

# Q-Learning ile Yol Planlaması

Kocaeli Üniversitesi Bilgisayar Mühendisliği Programlama Labaratuvarı 3 – 3. Proje

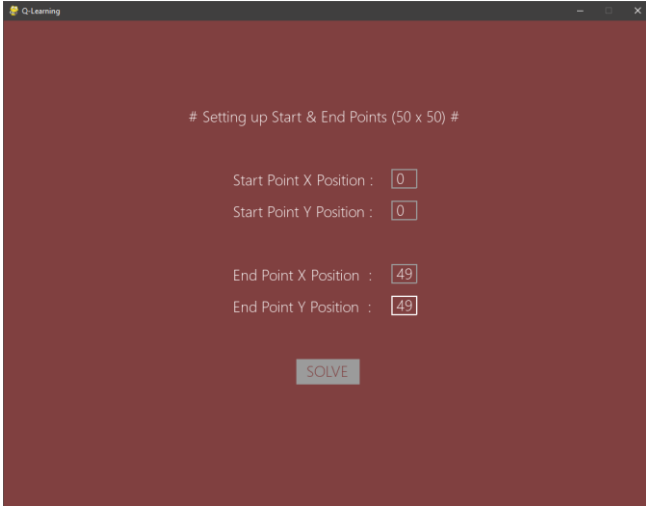
Emre Yelbey  
Kocaeli Üniversitesi  
Bilgisayar Mühendisliği  
180202043  
emre\_yelbey@hotmail.com

Ege Özeren  
Kocaeli Üniversitesi  
Bilgisayar Mühendisliği  
180202047  
ege\_ozeren@gmail.com

*Hazırlanan program 50 x 50 boyutunda belirli sayıda engeller içeren bir matris üzerinde Reinforcement Learning yaklaşımı kullanılarak yol bulmayı sağlayan bir uygulamadır.*

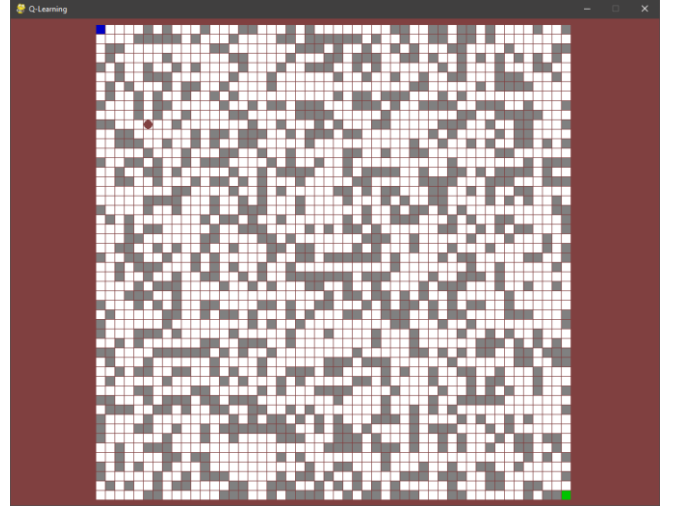
## I. GİRİŞ

Program çalıştırıldığı anda kullanıcıyı ajanın başlangıç noktasının ve ulaşacağı hedefin koordinatlarını girebileceği menü karşılar. Kullanıcı bu değerleri girip “solve” butonuna tıklayarak ya da değer girmeden doğrudan butona tıklayarak programı varsayılan değerler ile (0, 0 – 49, 49) başlatabilir.



