

KOCAELİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ



PROGRAMLAMA LABORATUVARI II

1.PROJE

-ŞİRİNLER-

200202098

M.Numan MERCAN

Projede ilerlerken kaydettiğim bazı aşamaları da görsel olarak paylaşıyor olacağım.

- Arayüzü Oluşturma

Haritayı oluşturmak için önce "harita.txt" dosyasını Java da okuttum. Ardından dosyanın içinde bulunan matris kısmını labirenti oluşturmak için okudum. İlk 2 satırdaki veri düşmanın kim olacağı ve labirentin hangi kapısında başlayacağını belirten veriydi. Bu veriyide ayrı bir kısımda düşmanın yerini belirtmek için okudum. Haritayı "Paint()" kütüphanesi ile çizdirdim.

Şekil 2 de "harita.txt" deki matris kısmı bulunmaktadır

0 -> Duvarlar

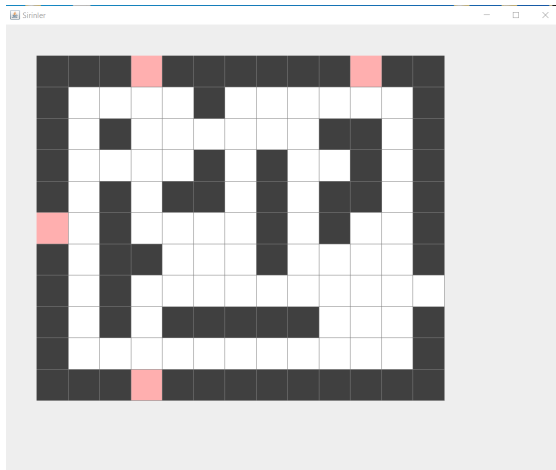
1 -> İlerlenebilecek Yollar

0	0	0	1	0	0	0	0	0	0	1	0	0
0	1	1	1	1	0	1	1	1	1	1	1	0
0	1	0	1	1	1	1	1	1	0	0	1	0
0	1	1	1	1	0	1	0	1	1	0	1	0
0	1	0	1	0	0	1	0	1	0	0	1	0
1	1	0	1	1	1	1	0	1	0	1	1	0
0	1	0	0	1	1	1	0	1	1	1	1	0
0	1	0	1	1	1	1	1	1	1	1	1	1
0	1	0	1	0	0	0	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0

Şekil 2. (Matris)

Şekil 3 te dosyadan alıp oluşturulan haritamız bulunmaktadır.

Pembe ile belirtilmiş bloklar düşmanların olası başlangıç base'i dir.



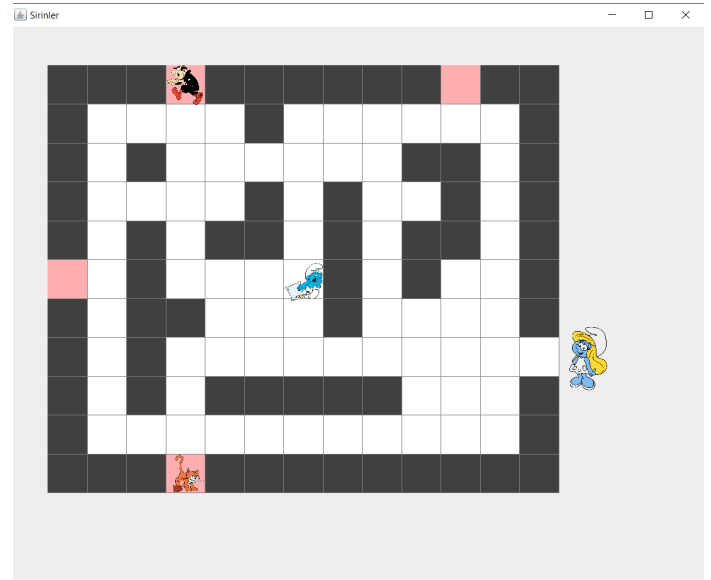
Şekil 3. (harita)

Ardından karakterleri sınıflara implement edip arayüzde yine Paint metodu ile çizdirdim. Karakterlerin belirli İmagelerini BufferedImage değişken türü ile her karakterin kendine ait sınıfında oluşturdum ve Oyun (JPanel) sınıfımda değişkenlerini oluşturarak resimlerini arayüzüme aktardım.

```
Karakter:Gargamel,Kapi:A  
Karakter:Azman,Kapi:D
```

Şekil 4. (Düşman ve base belirleme)

Şekil 5 te Artık dosyadan okunmuş bir haritanın son hali bulunuyor.

