

# ŞİRİNLER

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

PROGRAMLAMA LABORATUVARI PROJESİ

YASİN ÖMER KARA-TAHİR UMUT DOLAŞ

180201077-180201053

## ÖZET

Bu proje ile nesneye yönelik programlama ve veri yapıları algoritmalarını kullanarak Şirinler oyunu tasarlamamız beklenmektedir. Seçilen oyuncunun Labirent içerisinde puanlarını bitirmeden önce Şirine'ye ulaşması gerekmektedir.

karakter bulunmaktadır. Bu düşmanların da şirinler gibi kendilerine has özellikleri vardır.



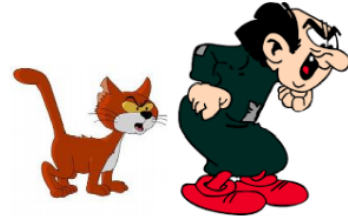
## 1.GİRİŞ

Projede bizden öncelikle bir arayüz tasarlamamız istenmiştir.

Kullanıcıdan oyun başlamadan önce iki farklı oyuncudan birini seçmesi beklenmektedir. Bu oyunculardan biri *Gözlüklü Şirin* diğeri ise *Tembel Şirindir*.

Seçilen her iki şirinin de farklı özellikleri bulunmaktadır.Bu özelliklere göre oyun içinde davranışları değişecektir.

Bize verilen harita.txt dosyasında düşman bilgileri bulunmaktadır.Oyunda *Gargamel* ve *Azman* isimli iki farklı düşman



Şekil 2. Karakterler

Oyundaki başlıca karakterlerimiz şekil 2'de belirtilmiştir.Oyunun işleyişi düşman karakterlerin seçilen şirini yakalamak ve şirineye ulaşması üzerine kuruludur.

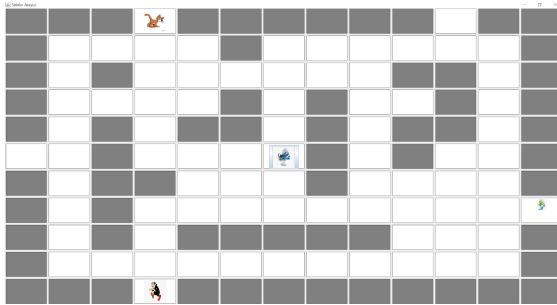
Düşman karakterler en kısa yolu bularak oyuncuyu engelleyeceklerdir.

## 2.YÖNTEM

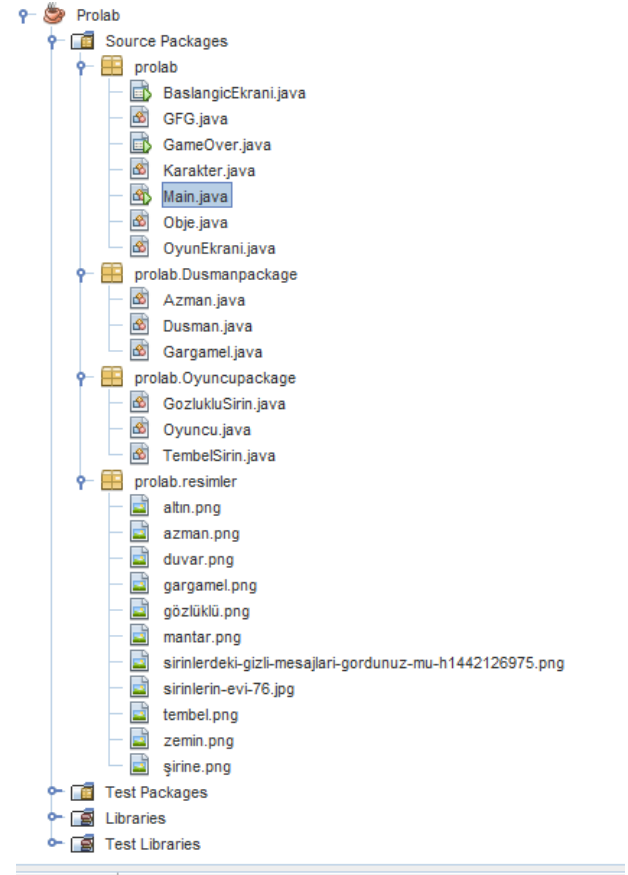
Oyuncu Karakter Seçim ekranında istediği oyuncuyu seçecektir.Bunun için bir JForm üzerinde BaşlangicEkranı() tanımlaması yapılmıştır.



Seçilen oyuncuya göre Java OOP yapısı kullanılarak, şirinlere özel classlardan özellik ataması yapılmış ve OyunEkranı() 'nında bu özelliklere göre oyunun şekillenmesi sağlanmıştır.



