

STAR WARS

Şeyma Nur MUTLU ve Melike OĞUZ

Bilgisayar Mühendisliği

Kocaeli Üniversitesi

{170201004 & 170201028}@kocaeli.edu.tr

ÖZET

"Star Wars" adlı programda 2 iyi karakter, 3 kötü karakter bulunmaktadır. Kötü karakterler, Darth Vader, Kylo Ren, Stormtrooper ; iyi karakterler Master Yoda, Luke Skywalker'dır. Kötü karakterler grafik üzerindeki kapılardan giriş yapıp, iyi karakteri bitiş çizgisini gelmeden yakalamaya çalışmaktadır. Program, iyi karakterin konumuna göre kötü karakterlerin iyi karaktere olan en kısa uzaklığını bulmaktadır.

GİRİŞ

Star Wars projesinde kullanılmak üzere "harita.txt" adında bir txt bulunmaktadır. Txt içinde programda grafik üzerine bastırılacak olan kötü karakterlerin isimleri, başlangıç kapıları ve haritanın bilgisi şekilde yer almaktadır.

Program, bu bilgiler ışığında grafiği oluşturup kullanıcıya hangi iyi karakterle başlamak istediğini sorar. Seçim işlemi gerçekleştikten sonra iyi karakter, grafik üzerinde belirtilen sarı hücreden oyuna başlar. Kullanıcı, iyi karakterin hareketini klavyedeki yön tuşlarını kullanarak sağlamaktadır. Kapılarda yer alan kötü karakterlerin hareketleri, iyi karaktere olan en kısa mesafe leri olarak hesaplanır ve yürütülür bu işlemlerin hepsi program tarafından yapılmaktadır.

Kullanıcının kontrol ettiği iyi karakter canları bitmeden önce kupaya ulaşmalıdır aksi

halde iyi karakter için oyun yenilgi ile sonlanır ve ekrana 'Game Over' yazısı basılır.

2. Temel Bilgiler

Program Java programlama dilinde geliştirilmiş olup, tümleşik geliştirme ortamı olarak "NetBeans IDE"

kullanılmıştır. Programın yazılma esnasında git versiyon kontrol sistemi olarak GitHub kullanılmış olup, kodlarımızın gizliliği için GitHub Education hesabımızın bize sağlamış olduğu private repository özelliğinden faydalandık.

3. Tasarım

Star Wars programının programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

3.1. Hiyerarşik Yapı :

Star Wars Projesinde, DarthVader, KyloRen, Stormtrooper, MasterYoda, LukeSkywalker sınıfları özelliklerini temel ata sınıfları olan Character sınıfından almıştır. Bunun yanı sıra hareketliliği kolay sağlamak adına Location sınıfı mevcuttur. Karakter sınıflarının her birinde aktif olarak Location sınıfının nesneleri kullanılmaktadır.

3.2. Sınıfların Özellikleri :

Location (class) : Koordinat Düzlemine göre x ve y kordinatlarını tutan, Matris sistemine göre satır ve sütunları tutan değişkenlere, en kısa yol bulunurken yolun tamamını tutmak için Location tipinde adı path olan bir arrayliste ve en kısa yolun uzunluğunu tutacak distance adında bir değişkene sahiptir. Bu sınıfın iki tane yapılandırıcı methodunu kullanmaktayız. İyi karakterler default yapılandırıcıyı kullanırken kötü karakterler başka bir yapılandırıcı kullanır. Değişkenlerin getter ve setter fonksiyonları mevcuttur.

Character (class) : DarthVader, KyloRen, Stormtrooper, MasterYoda, LukeSkywalker sınıflarının atasıdır. Karakterin adını tutacak Ad, türünü (iyi/kötü) tutacak kind, ve karakterin ilerlediği koordinatları tutacak Location değişkenlerine sahiptir. Constructor, Get, Set ve EnKısaYol metotları yer almaktadır.

DarthVader (class) : Character sınıfı kalıtım olarak ata sınıfıdır. Constructor, Get, Set ve EnKısaYol metotları yer almaktadır. EnKısaYol, tüm matris elemanlarının bir olacağı göz önüne alınarak hesaplanmıştır.

KyloRen (class) : Character sınıfı kalıtım olarak ata sınıfıdır. Constructor, Get, Set ve EnKısaYol metotları yer almalıdır. EnKısaYol, tek harekette iki birim birden gideceği göz önüne alınarak hesaplanmıştır.

