

Programlama Laboratuvar Raporu

ŞİRİNLER

Berke
YILDIZ
Kocaeli
Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği
190202049@kocaeli.edu.tr

Çağrı ŞENTRÜK
Kocaeli
Üniversitesi
Mühendislik
Fakültesi

Bilgisayar Mühendisliği
190202055@kocaeli.edu.tr

Özet-Şirinler programımızda 2 kötü karakter,2 iyi karakter bulunmaktadır. Kötü karakterler Azman ve Gargamel. İyi karakterler Tembel Şirin ve Gözlüklü Şirindir. Kötü karakterler belirtilen kapılardan başlayarak. Şirineye ulaşmadan iyi karakterleri yakalamaya çalışmaktadır. Kötü karakterlerin iyi karakterlere olan en kısa uzaklığını bulmada Dijkstra algoritması kullanıldı.

I. Giriş

Projemizde kullanılan “harita.txt” bulunmaktadır. Txt içinde programda grafik üzerine bastırılacak olan kötü karakterlerin isimleri, başlangıç kapıları ve haritanın bilgisi içerisinde yer almaktadır.

Kullanıcıya hangi karakteri seçim yapılacağı sorulur. Txt den elde edilen bilgiler ve yapılan seçim sonucunda grafiğe yerleştirilir. Kullanıcı, iyi karakterlerin hareketini klavyedeki yön tuşları ile sağlamaktadır. Kapılarda yer alan kötü karakterlerin hareketleri, iyi karaktere olan en kısa mesafe olarak hesaplanır ve program tarafından yürütülür.

Kullanıcı haritada rastgele alanlarda belirli saniye aralıklarında çıkacak olan altın ve mantarları toplayarak puanını artırabilir.

Kullanıcı iyi karakterin puanları bitmeden Şirine’ye ulaşmalıdır yoksa oyun biter ve ekrana kaybettiniz yazısı basılır.

II. Yöntemler

Bu başlık altında giriş kısmında söylenen işlemlerin hangi metotlarla yapıldığı açıklanacaktır.

A.Dosya Okuma

“Harita.txt” dosyamızı
BufferedReader
kullanarak satır satır
okuyup. Parçalara
ayırdık if else yapısı ile
hangi kötü karakterin
olduğu ve nereden
başlayacağını
değişkenlerimize
atadık. Bu bilgileri
kullanarak kötü
karakterleri
oluşturduk. Adlarına
göre array listimize
ekledik. Sonrasında
haritamızın bilgilerini
daha sonra
kullanılmak üzere bir
matrise aktardık ve
dosyamızı kapattık.

B.Arayüz Programlama

Projemiz
çalıştırdığımızda ilk
olarak ana seçim olarak
tanımladığımız bir
JFrame oluşturduk ve
bunun içerisine JPanel
açtık. Açtığımız JPanel
üzerinde Tembel Şirin ve
Gözlüklü Şirin butonu
ekledik ve
oluşturduğumuz arka
planımızı JLabel yardımı ile
konumlandırdık. Karakter seçimi
yapıldığında Labirent adıyla
oluşturduğumuz JFrame’e
yönlendirir. Buradan da
oyunumuzun işlevlerinin yapıldığı
Oyun adlı JPanel açılır. Burada
haritamızın çizimi, karakterimizin

skoru ,karakterlerin bulunduğu
konumlar ve kötü karakterlerin iyi
karaktere olan en kısa yolu Graphic
kütüphanesi içerisinde bulunan
drawImage, paint yardımı ile
çizdirilir. KeyAdapter sınıfından
yararlanarak yön tuşları ile
karakterimizin hareketini sağladık.

C.Sınıf ve NesneYapısı

Projemizde Java ve nesneye yönelik
programlama yapısını kullandık.

Lokasyon (Class) : Her karakterde ve objede
kullanacağımız haritada bulundukları
konum bilgisini taşıyan sınıfımızdır. X ve Y
koordinatlarını tutacak değişkenlere
sahiptir. Constructor, get, set, metodları yer
almaktadır.

Harita (Class) : Haritamızın okunduğu
sınıfımızdır. Kötü karakterlerimiz
oluşturulup array liste eklendiği kısımdır.
Sınıfımız içerisinde ek olarak yol, duvar ve
şirine resimlerini içerir. Constructor, get,
set, metodları yer almaktadır.

