Programlama Lab. II, 2020-2021 Dönemi Proje 1: Şirinler Labirent Oyunu

Kubilay Kaplan 180201123

kubilay.kaplan@yahoo.com

Bilgisayar Mühendisliği Bölümü Kocaeli Üniversitesi

Özet: Bu makalede; nesneye yönelik programlama yapısı kullanılarak ve çeşitli veri yapıları disiplinlerinden faydalanılarak Java dilinde geliştirilen, labirent oyunu timeline dayanan bir uygulamanın geliştirme süreci incelenmiştir.

I. GİRİŞ

Geliştirilen uygulama; kullanıcı tarafından kontrol edilen iyi karakterin, bilgisayar tarafından kontrol edilen kötü karakter veya karakterlerden kaçarak, labirent içerisindeki hedefe ulaşması amacına dayanmaktadır. Uygulama; nesneye yönelik programlama yapı ve ilkeleri doğrultusunda, çeşitli veri yapıları disiplinlerinden faydalanılarak Java dili ile geliştirilmiştir.

Labirent içerisinde hedefe doğru giderken, iyi karakteri birçok zorluk beklemektedir. Kullanıcı, oyun başlamadan önce konsol üzerinden seçtiği iyi karakterlerden (Gözlüklü Şirin / Tembel Şirin) birini, klavyedeki yön tuşları vasıtasıyla etmekte olup, hedefe (Şirine) calısmaktadır. Kullanıcının kontrol ettiği iyi karakteri, Harita.txt dosyasında belirtilen kötü karakterlerden (Azman / Gargamel) biri veya her ikisi durdurmaya çalışmaktadır. Kullanıcının kontrol edebileceği Gözlüklü Şirin ve bilgisayarın kontrolünde olan Gargamel, tek hamlede 2 birim ilerleyebilirken; diğer iki karakter ise tek hamlede 1 birim ilerleyebilmektedir. Herhangi bir karakter, labirent içerisindeki duvarlardan atlayamamaktadır. 20 puan ile oyuna başlayan kullanıcı; Azman'a yakalanması halinde 5, Gargamel'e yakalanması halinde 15 puan kaybetmektedir. Kullanıcı, puanı sıfırlanmadan Şirine'ye ulaşması halinde oyunu kazanacak, aksi halde kaybedecektir.

II. PROBLEM ÇÖZÜM VE UYGULAMA

A. Yararlanılan Teknolojiler

Uygulama; Java programlama dili kullanılarak, Apache NetBeans geliştirme ortamında oluşturulmuştur.

B. Probleme Çözüm Yaklaşımı

Oyunda bulunan 5 karakter ve kullanıcının oyunu kaybetmesi halinde gözükecek ekran için görseller hazırlandı. Tembel Şirin ve Gözlüklü Şirin kullanıcının control edebileceği iyi karakterler, Azman ve Gargamel de bilgisayarın kontrolünde olan kötü karakterlerdir. Şirine karakteri ise; kullanıcının ulaşmaya çalışacağı hedef olup; görsel haricinde işlevsizdir ve herhangi bir kod altyapısı bulunmamaktadır.

Oyun başlamadan önce, kullanıcı konsol vasıtasıyla kontrol edeceği iyi karakteri seçmektedir. Labirent haritası ve kötü karakter bilgileri ise Harita.txt dosyasından çekilir.

Oyundaki işlevsel 4 karakterin de miras alacağı ana Abstract Karakter sınıfı bulunmaktadır. Karakter sınıfında ad, tür, ID, lokasyon özellikleri ile kısaYol metodu tanımlanmıştır.

Oyuncu ve Dusman sınıfları, abstract Karakter sınıfından kalıtım alır. Karakter sınıfında tanımlanan özellikler, bu her iki sınıfta da super metoduyla çağrılmaktadır. Farklı olarak; Dusman sınıfında kapı, Oyuncu sınıfında ise skor özellikleri tanımlanmıştır. GozlukluSirin ve TembelSirin classları Oyuncu sınıfından, Azman ve Gargamel sınıfları ise Dusman sınıfından kalıtım almaktadır. Gene super metodu kullanılarak, karakter sınıfındaki özellikler, her dört sınıfta da tekrar çağrılmaktadır.

Abstract Karakter sınıfında tutulan lokasyon bilgisi, esas olarak ayrı bir Lokasyon classı oluşturularak tanımlanmıştır. EnKisaYol sınıfında; kötü karakter veya karakterlerin, kullanıcının kontrolündeki iyi karakteri en kısa yoldan takip edeceği algoritma tutulmuştur. Harita sınıfında dosya okuma, harita-grafik işlemleri ve oyun akışı gerçekleştirilirken; Sirinler sınıfı ise oyun ekranının açılmasında kullanılmaktadır.

C. Kod Yapısı

Uygulamada kalıtımsal hiyerarşide "1 => 2 => 4" şeklinde 6; toplamda ise 13 class bulunmaktadır. Okunma kolaylığı amacıyla bırakılan boş satırlar ve yorum satırları çıkartılmadan, uygulama toplam 967 satır koddan oluşmaktadır.