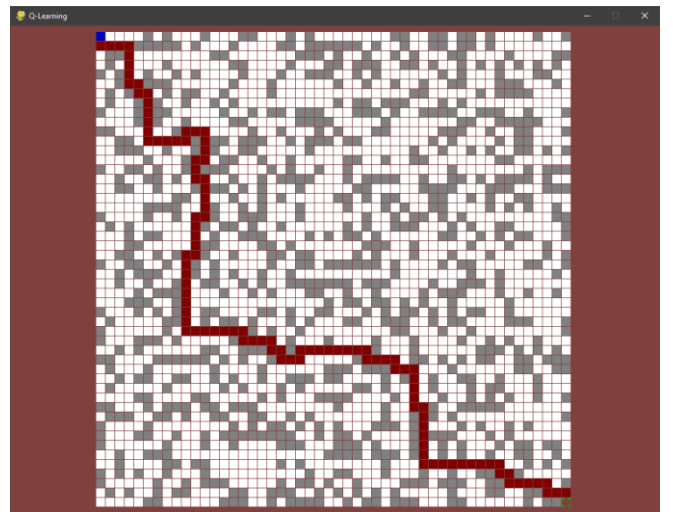
Screenshot 1: Değerlerin girilebildiği ana menü

“Solve” butonuna tıklandıktan sonra ajan Q-Learning algoritmasını baz alarak oyun alanı üzerinde hareket etmeye başlar. Ajan hedefe optimum yoldan ulaşacak şekilde eğitilene kadar hareketine devam eder.



Screenshot 2: Ajan hedefe ulaşan yolu Q-Learning algoritması ile hesaplar

Hedefe ulaşan yol hesaplandığında Q-Learning algoritmasının görevi sonlanır ve oyun alanı üzerinde hesaplanan yol çizdirilir. Ek olarak her episode için kaydedilen cost ve step değerleri grafik üzerinde gösterilir.



Screenshot 3: Hesaplanan yolun oyun alanı üzerinde gösterimi

## II. TEMEL BİLGİLER

Program **Python** programlama dilinde geliştirilmiş olup geliştirme ortamı olarak **Visual Studio Code 1.56.2** kullanılmıştır.

## III. TASARIM

Q-Learning ile yol planlaması programının geliştirme aşamaları belirtilen başlıklar altında açıklanmıştır.

### A. Algoritma

Programda kullanılan Q-Learning algoritması Reinforcement Learning yaklaşımı üzerine kurulu bir makine öğrenmesi algoritmasıdır. Reinforcement Learning yaklaşımında amaç; dinamik bir ortamda bir takım eylemler gerçekleştiren bir ajanın, ödül miktarını maksimuma çıkarmaktır. Bu yaklaşımda ajanın yeni eylemleri önceki eylemlerinin sonuçlarına dayalıdır. Bir diğer deyişle ajan önceki eylemlerinden ders çıkarır ve yeni eylemlerini maksimum ödül elde edebilme doğrultusunda buna göre meydana getirir.

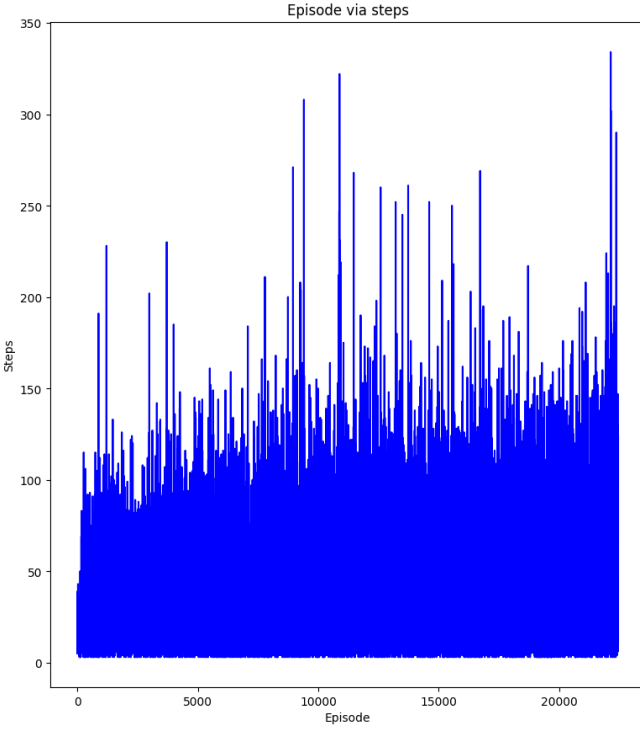
Q-Learning algoritması bu yaklaşım üzerine kurulmuş bir Reinforcement Learning algoritmasıdır. Algoritmanın adında bulunan “Q” harfi “quality” kelimesini temsil eder. Buradaki “quality” gerçekleştirilen eylemin ödül kazanma anlamında ne kadar etkili olduğu anlamına gelir.