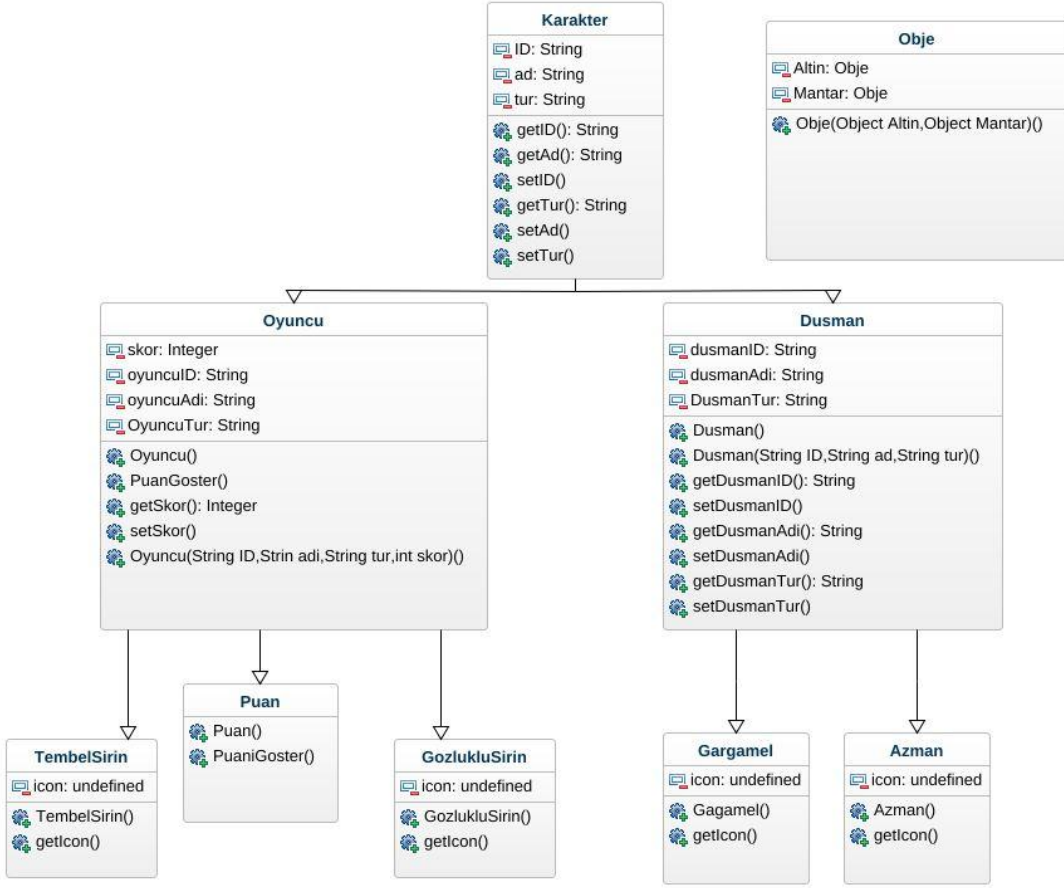


Şekil 5.(Haritanın tam hali)

- Sınıfların Oluşturulması

Sınıfları oluştururken UML diagramı çizimi işimizi oldukça açık ve kolay hale getiriyor. Fakat UML diagram çizimimi projeye başladığım aşamada çizmedim , yine de daha rahat görebilmek açısında ilerleyen vakitlerde diagramı uluşturdum ve yararlandım.

Şekil 6 da projenin UML diagramına yer verilmiştir.



Şekil 6.(UML diagram)

Şekil7.(ilk kısım)

-Algoritma Oluşturma

Önce kaba kodumuzu ve bu kodun yardımıyla Akış Şemamızı oluşturduk.Bu aşama projenin ilerleyişi açısından çok büyük bir önem taşıyor , projemin tam olarak yetişmemesinin en büyük sebebi bu aşamayı da olması gerekenden geç yapmam.

