 2. Bölüm Bilgisayar alıştırmaları

Yaklaşık süre – 30 dakika

Haritadaki görevleri tamamlayın:

- Görev 3
- Görev 4
- Görev 7
- Görev 8
- Görev 9
- Görev 11
- Görev 12
- Görev 20
- Ekstra Görev 6
- Görev 46
- Görev 47
- Görev 50

Görevleri yaparken, gerekli yerlerde öğrencilere tek tek açıklamalarda bulunun.

### 3. Bölüm Kapanış

Yaklaşık süre – 5 dakika

**Öğretmen:** «Bugün su altı istasyonundaki boru hattını ve çatlakları tamir etmek için bilgisayar kodu kullandık. Harika bir iş çıkardık, ama hala birçok bozuk şey var. Umarım ekibimiz yeni ve heyecan verici görevleri gerçekleştirmek için tekrar bir araya gelir.»

## Ders planı (2. Versiyon)

### 0. Bölüm Tanışma

Herkes kendisini tanıtır ve birbiriyle tanışır.

### 1. Bölüm Giriş

Yaklaşık süre – 10 dakika

Başlangıç videosunu gösterin ve hikayeyi tanıtın.

**Öğretmen:** «Bugün çok önemli bir görevdeyiz! Su altı istasyonunun işletim sistemini yeniden kurmalı ve çalışır hale getirmeliyiz. Zorluklarımızı tamamlarken, farklı nesnelere kontrol etmek için bilgisayar komutlarını nasıl kullanacağımızı öğreneceğiz».

### 2. Bölüm Bilgisayar alıştırmaları

Yaklaşık süre – 55 dakika

**Öğretmen:** «Kontrol panellerimizi açalım ve nasıl çalıştığını görelim.»

- Görev 3'e gidin

Bu görevden önce, komut sistemini tanıtın:

#### ***Temel Kavram:***

#### **KİM + NE + NASIL**

Nesneleri kontrol etmenin en basit yolunun komut sistemiyle olduğunu açıklayın:

KİM + NE (yapılması gereken) + NASIL (kaç adım, hangi yön, vb.)

Fiziksel bir oyunla alıştırmaya yapın «**Seni programlayacağım**»

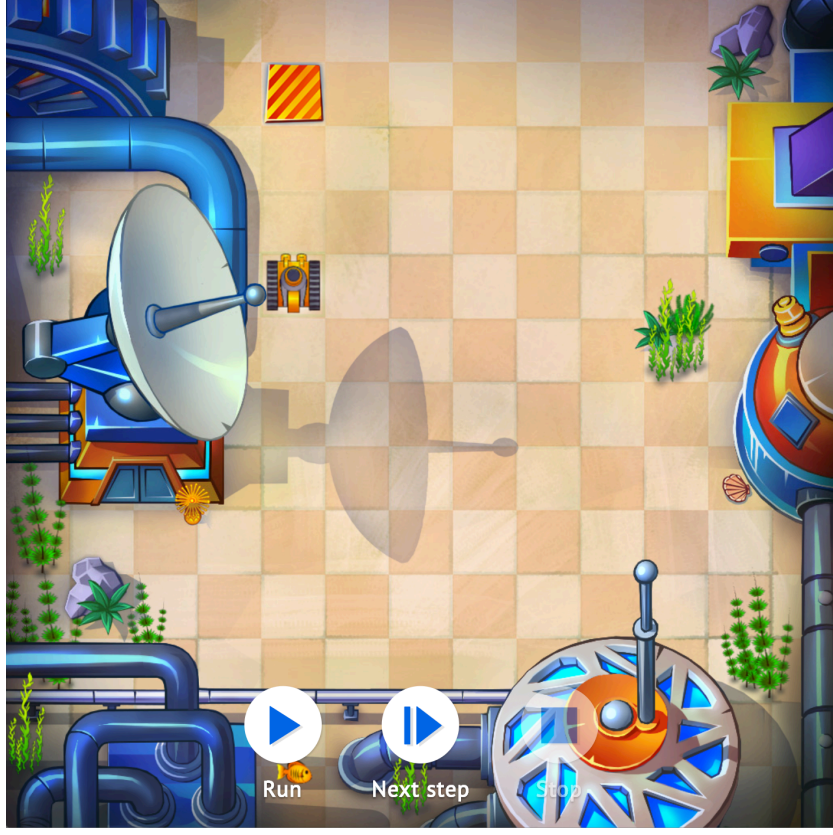
**1. Versiyon** Öğrenciler öğretmeni «KİM + NE + NASIL» kavramını kullanarak programlarlar.

Çocuklar komutu söyler - öğretmen uygular. (Ör. Öğretmen dön sağa)

**2. Versiyon** Öğrenciler tek tek tahtaya gelirler, ellerinde sembolik bir robot olur. Diğer çocuklar «robotu programlarlar», öğretmen komutları tahtaya yazar.

