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 Şirinler oyunu tasarlandı. Seçilen oyuncunun Labirent içerisinde puanlarını bitirmeden önce Şirine'ye ulaşması gerekmektedir. Düşman karakterlerin oyuncumuzau yakalayabilmesinde en kısa yol algoritması kullanılacaktır.

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.

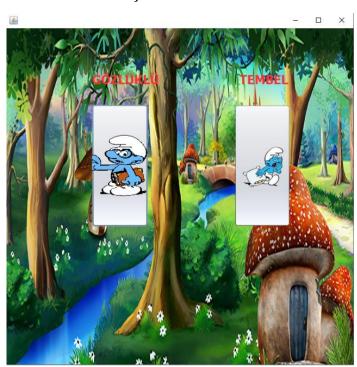
YÖNTEM

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 ettkileri resim değişkenleri tutulmuştur. Girişte kullanıcıyı bir arayüz karşılayacaktır. Bu arayüzde iki şirinden biri seçilecektir. Bu arayüz tasarlandı.

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

DENEYSEL SONUÇLAR



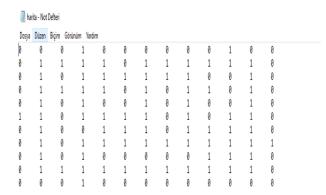
İlk olarak kullanıcıyı seçmesi üzerine fotoğraftaki giris ekranı karşılar.

*karakterler - Not Defteri

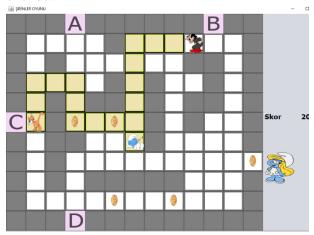
Dosya Düzen Biçim Görünüm Yardım

İlk Dusman : Gargamel : Kapi : B

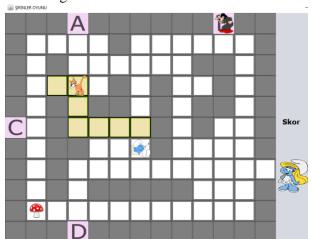
İkinci Dusman : Azman : Kapi : C



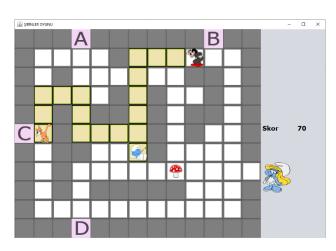
Txt dosyalarından harita ve karakterler okundu.



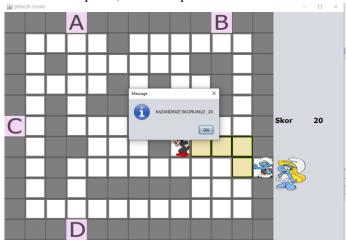
Düşman karakterler en kısa yoldan şirine ulaşmaya çalışır. Bu sırada en kısa yol harita üzerinde gösterilir.



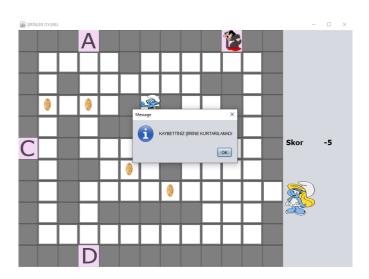
Düşman şirini yakaladığında düşman karakterin türüne göre puan azalır ve o düşman kapısına geri döner ve tekrar başlar.



Mantarlar 20 puan, altınlar 5 puan kazandırır.



Şirinimiz şirineye ulaştığında oyunu kazanırız.



Skorumuz 0 ve altına düşerse oyunu kaybederiz.

EnKisaYol

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

static boolean pozGecerliMi (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)

Node

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

Komşuluk

public Node hedefVertex; public int mesafe; int x,y;

public Komsuluk(Node hedefVertex, int mesafe, int x, int y)

Altin(extends Thread)

public int x,y;
 Random rnd;
 public Image image;
 int [][] labirent = new int[11][13];
 int dizi(][);

public void run() public Altin(int dizi[][])

Mantar(extends Thread)

public int x,y;
 Random rnd;
 public Image image;
 int [][] labirent = new int[11][13];
 int dizi[][];

public void run() public Altin(int dizi[][])

