# YAZILIM LABORATIVARI 2 QLEARNING ALGORITM

140202066-Ferdi ELİK,140202034-Furkan AYDIN

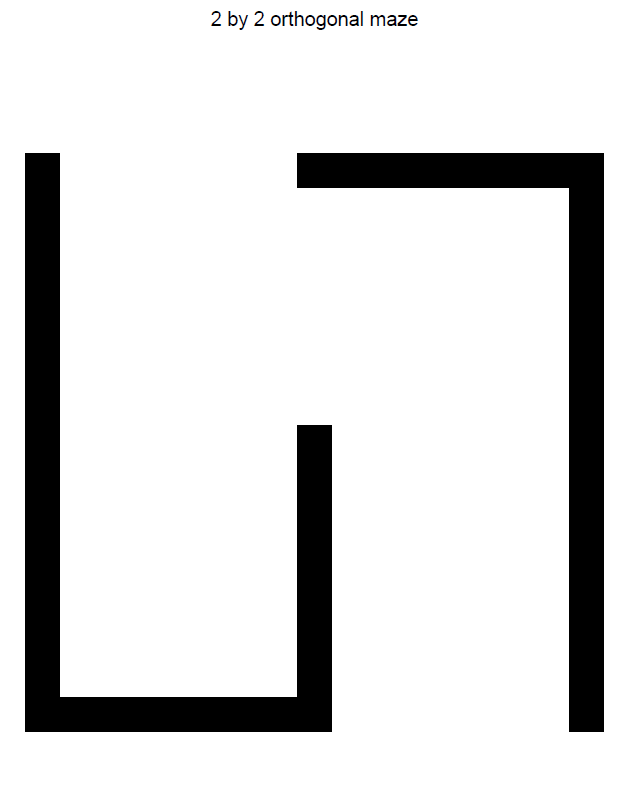
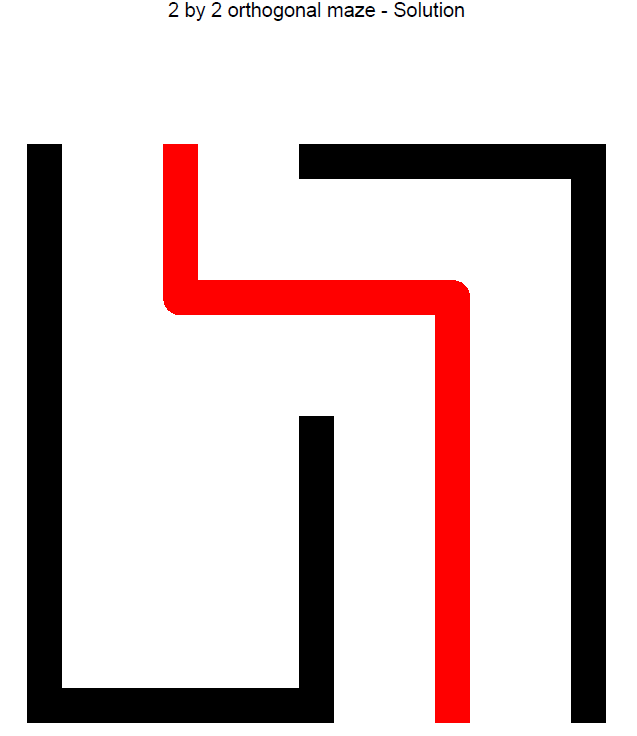
**Projenin Tanımı**