### B. Yöntem

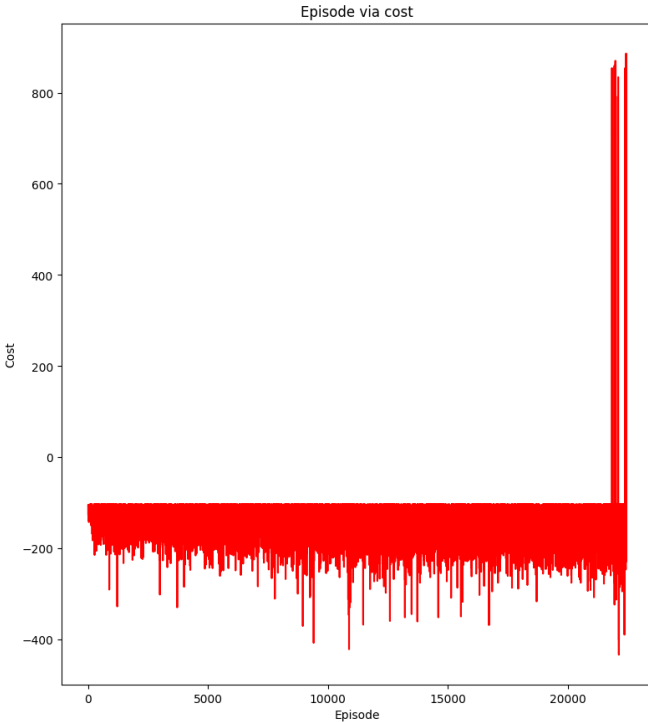
Programda ajan 50x50 boyutunda üzerinde engellerin bulunduğu matriste hedefe ulaşan optimum yolu Q-Learning algoritmasıyla bulmaya çalışır.

Bunun için öncelikle bir Q-tablosu oluşturulur. Q-tablosu her durumda gelecekteki maksimum ödülü hesaplamak için kullanılan veri yapısıdır. Daha açık ifadeyle Q-tablosu ajanı her durum için ödül miktarı bakımından en karlı eyleme yönlendirir. Q-tablosu başta tamamen sıfırlardan oluşur ve ajan eylemlerine devam ettiği süre boyunca güncellenmeye devam eder. Q-tablosunu güncellemek için Bellman denklemi kullanılır. Bu denklem şu şekildedir:

$$Q(s, a) = r(s, a) + \gamma \max_a Q(s', a)$$



Screenshot 4: Her episode için atılan adım sayısı grafiği



Screenshot 5: Her episode için maliyet miktarı grafiği

Bellman denkleminde:

*s*: *state* (durum)

*a*: *action* (aksiyon)

*y*: *discount rate* (azaltma katsayısı)

ifade eder.

Q-Learning algoritması Q-tablosu optimum değerlere ulaştığında sonlanır. Q-tablosunun optimum değerlere ulaşması için aşağıdaki adımlar uygulanır:

1. *Bir aksiyon seçilir:*  
Programa göre bu aksiyonlar “sağa, sola, aşağı, yukarı” olmak üzere 4 tanedir.
2. *Aksiyon uygulanır:*  
Ajan seçilen aksiyonu gerçekleştirir.
3. *Ödül hesaplanır:*  
Gerçekleştirilen aksiyon sonucu elde edilen ödül miktarı tespit edilir.
4. *Q-tablosu güncellenir:*  
Seçilen aksiyon ve elde edilen ödül miktarına göre Q-tablosu güncellenir.

