

# Qlearning ile Yol Planlaması

Yağızhan Şimşek– Emre Nihat Durgun

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

[180202087@kocaeli.edu.tr](mailto:180202087@kocaeli.edu.tr)

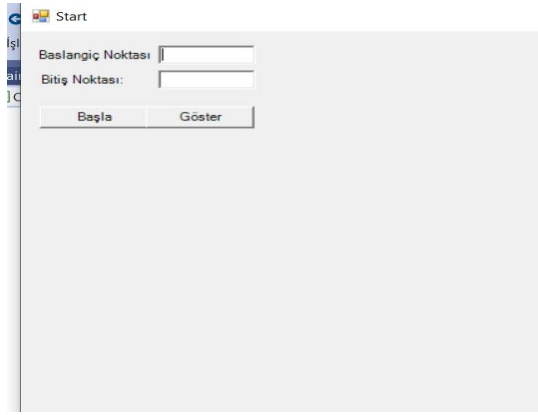
[180202015@kocaeli.edu.tr](mailto:180202015@kocaeli.edu.tr)

**Özet**–Projemizin temel amacı Qlearning yol algoritmasını kullanarak brute force bir şekilde hedefe ulaşan en optimize yolu bulmak

**Anahtar Kelimeler** – Qlearning, machine learning, C#, yol planlama, kısa yol algoritmaları, brute force

## I. AÇIKLAMA

Program ilk başladığında kullanıcıya bir başlangıç ekranı sunulmaktadır.



Ardından eğer kullanıcı göster tuşuna basarsa ise matrisimiz ve içindeki engeller ekrana yansıtılmaktadır.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

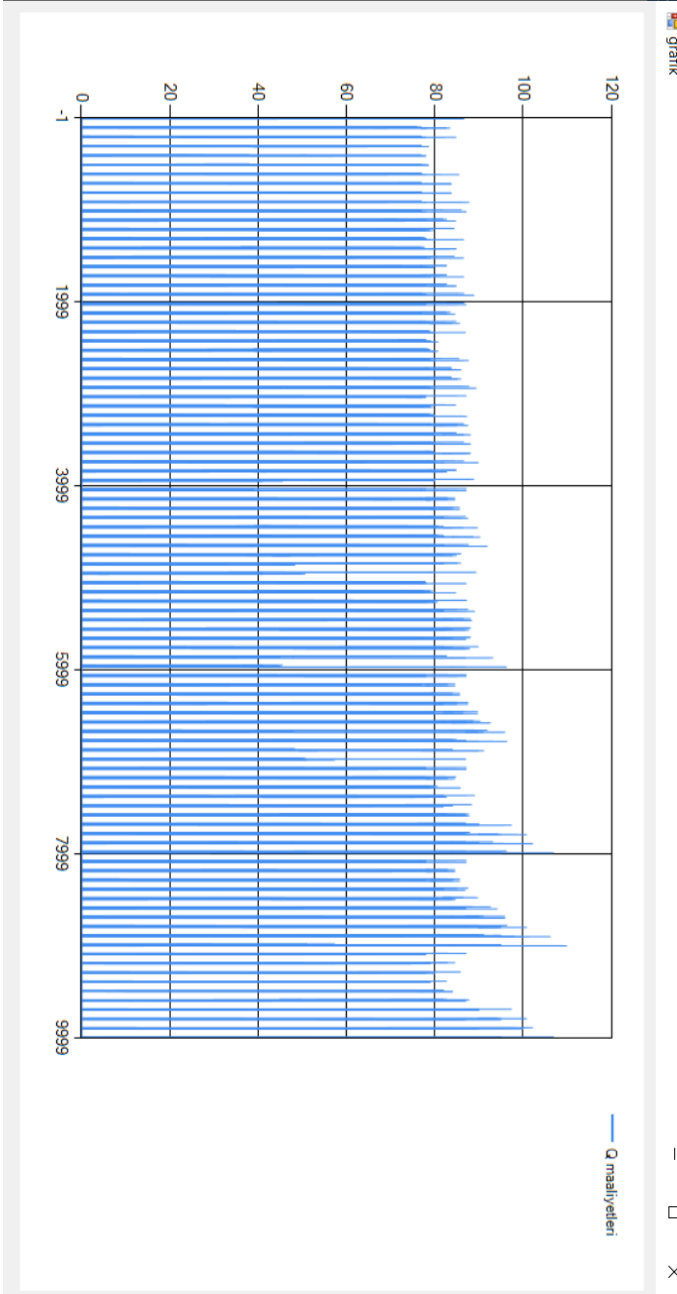
Eğer kullanıcı Başla butonuna basarsa ise program hesaplamaya başlar. Bu işlem birkaç dakika sürebilmektedir. Raporumuz için örnekte 0. Noktadan 99. Noktaya gidilmek istenmektedir.

