Star Wars

\*Ceren Genç, \*Berkehan Öztürk

\*Bilgisayar Mühendisliği Bölümü, Kocaeli Üniversitesi

Kocaeli, Türkiye

[cerengenc17@gmail.com](mailto:cerengenc17@gmail.com)

[berkehanozturk98@gmail.com](mailto:berkehanozturk98@gmail.com)

**Anahtar Kelimeler – “Karakter”, ”Lokasyon”, “Master Yoda”, “Luke Skywalker”, “Kötü Karakter”**

Özet— Programımız “harita.txt” dosyası içinden oynayacak kötü karakterleri, oyun haritasını okur ve kullanıcının istediği iyi karakterin seçilmesi ile tüm karakterler yerlerine konulur. Kullanıcı yön tuşları ile iyi karakteri hareket ettirerek kötü karakterlere yakalanmadan labirentten çıkmaya çalışır.

I.GİRİŞ

Oyunumuzda 5 adet karakter bulunmaktadır. Karakterlerden iki tanesi iyi karakter olup kullanıcının oyuncu olarak seçebileceği karakterlerdir. İyi karakterlerin görevi kötü karakterlere yakalanmadan labirentten kaçmaktır. Master Yoda ve Luke Skywalker oyun içindeki iyi karakterlerdir. İyi karakterler sadece yollardan gidebilmektedir. Her iki karakterinde 3 adet canı bulunmaktadır. Kötü karakterlere yakalandıkları durumlarda Luke Skywalker’ın bir adet, Master Yoda’nın ise yarım canı gitmektedir. Duvara çarpılması durumunda hareket etmemektedirler. Kötü karakterler “harita.txt” dosyası içinde seçilir. Oyuna giriş yapacakları kapılar txt dosyası içinde yer almaktadır. Görevleri iyi karakteri yakalamaktır. Kötü karakterlerden ilki Kylo Ren olup özelliği çok hızlı olmasıdır. Bir harekette iki kare ilerlemektedir. Bir diğer kötü karakter Stormtrooper olup bir özelliği bulunmamaktadır. Son kötü karakterimiz ise DarthVader olup özelliği duvarları yıkabilmesidir. Oyun içinde duvarlardan geçebilen tek karakterdir. Programımız kullanıcıdan iyi karakterlerden hangisi ile oynamak istediğinin sorulması ile başlar. Şekil sınıfının içinde “harita.txt” dosyasından okuma işlemi gerçekleştirilir. Oyunda yer alacak tüm karakterler yerlerine yerleştirilir. Harita ekranı açılır. Kullanıcıdan yön okları ile hareket etmesi istenir. Kullanıcının her hareketi ile birlikte kötü karakterler oyuncuya en kısa yoldan ulaşmaya çalışır. Oyuncunun her hareketinde lokasyonu yeni değerlerine güncellenir, kötü karakterlerin oyuncuya olan uzaklıkları yeniden hesaplanır ve yeni yola göre hareket ettirilirler. Oyuncu ile kötü karakterlerden herhangi birinin lokasyonlarının çakışması durumunda oyuncunun canı azaltılır ve can bitmemiş ise oyun başa alınır. Oyuncunun canı bittiğinde ise oyun sonlanır. Oyuncunun labirentten kaçmayı başarması durumunda oyun biter. Oyun esnasında arka planda müzik çalmaktadır. Oyun sonunda kazanma ekranı ve oyunu kaybetme ekranları farklıdır.

II. TEMEL BİLGİLER

Programımız Java dilinde yazılmıştır. Geliştirme ortamı olarak “NetBeans 8.2” kullanılmıştır. Windows 10 ve Windows 7 işletim sistemlerinde sorunsuz olarak çalıştırılmıştır.