**Öğretmen:** «Hadi aynı şeyi bilgisayarlarımızda yapalım – Görev 3 ve Görev 4'ü yapın.»

## Görev N°3



**Görev:** İşaretli kareye ulaş. Kodu kendin yaz!

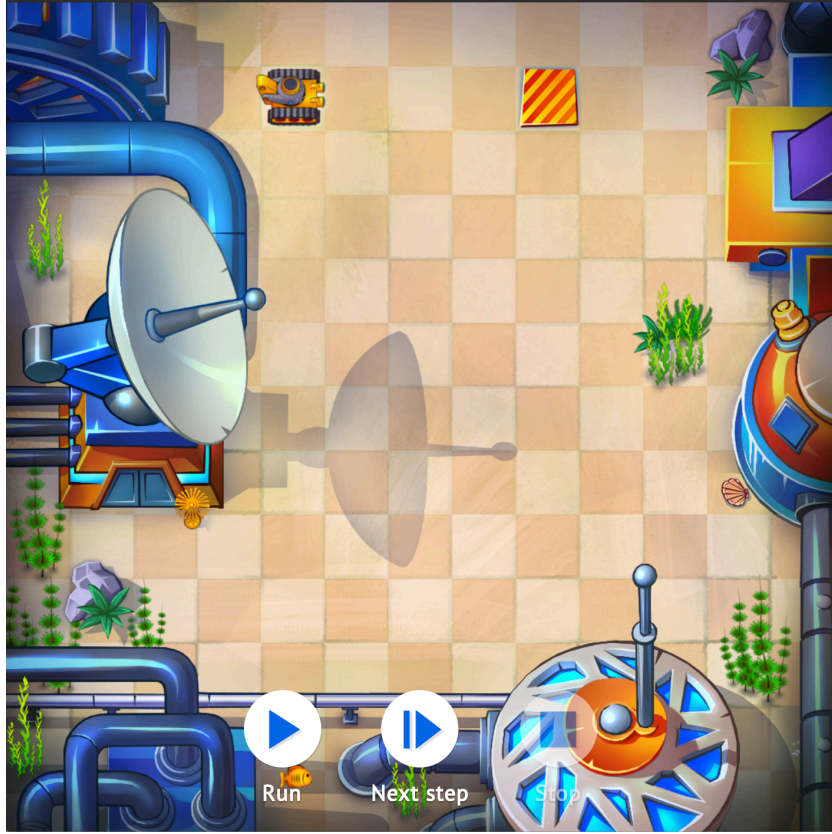
*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı ve «Çalıştır»a tıklayarak programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. robot.move 3

## Task N°4



**Görev:** İşaretli kareye ulaş. Nasıl yapacağını biliyorsun!


*Bu görevde kod yazılmamıştır. Öğrenciler kodu kendileri yazmalı ve programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. `robot.move 4`

## İpuçları & Hileler:

- Öğrenci Görev 4'e geçemezse – harita tuşunu gösterin 
- Öğrenci mesafeyi hesaplayamazsa – insan adımlarından ya da satrançtan bir örnek verin

Görevler tamamlandığında, ekranda Görev 4'ün nasıl çözüleceğini gösterin. Görevi yaparken kasten kodda bir hata yapın. Hatalı satırın rengi değişecektir:



Çocukların dikkatini robot olmadığımız ve sıklıkla hata yapabildiğimiz gerçeğine çekin. Bilgisayar dünyasında hatalar dostumuzdur. Hedefe ulaşabilmek için neyin düzeltilmesi gerektiğini bize gösterirler. Codewards sisteminin hatalı çizgiyi nasıl vurguladığını gösterin.

**Öğretmen:** «Kontrol ettiğiniz nesne yalnızca tek bir komut bildiğinde bu çok kolaydır. Ancak bazı mekanizmalar daha karmaşıktır. Örneğin çamaşır makinesi yünlüleri yıkamak için bir program kullanırken, kotları yıkamak için başka bir program kullanır. Bir mikrodalga fırın, yiyecekleri hem ısıtabilir hem de buzlarını çözebilir, vb.»

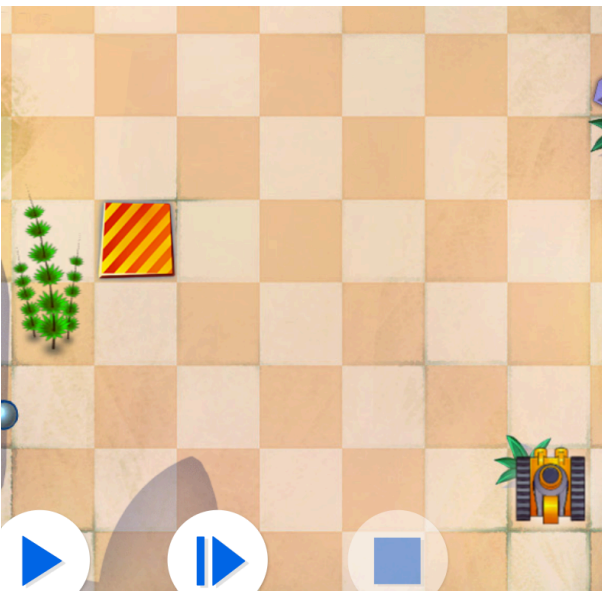
**Eğer mekanizma birçok farklı eylemi gerçekleştirebiliyorsa, komut sisteminin hangi «bölümleri» birden fazla seçeneğe sahip olabilir: «kim» mi, «ne» mi, «nasıl» mı?**

Öğrencilerin çoğunluğu «ne» diyerek doğru cevabı verecektir. Onları tebrik edin ve artık iki farklı «değişken alanı» oluştuğuna dikkat çekin:

1. «nasıl» (önceki görevlerde gördüğümüz gibi robotun adım sayısı değişebiliyor)
2. «ne» (bazı durumlarda robot çeşitli komutları gerçekleştirebiliyor).

Aşağıdaki görselde robotun hedefe ulaşması için kaç adet eylem gerçekleştirmesi gereklidir? Robot üç farklı eylem gerçekleştirmelidir - toplamda iki farklı komut kullanarak eylemleri tek tek gerçekleştirmelidir.

Önceki görevlerde hedefe ulaşmak için yalnızca tek bir komut kullanıyorduk.



Komutları, mekanizmayı hedefe götüren bir sırada yazdığımızda, bu diziye **algoritma** - tek tek gerçekleştirilen komut dizisi - denir.

- **Görev 7'ye gidin**

Yeni bir komut öğrenelim – **rotate (dön)**

`rotate` – dön. Bu komut yönlerle birlikte kullanılır - ne tarafa dön? Left (sol) ya da right (sağ).