Kullanıcının önüne gelen ekranlarından ilki optimum yolun ekrana çizdirilmiş halidir. Bu ekranda Qlearning algoritması kullanılarak en optimum yol bulunmaktadır. Şekilde görüldüğü üzere 0. Noktadan 99. Noktaya engellere çarpmadan geçilmiştir.

0(1)	1(2)	2	3	4	5	6	7	8	9
10	11	12(3)	13	14	15	16	17	18	19
20	21	22	23(4)	24	25	26	27	28	29
30	31	32	33(5)	34	35	36	37	38	39
40	41	42	43	44(6)	45	46	47	48	49
50	51	52	53	54	55(7)	56	57	58	59
60	61	62	63	64	65	66(8)	67	68	69
70	71	72	73	74	75	76	77(9)	78	79
80	81	82	83	84	85	86	87	88(10)	89
90	91	92	93	94	95	96	97	98	99(11)

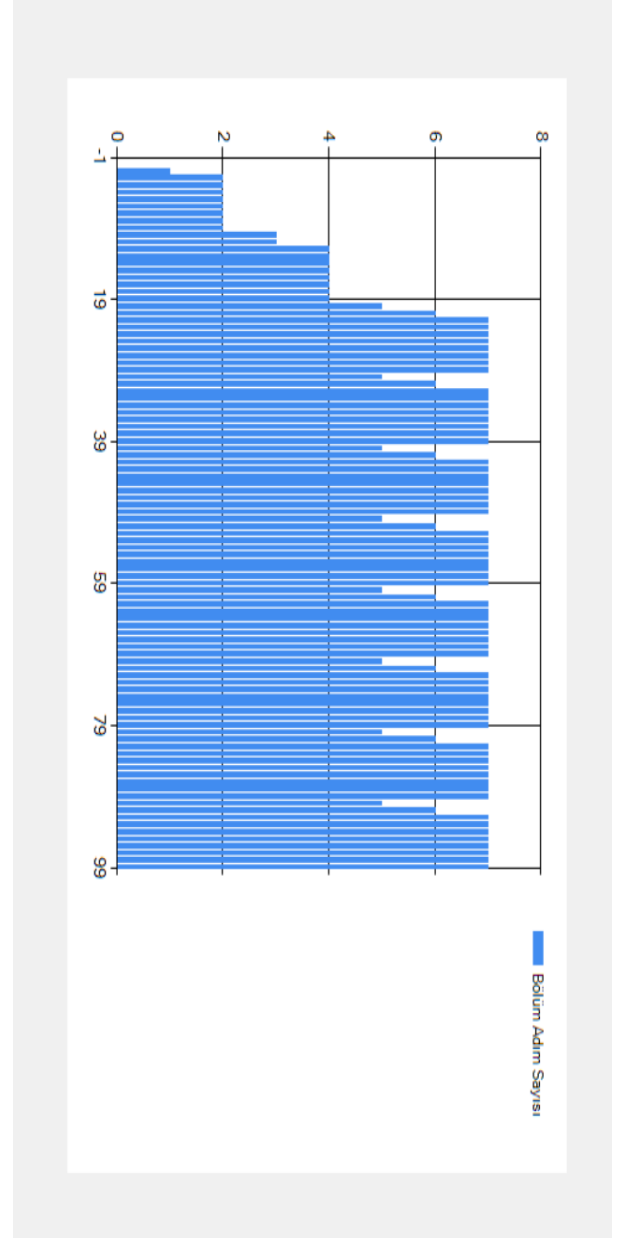
```
e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e
```