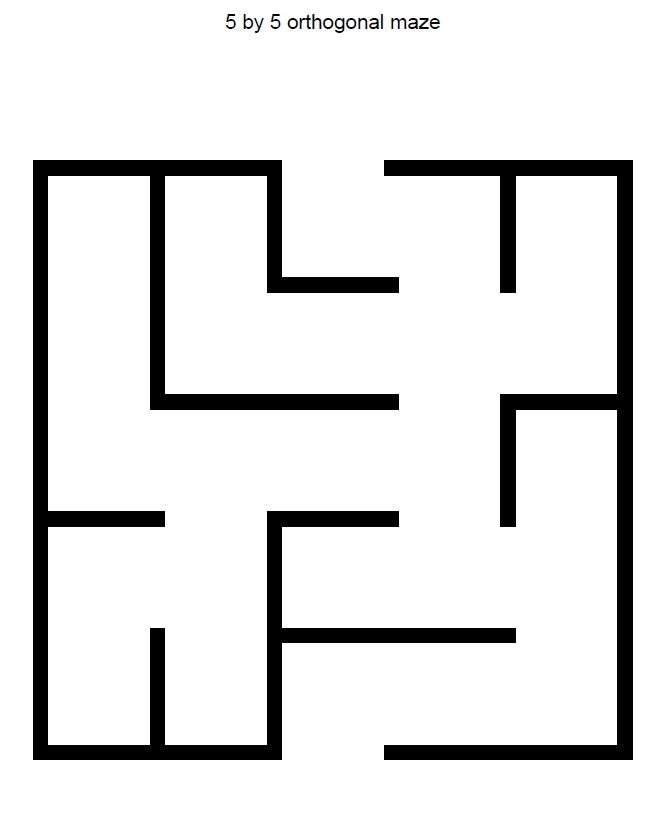
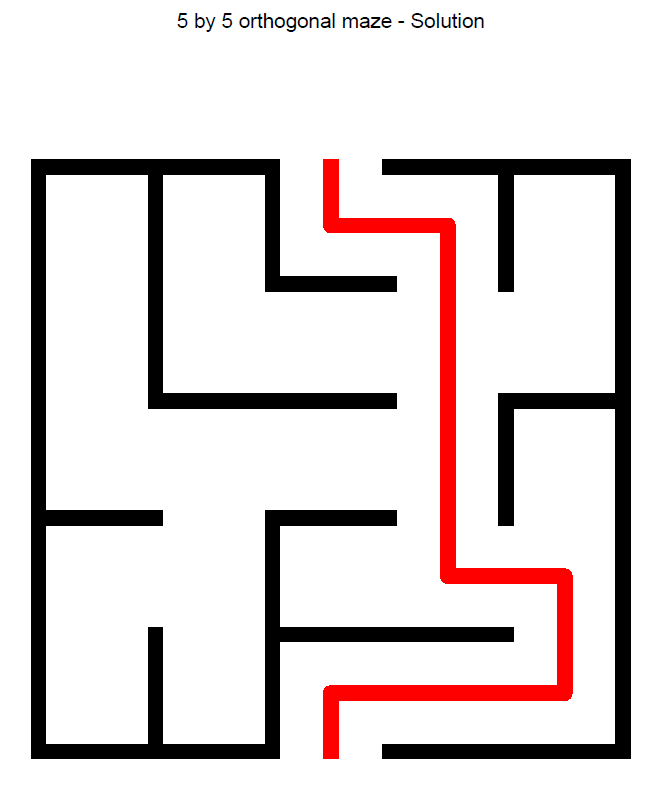
Algoritma hakkında biraz bilgi verecek olursak eğer: QLearning algoritması ödül ve cezaya dayanan bir öğrenme algoritmasıdır.QLearning algoritmasını labirentte dolaşan bir robot üzerinde anlatalım.Başlangıçta robot, labirent hakkında sadece yollar arasında geçiş var mı yok mu onu bilmektedir.Ancak çıkışa ulaştıran yolu bilmemektedir.Robot,rastgele yollara girerek girişten çıkışa ulaşmaya çalışacaktır.Bu gezinti esnasında çıkış bulunduğu takdirde çıkıştan bir önceki yola belirli bir kesinti yapılarak (çünkü en yüksek puan çıkışındır.) puan verilir.

**Yapılan Araştırmalar ve Kullanılan Metotlar**

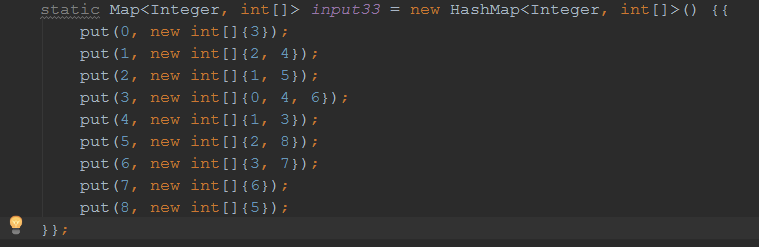
Projemizin ana algoritması olan QLearning Algoritmasıyla[1] ilgili araştırmalar yaptık.Proje ile ilgili verilen dokümanlardan yararlandık.Grafik kısmı[2] için araştırmalarımız oldu nasıl bir şekil de Maze çizimi yapabiliriz bunu araştırdık.Q matrisini oluşturmak için kullanılan formül Q(durum; aksiyon) = R(durum; aksiyon)+ *γ*  x MaxQ(sonrakidurumlar; tumaksiyonlar) burada bahsedilen R matrisi bizim komşuluk matrisimizdir.Bu matrisi elde etmek için gidiş yönleri kontrol edilir eğer bir gidiş yolu mevcutsa 0 değeri verilir gidiş yolu mevcut değil ise -1 değeri verilir.Eğer gidiş yolumuz hedef ise kullanıcı tarafından belirlenen puanı alır. *γ*  sabit değerdir ve proje de 0,8 aldık.