Karakter (Class) : Oyuncu, Düşman
sınıflarının atasıdır. Karakterin adını, türünü,
id sini ve karakterin konumunu tutacak
lokasyon değişkenlerine sahiptir.
Constructor, get, set, rotaBul, sirinTakip,
karakterlerin hareket etmesini sağlayacak
ilerle metodları yer almaktadır.

Oyuncu (Class): Karakter sınıfından kalıtım
alarak oluşur. Oyuncunun adını tutacak
oyuncuAdi, oyuncunun türünü tutacak
oyuncuTur, oyuncunun id sini tutacak
OyuncuID, oyuncunun skorunu tutacak skor
değişkenlerine sahiptir. . Constructor, get,
set, PuaniGoster metodları yer almaktadır.

Düşman (Class): Karakter sınıfından kalıtım
alarak oluşur. Düşmanın adını tutacak
dusmanAdi, oyuncunun türünü tutacak
dusmanTur, oyuncunun id sini tutacak
dusmanID tutacak değişkenlerine sahiptir.
Constructor, get, set, metodları yer
almaktadır.

GözlüklüŞirin (Class): Oyuncu sınıfından

kalıtım olarak oluşur. Gözlüklü Şirin'in resminin tutulacağı Image gözlüklüşirin adlı değişkende tutulur. Constructor, get, set, ilerle fonksiyonun 2 birim gidecek şekilde düzenlenmiş metodları yer almaktadır.

TembelŞirin (Class): Oyuncu sınıfından kalıtım olarak oluşur. Tembel Şirin'in resminin tutulacağı Image tembelsirin adlı değişkende tutulur. Constructor, get, set, ilerle fonksiyonun 1 birim gidecek şekilde düzenlenmiş metodları yer almaktadır.

Azman (Class): Düşman sınıfından kalıtım olarak oluşur. Azman'ın resminin tutulacağı Image azman adlı değişkende tutulur. Azman'ın ilk konumunun tutulacağı lokasyon değişkeninde tutulur. Constructor, get, set, ilerle fonksiyonun 2 birim gidecek şekilde düzenlenmiş metodları yer almaktadır.

Gargamel (Class): Düşman sınıfından kalıtım olarak oluşur. Gargamel resminin tutulacağı Image Gargamel adlı değişkende tutulur. Azman'ın ilk konumunun tutulacağı lokasyon değişkeninde tutulur. Constructor, get, set, ilerle fonksiyonun 1 birim gidecek şekilde düzenlenmiş metodları yer almaktadır.

Obje (Class): Altın ve mantarın kalıtım aldığı sınıftır. Objenin adının, puanın, objenin konumunu tutacak lokasyon değişkenlerine sahiptir. Constructor, get, set metodları yer almaktadır.

Altın (Class): Obje sınıfından kalıtım olarak oluşur. Altın resminin tutulacağı Image Altın adlı değişkende tutulur. Constructor, get, set metodları yer almaktadır.

Mantar (Class): Obje sınıfından kalıtım olarak oluşur. Mantar resminin tutulacağı Image Mantar adlı değişkende tutulur. Constructor, get, set metodları yer almaktadır.

C.Oyun Mantığı

Kullanıcı seçtiği Şirin ile haritanın merkezinde oyuna başlar. Kullanıcının karakteri hareket ettirdiği her adımda düşmanlar karakterimize doğru en kısa

mesafeden hareket eder. Düşmanlar karakteri yakaladığında düşmanın özelliğine göre kullanıcı puan kaybeder. Kullanıcının oyunu kazanması için 0 puana düşmeden önce şirineye ulaşması gerekmektedir. Şirineye ulaşmadan 0 puan veya altına düşer ise oyunu kaybeder. Ayrıca farklı zamanlarda ortaya çıkan altınlar ve mantarlar kullanıcıya puan kazandırarak oyunu kazanması için imkan sağlamaktadır. Altınlar konumları rastgele olmak üzere her 10 saniyede 5 adet oluşur ve her biri 5 saniye boyunca alınabilir. Benzer şekilde mantarlar ise konumları rastgele olmak üzere her 20 saniye 1 adet oluşur ve 7 saniye boyunca alınabilir. Kullanıcımız bu kurallar ve şartlar doğrultusunda oyunu kazanmaya çalışmaktadır.