Kaba kodları Şekil 7 ve 8 deki gibidir.

```

}

//klavye hareketleri için KeyListener implement edildi
//oyuncu sirinenin olduğu bloğa ulasana kadar ya da yanana kadar
if(basilanTus.kod==37)
if(basilanTus.kod==38)
if(basilanTus.kod==39)
if(basilanTus.kod==40)
dusmanın hareketleri için dijkstra algoritması
if(dusman.lokasyon==şirine.lokasyon)//şirinin ve dusmanın tipine göre puan eksilir
if(secilenşirin.lokasyon==şirine.lokasyon && puan > 0)
sout("Tebrikler oyunu kazandınız");
else if(secilenşirin.lokasyon != şirine.lokasyon && puan >0)
//oyun devam etsin
else
sout("Maalesef kaybettiniz, lütfen bir daha deneyin");
}

```

Şekil8.(ikinci Kısım)

```

- Başlat
- Oyuncu seç
- "harita.txt" den veriyi okumaya başla
while(harita.txt.hasNextLine()){okunacak satir olduğu sürece
//harita.txt siradaki satiri oku
String harita = sc.nextLine();
//ilk 2 satir düşman ve doğacağı kapıyı belirler//ona sonra bakacağız
//2.satirdan sonra haritayı oku
//paint metodu ile çizdiriyoruz
char[] array = harita.split("\t"); // satırı tab kısımlarından ayır diziye koy
for(i=0;i<satirinuzunluđu;i++){
if(array[i].equals("1")) //yolu bloklar halinde çizdirelim
g.fillRect(x * 50, y * 50, 50, 50);
else if(array[i].equals("0"))//duvarları oluşturalım
g.fillRect(x * 50, y * 50, 50, 50);
}
if(secim==1) //seçilen şirini çizelim
g.drawImage(imageGozluklu,sirinX, sirinY,genislik/15,yükseklik/15, this);
else if(secim == 2) g.drawImage(tembelsirin);
-----
if(dusman1=='')//dusman 1 hangi karakterse sec ve çizdir
if(dusman1.kapi=='') hangi kapıdan doğacağını seç
if(dusman2=='')//dusman 2 hangi karakterse sec ve çizdir
if(dusman1.kapi=='') hangi kapıdan doğacağını seç

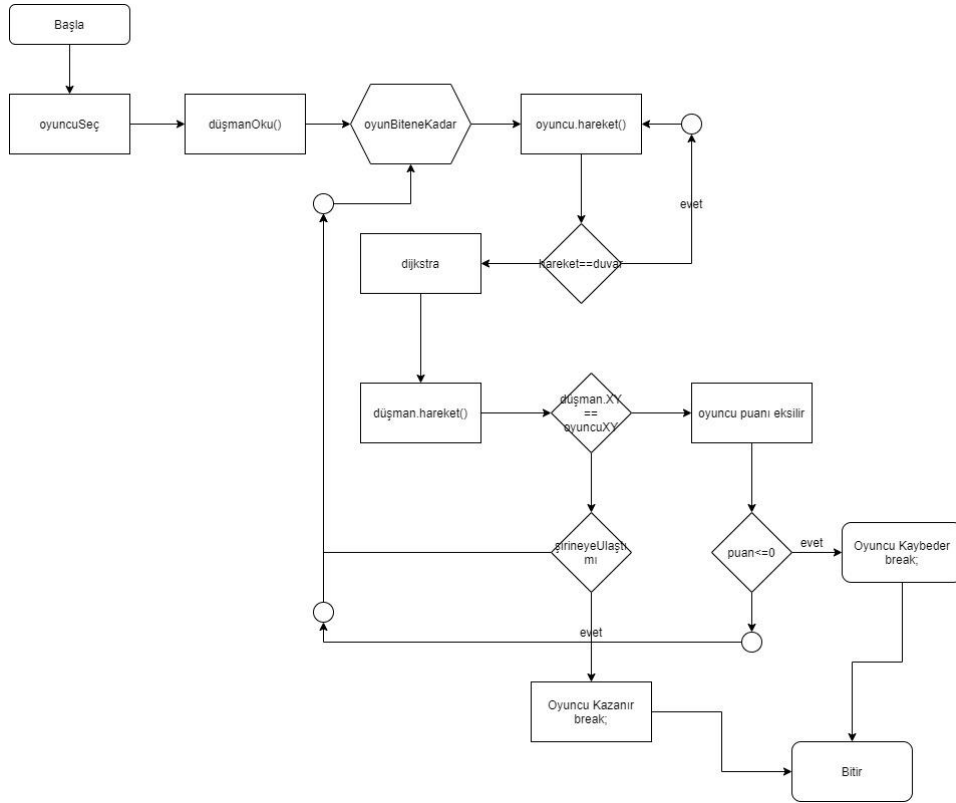
```

- Akış Şemaları

Proje için 2 adet akış şeması oluşturduk.Bunlardan biri Oyun için diğer şema ise arayüz için oluşturuldu.