Stormtrooper (class) : Character sınıfı kalıtım olarak ata sınıfıdır. Constructor, Get, Set ve EnKısaYol metotları yer almaktadır. En kısa yol algoritmasında kullanacağı fazladan bir özelliği yoktur.

MasterYoda (class): Character sınıfı kalıtım olarak ata sınıfıdır. Constructor, Get, Set ve EnKısaYol metotları yer almaktadır. Ek olarak Can sayısını tutmak için can değişkeni vardır. Objesi ilk oluştuğunda 6 can hakkı bulunur.

LukeSkywalker (class) : Character sınıfı kalıtım olarak ata sınıfıdır. Constructor, Get, Set ve EnKısaYol metotları yer almaktadır. Ek olarak Can sayısını tutmak için can değişkeni vardır. Objesi ilk oluştuğunda 3 can hakkı bulunur.

3.3 Grafik İçin Kullanılan Sınıflar :

public class Graphic extends JPanel implements KeyListener, ActionListener : Bu sınıfta iyi karakterlerin hareketi grafik olarak gösterileceği için KeyListener ve ActionListener sınıflarını kullandık.

4. Karşılaşılan Sorunlar ve Çözümleri

4.1. Dinamik Arayüz Kullanımı : Arayüz üzerinde devamlı hareket olacağı için dinamik bir arayüz yapmak zorundaydık projeye başlamadan önce java ile yapılmış büyük ölçekli olmayan projeler okuduk. Dinamikliği sağlamak için repaint methodunu kullandık.

4.2. En Kısa Yolu Bulma Algoritması : Daha öncesinde de görmüş olduğumuz Dijkstra Algoritması, Bellman-Ford Algoritması gibi en kısa yolu bulma algoritmaları vardı fakat bizim üzerinde çalıştığımız ikilik sayılardan oluşan labirent için koda nasıl uygulanacağını bilmiyorduk. Arama motorunda “Shortest Path in a Binary Maze” olarak aratınca Lee Algoritması’na ulaştık. Projemizdeki En Kısa Yol Algoritması, Lee algoritması tipinden yazılmıştır. Eklemeler ve bazı kısımların çıkarılması ile kodumuza uygun entegre edilmiştir (<https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/>).