Komutun tamamlanmış hali şu şekildedir:

```
robot.rotate left  
robot.rotate right
```

Eğer robotu ne tarafa döndüreceğinizi bilmiyorsanız, öncelikle robotun ne yöne baktığını bulun, daha sonra kendinizi robotun yerine koyarak ne tarafa döneceğinizi düşünün.

Programı yazmadan önce, robotun istenilen noktaya hangi yoldan gideceğini hayal etmeniz gerektiğini unutmayın. Daha sonra bu işlemi robotun bildiği komutlardan oluşan net komut dizilerine bölün. Ve son olarak bu adımları kodlama dilinde yazın.

Temelde tüm programcıların yaptığı budur - planladıklarını robotlara yaptırmak için özel algoritmalar (komut dizileri) yazarlar.

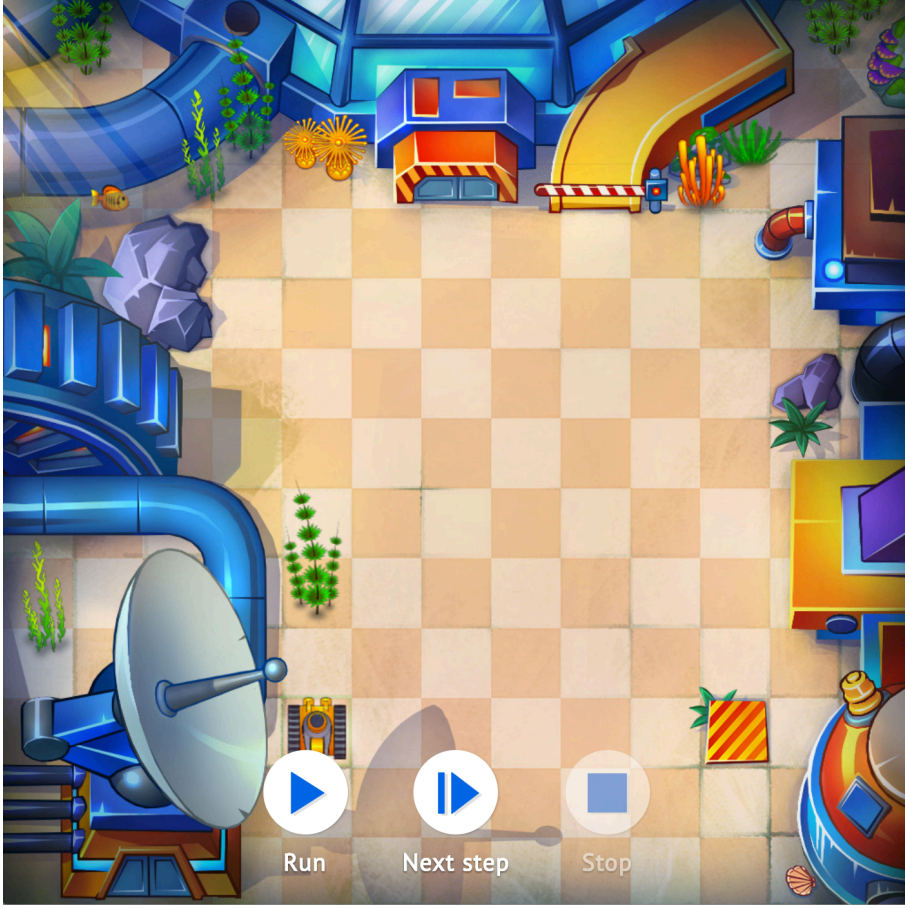
Görev N°7, N°8 ve N°9'u yapın.

Öğrencilere görevleri daha hızlı yapmalarında yardımcı olabilmek için aşağıdaki soruları sorabilirsiniz:

- Robotun kaç eylem gerçekleştirilmesi gerekiyor?
- Eylemi gerçekleştirebilmesi için hangi komutlar gerekiyor?
- Komut nasıl değiştirilmelidir?



## Görev N°7



**Görev:** İşaretli kareye ulaş. Programda bir şeyler eksik görünüyor.

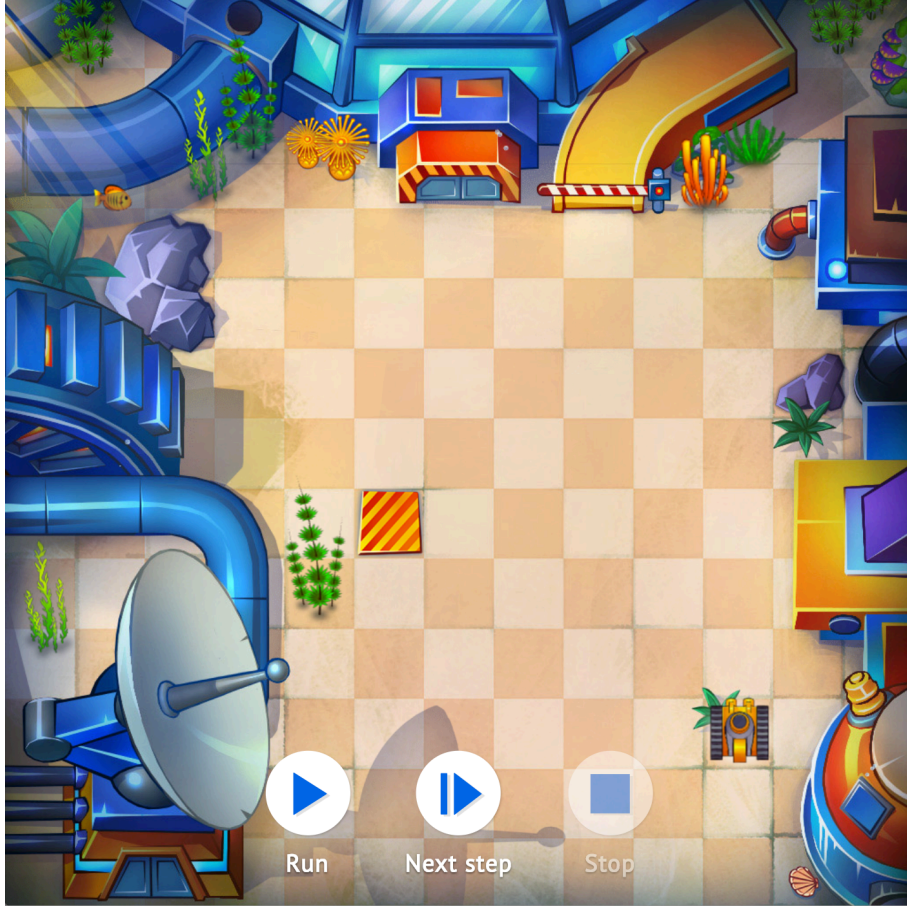
*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. `robot.rotate right`
2. `robot.move 6`

