

ŞİRİNLER

ÖMER KAÇAMAK

190201109

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

190201109@kocaeli.edu.tr

ÖZET

Bu proje ile nesneye yönelik programlama ve veri yapıları algoritmalarını kullanarak Şiriner oyunu tasarlandı. Seçilen oyuncunun Labirent içerisinde puanlarını bitirmeden önce Şirine'ye ulaşması gerekmektedir. Düşman karakterlerin oyuncumuzau yakalayabilmesinde en kısa yol algoritması kullanılacaktır.

Arkaplan oluşturuldu. Dosyaİslem adında yeni bir class oluşturuldu. Bu classta iki tane text dosyasından bilgiler alındı. En kısa yol algoritması oluşturuldu ve projeye eklendi.

GİRİŞ

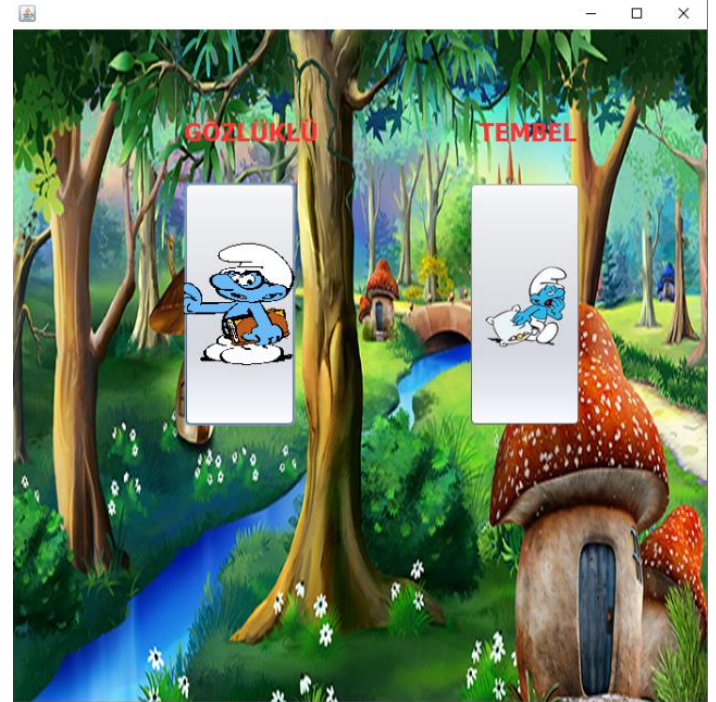
Oyun haritayı oyun klasörü içinde bulunan harita.txt dosyasından alır. Düşman karakterleri ve onların bulundukları kapıları ise karakterler.txt doyasından alır. Oyuna başlarken iki karakterden biri seçilmesi isteniyor ve bu işlem arayüzde kullanıcı tarafından seçilebiliyor. Düşman karakterlerimizden azman ve gargamel kullanıcının seçtiği şirini yakalaması için en kısa yolu seçmesi gerekmektedir.Bu işlem için en kısa yol algoritması kullanılacaktır.

YÖNTEM

Program NetBeans ortamında Java programlama dili ile yazılmıştır. Projede Inheritance , Polymorphism, Veri Yapıları, ve Swing kavramları kullanılmıştır. Kodda Constructor (Yapıcı) metotlar tanımlanmıştır.

Öncelikle karakter adında bir class oluşturuldu. Bu class düşman ve oyuncu classlarına kalıtım verecektir.Bu classlarda ad, isim, x,y değişkenleri ile birlikte temsil ettikleri resim değişkenleri tutulmuştur. Girişte kullanıcıyı bir arayüz karşılayacaktır. Bu arayüzde iki şirinden biri seçilecektir. Bu arayüz tasarlandı.

DENEYSEL SONUÇLAR



İlk olarak kullanıcıyı seçmesi üzerine fotoğraftaki giris ekranı karşılar.

