

**KOCAELİ ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABARATUVARI 2**

**2. PROJE**

Q-learning ile Labirentte Yol Bulma

140201082 SÜMEYRA ÇAKMAK

140201083 MELEK PAZARBAŞI

Q-learning ile Labirentte Yol Bulma

1.Melek Pazarbaşı 2.Sümeyra Çakmak

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

melekp30@gmail.com sumeyra.cak@gmail.com

Özet

*Q-learning ile labirentte yol bulma programı giriş ve çıkışı bilinen bir labirentin girişinden çıkışına en kısa ve mantıklı şekilde nasıl gidileceğini bulan ve bunu bir arayüzle kullanıcıya sunan bir programdır.*

*Program açıldığında ekrana önce bir arayüz gelir ve bu arayüzden kullanılacak input dosyası seçilir. Input dosyasının seçimi yapıldıktan sonra kullanıcıdan başlangıç-bitiş noktası ve uygulanacak iterasyon sayısı istenir. Bu istenilen başlangıç bitiş noktalarına ve iterasyon sayısına göre Q ve R matrisleri oluşturulur. Oluşturulan matrisler textboxla kullanıcıya da gösterilir. Verilen iterasyon sayısına göre oluşturulan Q matrisinin bize verdiği değerlere göre labirentin yolu sayısal olarak bulunur ve ekranda textbox içinde kullanıcıya da gösterilir. Bu işlemler tamamlandıktan sonra Q matrisini kaydetme, R matrisini kaydetme ve bulunan pathi kaydetme adında 3 tane buton var. Bu butonları kullanarak istenilen matrisi ya da pathi bilgisayarda istenilen yere kayıt edebilme özgürlüğü kullanıcıya verilmiştir.*

*Kullanıcı labirentin çizimini görmek istediğinde ÇİZİMİ GÖR butonuna tıklayarak labirentin yolunu görebilir. Çizimi gör denildiğinde başka bir form sayfasına geçerek labirentin çizimini görür ama burada yol çizili değildir. Yolu çizdirmek için ise YOLU ÇİZ butonu kullanılarak labirentte başlangıçtan bitişe gidilecek en doğru yolu bulan yol ekrana çizilir.*

*Projenin kullanıcı dostu olabilmesi açısından kullanıcı yeni bir labirent çizdirmek istediğinde kullanıcıyı yönlendirecek butonlar projeye eklenmiştir. Ayrıca seçilen input dosyasına göre başlangıç ve bitiş noktasını kullanıcıdan alma işlemine sınırlandırmalar getirilmiştir. Örneğin input dosyası 5\*5lik bir dosyaysa başlangıç ve bitiş değerleri 25ten büyük girilemez.*

# Giriş

Kullanıcıdan başlangıç ve hedef düğümlerinin alınması sonucunda Q learning algoritmasının işletimine bağlı olarak kullanıcıya doğru yolun verilmesi ve bu yolun grafiksel olarak kullanıcıya gösterilmesi.

# Temel Bilgiler

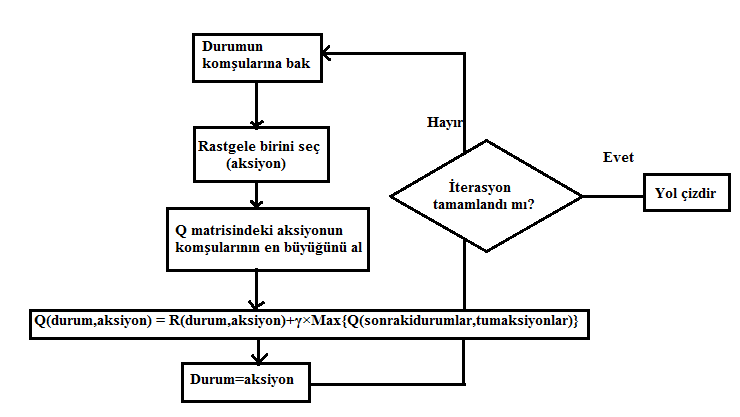
Q learning algoritması kullanılarak yapılmış olan labirentte yol bulma projesi Visual Studio ortamında C# dili kullanılarak hazırlanmıştır.

C# Programlama Dili, Microsoft'un geliştirmiş olduğu yeni nesil programlama dilidir. Yine Microsoft tarafından geliştirilmiş . NET Teknolojisi için geliştirilmiş dillerden biridir.

Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). [Microsoft Windows](https://tr.wikipedia.org/wiki/Microsoft_Windows), [Windows Mobile](https://tr.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://tr.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://tr.wikipedia.org/wiki/.NET_Framework), [.NET Compact Framework](https://tr.wikipedia.org/w/index.php?title=.NET_Compact_Framework&action=edit&redlink=1) ve [Microsoft Silverlight](https://tr.wikipedia.org/wiki/Microsoft_Silverlight) tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır.

Pekiştirmeli öğrenme olan Q learning algoritması, öznelerin bir görevi en yüksek kazançla tamamlayabilmek için hangi eylemleri gerçekleştirmeleri gerektiği ile ilgilenen bir makine öğrenmesi tekniğidir.

# Geliştirilen Mimari



*ŞEKİL 0 :* AKIŞ ŞEMASI

1- private void button3\_Click(object sender, EventArgs e)

Kullanıcı tarafından seçilen input dosyası ve kullanıcının girdiği başlangıç ve hedef değerlerine göre projeye giriş yapılmış olur.Bu input dosyasından verileri parçalayarak çeker. Bu input dosyasına göre oluşturulacak olan R ve Q matrisinin boyutları bulunur. Bu R ve Q matrisi oluşturulması şu şekilde sağlanır. R matrisi için tüm değerler başlangıç olarak -1, Q matrisi için ise başlangıç olarak tüm değerleri 0 olarak atanır. Daha sonrasında ise inputtan dosyaya veri çekilmeye başlanıyor . Bu çekilen değerler R matrisini yeniden güncelleyerek komşuluk matrisi oluşturuluyor. R matrisine göre komşuluk olan her sütun için ilgili satır ve sütuna 0 değeri atanıyor.

2- private void button1\_Click(object sender, EventArgs e)

Başlangıç ve bitiş noktaları alınan ve BAŞLA butonuyla işleme başlayan kod R matrisini bitiş noktasına göre tekrardan düzenleyerek kazanç matrisi oluşturur.Kodun devamlılığı şu şekilde olur; girilen bitiş değeri eğer komşuluk matrisinde sıfıra karşılık geliyorsa ilgili satır-sütun ve o sütun indexine ait satır sütun 100 olarak yenilenir.Bu şekilde R matrisi kazanç matrisine dönüştürülmüş olur. Kullanıcıdan başlangıçta alınan iterasyon sayısına göre döngü içerisinde durum bitişe eşit olana kadar döngü devam eder.

Bu döngü içerisinde Q learning algoritması işletilir.

2.1-public static int Random\_aksiyom(int[,] R, int baslangic,int ilk\_aksiyom)

Girilen başlangıç değerine göre R matrisinde 0 olan satırların içerisinden algoritmik olarak işletilen kodlarla random olarak bir adet sıfır a ait index seçer.Ve bu döndürülen değeri aksiyom değişkenine atar.

Qlearning algoritmasına atanan bu aksiyom değişkenine bağlı olarak Q kazanç yol matrisi yenilenir.

2.2-public static double Max(int secilen, int[,] R, double[,] Q)

Seçilen aksiyomun için R matrisinde aksiyom değerine karşılık gelen satırlardaki 0 veya 100 değerlerini bir diziye atıyor.

Bu diziyi ve bu dizinin boyutunu Max\_Deger\_Dondur() adlı fonksiyona göndererek en büyük değeri max fonksiyonuna dönüşümünü sağlıyor.

3-Yol bulma kısmında Q matrisi üzerinden başlangıç değerinden başlanarak o satırda hangi sütunun değeri daha büyükse ilgili sütun değeri bir sonraki adımda ilgili satır değeri oluyor ve kod algoritmik olarak hedefe ulaşılıncaya kadar devam ediyor.

4-private void button4\_Click(object sender, EventArgs e)

Oluşturulan R matrisinin bilgisayara txt biçiminde kaydedilmesini sağlayan butondur.

5-private void button5\_Click(object sender, EventArgs e)

Oluşturulan Q matrisinin bilgisayara txt biçiminde kaydedilmesini sağlayan butondur.

6-private void button6\_Click(object sender, EventArgs e)

Kullanıcıdan alınmış olan bilgilere ve iterasyona göre belli işlemler kullanarak elde edilmiş olan pathi txt dosyası olarak bilgisayara kayıt etmeyi sağlayan butondur.