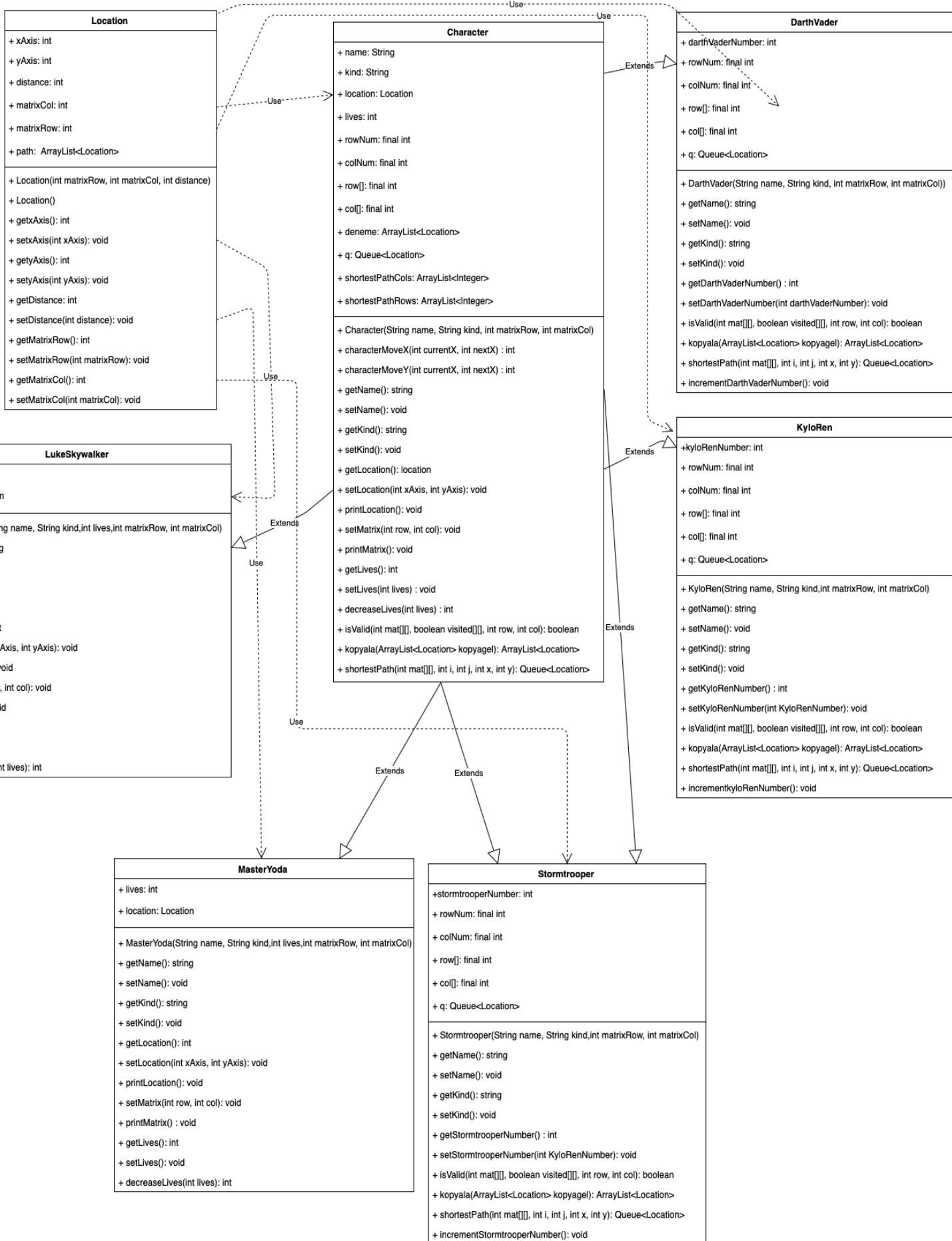
5. Kazanımlar

5.1. Swing Kütüphanesi bazı methodları

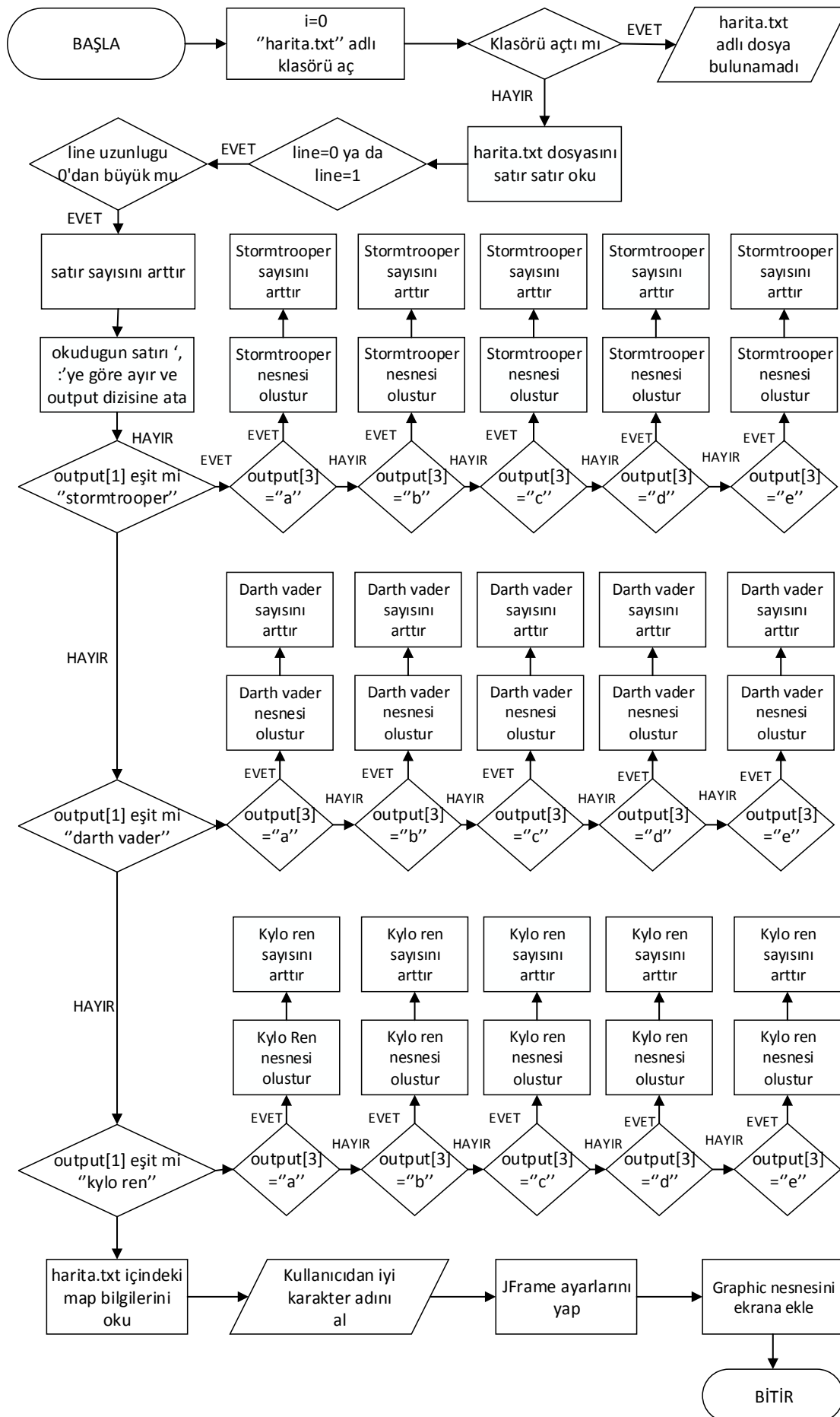
5.2. Git sürümleri - GitHub Kullanımı

5.3. Dijkstra Algoritmasının çalışma mantığı

6. UML Diyagramı



7. Akış Şeması



8- Ekran Çıktıları

```
melike2.1.0 - NetBeans IDE 8.2
Output - melike2.1.0 (run)
Master Yoda karakteri 6 can hakkına sahiptir. MASTER YODA ICIN 4 HAFINE BASIN !
