PROGRAMLAMA LABORATUVARI 1. PROJE

Buğra Oktay Ateş

I. Özet

Bu doküman Programlama Laboratuvarı 2 dersinin 1. Projesini açıklamaya yönelik oluşturulmuştur. Dökümanda giriş, yöntem, sonuç, proje hazırlanırken kullanılan geliştirme ortamı gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemi hazırlarken kullandığım kaynaklar ve proje derlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

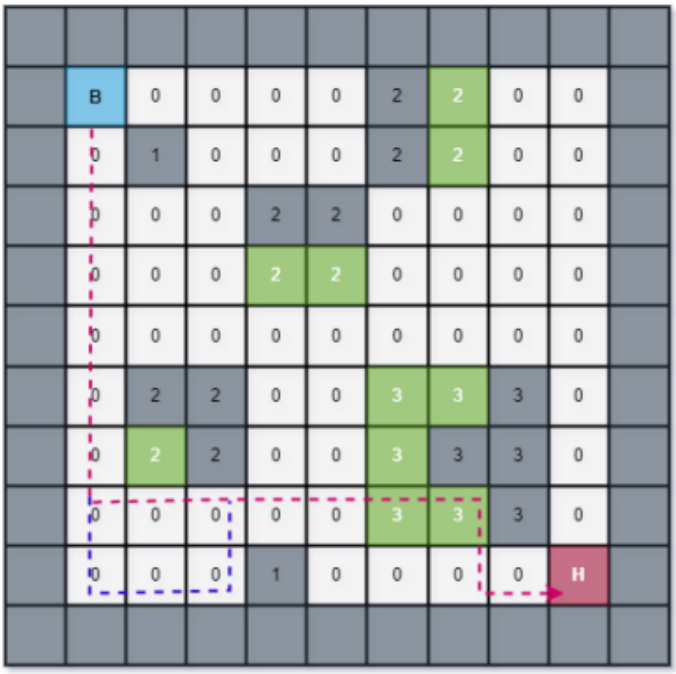
II. Giriş(Proje Tanitimi)

Amaç:Proje gerçekleştirimi ile öğrencilerin nesneye yönelik programlama ve veri yapıları bilgisinin pekiştirilmesi ve problem çözme becerisinin gelişimi amaçlamaktadır.

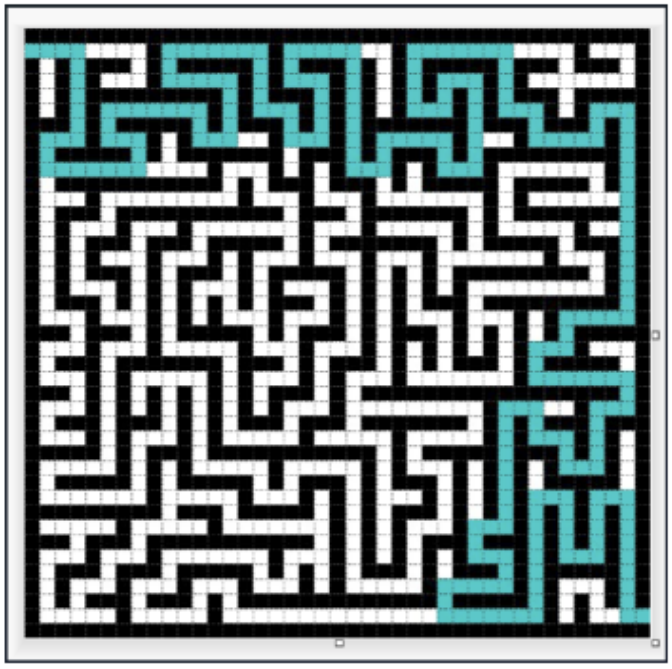
Projenin konusu belirli kurallara göre hareket eden bir robotun önündeki engelleri aşarak istenen hedefe ulaşmasını sağlayan bir oyun tasarlanması beklenmektedir. Oyunda iki adet problemin çözülmesi gerekmektedir.