7-private void canvas\_Paint(object sender, PaintEventArgs e)

Üzerine çizim yapabilmek için projenin tasarım kısmına panel ekledik ve o paneli paint olarak kullanmak için gerekli tanımlamaları yaptık.

8-private void path\_olustur()

Diğer formda yapılmış olan işlemlere göre bulunan pathi o formdan çekip path[] adıında diziye kaydeden fonksiyon.

9- private void paint\_lab(int boyut)

Seçilmiş olan input dosyasının boyutuna alıp fonksiyona yolladıktan sonra verilen değerlere göre sadece labirenti çizen fonksiyon

10- private void paint\_path(int[] path, int boyut)

Diğer formda bulunmuş ve bir matrise atanmış pathe göre daha önce çizilmiş olan labirentte yolu çizmeye yarayan fonksiyon

11- private void drawLineDikey()

Canvas adını verdiğimiz panele 50 pixel boyutunda dikey çizgiler çizen fonksiyon.

12- private void drawLineYatay()

Canvas adını verdiğimiz panele 50 pixel boyutunda yatay çizgiler çizen fonksiyon.

13- private void button1\_Click\_2(object sender, EventArgs e)

Pathi çizmek için canvas adını verdiğimiz panel üzerinde belli işlem yapan fonksiyon. Canvasa pathi çizdirmek için belli fonksiyonlar çağırılır.

14- private void button2\_Click(object sender, EventArgs e)

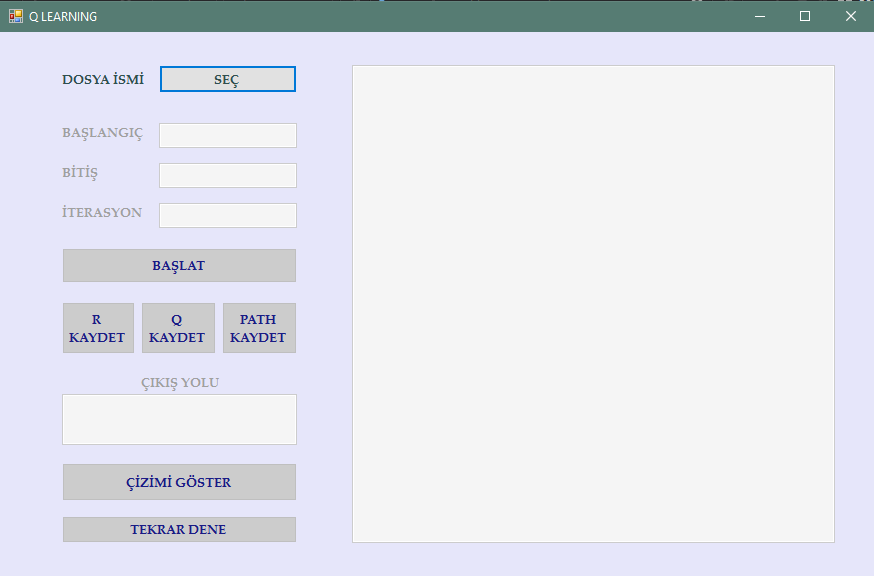
Kullanıcı dostu olması açısından kullanıcının istediği zaman diğer forma dönebilmesi için GERİ DÖN butonunun kodudur.