*karakterler - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
İlk Dusman :Gargamel:Kapi:B
İkinci Dusman:Azman:Kapi:C

EnKisaYol
<pre> public static final int M = 11; public static final int N = 13; public static final int satir[] = { -1, 0, 0, 1 }; public static final int sutun[] = { 0, -1, 1, 0 }; int [][] ziyaret = new int [M][N]; Queue<Node> q int minDist </pre>
<pre> static boolean pozGecerilMi (int mat[], int ziyaret[],int satir, int sutun) static void KisaYol (int mat[], int i, int j, int x, int y) public static void yollariGoster(Node node) </pre>

Node
<pre> public ArrayList<Komsuluk> komsular int x, y, dist; Node parent; int duvar; public int minMesafe public boolean kullanildiMi </pre>

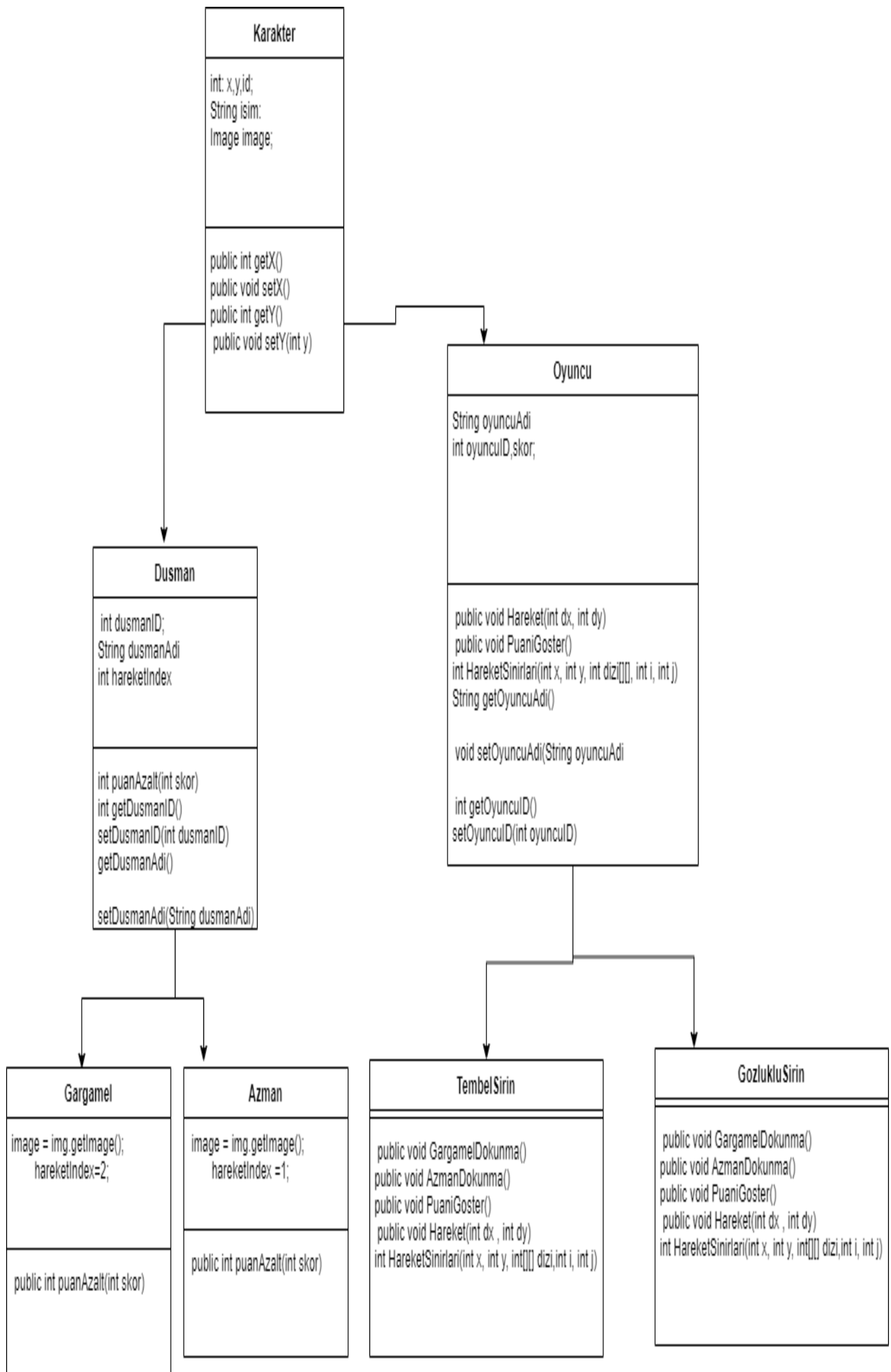
Komsuluk
<pre> public Node hedefVertex; public int mesafe; int x,y; </pre>
<pre> public Komsuluk(Node hedefVertex, int mesafe, int x, int y) </pre>