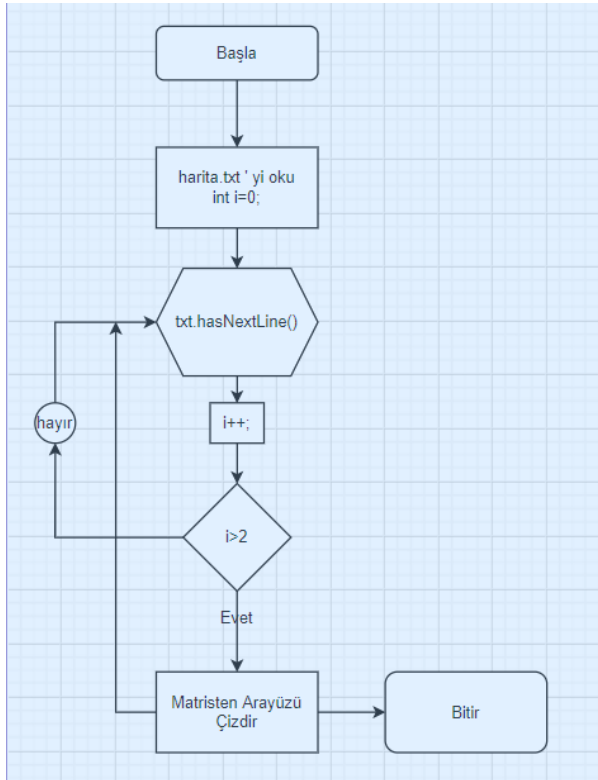
Şekil 9 da Oyun için oluşturulan şema gözükmemektedir.

Şekil 10(Arayüz Akış Şeması)



Şekil9.(Oyun Akış Şeması)

Şekil 10 da Arayüz çizimi için oluşturulan şemayı göreceksiniz.



-Son Söz

Not : Proje de bazı eksiklikler bulunmaktadır bunlardan en önemlisi zaman yönetimi başarısızlığı dolayısıyla düşman yol algoritmasını ve haliyle hareket kodlarını oluşturamamam, bu yüzden projemin fazlasıyla eksikleri olduğunu bilerek fakat ona rağmen elimden geleni yaptığımı da belirterek projemi açıklamak istedim

Projenin belli aksaklık ve tecrübesizlikler dolayısıyla tam anlamıyla bitmediğini belirtmem gerek. Proje ye yeterli kavrama ve planlama yapılmadan oluşturma aşamasına girmiş olmam günler azaldıkça yanlış bir strateji izlediğimi kanıtladı. Projenin en büyük eksiği dijkstra algoritmasını entegre edememek. Bu eksik azımsanmayacak büyüklükte olduğunu kabul etmekte fayda var fakat geriye kalan isterlerin büyük çoğunluğunu proje tamamlamış olup sunuma çıkma cesaretini gösterecektir.

KAYNAKÇA

- https://www.youtube.com/watch?v=9XJicRt_Fal
- **Sıfırdan Komple Java Geliştirici Kursu** -> Mustafa Murat COŞKUN (Udemy) (Bazı unutulmuş kavramlar için)
- <https://www.javatpoint.com/java-swing>
- <https://www.draw.io>
- <https://www.javatpoint.com/java-keylistener>
- https://www.youtube.com/watch?v=dyrvXiMumXc&ab_channel=BroCodeBroCode -> Leo ONO
- <https://github.com/irealva/maze-gui/blob/master/Maze.java>
- <https://stackoverflow.com/>
- https://www.youtube.com/watch?v=q3yKyE19OR0&ab_channel=SureshSrivastava -> for Dijkstra Algorithm
-
- https://www.youtube.com/watch?v=uPXlsSd_wK4&list=WL&index=9&ab_channel=coderodde
- <https://github.com/yunusemregul>
- <https://www.geeksforgeeks.org/java-program-for-rat-in-a-maze-backtracking-2/>
- <https://blog.emrahkahraman.com.tr/java-dosyadan-istenilen-satiri-okuma/>
- <https://medium.com/gokhanyavas/javada-i-o-i-CC%87%C5%9Flemleri-abde319f0544>