

Programlama Laboratuvarı II

PROJE#1 ŞİRİNLER

Proje Teslim Tarihi : 28.03.2021

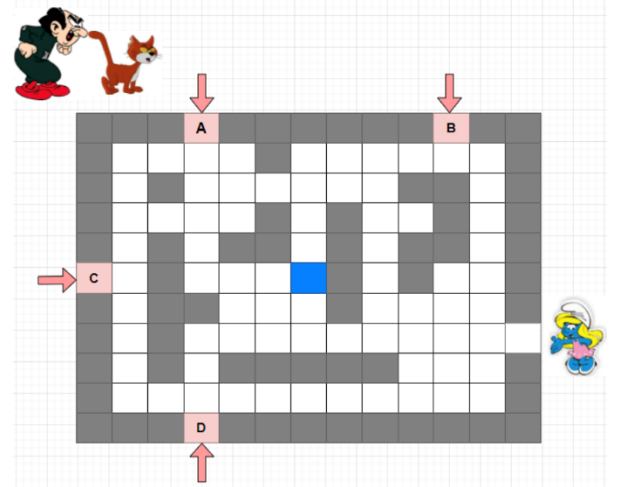
Özet

Bu projede oyunda kullanılacak labirentin haritası, oyunda yer alacak düşman karakterlerin ismi ve başlayacağı kapının ismi bir txt dosyası aracılığıyla okunur. Oyunda oynanacak karakter başlangıç ekranında yapılacak seçime göre belirlenir. Labirent txt dosyasından alınan '1'(hareket alanı) ve '0'(duvar) değerlerine göre labellar aracılığıyla oluşturulur. Düşman karakterleri dosyadan okunan kapı ismine göre haritadaki ilgili kapının konumuna yerleştirilir. Şirine ise labirentin belirli bir konumuna yerleştirilmiştir. Oyuncunun hareketi yön tuşları aracılığıyla gerçekleştirilir. Oyun içerisinde bulunan düşman karakterler oyuncu ile arasındaki en kısa yolu hesaplayarak oyuncuya ulaşmaya çalışır. Oyuncu oyuna 20 puan ile başlamakta olup oyun içerisinde yer alan objeleri toplayarak puan kazanabilmektedir. Düşman karakterlerle olan temasında ise puan kaybeder. Skorun 0 veya altına inmesi durumunda oyun kaybedilir eğer oyuncunun skoru 0'ın üstündeyken şirineye ulaşırsa oyun kazanılır.

Programlama dili olarak java, geliştirme ortamı olarak eclipse kullanılmış olup arayüz tasarımı için java swing tercih edilmiştir.

Giriş

Bu proje ile nesneye yönelik programlama ve veri yapıları algoritmalarını kullanarak Şirinler oyunu tasarlanacaktır. Seçilen oyuncunun Labirent içerisinde puanlarını bitirmeden önce Şirine'ye ulaşması gerekmektedir. Hedefe doğru giderken oyuncuyu birçok zorluk beklemektedir. Oyun içerisinde iki farklı oyuncu ve iki farklı düşman karakteri olacaktır. Kullanıcı oyun başlamadan önce seçtiği oyunculardan birini klavye yardımı ile kontrol edecek ve Şirine'ye ulaştırmaya çalışacaktır. Yine oyun başlamadan önce seçmiş olduğu düşman karakterlerden biri veya birden fazlası da onu durdurmaya çalışacaktır.



Projede yukarıda gösterilen harita kullanılacaktır. Mavi renkli kutu oyuncuların (yani kullanıcı tarafından kontrol edilecek olan karakter) başlama noktası olacaktır. Kırmızı oklar ise düşman karakterlerin labirente giriş yapabileceği kapılar olacaktır. Kullanıcının kontrol ettiği oyuncu puanları bitmeden önce Prenseşe ulaşmalıdır aksi halde oyuncu başarısız olur.

Yöntem

Öncelikle projeye bizden istenen classlar, değişkenler ve metodlar oluşturularak başlanmıştır. Bunlar;

Karakter Sınıfı

Yapıcı metotları parametrelili ve parametresiz olacak şekilde iki adet oluşturulmuştur. Parametreler id(int), ad(String), tur(String), lokasyon(Lokasyon) olarak oluşturulmuştur. Oluşturulan parametrelerin getter ve setter metotları oluşturulmuştur.