## Görev N°8



**Görev:** İşaretili kareye ulaş. Kodu kendin yaz.

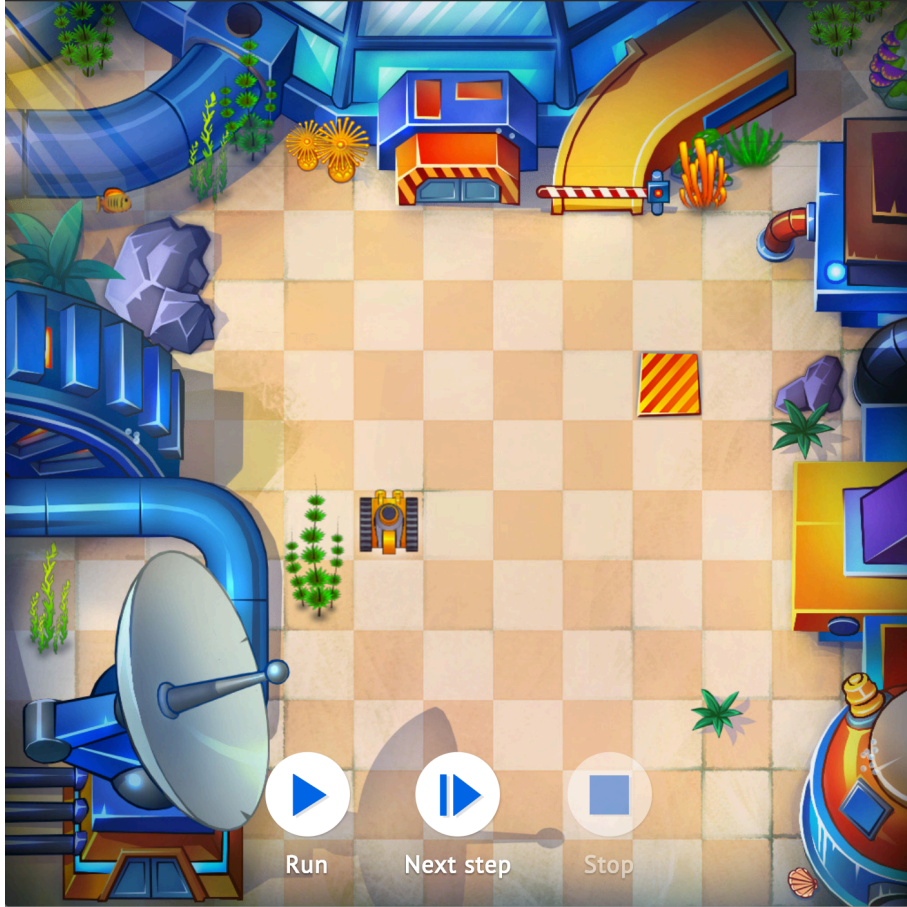
*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. robot.move 3
2. robot.rotate left
3. robot.move 5

## Görev N°9



**Görev:** İşaretli kareye ulaş. Hadi!

*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. robot.move 2
2. robot.rotate right
3. robot.move 4

## İpuçları & Hileler

- Öğrenci yanlış veya farklı bir şey yazdıysa ve metni nasıl sileceğini bilmezse, metni seçerek Backspace (klavyedeki silme tuşunu) kullanmayı gösterin.
- Öğrenci farklı ama çalışan bir algoritma yazdıysa ve görev için 3 yıldızdan daha az aldıysa, öğrenciye her şeyin yolunda olduğunu ve daha sonra bu göreve geri döneceğinizi söyleyin. Bu gibi durumlarda büyük ihtimalle yazılan algoritma gerekenden fazla satırdan oluşmaktadır.
- Eğer öğrenci yazdığı her satırdan sonra programı çalıştırıyorsa, Codewards sistemi «Tekrar dene» uyarısını gösterecektir. Programı çalıştırmadan önce tüm komut dizisini yazmaları gerektiğini onlara anlatın. Burada öğretilen temel beceri budur.