III. MİMARİ

Her karakter için özel sınıf oluşturulmuştur. Bu sınıfların hepsi oluşturulmuş Karakter sınıfından extends edilmiştir. Karakter sınıfının yapılandırıcısını kullanılıp, fonksiyonları override edilmiştir. Karakter sınıfının içinde karakterin ismi, iyi veya kötü karakter olduğu bilgisi, oyun içindeki koordinatları tutulmuştur. İyi karakterlerin kendi sınıfları içinde karakter sınıfının içindekilere ek olarak can değişkeni tutulmuştur. Şekil sınıfı JPanel sınıfını extends edip, KeyListener ara yüzünü implement etmiştir. Grafik çizimlerini gerçekleştirmek için paintComponent fonksiyonunu override etmiştir. KeyListener ara yüzünün fonksiyonları oyuncunun hareketlerini yön tuşları ile gerçekleştirebilmek için şekil sınıfı içinde override edilmiştir. Kötü karakter bilgilerinin tutulması ve en kısa yolların nereden geçtiğini tutmak için ArrayList yapıları kullanılmıştır. İyi karakter için oyuncu1 ve oyuncu2 olmak üzere iki ArrayList tutulmuştur. Oyuncu1 Master Yoda için Oyuncu2 ise Luke Skywalker için kullanılmıştır. Karakter resimlerinin ekranda gösterilebilmesi için İmage ve İmageIcon’lar tanımlanmıştır. Lokasyon sınıfı açılmıştır. En kısa yol bulma işlemini gerçekleştirmek için node sınıfı oluşturulmuştur.

IV.KULLANILAN FONKSİYONLAR

1. *Void Setx():* Oyuncunun ok yönleri ile hareketinden sonra x lokasyonunu yeni haline günceller. Kötü karakterlerin en kısa yol algoritması için açılan node sınıfını çağırır. Kötü karakterler ile oyuncunun lokasyonlarının çakışıp çakışmadığını ve kupaya ulaşılma durumlarını kontrol eder.
2. *Void Sety():* Oyuncunun ok yönleri ile hareketinden sonra y lokasyonunu yeni haline günceller. Kötü karakterlerin en kısa yol algoritması için açılan node sınıfını çağırır. Kötü karakterler ile oyuncunun lokasyonlarının çakışıp çakışmadığını ve kupaya ulaşılma durumlarını kontrol eder.
3. *Boolean end():* Setx ve Sety fonksiyonları içinden çağırılır. Oyunun kazanma durumunu kontrol eder.
4. *Void PaintComponent():* Override edilmiştir. Grafik çizimlerini gerçekleştirir.
5. İnt MatrisOku(): “harita.txt” dosyası içinden harita bilgilerini okur.
6. *Void Music():* Oyun esnasında çalan müziği ekler.
7. *ArrayList karakterOku():* “harita.txt” dosyası içinden kötü karakteri ve oyuna dahil olacakları kapıları okur.
8. *Void KeyPressed():* Override edilmiştir. Oyuncunun yön okları ile hareket edebilmesini sağlar.
9. *Void kotuSet():* Kötü karakterlerin lokasyonlarını günceller.
10. *ArrayList enKısaYol():* Kötü karakterlerin oyuncuya ulaşabileceği en kısa yolu bulur.
11. *ArrayList enKısaYolVader****():*** DarthVader karakteri duvarlardan geçebildiği için her yer yolmuş gibi düşünülerek bu karakter için özel en kısa yol hesaplanır.
12. *Void setLokasyonX():* Karakterin x lokasyonunu günceller.
13. *Void setLokasyonY():* Karakterin y lokasyonunu günceller.
14. *Void setCan():* İyi karakterin can değerini azaltır.
15. *Void setTur():* Karakterin türünü günceller.
16. *Void setName():* Karakterin adını günceller.
17. *String getName():* Karakterin adını döndürür.
18. *İnt getLokasyonx():* Karakterin x lokasyonunu döndürür.
19. *İnt getLokasyonx():* Karakterin y lokasyonunu döndürür.
20. *İnt getCan():* İyi karakterlerin can değerlerini döndürür.
21. *String getTur():* Karakterin türünü döndürür.