Oyuncu Sınıfı

Karakter sınıfından kalıtım alınmıştır. Yapıcı metotları parametrelili ve parametresiz olacak şekilde iki adet oluşturulmuştur. Base classtan id,ad,tur,Lokasyon bilgileri alınmış olup skor(int), hareketHizi(int), gargamelTemas(int),azmanTemas(int) şeklinde yeni parametreler oluşturulmuştur. PuaniGoster ve DusmanTemas metodları sonradan override edilmek üzere oluşturulmuştur. Oluşturulan parametrelerin getter ve setter metotları oluşturulmuştur.

Düşman Sınıfı

Karakter sınıfından kalıtım alınmıştır. Yapıcı metotları parametrelili ve parametresiz olacak şekilde iki adet oluşturulmuştur. Base classtan id,ad,tur,Lokasyon bilgileri alınmış olup imajelcon(Imajelcon), kapi(String) şeklinde yeni parametreler oluşturulmuştur. Düşman karakterlerin oyuncuya ulaşmak için kullanacağı yolu EnKisaYol metodu dijkstra algoritması kullanılarak oluşturulmuştur. DijkstraList metodu ile bulunan pathlerden en kısa yolun seçilmesi sağlanmıştır.

Oluşturulan parametrelerin getter ve setter metotları oluşturulmuştur.

Lokasyon Sınıfı

Yapıcı metotları parametrelili ve parametresiz olacak şekilde iki adet oluşturulmuştur. Oluşturulacak nesnenin koordinat bilgileri x(int),y(int) şeklinde bu classtan faydalanılarak oluşturulmuştur. Oluşturulan parametrelerin getter ve setter metotları oluşturulmuştur.

Objekt Sınıfı

Yapıcı metotları parametrelili ve parametresiz olacak şekilde iki adet oluşturulmuştur. Parametreler lokasyon(Lokasyon), puan(int), imajelcon(Imajelcon), isim(String) olarak oluşturulmuştur. Statik olarak oluşturulacak altın ve mantar miktarları değişkenlerde tutulmuştur. Oluşturulan parametrelerin getter ve setter metotları oluşturulmuştur.

Puan Sınıfı

Oyuncu sınıfında bulunan PuaniGoster() ve DusmanTemas() metotları bu class içerisinde override edilmiş olup her bir oyuncu için özelleştirilmesi sağlanmıştır.

Başlangıç Penceresi Sınıfı

Proje başlatıldığında ilk karşılaşılabilecek ekrandır. Oyuncudan tembel şirin veya gözlüklü şirin arasından buton yapısı kullanılarak tercih yapması istenmektedir. Yapılan tercihe göre labirent sınıfına gözlüklü şirin için '1' tembel şirin için '2' değeri gönderilir. Tercihin yapılmasının ardından bu pencere kapanır ve oyun ekranı açılır.

Labirent Sınıfı

Parametrelili olacak şekilde bir adet yapıcı metot oluşturulmuştur.

arayuzOlustur() : Oyun ekranının boyutları buradan belirlenmiş, skor değişkenini gösterecek label oluşturulmuştur. Labirentin çıkış noktasını göstermek amacıyla şirinenin resmi eklenmiştir.

karakterSecim() : Başlangıç penceresinden gelen sayıya göre gözlüklü veya tembel şirinin nesne oluşturma işlemi yapılmıştır. Oyuncular oluşturulurken id,isim,tür,lokasyon ve skor bilgileri yapıcı metod aracılığıyla verilmiş, hareket hızı ve düşmanlar ile temasında kaybedecek puanı set metodu ile yapılmıştır. Önceden oluşturulan oyuncucon değişkenine seçilmiş karakterin ikonu verilmiştir.

labirentOku() : Txt dosyasında verilmiş olan labirentin haritasını okuma işlemi buradan yapılmıştır. Dosyadan scanner fonksiyonu aracılığıyla okuma işlemi yapılmış olup okunan değerler labirent ismindeki matrisin içinde tutulmuştur.

labirentOlustur() : labirentOku() fonksiyonu ile okunan değerler aracılığıyla labirentin oluşturulması bu fonksiyonda sağlanmıştır. Oluşturulan labirentin genişliği, rengi gibi detaylar buradan belirlenmiştir.

karakterOku() : Txt dosyasında düşman karakterlere ait isim ve kapı bilgisi buradan scanner fonksiyonu aracılığıyla okunmuştur. Düşman karakterlerin oluşturulma işlemleri bu metod içerisinde yapılmıştır. Oluşturulan düşmanlara ait id,isim,tür,lokasyon,ikon ve kapı bilgisi buradan yapıcı metod ile oluşturulmuştur. Oluşturulan düşmanlar , “dusmanlar” isminde bir arraylist ile tutulmuştur.