Skor

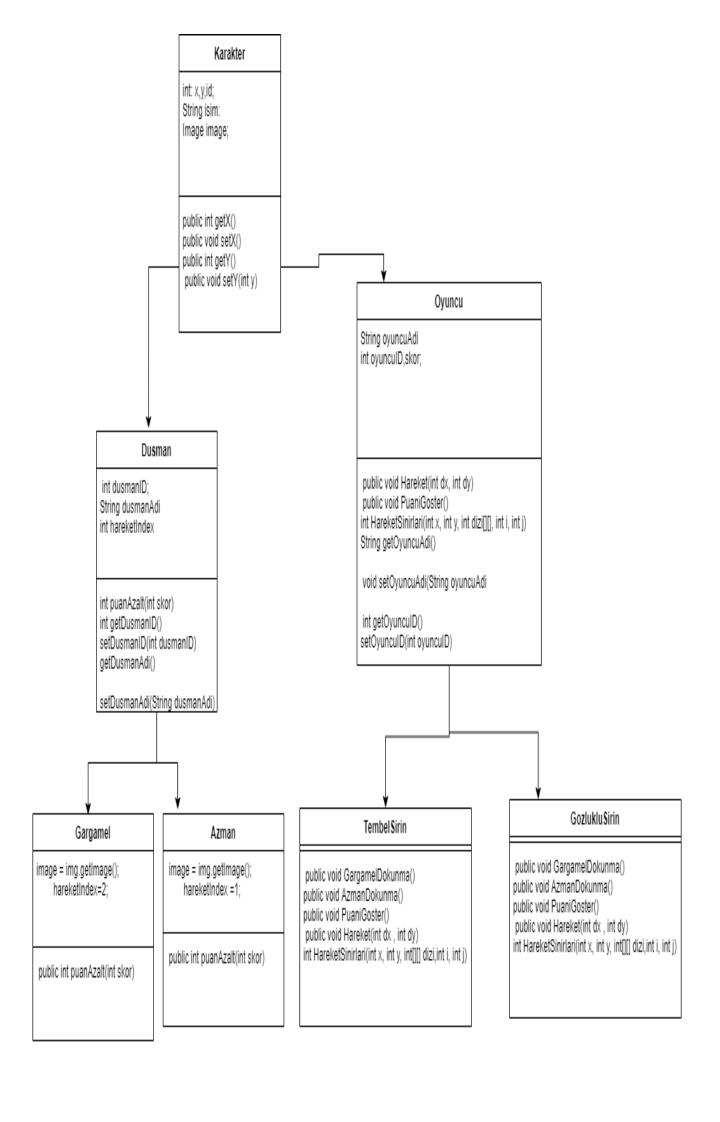
public String skorYazi

public Skor()

Yol

int x,y; public Image image;

+ method(type): type



GirisEkran

JButton jButton1; JButton jButton2;

private void jButton1ActionPerformed private void jButton2ActionPerformed

Main

int mGenislik=8000; int mYukseklik=700; JFrame pencere;

public void SetDimension() public Main()

<<Interface>> Lokasyon

public static ArrayList<Integer> yKoordinat = new ArrayList<>();
 public static ArrayList<Integer> xKoordinat = new ArrayList<>();
 public static ArrayList<Integer> yolX = new ArrayList<>();
 public static ArrayList<Integer> yolY = new ArrayList<>();
 public static ArrayList <String> isimTutucu = new ArrayList<>();
 public static ArrayList <Dusman> dusmankarakterimiz = new ArrayList<>();
 public static ArrayList<Oyuncu> sirinimiz = new ArrayList<>();
 public static ArrayList<String> kapiTutucu = new ArrayList<>();
 public static ArrayList<Integer> kapiX = new ArrayList<>();
 public static ArrayList<Integer> kapiY = new ArrayList<>();
 public static ArrayList<Integer> xKorlki = new ArrayList<>();
 public static ArrayList<Integer> yKorlki = new ArrayList<>();

OyunYonetimi

Image yol, duvar, sirine;
Dosylslem dosya;
int pix = 60;
Oyuncu sirinimiz;
Dusman dusman1, dusman2;
EnKisaYol enKisa;
Yol gidilecekYol1, gidilecekyol2;
Skor skr;

Timer timer;

Altin altin, altin2; ArrayList<Altin> altinListe;

Mantar mantar; Main main;

int[][] labirent = new int[11][13];

JButton jb= new JButton("SKOR");

public OyunYonetimi()

public void paint(Graphics g)

public void AltinEkle(Graphics g)

public void oyunKazanır()

public void GameOver()

public class Kontrol extends KeyAdapter class Aksiyon implements ActionListener

Dosyalslem

public void DosyaDizi(int dizi[][]) public void Dosya ()

KAYNAKÇA

- Veri yapıları ve algoritma dersi slaytları
- https://www.geeksforgeeks.org/dijkstra s-shortest-path-algorithm-greedyalgo-7/
- https://www.tasarimkodlama.com/java-programlama/java-jframe-programlama-ornekleri/