V. SONUÇ

Tüm isterler gerçekleştirilmiştir. İstenilen sınıflar oluşturulmuş getter ve setter fonksiyonları eklenmiştir. Her karakter için en kısa yol bulunmuş ve oyun ekranında gösterilmiştir. Oyuncu yön okları ile hareket ettirilmiştir. Oyuncunun canları gerektiği gibi azaltılmış, kullanıcıya gösterilmiştir. DarthVader karakteri duvarlardan hareket ettirilmiştir. Kylo Ren karakteri iki birim hareket ettirilmiştir.

VI. KABA KOD

1. Başla
2. JFrame tanımla
3. Oynamak istediğiniz karakterin adı?
4. Şekil sınıfını çağır
5. karakterOku() fonksiyonunu çağır
6. Oyuncuya olan uzaklıkları hesapla
7. matrisOku() fonksiyonunu çağır
8. Oyun ekranını çiz
9. Karakterleri yerine çiz
10. İf(Sağ yön oku || Sol yön oku)
11. Harita dışına çıkıyor mu?
12. SetX() fonksiyonunu çağır
13. Oyuncu lokasyonlarını güncelle
14. Yeni lokasyonu kullanarak en kısa yol bul
15. İf(Yukarı yön oku || Aşağı yön oku)
16. Harita dışına çıkıyor mu?
17. SetY() fonksiyonunu çağır
18. Oyuncu lokasyonlarını güncelle
19. Yeni lokasyonu kullanarak en kısa yol bul
20. En kısa yolları göster
21. Win==True
22. Oyun sonu resmini çiz
23. Lose==True
24. Oyunu sıfırla
25. Oyuncu canı==0
26. Game Over resmi çiz