karakterYerlestir() : Oluşturulan karakterlerin labirent üzerine yerleştirilme işlemleri buradan yapılmıştır

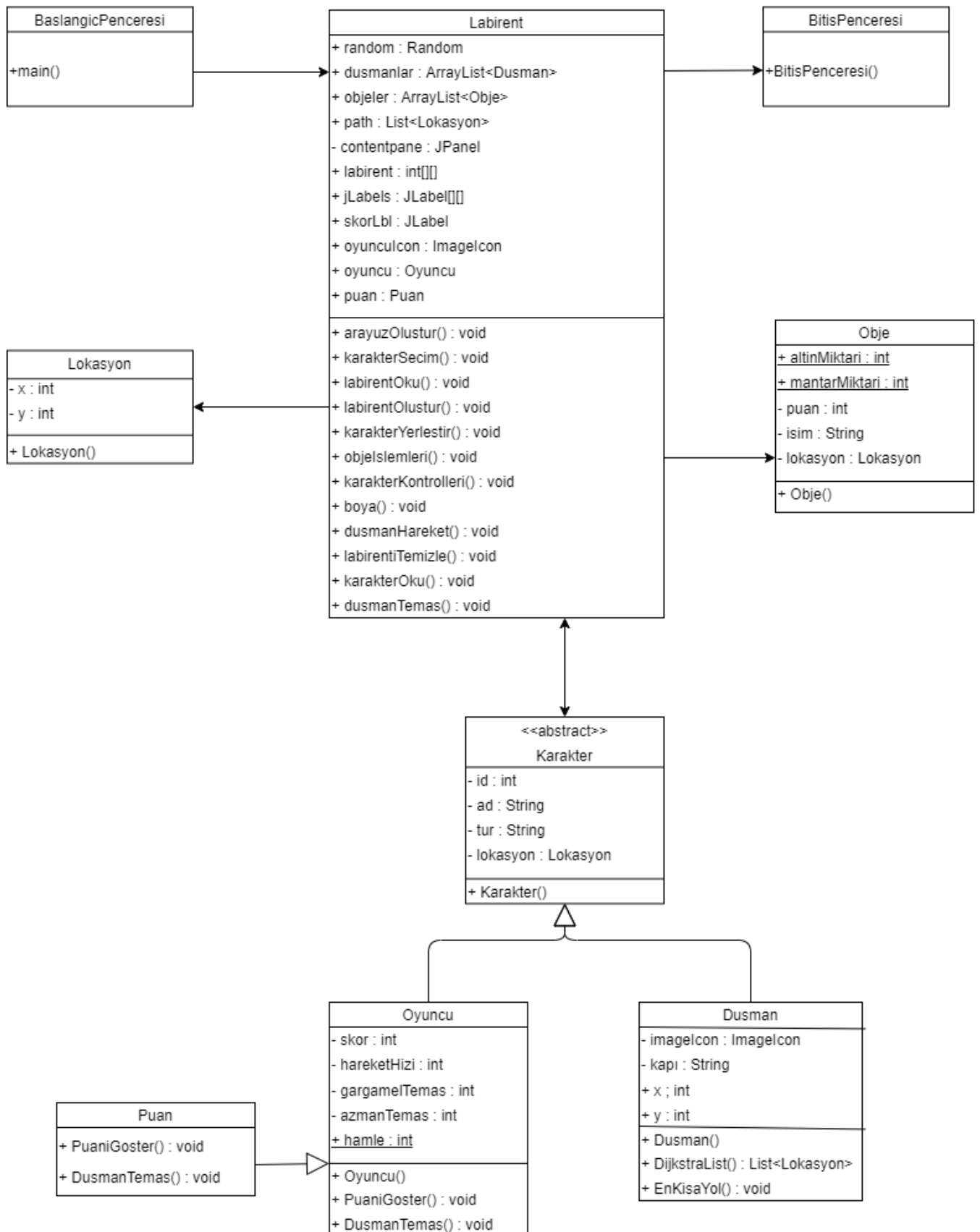
objeislemleri() : Oluşturulacak altın ve mantar objelerini oluşturmak için lokasyon tipinde dizi oluşturulmuştur. Oluşması gereken altın ve mantar miktarlarına göre labirentte ‘1’ noktasına gelecek rastgele bir lokasyon oluşturulmuş diziler içerisinde saklanmıştır. Bu lokasyon bilgileri ve timer fonksiyonu aracılığıyla belirli süreler arasında objeler oluşturulma ve silinme işlemleri yapılmıştır. Oluşturulan objeler, “objeler” ismindeki bir arraylist aracılığıyla tutulmuştur.