**Tasarım ve Görsel Örnekler**

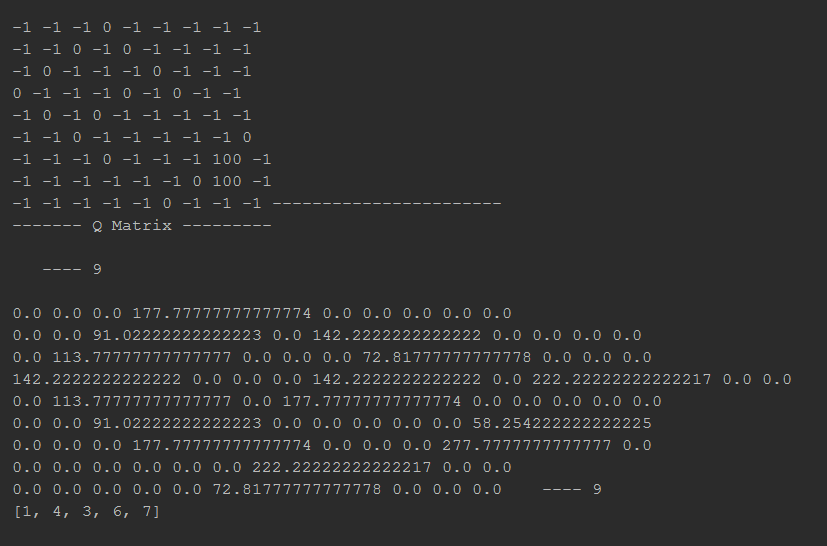
 

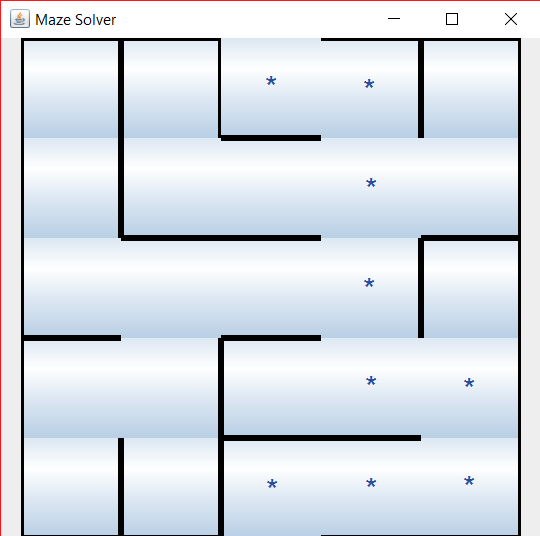
Bunlar bize verilen örneklerin input ve output görselleri.



Verilen input girişi.



Çıkan R,Q matrisi ve gidilecek yol.



Görsel Yol

**Referanslar**

Websites

-[1] http://artint.info/html/ArtInt\_265.html

-[2] https://docs.oracle.com/javase/7/docs/api/javax/swing/JPanel.html

,stackoverflow.com

- http://www.yazilimdefteri.com/KonuCevap.aspx?id=830&mid=105&c=19 (Türkçe kaynak)

- https://github.com/

**Communication**

Discord

Teamviewer

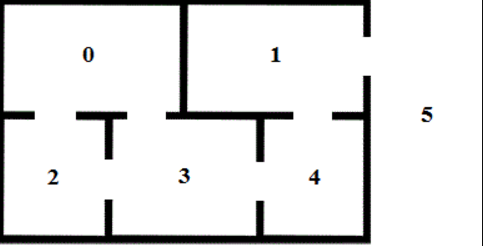
**Connection**

Git Hub

**Genel Yapısı**

Projenin genel yapısından bahsedecek olursak Qlearning tamamen rastgele davranışlar ve deneme yanılmaya dayalı bir algoritma. Programımız önce, belirli sayıda rastgele girişimlerde bulunur. Bu girişimler sonucunda hangi durumda ne yapması gerektiği konusunda bir tablo oluşturur. Öğrenme kısmı bu kısımdır. Bunları yaptıktan sonra da istediğiniz bir başlangıç koşulu verdiğinizde, öğrendiği tabloya göre hareket eder ve sorunu çözer.

Bu konuyu anlatmak için aklıma gelen en iyi örnek, programımıza bir evden dışarı çıkmayı öğretmek oldu. 5 odalı bir evimiz var ve evin kuş bakışı planı kabaca şu şekilde.