VII.KAYNAKÇA

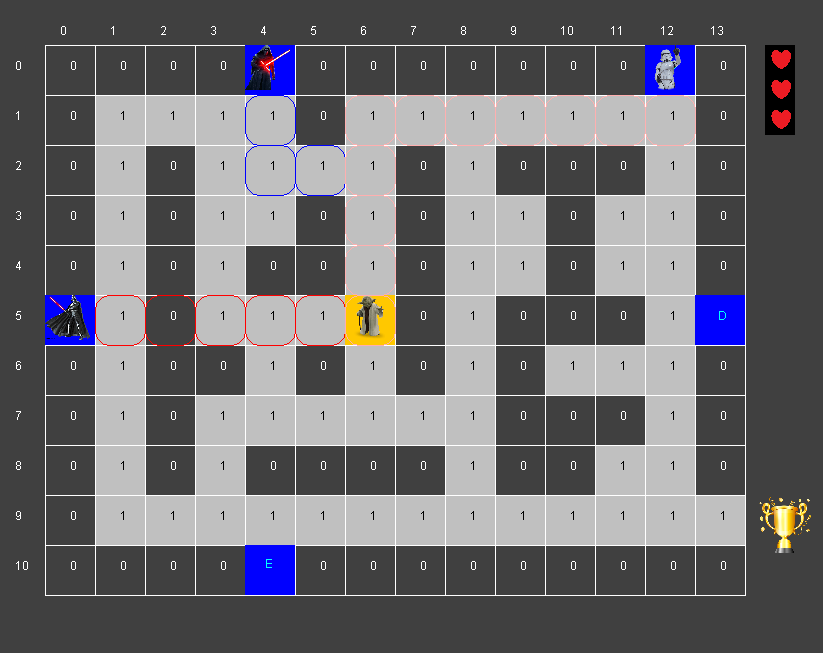
htpps://youtu.be/3q4f6I5zi2w

https://frmtr.com/java/4524819-java-resim-ekleme-yardim-.html

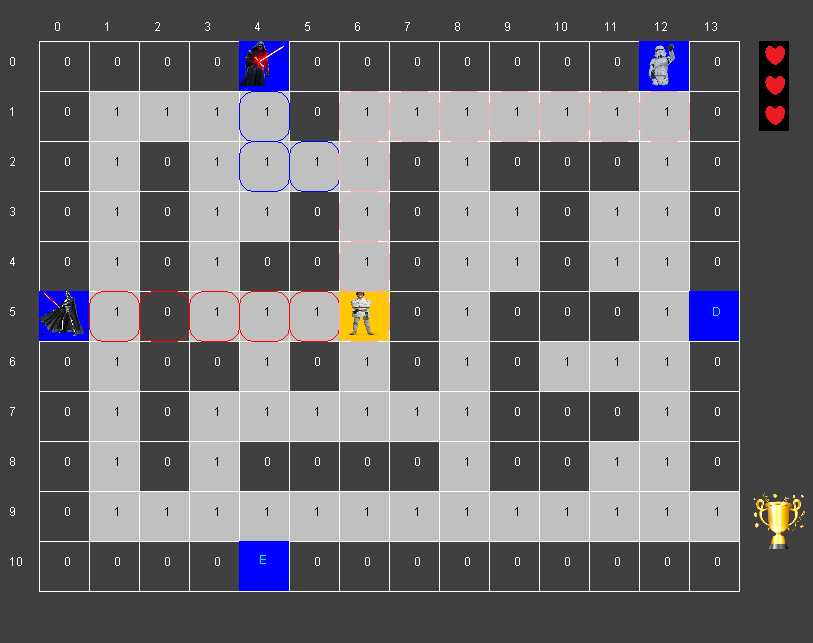
VIII.EKRAN GÖRÜNTÜLERİ



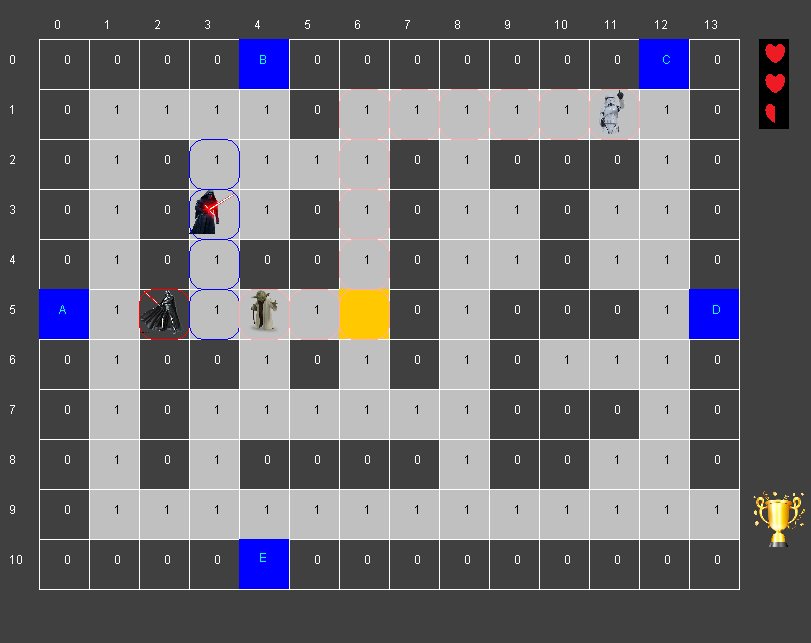
Resim 1-Karakter Seçim Ekranı



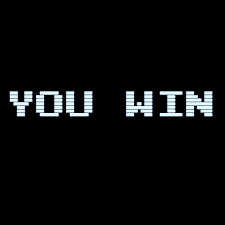
Resim 2- Oyun başlangıç ekranı(Master Yoda)



Resim 3- Oyun başlangıç ekranı(Luke Skywalker)



Resim 2- Canın azalması ve oyun hareket anı



Resim 3-Oyun sonu ekranı Resim 4-Oyun sonu ekranı