### C. Kullanılan Bazı Metodlar ve Classlar

- *def createGameBoard(start\_x, start\_y, end\_x, end\_y):*

Bu metod parametre olarak aldığı *start* ve *end* pozisyonlarına uygun olarak belirli sayıda engellerin bulunduğu bir oyun alanı oluşturur ve oluşturulan oyun alanını *engel.txt* dosyasına yazdırır.

- *def learn(self):*

Bu metod Q-Learning algoritması kullanarak ajanı optimum şekilde hedefe ulaşacak seviyede eğitir.

- *class MainMenu:*

Bu class program ve kullanıcı arasındaki etkileşimi sağlayan GUI komponent ve methodlarını içerir.

- *class Squares:*

Bu class oyun alanındaki her bir karenin oluşturulmasında kullanılan classtır.

- *class Environment:*

Bu class Q-Learning algoritmasında kullanılan değişken ve metodlara sahip olan classtır.

### D. Karşılaşılan Bazı Sorunlar

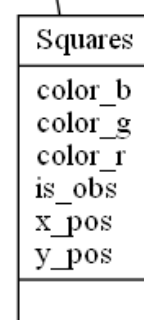
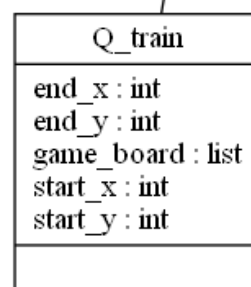
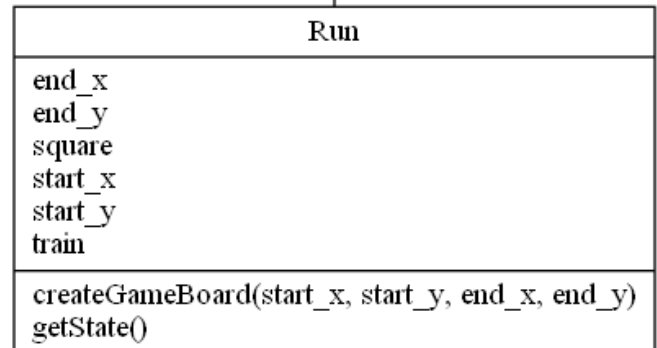
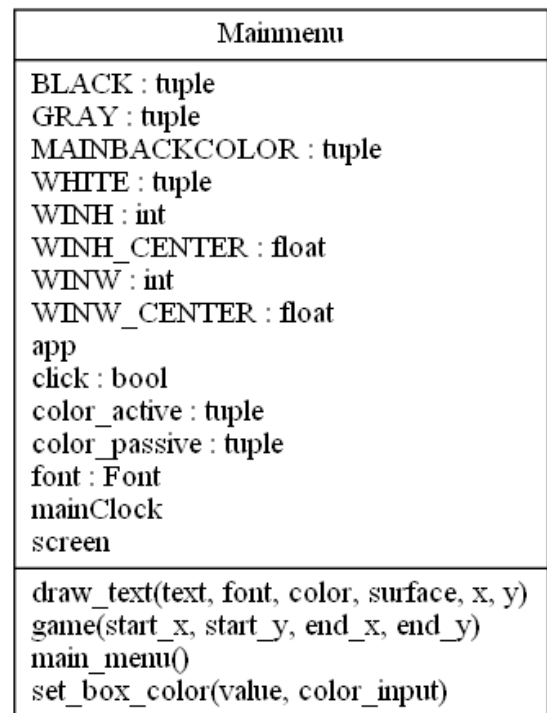
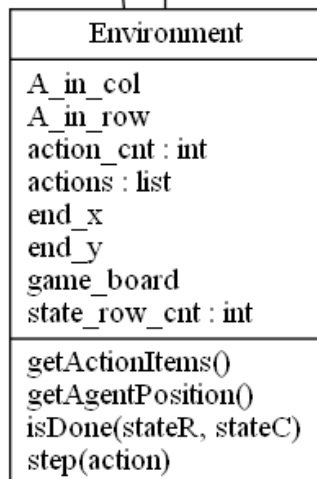
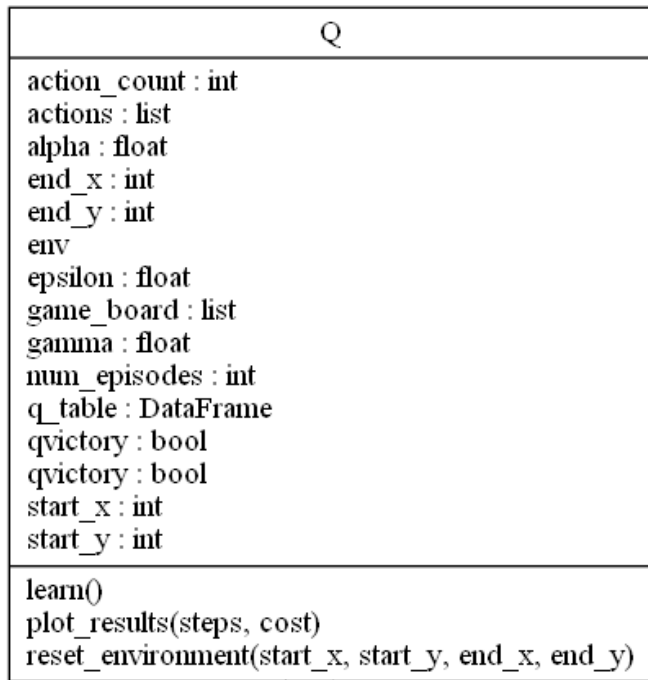
- Karşılaştığımız ilk sorun programın matris boyutu büyüdüğünde oldukça yavaş çalışmasıydı. Bunun nedenini anlayıp problemi çözmemiz biraz zaman aldı.
- İkinci bir sorun ise ajanın matris büyüdüğünde ilerlediği yolların hedefe ulaşmaktan uzak olmasıydı. Bu sorunu Q-Learning algoritmasında kullandığımız bazı parametreleri değiştirerek çözdük.