4.DJİKSTRA

Kötü karakterimiz iyi karakterimize kadar olan tüm yolları kontrol ederek gidebileceği en kısa mesafeyi bulur ve onu değişkene kaydederek rotasını oluşturur.

Dijkstra zaman

karmaşıklığı:

```
for (int j = 0; j < AdimSayisi; j++) {  
    for (int x = 0; x < 11; x++) {  
        for (int y = 0; y < 13; y++) {  
            harita[x][y] = 100;  
        }  
    }  
}
```

Resim(1)

Dış for'da n kere döner
içerisindeki for'lar belirli sayı
olduğundan dahil edilmez.

```

for (int i = mesafe - 1; i > -1; i--) {
    if (harita[yol[i + 1][0] + 1][yol[i + 1][1]] == i) {
        yol[i][0] = yol[i + 1][0] + 1;
        yol[i][1] = yol[i + 1][1];
    } else if (harita[yol[i + 1][0] - 1][yol[i + 1][1]] == i) {
        yol[i][0] = yol[i + 1][0] - 1;
        yol[i][1] = yol[i + 1][1];
    } else if (harita[yol[i + 1][0]][yol[i + 1][1] + 1] == i) {
        yol[i][0] = yol[i + 1][0];
        yol[i][1] = yol[i + 1][1] + 1;
    } else if (harita[yol[i + 1][0]][yol[i + 1][1] - 1] == i) {
        yol[i][0] = yol[i + 1][0];
        yol[i][1] = yol[i + 1][1] - 1;
    }
}

```

Resim(2)

Buradaki for'da n kere
döneceğinden resim (1) deki
for'un içerisinde
bulunduğunda çarpılır ve
sonuç n^2 hesaplanır.

Zaman karmaşıklığı

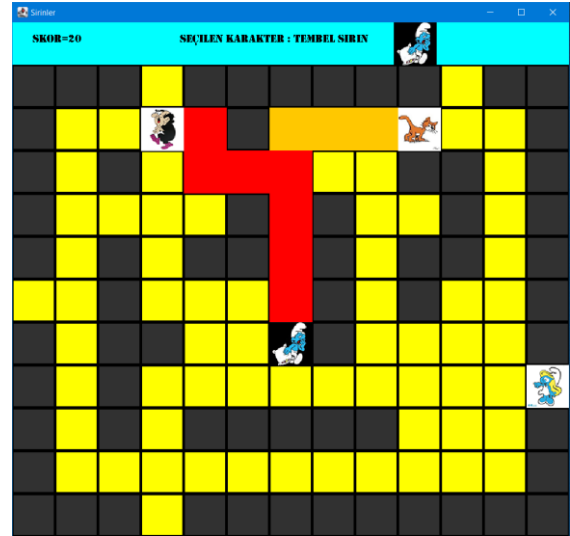
$O(n^2)$ dir.

Dijkstra bellek

karmaşıklığı:

10 tane int olduğundan
dolayısı 40 byte alan kaplar.

olmak üzere 2 sahne yer alıyor.



III.Geliştirme

Ortamı ve

Kullanılan Dil

Projeyi Java programlama dilinde
Windows işletim sistemi üzerinde
gerçekleştirirken, geliştirme ortamı olarak
JavaSE-15 JDK konfigürasyonlu Eclipse
idesini kullandık.

IV.Programı

n Genel

Yapısı Ve

Tasarımı

Program temel olarak seçim, oyun

V.Deneysel Sonuçlar

Java Swing kütüphanesinden
yararlanarak arayüz oluşturmak onu
oyun mantığının içine katmak gibi bir
çok konu hakkında bilgi sahibi olduk ve
uyguladık. Dijkstra algoritmasını oyun

yapısına katarak bu algoritmanın mantığını ve işleyişini uygulamalı olarak öğrendik.

VI. Sonuç

Projemiz basit bir mantık üzerine kurulu gözükse de bu fonksiyonları koda dökmek ve eksiksiz çalışacak hale getirmek için bir çok konu hakkında araştırma yapmamız gerekti. Bu nedenle projemiz kendimizi geliştirmemiz açısından bize oldukça katkı sağladı. Ayrıca daha önceden bildiğimiz nesneye yönelik programlama konusunda tecrübelenmemize olanak sağladı.