- **Görev 11'i açın**

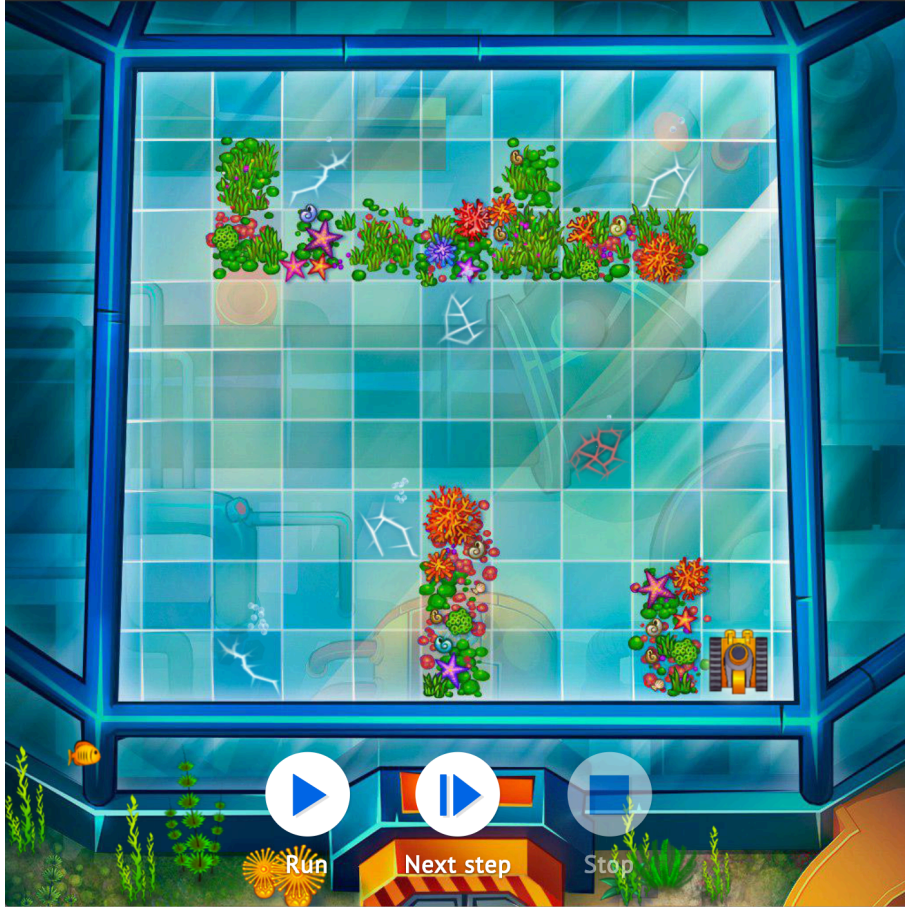
Aşağıdaki soruları sorun:

- Robot hedefe ulaşıyor mu? - evet
- Robot toplamda kaç kare ileri gidiyor? - 5
- Algoritmada kaç adet komut var? - 3
- Programı daha az satır kullanarak veya robotu daha az karede ilerleterek tekrar yazmak mümkün müdür? - hayır

Robotun daha verimli çalışmasını istiyorsanız **ideal** programı yazmanız gerektiğini, bu şekilde zaman ve diğer kaynaklardan tasarruf etmiş olacağınızı açıklayın.

Görev N°11 ve N°12'yi yapın.

## Görev N°11



**Görev:** Çatlağı tamir etmek için programı çalıştır!

*Bu görevde kod hazır olarak yazılmıştır. Öğrencinin yalnızca «Çalıştır»a basması ve programı çalıştırması gereklidir.*

*Çatlağı tamir etmek için robotun çatlağın olduğu kareye ulaşması gerektiğini açıklayın. Tamir edilmesi gereken çatlak, rengi değiştirilerek vurgulanmıştır.*

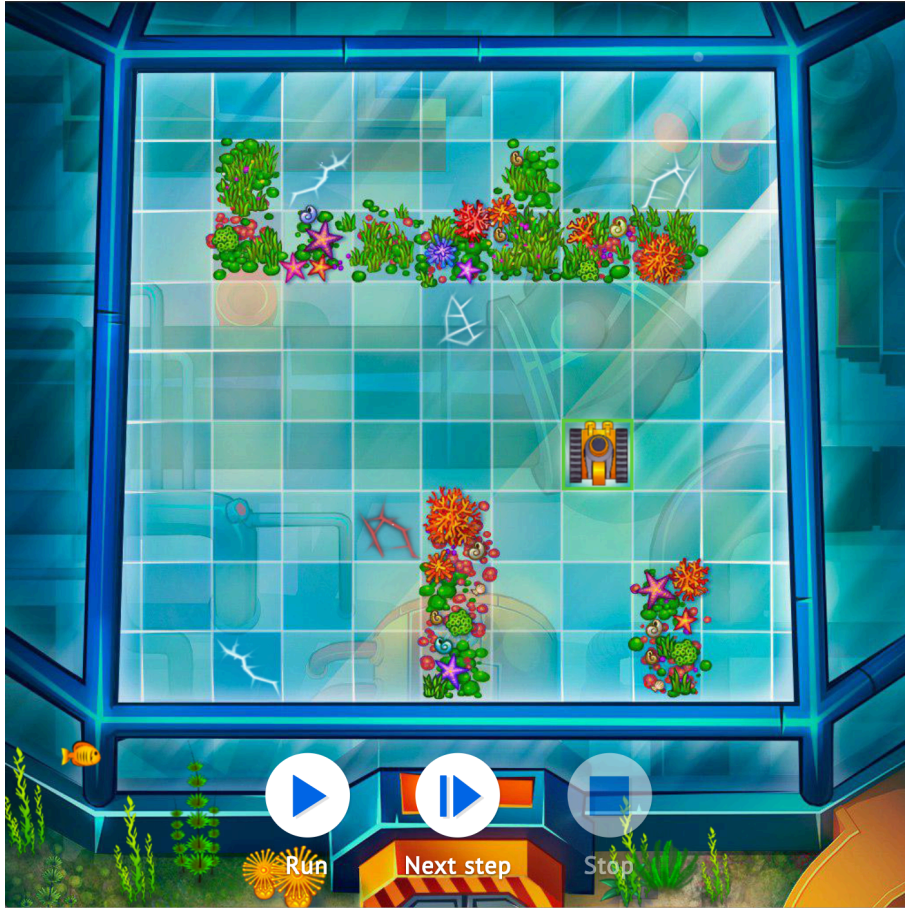
### Kaynak kod:

1. robot.move 3
2. robot.rotate left
3. robot.move 2

### Çözüm kodu:

1. robot.move 3
2. robot.rotate left
3. robot.move 2

## Görev N°12



**Görev:** Çatlağı tamir et. Eksik kodları yaz.

*Bu görevde kod tamamlanmamıştır. Öğrenciler eksik satırları tamamlayarak programı çalıştırmalıdır. Tamir edilmesi gereken çatlak, rengi değiştirilerek vurgulanmıştır.*

### Kaynak kod:

1. \_\_\_\_\_
2. robot.move 3
3. \_\_\_\_\_
4. robot.move 1

### Çözüm kodu:

1. robot.rotate left
2. robot.move 3
3. robot.rotate left
4. robot.move 1

**Öğretmen:** «Bugün öğrendiklerimizi gözden geçirelim»:

- Mekanizmalar ve robotlar, aşağıdaki yapıya sahip komutlarla kontrol edilir:  
KİM + NE + NASIL
- Bazı mekanizmalar yalnızca tek bir komut gerçekleştirebilir, ancak çoğu mekanizma birden fazla komut gerçekleştirebilir.
- Belirli bir hedefe ulaşmak için yazılan komutlar dizisine **algoritma** denir.
- Bir algoritma **ideal** olabilir veya olmayabilir. İdeal olmayan algoritma daha fazla kaynak kullanır. Vaktimizi ve enerjimizi korumak için yalnızca ideal algoritmalar yazmalıyız.

**Öğretmen:** «Geriye son bir önemli görev kaldı - Hadi!»

Bu görevde yeni eklenenler:

1. Bu robot birçok komutu gerçekleştirebilir.
2. Yeni komutlar: **load** (yükle) - boru parçasını alır, **put** (koy) - boru parçasını bırakır / yerleştirir.
3. Bu görevin birden fazla çözümü vardır - çocuklar ideal olanı bulmalıdır.

## Görev N°20



**Görev:** Boru parçasını "load" komutu ile vince yükle. İşaretli kareye ulaşınca "put" komutu ile boru hattına yerleştir. Kodu kendin yaz.

*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.*

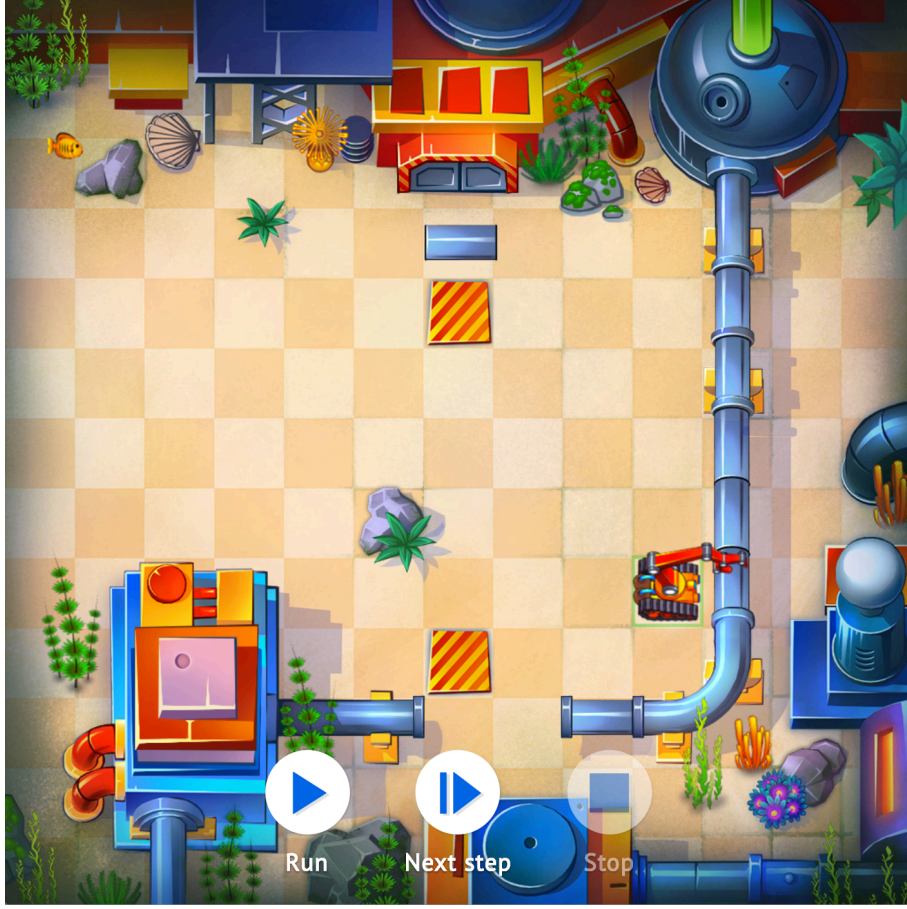
**Kaynak kod yok**

**Çözüm kodu:**

1. crane.load
2. crane.move 3
3. crane.put



## Ekstra Görev N°6



**Görev:** Yukarıdaki boru parçasını işaretli kareden vince yükle (load). Aşağıdaki işaretli kareye gelerek boru hattına yerleştir (put). Kodu kendin yaz.

*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. crane.rotate right
2. crane.rotate right
3. crane.move 3
4. crane.rotate right
5. crane.move 4
6. crane.load
7. crane.rotate right
8. crane.rotate right
9. crane.move 5
10. crane.put

**Öğretmen:** «Hayatımızda bazı şeylerin tekrar ettiğini fark ettiniz mi? Düzenli olarak gerçekleşen şeyleri bir düşünelim. Bunlar nelerdir?»

Örneğin: mevsimler, tatiller, vb.

Doğum günü şarkısından bir bölümü inceleyelim:

*Mutlu yıllar  
Mutlu yıllar  
Mutlu yıllar  
sana.*

Tekrarlanan satırları işaretleyin.