Kullanıcının önüne gelen diğer ekran ise uygulama öğrenme aşamasında her bir yol denediğinde hesapladığı maliyetleri hesaplama sırasına bir grafik haline getirmesini gösterir.



Bu ekranda resimde de görülebileceği üzere bir maliyet ve hamle grafiği bulunmaktadır.

Bir sonraki ekranda ise prgram hesaplama yaparken öğrene aşamısında her seferinde gittiği adım sayısı ekrana grafik halinde gösterilmektedir.



Bu ekranda resimde de görülebileceği üzere adım ve hamle grafiği görünmektedir.

Ayrıyeten program başlatıldığında engeller.txt isimli bir dosyaya engeller oluşturulurken her bir engelin yeri kayıt edilmektedir.

1. engel 25  
2. engel 53  
3. engel 81  
4. engel 79  
5. engel 29  
6. engel 47  
7. engel 77  
8. engel 11  
9. engel 35  
10. engel 52  
11. engel 67  
12. engel 51  
13. engel 45  
14. engel 54  
15. engel 13  
16. engel 88  
17. engel 95  
18. engel 56  
19. engel 58  
20. engel 37  
21. engel 35  
22. engel 76  
23. engel 29  
24. engel 0  
25. engel 90  
26. engel 52  
27. engel 91  
28. engel 50  
29. engel 81  
30. engel 95

## II. SÖZDE KOD

### START:

- 1 - program start formunda başlar.
  - 2- program engel üret fonksiyonunda engelleri üretir.
  - 3- üretilen engelleri block dizisine at
- başlangıç ve bitiş noktalarını girdikten sonra göster  
botton una bastığımızda:
- EngelUret Fonksiyonu :
- random değerleri (30 adet) oluştur ve başlangıçta -99 değerini üret
  - 30 defa random değer oluştur ve kontrol ederek bu değerleri engel dizisine ekle
  - ardından üretilen değerleri teker teker txt dosyasına yazdır.
  - engeli dizisini döndür.

göster buttonuna basıldığında :

- 1- counter isimli bir sayac oluştur.
- 2- 10 a 10 bir buton matrisi oluştur.
- 3- eğer block dizisi ile sayacdaki değeri içeriyorsa, butonun rengini kırmızı yap ve bastır.

4-counter'ı 1 arttır.

başlat buttonuna basıldığında :

1-qlearning classından bir değişken oluşturulur.

2- oluşturduğu classın constuructoruna 0,8 (delta) ve r matrisi oluşturmak için kullandığımız roomsProblem classından bir obje gönderilir.

(bu iki class'da daha sonra detaylı olarak açıklanacaktır.)

3- qmatrisini normalizedMatrix isimli bir değişkende tutturulur.

4- matrisin öğelerini bir arraylistte durulur.

5- arayüz kısmına önceden aldığımız başlangıç noktasını, bitiş noktasını, adımları ve engelleri göndeririz.

6- ilk grafiğin yazdırılacağı arayüze de daha önceden qmatrisinin ogelerini ve sayacı göndeririz.

### MAİN:

1- Generate fonksiyonunu çalıştır.

Generate fonksiyonunda:

- 1- buttonlar oluşturulur.
- 2- sayac yardımıyla, gönderilen blocks dizisi sayacımızı içeriyorsa buttonu kırmızıya boya
- 3- sayac bitiş noktasına eşitse blocku yeşile boya
- 4- sayac başlangıç noktasına eşitse blocku yeşile boya
- 5- adımlar listesi sayacı içeriyorsa ve sayac bitiş noktasına eşit değilse, yolu mora boya.
- 6- sayacı arttır.