Proje bizden iki adet problemi çözmemizi istemektedir.

PROBLEM 1: Bu problemde bizden Url üzerinden alınan bir sayı matrisine göre bir ızgara oluşturmamız beklenmektedir. Bu matristeki 0 lar gidilebilecek yolları 0 harici sayılar ise engelleri temsil etmektedir. Bizim amacımız oluşturulan bu ızgara yapısında rastgele bir başlangıç ve bir bitiş noktası belirleyip bu noktalar arası yolu gidileblecek en kısa yoldan giderek bulmaktır. Problem 1 örnek Izgara şekildeki gibidir.



PROBLEM 2: Bu problem ise bizden kullanıcıdan alınan boyut bilgilerine göre bir labirenet oluşturmamızı istemektedir. Oluşturulan labirentin başlangıcından başlayan robot yolları deneyerek çıkışı bulması amaçlanmaktadır.



A. Oluşturulması Gereken Sınıflar

Robot Sınıfı: Robot sadece yukarı-aşağı ya da sağ-sol doğrultusunda hareket edebilmeli, çapraz yönde hareket etmemelidir. Robot ızgara dünyasında bulunduğu konumdan sadece bir birim sonrası ile ilgili bilgileri görebilir. Haritanın tümünü görmemelidir.

Izgara Sınıfı: Bu sınıfta problem 1 için ızgara tasarımı verilen url adresindeki text dosyasına göre oluşturulurken problem 2 için ızgara kullanıcıdan alınacak boyut bilgisine göre oluşturulmalıdır.

Engel Sınıfı: Engel sınıfında problem 1 için üç farklı tipteki nesneler kullanarak engellerin oluşturulması url adresindeki text dosyasına göre yapılırken problem 2 için labirent oluşumu 1 nolu tek bir nesne türü kullanılarak uygulama içerisinde rastgele oluşturulacaktır.

Uygulama Sınıfı: Uygulama içerisinde robotun problem 1 ve problem 2 deki hedefe ulaşma süresi, kaç kare üzerinden geçildiği gibi bilgilerin tutulduğu ve ekranda gösterilmesi fonksiyonlarını sağlamalıdır

III. Yöntem

A. Oluşturduğum Sınıflar

Projeye ilk olarak oluşması gereken sınıflar başlığı altındaki sınıfları oluşturmakla başladım.

Izgara Sınıfı:

Problem 1 için verilen Url ye göre ızgara oluşturmaktadır.

Robott Sınıfı:

Problem 1 için robotun hedefi bulmasını sağlar.

BFS Sınıfı:

Problem 1 için en kısa yolu bulmaktadır.

Labirent Sınıfı:

Problem 2 için labirent oluşturmaktadır.

Cell Sınıfı:

BFS ve Labirent sınıfları için gerekli bir sınıftır.

Sayi Sınıfı:

Diğer sınıflarda kullanmak için değişkenleri tutan sınıftır.

B. Programın Genel Çalışma Mekanizması

Programı ilk çalıştırdığımız anda karşımıza Problem 1 ve Problem 2 Adında iki buton çıkmaktadır. Problem 1 butonuna bastğımız anda açılan pencerede 1. Urlye göre oluşturulmuş Izgara yapısı karşımıza çıkıyor. Çalıştır butonuna bastığımızda ise oluşan ızgara yapısında başlangış ve bitiş noktaları belirleniyor ve 4 buton daha aktif hale geliyor. Bu butonların isim ve işlevleri şu şekilde:

En kısa yol butonu en kısa yolu çiziyor.