karakterKontrolleri() : Oynanacak karakterin hareket fonksiyonları burada KeyListener classı ile yapılmaktadır. Karakter yön tuşları ile labirentin içerisinde gezinebilmektedir. Karakterin ikonunun taşınma işlemleri ve eski konumundaki ikonun kaldırılma işlemleri de buradan yapılmaktadır. Karakterin objeler ile temasında kullanıcıya puan kazandırması ve temas edilen objenin silinme işlemi buradan yürütülmüştür. Kullanıcının her hareketinden sonra her bir düşman için enKisaYol() metodu buradan çağırılıp ilgili labelların boyanma işlemleri boya() metodu ile yapılmaktadır. Eğer oyuncu hedefe ulaşmış, skoru 0 veya altına inmişse BitisPenceresi() buradan çağırılmıştır.

boya() : Her bir düşman ile oyuncu arasındaki en kısa yolu temsil eden labelların boya işlemleri buradan yapılmaktadır. Her bir düşman karakteri için farklı renkler kullanılmıştır.

dusmanHareket() : Düşman karakterin hareket fonksiyonları buradan yürütülmüştür. Oyuncu ile temasında dusmanTemas() metodu ile düşman karakterin başlangıç konumuna geri dönmesini ve puan işlemleri buradan yapılmıştır.

UML SINIF DİAGRAMI:



Kaba Kod:

1- Karakter classı abstract olarak oluşturulur. Kendisinden miras alacak olan oyuncu ve düşman classlarının ortak özellikleri (id,isim,tür,lokasyon) yazılır.

2- Oyuncu classı oluşturulur. Karakter classından kalıtım alınır. Oyuncu classına has özellikler (skor,hamle) oluşturulur. PuaniGöster ve DusmanTemas metodları override edilmek üzere oluşturulur.

3- Dusman classı oluşturulur. Karakter classından kalıtım alınır. Dusman sınıfına has özellikler (imajelcon,kapi) oluşturulur. Oyunda oluşturulacak düşmanlara ait en kısa yol oluşturma işlemi enKisaYol ve dijkstraList metotları ile sağlanmaktadır.

4- Puan classı oluşturulur. Oyuncu classından kalıtım alınır. Oyuncu classına ait PuaniGoster ve DusmanTemas metodları override edilerek her bir oyuncu için özelleştirilir.

5- Lokasyon classı oluşturulur. Lokasyon classına has özellikler (x,y) oluşturulur. Oyunda oluşturulacak obje ve karakterlerin koordinat bilgileri bu class aracılığıyla tutulur.

6- Obje classı oluşturulur. Obje classına has özellikler (puan,imajelcon,lokasyon,isim) oluşturulur. Oyunda oluşturulacak objelerin oluşma miktarları bu class içerisinde static bir değişkende tutulur.

7- BaslangicPenceresi classı oluşturulur. Bu classta oyun esnasında oynanacak olan karakterin seçilme işlemi yapılmaktadır.

8- Labirent classı oluşturulur. Bu classta oyunun genel işleyişine dair her şey burada yapılmaktadır. İlk etapta oyunun oynanacak olan pencerenin oluşturulması ve skor labelının oluşma işlemi için arayuzOlustur metodu kullanılmıştır. BaslangicPenceresi classından seçilen karakterin oluşturulma işlemi için karakterSecim metodu kullanılmıştır. Harita.txt üzerinden okunması gereken labirentin okunup bir matrisin içine atılması için labirentOku metodu ve bu matristeki verileri kullanarak labirentin çizilme işlemi labirentiOlustur metodu ile yapılmıştır. Harita.txt dosyasında bulunan düşman ismi ve ortaya çıkacağı kapı karakterOku metodu ile Dusman tipinde oluşturulan bir arrayliste alınmış ve oluşturulan düşmanların labirentte gösterilme işlemi karakterYerlestir metodu ile yapılmıştır. objelslemleri metodu oluşturulmuş, belirli zaman aralıklarında belirli objeler ortaya çıkma ve silinme işlemleri buradan yürütülmüştür. Oyuncunun tüm hareket fonksiyonları, herhangi bir obje veya düşman ile temasının olup olmaması, düşmanın hareket fonksiyonları, oyuncunun puan durumunun her hareketten sonra güncellenme işlemi, düşman ile oyuncu arasındaki en kısa mesafenin boyanma işlemi karakterKontrolleri metodu ile yapılmıştır.