ROOMSPROBLEM : alınan engele ve bitiş noktasına göre bir R\_Matrisi oluşturur. Butonların üzerindeki numaralardan gider.

1- 10 a 10'lük bir matris için

2- öncelikle satırın tamamını yol yokmuş gibi gösterip -1 atar.

3- her bir numaraya bakar.

4- eğer yol varsa +3, engel varsa -5 yok varsa ve hedefse +5 değerini atar.

5- bu işlemleri r matrisinin 100 noktasına da uygular.

GetValidActions : bu metot matrisin engellere göre üretildiği yerdir.

1- üretilen matrisi, matris2 dizisine atar.

2- geçerli hareketleri tutmak için bir liste oluşturur ve bu hareketleri, oluşturulan matriste kontrol eder.

3- hareketleri listeye atar.

4- listedeki her adımı teker teker 2.grafiğe gönderir.

QLEARNİNG : Q matrisinin üretildiği yerdir.

QLearning Classında 2.grafik için gereken bölüm adım sayısı çalıştırılır.

1- constructor a proje dosyasında verilen 0,8 değerini ve qmatrisinin oluşturulacağı problemi (yani r matrisini) göndeririz.

2- qtablosunu hareket aksiyonlarına göre teker teker oluşturur.

3- qtablosunun işlemlerini TakeAction fonksiyonunda yapar.

TakeAction Fonksiyonu :

1- aksiyon alınır.

2- girilen değere göre (0,8) matematiksel işlemi yapar.

3- matematiksel değeri q tablosuna işler ve aksiyonu döndürür.

Run Fonksiyonu :

1- eğer gerekli değerler yoksa (başlangıç noktası) hata döndür.

2- başlangıç değerini al.

3- bitiş değerini al.

4- hareket değeri bitiş noktasına gelene kadar:

5- Adımları al.

6- Alınan adımları action metodunda tut.

7- adımları array formatında döndür.

GRAFİK-1 :

Q matrisinin maaliyet grafiğini sayac ve gönderilen dizinin değerlerinin grafiğini oluşturur.

GRAFİK-2 :

Gönderilen q matrisinin adım sayılarını ve kaçınıcı adımda bulunduğunun grafiğini yazdırır.

### III. KAYNAKÇA

- [1] Doğan, Ö. (2019, March 18). *Q-learning'e Giriş*. Medium. <https://medium.com/deep-learning-turkiye/q-learning-giris-6742b3c5ed2b>
- [2] Doğan, Ö. (2019, May 13). *Python ile Q-learning*. Medium. <https://medium.com/deep-learning-turkiye/python-ile-q-learning-ef6413aa896e>
- [3] *Q-learning*. (2004, December 15). Wikipedia, the free encyclopedia. Retrieved May 18, 2021, from <https://en.wikipedia.org/wiki/Q-learning>
- [4] *Q-learning için Yeni Başlayanlar Kılavuzu*. (2019, November 17). ICHI.PRO. <https://ichi.pro/tr/q-learning-icin-yeni-baslayanlar-kilavuzu-215378165392700>
- [5] Wang, M. (2021, February 1). *Deep Q-learning tutorial: MinDQN*. Medium. <https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc>
- [6] *Hands-on Q-learning with Python: Practical Q-learning with OpenAI gym, Keras, and TensorFlow: Habib, Nazia: Amazon.com.tr*. (n.d.). Amazon.com.tr: Elektronik, bilgisayar, akıllı telefon, kitap, oyuncak, yapı market, ev, mutfak, oyun konsolları ürünleri ve daha fazlası için internet alışveriş sitesi. <https://www.amazon.com.tr/Hands-Q-Learning-Python-Q-learning-TensorFlow/dp/1789345804>