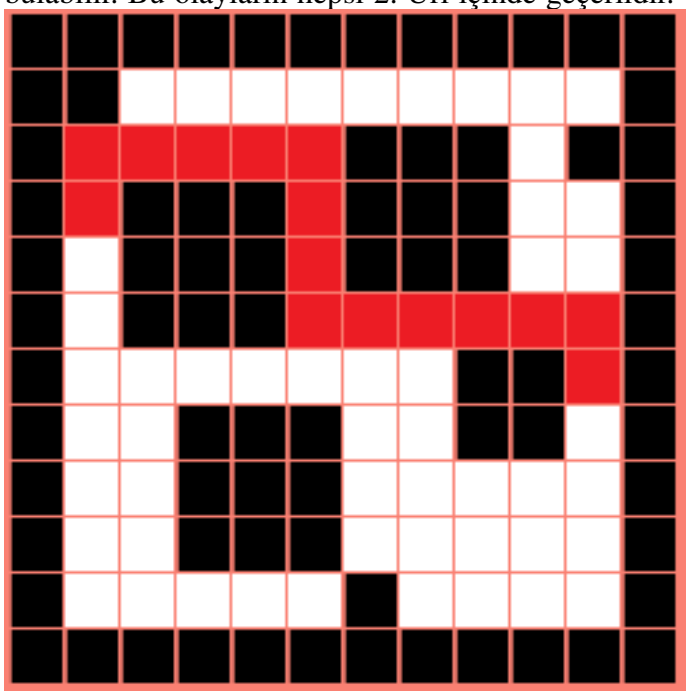
Adım Adım butonu robotum adım adım hareketeini gösteriyor.

Hepsini aç butonu robotun geçtiği tüm yolları gösteriyor.

Geçilen yollar buntonuda toplam kaç birim hareket edildiğini gösteriyor.

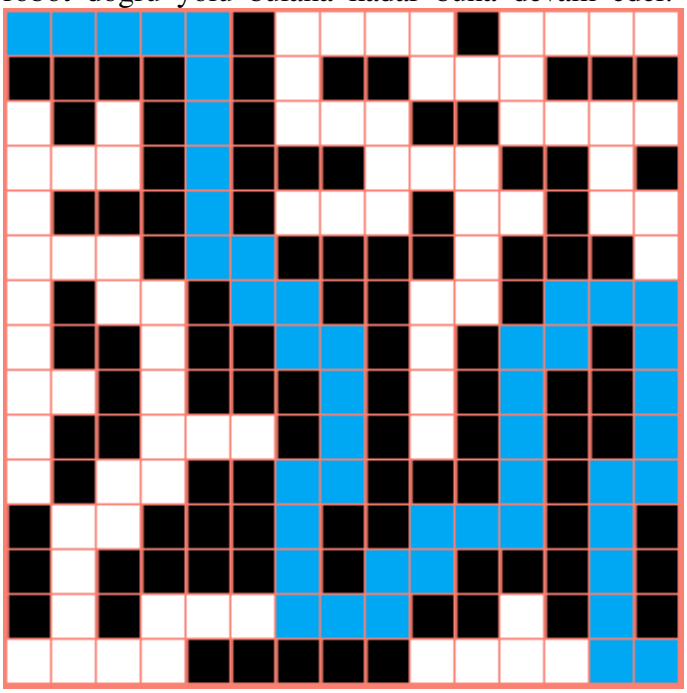
Ayrıca Url değiştir butonu ile 2. Url ye geçiş yapılabiliyor.

Şimdi bu olayların perde arkasına bakalım. İlk olarak Url den aldığım metni integer bir matrise çeviriyorum. Bu matrise göre JLabel'lerden oluşan bir dizi oluşturuyorum. Bu JLabel'lerın her biri robotun hareket edeceği kutucukları temsil etmektedir. Daha sonra matrise göre kutucukların içini engeller siyah, yollar beyaz olacak şekilde doldurdum. Bu sayede ekranda bir ızgara görüntüsü yakalamış oldum. Sonra rasgele belirlenen başlangıç konumundan yola çıkan robot ilk olarak gidilmemiş yolları tercih edecek şekilde ilerlemektedir. Gidilmemiş yol kalmayınca veya çıkmaz bir yola girince geri çıkmasını kullandığım Stack veri yapısı ile çözdüm. Bu sayede robot başlangıçtan bitişe kadar olan yolu kayedetim. Ayrıca robot artık bitiş noktasını bildiğine göre en kısa yoluda bulabilir. Bu olayların hepsi 2. Url içinde geçerlidir.