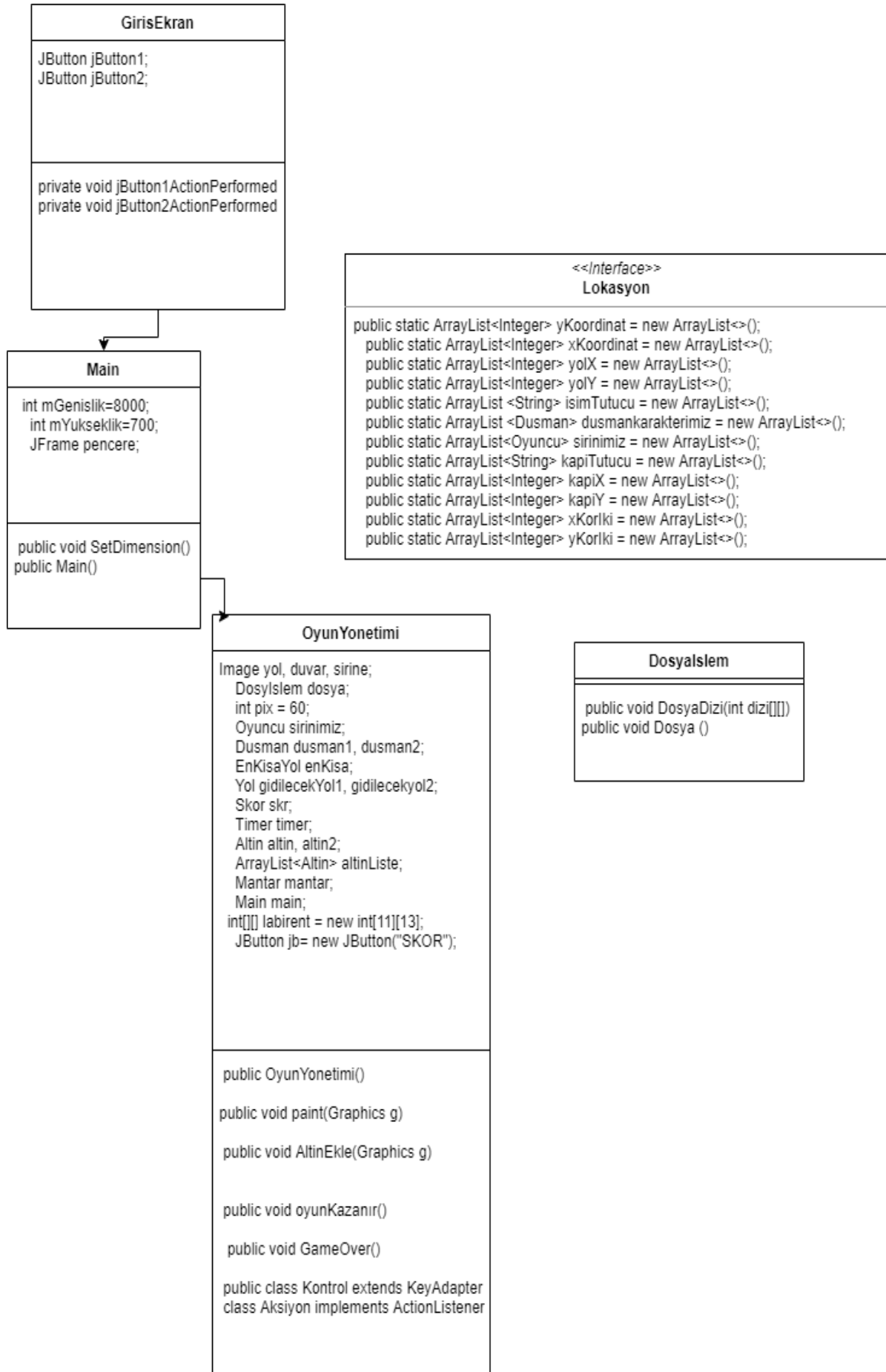
Altin(extends Thread)
<pre> public int x,y; Random rnd; public Image image; int [][] labirent = new int[11][13]; int dizi[][]; </pre>
<pre> public void run() public Altin(int dizi[]) </pre>

Mantar(extends Thread)
<pre> public int x,y; Random rnd; public Image image; int [][] labirent = new int[11][13]; int dizi[][]; </pre>
<pre> public void run() public Altin(int dizi[]) </pre>

Skor
<pre> public String skorYazi </pre>
<pre> public Skor() </pre>

Yol
<pre> int x,y; public Image image; </pre>
<pre> + method(type): type </pre>





KAYNAKÇA

- *Veri yapıları ve algoritma dersi slaytları*
- *<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>*
- *<https://www.tasarimkodlama.com/java-programlama/java-jframe-programlama-ornekleri/>*