# Kullanıcı Kataloğu

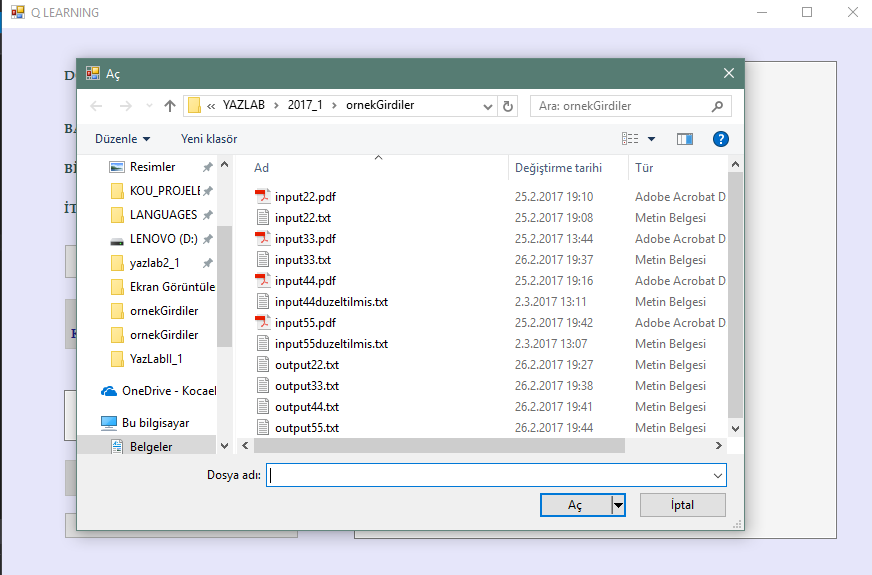
Proje açıldığında ekrandan önce input dosyası seçilmesi gerekiyor. Input dosyası seçilmeden diğer araçlar kullanılamaz. SEÇ butonuna tıklandıktan ve input dosyası seçildikten sonra başlangıç, bitiş ve iterasyon sayısı girilmesi gerekir. Başlangıç, bitiş ve iterasyon girildikten sonra BAŞLATa tıklanınca sağ taraftaki textBoxta R ve Q matrisinin önizlemesi görünecek bunun yanında sol tarafta çıkış yolu yazan textBoxta path görünecektir.

R kaydet, Q kaydet ve Path Kaydet butonları kullanılarak matrisler ve path kayıt edilebilir. Bu işlem de yapıldıktan sonra ÇİZİMİ GÖSTER denilerek diğer forma geçilir ve burada labirent görünür.

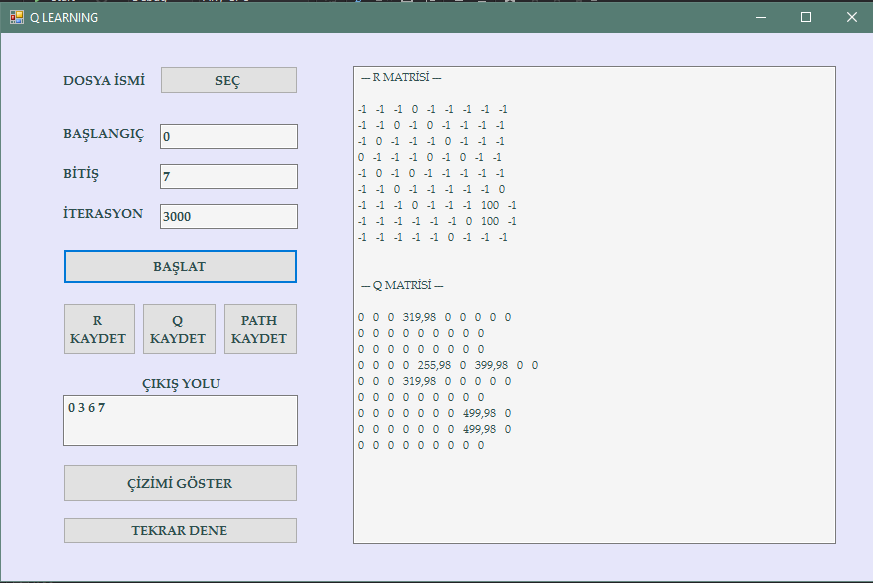
İkinci form sayfasına geçilince karşımıza seçilen input dosyasına göre oluşan labirent gelir. Burada yolu görebilmek için YOLU ÇİZ butonuna tıklamamız gerekecektir. Bu butona basılınca labirente en uygun yol ekranda çizilerek gösterilir.