**Öğretmen:** «Aynı satırları tekrar tekrar yazmak sıkıcı değil mi?»

Aynı satır üç kez tekrar ediliyor. Aynı şeyleri tekrar eden bir aracımız olsaydı harika olmaz mıydı?

Böyle bir araç aslında var, adı da **döngü**. Nasıl çalıştığını anlamak için bir oyun oynayalım.

Öğretmen nesne olur ve belirli bir noktaya ulaşması gereklidir. Bunun için de iki adet komutu dörder kez tekrarlaması gerekmektedir.

*öğretmen.ilerle 4  
öğretmen.dön sağ*

Dört kere olmalıdır.

**Öğretmen:** «Hadi programı çalıştıralım»:

Öğretmen: «Döngü açık. Kaç kere tekrarlanacak?»

Öğrenciler: «Dört»

Öğretmen: «Ne yapmalıyım?»

Öğrenciler: «Dört adım ilerlemelisin»

Öğretmen: «Başka?»

Öğrenciler: «Sağa dönmelisin»

Öğretmen belirlenen kareye ulaşır.

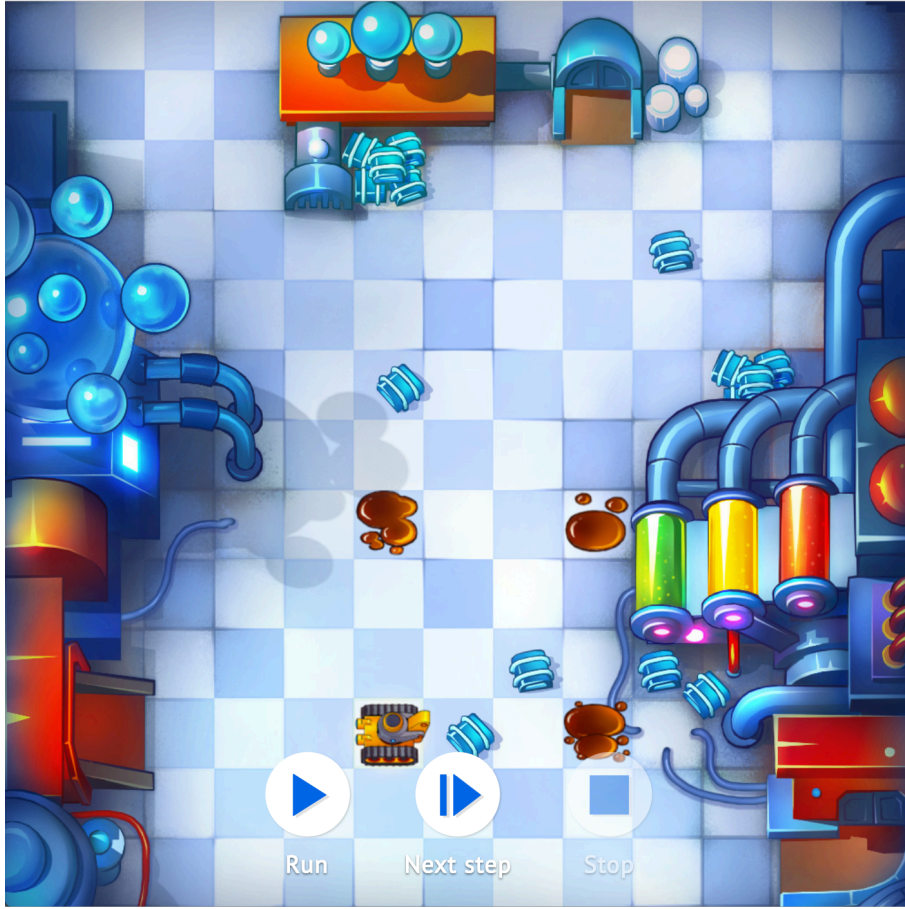
- **Görev 46'ya gidin.**

**Öğretmen:** «Hadi döngüyle birlikte kodun nasıl görüldüğünü ve robotun nasıl hareket ettiğini görelim.»

Belirtilen «loop» (döngü) sayısı kadar işlem yapılmadan, «son komut» işleminin yapılmadığına dikkat çekiniz. Program son komutu ancak döngü tamamlandıktan sonra gerçekleştirecektir.

Programın, tekrar eden komutlardan oluşmasını istiyorsak, bu komutları döngülerin içlerine yerleştirmemiz gerektiğini açıklayınız.

## Görev N°46



**Görev:** Robotun aynı eylemleri birden çok kez gerçekleştirmesi gerektiğinde, komutları tekrar yazmak yerine döngü (loop) kullanırız. Nasıl çalıştığını görelim. Kod zaten yazılı, programı çalıştır.

*Bu görevde kod hazır olarak yazılmıştır. Öğrencinin yalnızca «Çalıştır»a basması ve programı çalıştırması gereklidir.*

