****

**KOCAELİ ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**LABİRENT OYUNU**

Edanur TOSUN, Merve TAFRALI

160202098 - 160202100

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

eedanurtosun@gmail.com, mervetafralii@gmail.com

**1.1.Problem Tanımı**

Labirent Oyunu adlı projede amaç kullanıcıdan alınan boyuta göre matrisin elemanlarına rastgele 0 ve 1 atamak, oluşan bu matris üzerinden çıkış yolunu bulmaktır.

Program çalıştırıldıktan sonra kullanıcıdan rastgele olarak 0 ve 1 atanacak matrisin boyutu istenmektedir. Kullanıcı matrisin boyutunu girdikten sonra oluşturulan matris ekranda gözükmektedir.

Labirent matrisi oluşturulduktan ve kullanıcıya gösterildikten sonra proje adımlarında istendiği şekilde kullanıcıdan sırasıyla giriş ve çıkış koordinatları istenmektedir.

Kullanıcıdan alınan giriş ve çıkış kapılarına göre ve yolun olduğu yerler “1” yolun olmadığı yerler yani duvar olan yerler için “0” mantığı ile çalışarak çıkış bulmaktadır.

Bulunan çıkış yollarının koordinatları ve çözüm matrisi ekrana yazdırılmaktadır.

**1.2.Yapılan Araştırmalar**

Projemizin başlangıç aşamasında,kodun algoritmasının doğru olabilmesi için algoritma mantığını anlamamız ve Labirent Oyunun algoritmasını çıkartmamız gerekti.Yaptığımız araştırmalar sonucunda algoritmanın, belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol olduğunu[1] ve labirent oyunu çözmemiz için tasarlamamız gereken adımlar[2] olduğunu çıkarttık.

Bölüm 1.3’te anlattığımız akış şemasını çıkarttıktan sonra bağlantılı liste (link list) ve yığın mimarisini anlama ve koda dökme adımlarını yaptık,bunun sonucunda bağlı listenin herhangi bir tipten node’ların (düğümlerin) yine kendi tiplerinden düğümlere işaret etmesi (point) ile oluşan zincire verilen isim olduğunu[3-site] öğrendik. Edindiğimiz bilgilerle elimizde olan dizinin yığına eleman ekleme ve çıkarma işlemlerinin dizinin en son konumuna göre yapıldığı özel bir yapı olduğunu ve sıralı yapılarda olduğu gibi ara elemanlara doğrudan erişim yapılamadığını[4] öğrenip koda döktüğümüz yığın mimarisini bu bilgiye göre düzenledik..