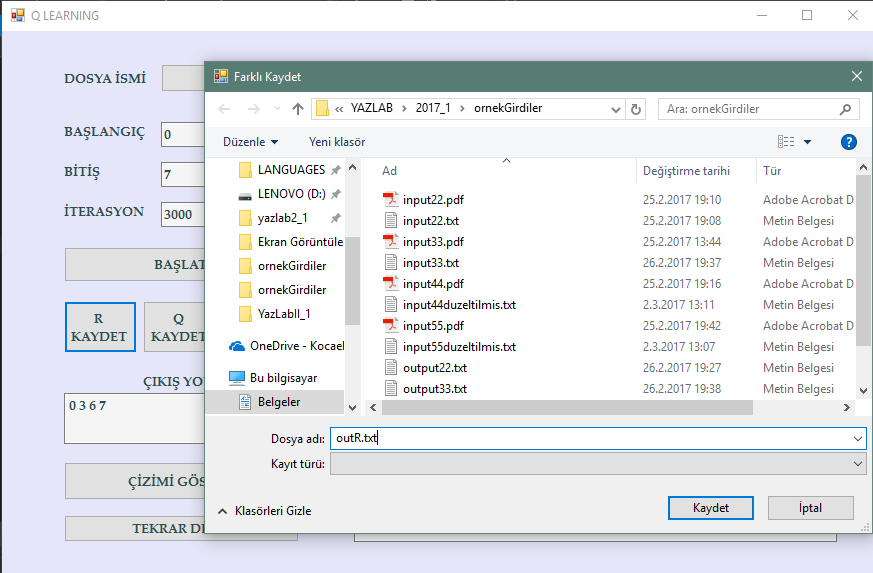
*Şekil 1 :* Proje çalıştırıldığında gelecek olan ilk form



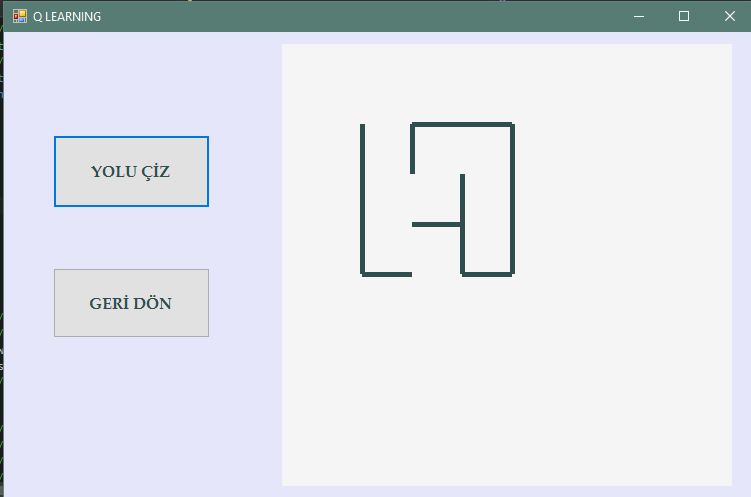
*Şekil 2 :* Dosya ismi için SEÇ butonuna tıklandığında gelecek olan dosya açma ekranı



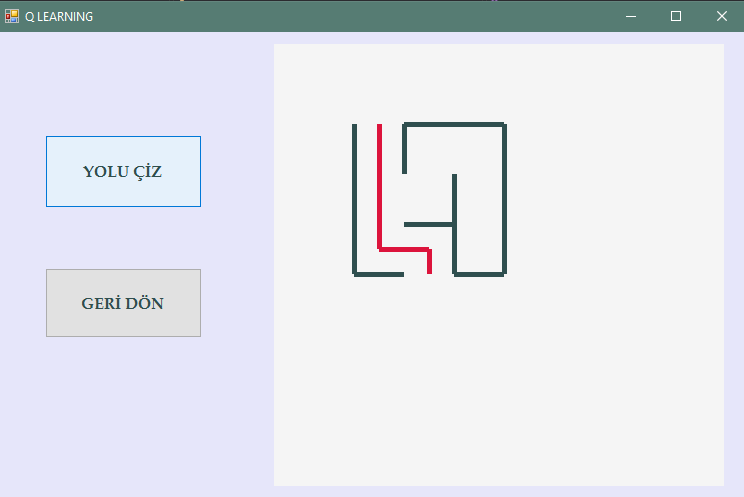
*Şekil 3 :* Verilen başlangıç-bitiş noktasına ve iterasyona göre oluşan Q ve R matrisleri



*Şekil 4 :* R, Q matrislerini ve pathi kayıt etme işlemi



*Şekil 5:*Verilen input dosyasına göre labirent çizimi



*Şekil 6:* YOLU ÇİZ butonuna basıldıktan sonra pathin çizilmiş hali

# Sonuçlar

Projemizde kullanıcıdan alınan başlangıç ve hedef girişleri kullanılarak Q-learning algoritmasına göre çıkan yol,kazanç değerlerine göre oluşturulup grafiksel olarak kullanıcıya çıktısını vererek sorunsuz bir şekilde çalışmaktadır.

# Kaynakça

1. https://en.wikipedia.org/wiki/Q-learning
2. https://www.youtube.com/watch?v=Jc5eXYwTROg