Problem 2 Butonuna bastığımızda ise karşımıza

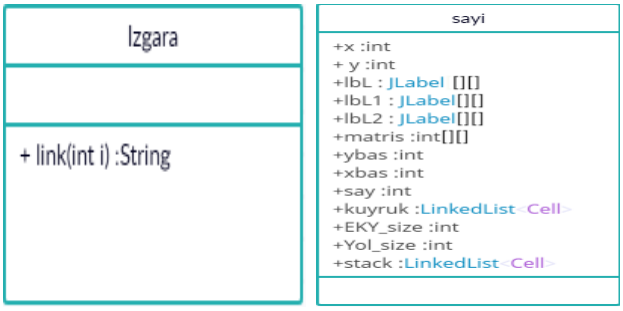
labirent boyutunu belirlemek için bir alan çıkıyor. İstedeğimiz boyutu girip labirent değiştir butonuna basınca istenilen boyutta bir labirent oluşturuyor. Bu labirentte problem 1 de olduğu gibi JLabellerden oluşmaktadır. Siyah kutucuklar duvarları beyaz kutucuklar ise yolları temsil etmektedir. Ayrıca panelde çalıştır butonu vardır. Bu butona basınca robotumuz bulunduğu konumdan beyaz kutuckları takip ederek bitiş noktasına gitmeye çalışmaktadır. Bu olayda aynı şekilde stack yapısı ile gerçekleşmektedir. Robot her geçtiği yolu bir stack yapısına ekler. Çıkmaz bir yola girdiğinde ise çıkmaz yoldan çıkana kadar stack yapısından eleman çıkartır. Bu sayede robot doğru yolu bulana kadar buna devam eder.

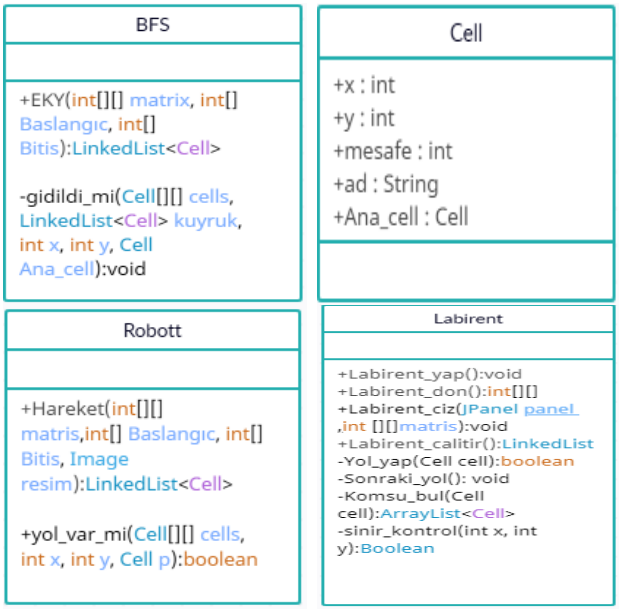


IV. Sonuç

Özetle ilk olarak sınıflarımızı tanımladık daha sonra tanımladığımız sınıfların metotlarını ve özelliklerini kullanarak arayüzle optimal olarak çalışan bir oyun yaptık.

V. Tablo





VI. Kaynakça

Java kulanırken karşılaştığım her türlü sorun için:
https://stackoverflow.com
https://www.geeksforgeeks.org/

Projeyi java dili kullanarak Eclipse ortamında geliştirdim.