Nesneye yönelik programlama için oluşturulan sınıflar:



gibidir.

### *Karakter Sınıfında Bulunan Özellikler*

- Karakterlerin adını tutacak ID, Ad, Türünü (oyuncu/düşman) tutacak.
- Karakterlerin ilerlediği koordinatları tutacak Lokasyon değişkenleri olmalıdır.
- Constructor, Get, Set ve En KısaYol metotları yer almalıdır.

### *Oyuncu Sınıfında Bulunan Özellikler*

Bu sınıfta karakter sınıfı kalıtım olarak verilecektir.PuaniGoster() methodu olacaktır.

### *Oyuncu Türlerine Alt Sınıflar (3 Adet):*

- **Gözlüklü şirin(Oyuncu 1):** Her defasında 2 birim ilerler. Azman'a dokunursa oyuncu 5 puan kaybeder. Gargamele dokunursa oyuncu 15 puan kaybeder.
- **Tembel Şirin(Oyuncu 2):** Her defasında 1 birim ilerler. Azman'a dokunursa oyuncu 5 puan kaybeder. Gargamela dokunursa oyuncu 15 puan kaybeder.
- **Puan :** Oyuncu sınıfında bulunan PuaniGoster() metodu override edilerek her bir oyuncu için özelleştirilecektir.

Projemizde algoritmayı düşmanların konumundan keyboard ile değiştirdiğimiz oyuncu konumu arasında en kısa yolu bulmak için kullandık.

En kısa yolu buldurduktan sonra renklendirme işlemi ile birlikte yolumuzu çizdirdik.

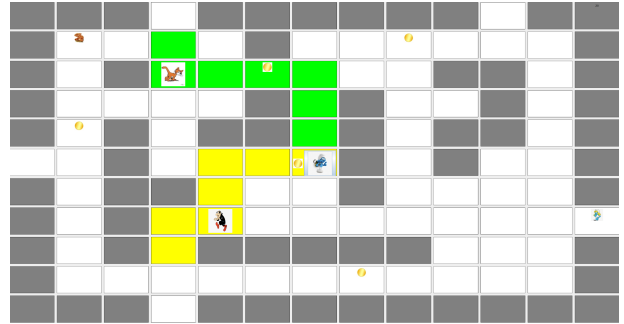


### *Düşman Sınıfında Bulunan Özellikler*

Bu sınıfta karakter sınıfı kalıtım olarak verilecektir.

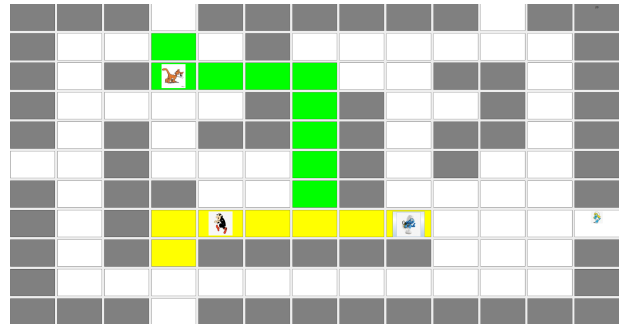
### *Düşman Türlerine Alt Sınıflar (2 Adet):*

- **Azman (Düşman 1):** 1 birim ilerler. Azman Gargamelin üzerinden atlayamaz.
- **Gargamel (Düşman 2):** 2 birim ilerler. Gargamel ,Azmanın üzerinden atlayabilir.



### *Dijkstra Algoritması:*

Bilgisayar bilimlerinde kullanılan ve algoritmayı literatüre kazandıran kişinin ismini taşıyan dijkstra algoritması, verilen bir şekilde (graph) en kısa yolu (shortest path) bulmak için kullanılır.