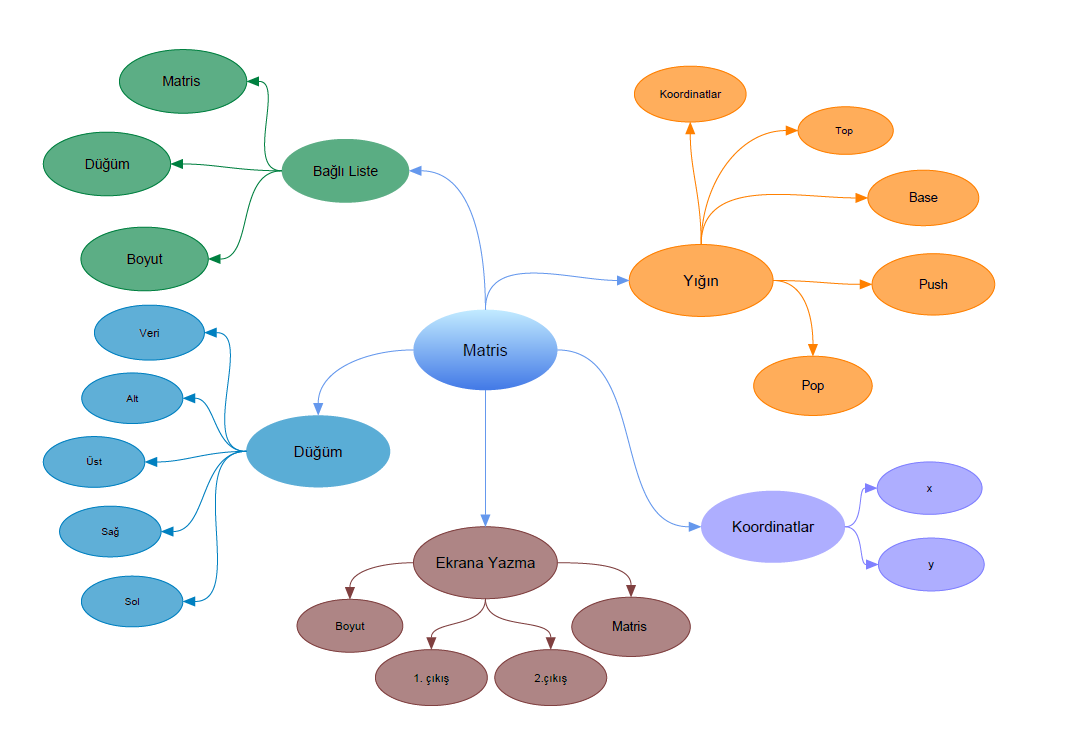
**1.3.Tasarım**

**1.3.1.Akış Şeması**

Akış seması ekte gösterilmektedir.

**1.3.2.Yazılım Mimarisi**

Projenin kodlanmasında kullanılan yapı şu şekildedir :



**1.4.Genel Yapı**

Labirent oyunu projemizde main fonksiyonu içerisinde öncelikle kullanıcıdan oluşturulacak matris için boyut bilgisini istiyoruz. (Matris tanımı main fonksiyonu içerisinde değil program içerisinde tanımlanmıştır.) Programda belirlediğimiz maksimum boyut 106’dır. ( #define MAXDEGER 106) Eğer kullanıcı bu boyuttan fazla bir değer girerse programı sonlandırıyoruz. Boyut alındıktan sonra matris elemanlarına rastgele 0 veya 1 atanıyor. Oluşturulan matrisi ekrana yazdırıyoruz. Giriş kapısının koordinatlarını kullanıcıdan istiyoruz. Kullanıcı boyutu aşacak bir x koordinat değeri girerse aldığımız boyuttan fazla bir değer olduğu için programı sonlandırıyoruz. Giriş kapısının sadece köşelerden olması için gerekli kontroller yapılıyor. Girilen x koordinat değeri doğru ise y koordinat değerini yani çıkış kapısını kullanıcıdan istiyoruz. Giriş kapısında yapılan tüm kontrolleri çıkış kapısı için de gerçekleştiriyoruz. Labirent içerisinde dolaşabilmek tanımladığımız yapılar :

1.struct node \*m[][]

2.struct node

3.struct node \* nnode

4.struct ko (koordinatlarımızı tutan yapı)

5.struct stack

6.pop

7.push

8.void dolink

9.void ekranayaz

Struct node tipinde matrisimiz var. Matrisimizi öncelikle dolink fonksiyonuna gönderiyoruz. Matris linklist haline getiriliyor. Daha sonra tanımlanan yapıları kullanarak linklist’te eleman değeri 1 olanları yığın yapımıza ekliyoruz. Main fonksiyonu içerisinde while(1) döngüsü içerisinde eleman değerinin 1 olup olmadığı kontrolleri yapılıyor ve push ve pop yapısı kullanılarak yığına ekleme veya çıkarma işlemi yapılıyor.

Çıkış değerine ulaşıldıktan sonra ekranayaz metodu ile bulunan yola 1, diğer elemanların hepsine 0 atayarak matrisi ekrana yazdırıyoruz.

**1.5.Referanslar**

1-<https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/v/what-are-algorithms>

2-Veri Yapıları ve Algoritmaları - Rifat Çölkesen

3-<http://bilgisayarkavramlari.sadievrenseker.com/2007/05/03/linked-list-linkli-liste-veya-bagli-liste/>

4-<http://www.zafercomert.com/IcerikDetay.aspx?zcms=75>

5-<http://web.karabuk.edu.tr/hakankutucu/BLM227/VER%C4%B0%20YAPILARI%20v.6.3.pdf>

6-<http://www.yildiz.edu.tr/~wwwhid/TR/algoritma3.htm> (akış şeması için)

7- ClickCharts Diagram Flowchart Software (akış şemasını ve yazılım mimarisinin yapıldığı program)

8-<https://mehmetsalihdeveci.net/2010/06/22/yigin-stack-veri-yapisi-push-pop/>

9-<http://bilgisayarkavramlari.sadievrenseker.com/2007/05/04/stack-yigin/>

10-<https://www.tutorialspoint.com/data_structures_algorithms/stack_program_in_c.htm>

11-<http://turgayozgur.com/c-ile-bagli-listestackyigin-ve-kuyruk-kullanimi.html>

12-<http://www.ckutuphanesi.com/c-programlamada-array-kullanarak-stack-islemleri-uygulamak/>

13-http://turgayozgur.com/c-ile-bagli-listestackyigin-ve-kuyruk-kullanimi.html