9- BitisPenceresi classı oluşturulur. Eğer oyuncu şirineye ulaşmış ise kazanma, skorun 0 veya altına inmesi durumunda kaybetme ekranını oluşması sağlanır.

Zaman Karmaşıklığı

Algoritmanın karmaşıklık analizini yaparken sabit zamanlı veriler göz ardı edilerek sadece oluşturulan iç içe döngülerin hesabı yapılmıştır. Big o analizine göre sadece iç içe döngülerin hesabının yapılması yeterlidir. Algoritma EnKisaYol metodu ile hesaplanmış olup özyinelemeli fonksiyon aracılığıyla her defasında tek bir değer için kendini çağırılmaktadır. Karmaşıklık analizi bu sebepten ötürü **$O(n)$** 'dir.

Şirinler

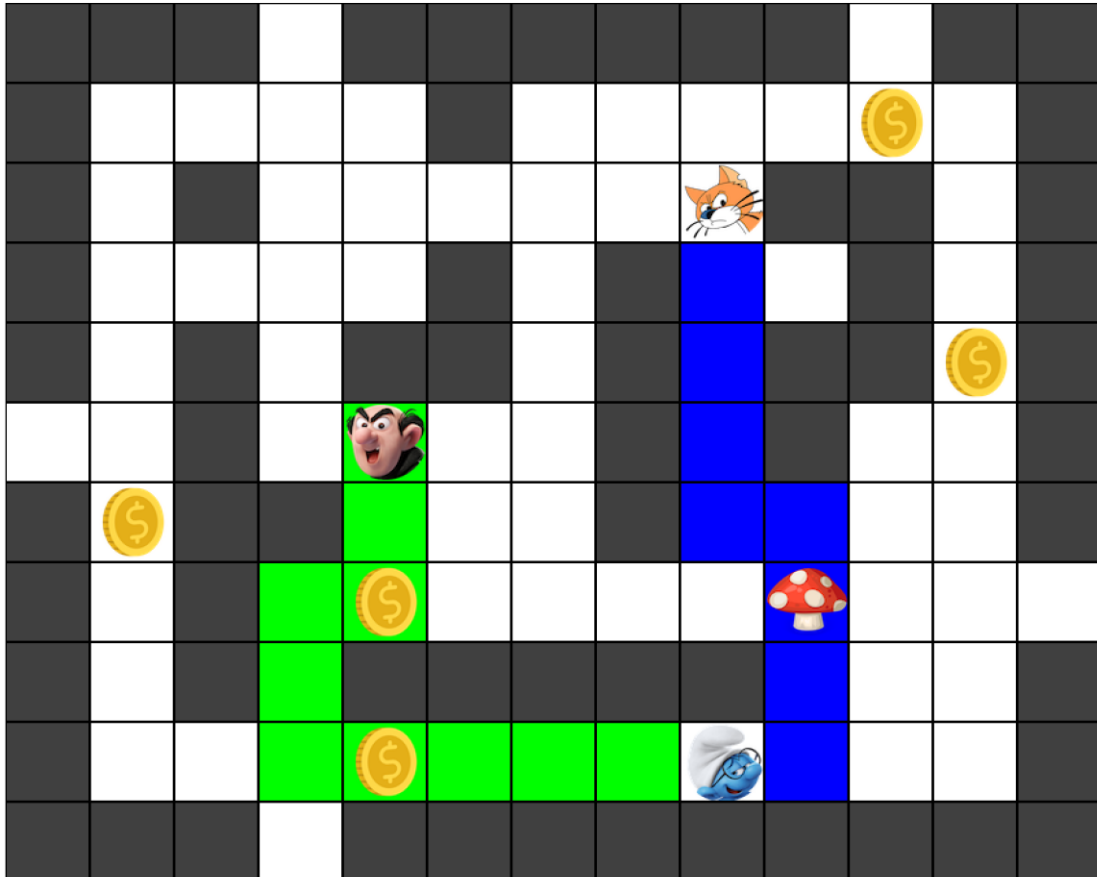


Gozluklu Sirin



Tembel Sirin

Skor = 20



Kazandınız

Skorunuz = 100

Yasin Çalışkan 180201060

Atakan Atalar 190201038



Kaynakça :

<https://www.javatpoint.com/java-swing>

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

<https://www.pngwing.com/en/search?q=smurfs>