6 tane alanı 0 dan başlayarak numaralandırdım. 5 numaralı alan da evin dışı, yani hedefimiz. Şimdi bu ev planına göre, her odada yapabileceği hamlelere göre bir tablo oluşturmamız gerek.

alan0 -> alan2, alan3

alan1 -> alan4, alan5

alan2 -> alan0, alan3

alan3 -> alan0, alan2, alan4

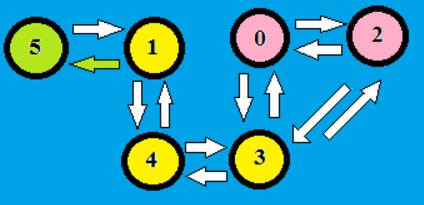
alan4 -> alan1, alan3

alan5 -> alan1

Bu tabloyu dizi haline çevirirsek,

hamleler = [ [2, 3], [4, 5], [0, 3], [0, 2, 4], [1, 3], [1] ]

Hamle tablomuzu elde ettik. Şimdi bir de ödül tablosu oluşturmamız gerek. Ödül tablosunda her durumdaki hamleler için puanlar olacak. Biz programı sonuca ulaştıran hamle veya hamlelere yüksek bir puan koyacağız, yapabileceği diğer hamleler 0 puan olacak.



Bu ödül tablosunu yukarıdaki şema gibi düşünebilirsiniz. Hedefimiz 5. duruma ulaşmak ve gördüğünüz gibi sadece 1. durumdan 5. duruma geçişte program problem çözüme ulaşıyor. Diğer geçişler ise başlangıçta etkisiz. Tabloyu oluşturuyorum, aralarında bağlantı olmayan durumlardaki hamleler için, -1 kullanacağım.

ödül\_alan0 -> [ -1, -1, 0, 0, -1, -1 ]

ödül\_alan1 -> [ -1, -1, -1, -1, 0, 100 ]

ödül\_alan2 -> [ 0, -1, -1, 0, -1, -1 ]

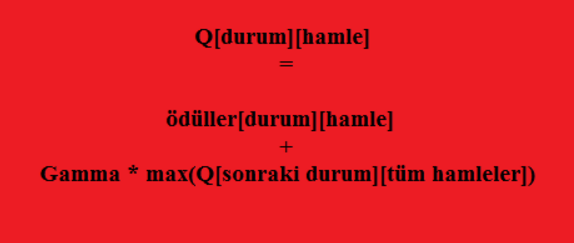
ödül\_alan3 -> [ 0, -1, 0, -1, 0, -1 ]

ödül\_alan4 -> [ -1, 0, -1, 0, -1, -1 ]

ödül\_alan5 -> [ -1, 0, -1, -1, -1, -1 ]

ödüller = [ödül\_alan0, ödül\_alan1, ödül\_alan2, ödül\_alan3, ödül\_alan4, ödül\_alan5]

Ödüller adındaki 2 boyutlu matrisimizi de elde etmiş olduk. Bu noktadan sonra öğrenme matrisi olan Q matrisini oluşturmaya başlayacağız. Yani burası programın öğrenmeye başladığı bölüm. Program rastgele seçimlerle ilerleyerek, tecrübelerinden bir şeyler öğrenecek. Bunu yapmak için bir formül kullanıyoruz.



Burada yeni bir terim giriyor işin içine. Gamma denilen öğrenme katsayısı. Gamma 0 ile 1 arasında bir sayı ve büyük olması, gelecek durumda alacağı ödüllere önem vermesini sağlarken, küçük olması, o anda yapmak istediği hamlenin ödülünü öne çıkarıyor. Formülde, şu an bulunduğumuz durumun içindeki, rastgele yapacağımız hamlenin ödülünü, ödül tablosundan alıyoruz, sonra Q matrisinde, bu hamleyle gideceğimiz sonraki durumdaki en büyük Q puanını alıp, gamma ile çarpıyoruz. En sonunda bu ikisini toplayıp, Q matrisinin bulunduğumuz durumda ve yapacağımız hamledeki yerine koyuyoruz.

**Kazanımlar**

Bu başlık altında projemizle ilgili kişisel kazanımlarımızı göstereceğim.Kişisel olarak her bir birey öncelikle ekip ile çalışmanın önemini ve çalışmamızın profesyonel hayatta işe yarar şeyler çıkarabileceğini gördük.Projemizde kullandığımız Qlearning Algoritması hakkında daha önce bir bilgimiz yoktu ve hakkında bilgi sahibi olduk.Java konusunda kendimizi geliştirdik.İlerleyen projelerde de kendimizi daha çok geliştireceğimize inanıyorum.