### E. Kazanımlar

- Pygame kütüphanesini kullanarak oyun yazmayı öğrendik.
- Pygame kütüphanesini kullanarak kendi GUI komponentlerimizi oluşturmayı öğrendik.
- Reinforcement Learning ve Q-Learning hakkında bilgi sahibi olduk.

## KAYNAKÇA

- [1] <https://realpython.com/pygame-a-primer/>
- [2] <https://www.youtube.com/watch?v=xxRhvyZXd8I&list=PLX5fBCkxJmm1fPSqgn9gyR3qih8yYLVmJ>
- [3] <https://towardsdatascience.com/a-beginners-guide-to-q-learning-c3e2a30a653c>
- [4] <https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56>
- [5] <https://medium.com/deep-learning-turkiye/q-learning-giri%C5%9F-6742b3c5ed2b>
- [6] <https://medium.com/@sddkal/python-ve-makine-%C3%B6%C4%9Frenmesi-q-learning-temelleri-181d29326782>
- [7] <https://medium.com/analytics-vidhya/introduction-to-reinforcement-learning-q-learning-by-maze-solving-example-c34039019317>
- [8] <https://becominghuman.ai/q-learning-a-maneuver-of-mazes-885137e957e4>
- [9] <https://www.youtube.com/watch?v=iKdIKYG78j4>
- [10] <https://www.youtube.com/watch?v=qhRNvCVVJaA>



env  
env

app

train

square

train

square