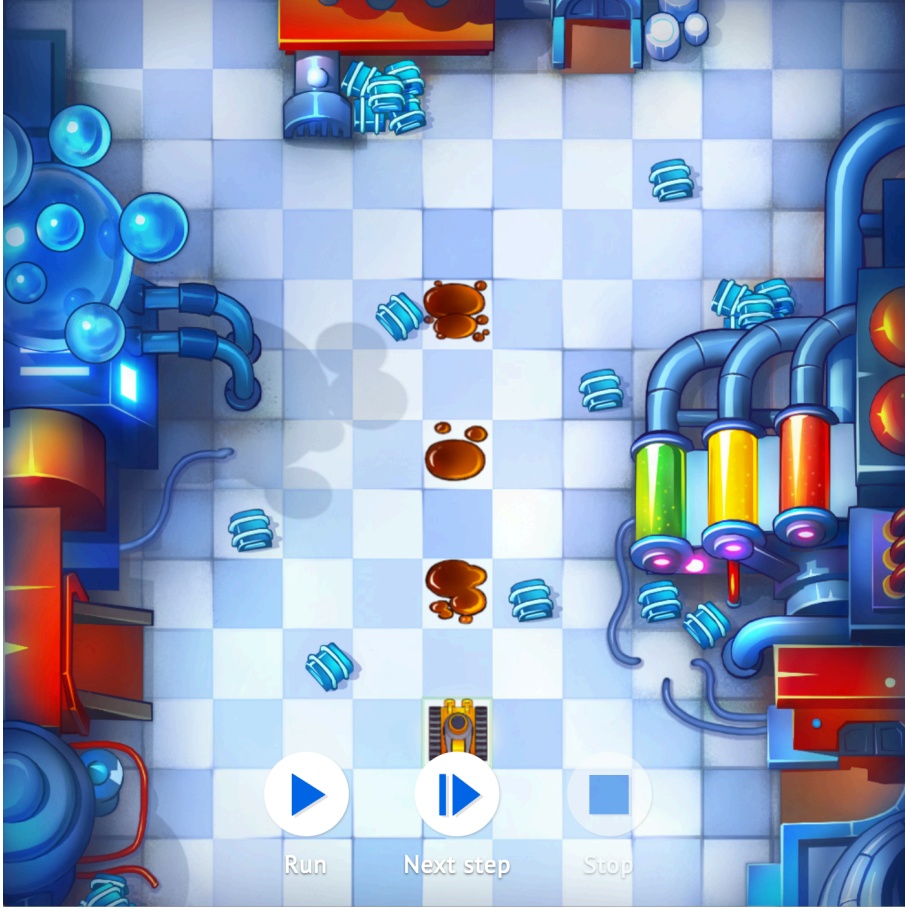
### Kaynak kod:

1. loop 3
2. robot.rotate right
3. robot.move 3
4. end

### Çözüm kodu:

1. loop 3
2. robot.rotate right
3. robot.move 3
4. end

## Görev N°47



**Görev:** Yağ lekelerini temizle. Programı döngü (loop) kullanarak yaz.

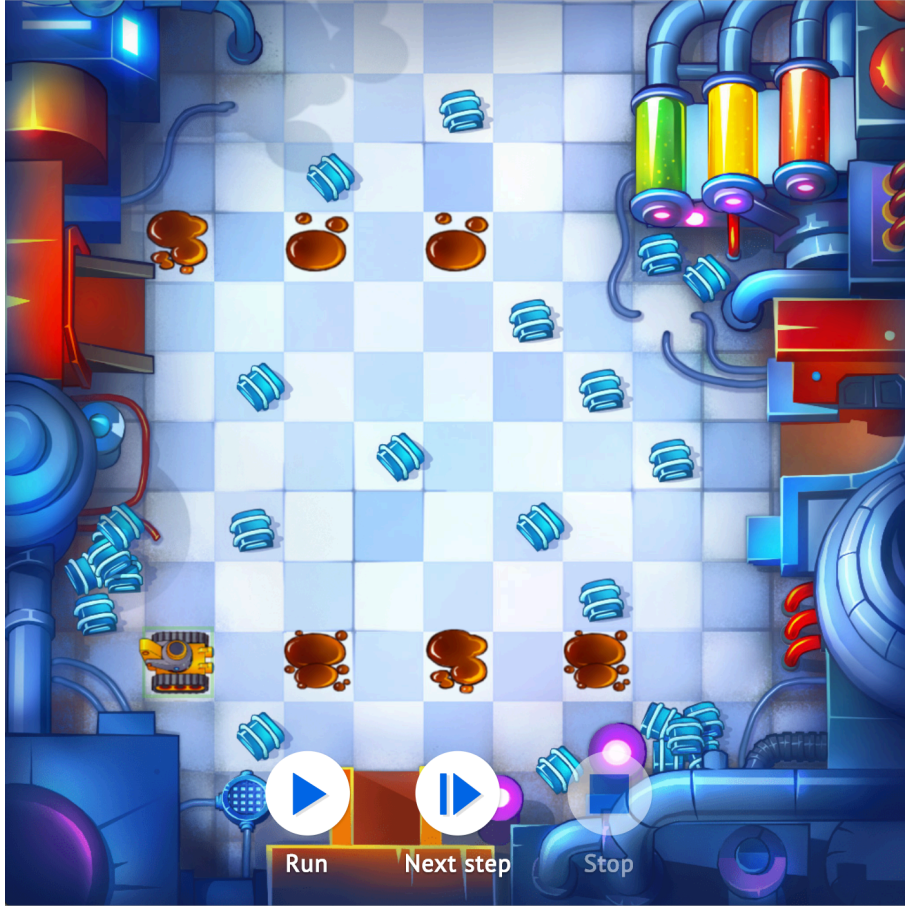
*Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.*

**Kaynak kod yok**

**Çözüm kodu:**

1. loop 3
2.     robot.move 2
3. end

## Görev N°50



**Görev:** Yağ lekelerini temizle. Kodu kendin yaz.

Bu görevde kod penceresi boştur. Öğrenciler kodu kendileri yazmalı programı çalıştırmalıdır.

**Kaynak kod yok**

**Çözüm kodu:**

1. loop 3
2. robot.rotate left
3. robot.move 6
4. robot.rotate right
5. robot.rotate right
6. robot.move 6
7. robot.rotate left
8. robot.move 2
9. end

### 3. Bölüm Kapanış

Yaklaşık süre – 5 dakika

**Öğretmen:** «Bugün su altı istasyonundaki boru hattını ve çatlakları tamir etmek için bilgisayar kodu kullandık. Harika bir iş çıkardık, ama hala birçok bozuk şey var. Umarım ekibimiz yeni ve heyecan verici görevleri gerçekleştirmek için tekrar bir araya gelir.»