VII.Kaynaklar

1.Java Swing

<https://youtube.com/playlist?list=PLzIWkToFwqHRjHdDsrJhQSWGaoJkAHlU4>

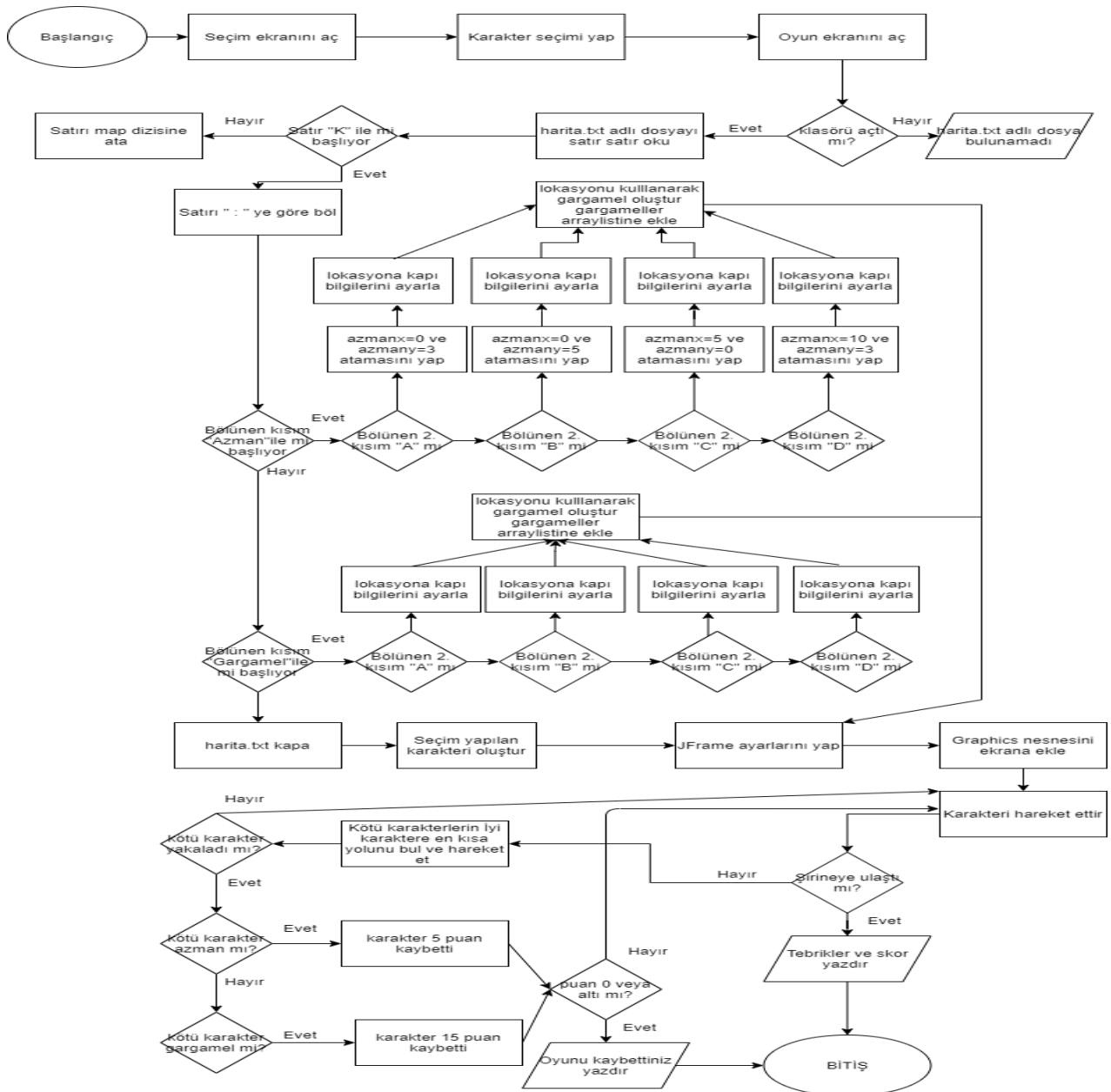
2.Java Temel Yapıları

https://www.udemy.com/share/101WziCUcbeVdRR3w=

3.Dijkstra Algoritması

<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

Akış Şeması



UML Diagramı