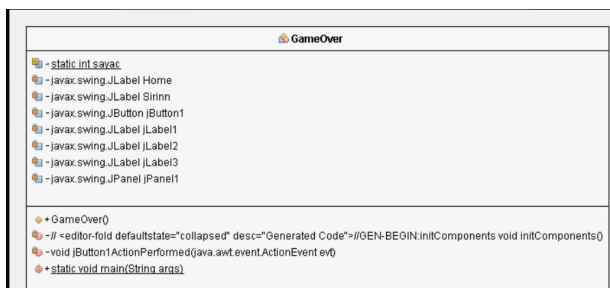
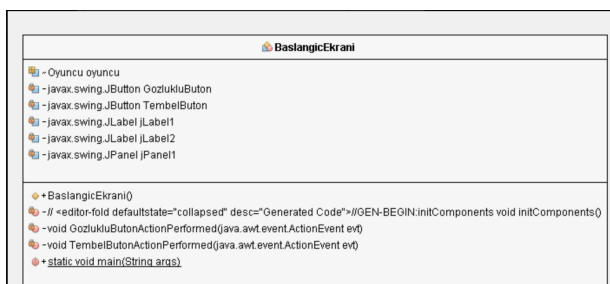
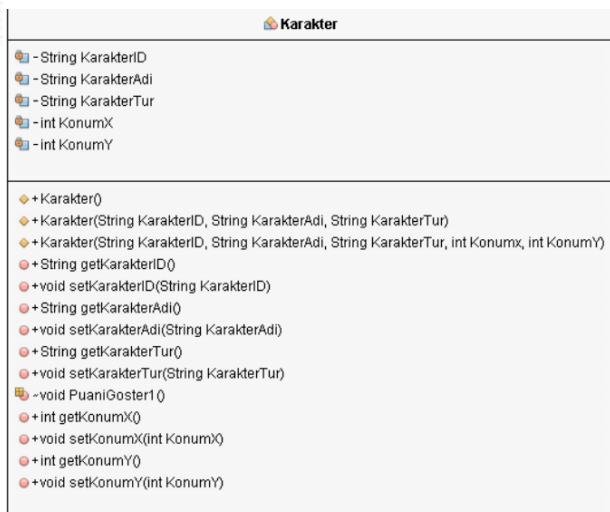
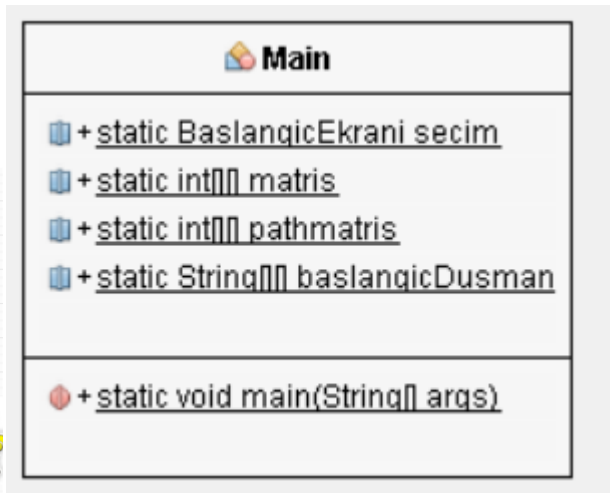


- Program; harita ve oyuncu bilgilerini tek bir txt dosyasından okuyacaktır. Harita.txt dosyasında hangi düşman karakterin oyunda olacağı ve hangi kapıdan giriş yapacağı bilgisi bulunmalıdır. Size verilen harita sabit kalacak fakat karakter bilgilerinin değiştirilebilir olması beklenmektedir bunun için dinamik yapı oluşturmalısınız. List, queue, stack veya dizi kullanabilirsiniz.
- Program çalıştırıldığında kullanıcıya harita grafiksel olarak sunulmalıdır
- Sol üst köşenin (0,0) noktası olduğunu unutmayınız bu durumda oyuncunun başlayacağı sarı nokta (5,6) noktası olarak belirlenmiştir. Mavi karede seçtiğiniz Gözlüklü Şirin ya da Tembel Şirin oyuncusu bulunacaktır. Bunun için kullanacağınız görselleştirme işlemi size bırakılmıştır fakat hangi karakterin oynadığı belirli olmalıdır.
- Program başlar başlamaz, düşman karakter oyuncuya en kısa kaç adımda ulaşabileceği bilgisini her adımda vermelidir. Örneğin Gözlüklü Şirin

A 15x15 grid with a 5x5 green square in the center. A blue robot is at the bottom center of the green square, and a red cat is at the top right. Four pink squares are labeled A, B, C, and D. A is at (2, 5), B is at (10, 10), C is at (5, 1), and D is at (10, 14). Gray squares are at the edges and in a pattern around the green square.

- ## 4.SONUÇ

## 5.UML DIAGRAMI



## 6.KAYNAKÇA

- <https://www.javatpoint.com/java-keylistener>
- <https://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyListener.html>
- <https://stackoverflow.com/questions/10876491/how-to-use-keylistener>
- <https://www.kodlamamerkezi.com/java/java-timer-zamanlayici-kullanimi/>
- <https://medium.com/gokhanyavas/javada-tip-dönüşümleri-ve-string-ifadeler-4d5457497c78>
- <https://www.tutorialspoint.com/java/button-click-event/>
- <https://stackoverflow.com/questions/616924/how-to-check-if-the-key-pressed-was-an-arrow-key-in-java-keylistener>