Luke Skywalker karakteri 3 can hakkına sahiptir. LUKE SKYWALKER ICIN 5 HAFINE BASIN !

KARAKTERINI SEC VE BASLA !
SECIM:
#
-----9 4
-----9 3
-----8 3
-----7 3
-----7 4
-----6 4
-----5 4
-----5 5
-----5 6

The shortest path from source to destination has length 9

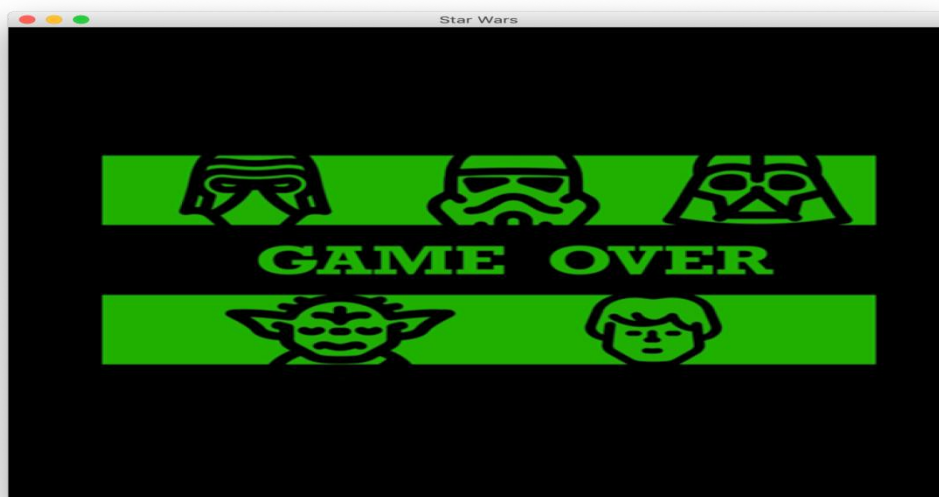
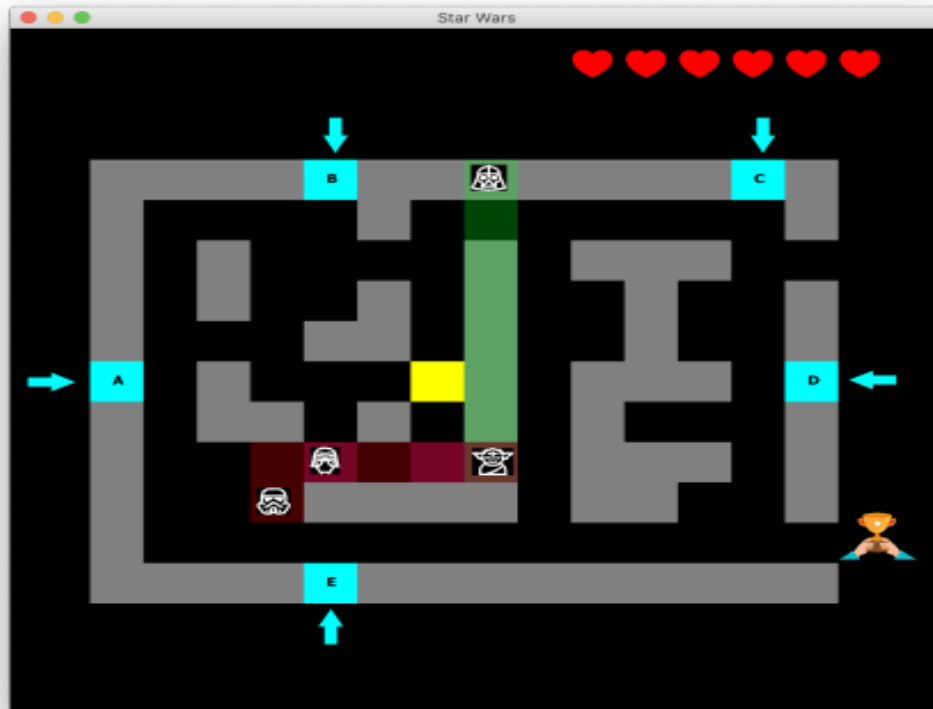
stormtrooperdayiz
-----0 5
-----0 6
-----1 6
-----2 6
-----3 6
-----4 6
-----5 6

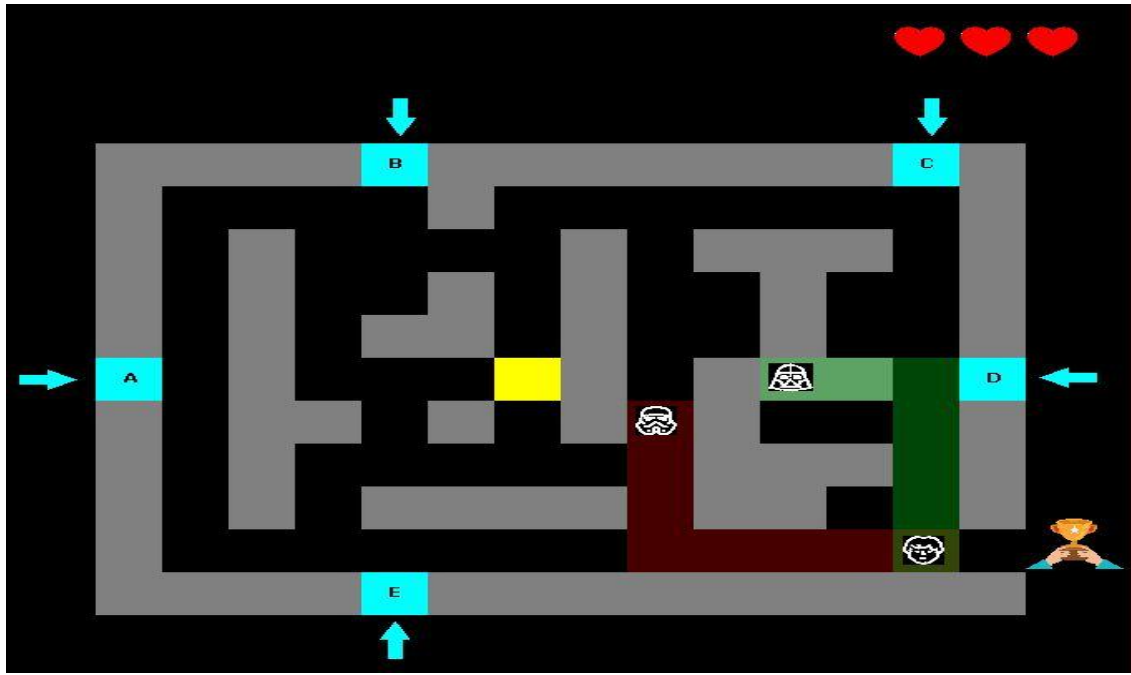
The shortest path from source to destination has length 7

dart vaderdayiz
-----5 1
-----4 1
-----4 2
-----4 3
-----5 3
-----5 4
-----5 5
-----5 6

The shortest path from source to destination has length 8

kylorundayiz
```





9.Kaynakça

[1] **En Kısa Yol Algoritması** : <https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/>

[2] **KeyListener** : <https://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyListener.html>