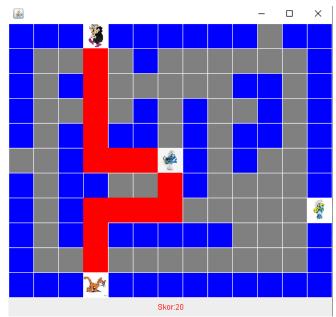
Grafik arayüzü için Swing ve Graphics kütüphanelerinden yararlanılmıştır.

Nesneye yönelik programlama (OOP) yapı ve felsefesine uygun olarak uygun olarak; Class, Inheritance, Abstraction, Encapsulation, Polymorphism yapıları, Constructor metotlar ile get, set metotları kullanılmıştır.

D. Oyunun Çalıştırılması

Uygulama açıldığında ilk olarak, konsol ekranında kullanıcıdan kontrol etmek istediği karakteri seçmesi beklenmektedir. Kullanıcının seçiminin akabinde, oyun ekranı açılmaktadır. Labirenti şekillendiren harita ve kötü karakter veya karakterlere ait bilgiler, Harita.txt dosyasından

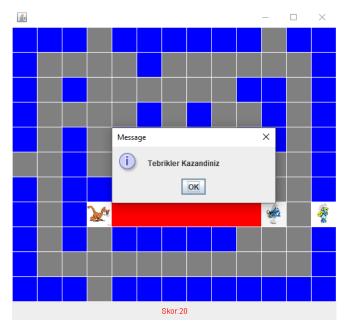
okunmaktadır. Buna göre Azman ve Gargamel kötü karakterlerinden biri veya her ikisi; A(0,3), B(0,10), C(5,0), D(10,3) kapılarından oyuna giriş yapabilir. Oyun başlangıcında; kullanıcının kontrolündeki iyi karakterin konumu (5,6) ve ulaşılmak istenen hedef Şirine'nin konumu (7,12) olmakla beraber, bu iki başlangıç konumu her oyunda sabittir.



Verilen örnek oyun içi görüntüsünde; Gargamel'in A kapısından (0,3), Azman'ın ise D kapısından (10,3) giriş yaptığı bir oyun başlangıcı görülmektedir.

Gri hücreler, karakterlerin hareket edebileceği labirent yolları olup, mavi renkli hücreler ise labirent içerisinde karakterlerin gezinemeyeceği ve üstünden de atlayamayacağı engelleri/duvarları belirtmektedir. Kırmızıya boyalı hücrelerin oluşturduğu şeritler ise; kötü karakterlerin, iyi karaktere olan en kısa mesafesini göstermektedir. Bilgisayar kontrolündeki kötü karakter veya karakterler; iyi karakteri takip ederken bu kırmızı şeritler halinde görülebilmekte olan yolları izleyeceklerdir.

Kullanıcı klavyede bulunan yön tuşları vasıtasıyla, seçtiği iyi karakteri kontrol etmektedir. Gözlüklü Şirin ve Gargamel karakterleri tek hamlede 2 birim ilerlerken, Tembel Sirin ve karakterleri ise tek hamlede ilerlevebilmektedir. Hicbir karakter labirentteki mavi renk ile temsil edilen engeller üzerinden yürüyemez veya atlayamaz. 20 puan ile oyuna başlayan kullanıcı; Gargamel'e yakalandığında 15, Azman'a yakalandığında ise 5 puan kaybetmektedir. Kullanıcının kötü karaktere yakalanmasından sonra, skoru azalır ve karakterler başlangıç pozisyonlarına geri döner. Kullanıcı, puanı sıfırlanmadan Sirine'ye varabilirse oyunu kazanacaktır.



Verilen örnek oyun içi görüntüsünde, kullanıcıya oyunu kazandıran son hamleden önceki pozisyonlar görülmektedir. Hamlenin akabinde ise oyun kazanılmakta ve kullanıcıya bu durumu bildiren diyalog kutusu çıkmaktadır.



Kullanıcının puanının sıfırlanması halinde, oyunun sona erdiğini gösteren "Game Over" ekranı gözükecektir.

E. En Kısa Yol Algoritması

En kısa yol hesaplaması için "Lee Algoritması" kullanılmıştır.

F. Akış Şeması

Ek-1'de bulunmaktadır.

G. UML Sınıf Diyagramı

Ek-2'de bulunmaktadır.

III. SONUÇ VE DEĞERLENDIRME

Geliştirdiğimiz bu uygulama ile, Object Oriented Programming (OOP) ilkelerini daha iyi idrak ettiğimizi ve pratik kazandığımızı; en kısa yol algoritmalarını araştırmak ve deneyimlemek suretiyle veri yapılarını artık daha iyi kullanabildiğimizi düşünüyoruz. Geliştirdiğimiz uygulama; "Harita.txt" dosyasında bulunan harita ve tekli veya çoklu kötü karakter bilgilerini okuyarak labirenti doğru bir şekilde oluşturabilmekte, bilgisayar ve kullanıcı kontrolündeki karakterlerin labirent içerisindeki hareketlerini doğru bir şekilde sağlayabilmekte, Lee algoritması ile kötü karakter veya karakterlerin iyi karaktere olan en kısa mesafesini hesaplayabilmekte, bu yolu çizdirebilmekte ve olumlu/olumsuz olarak oyun sonunu getirmektedir. Altın ve mantar isterleri ise, projede gerçekleştirilememiştir.

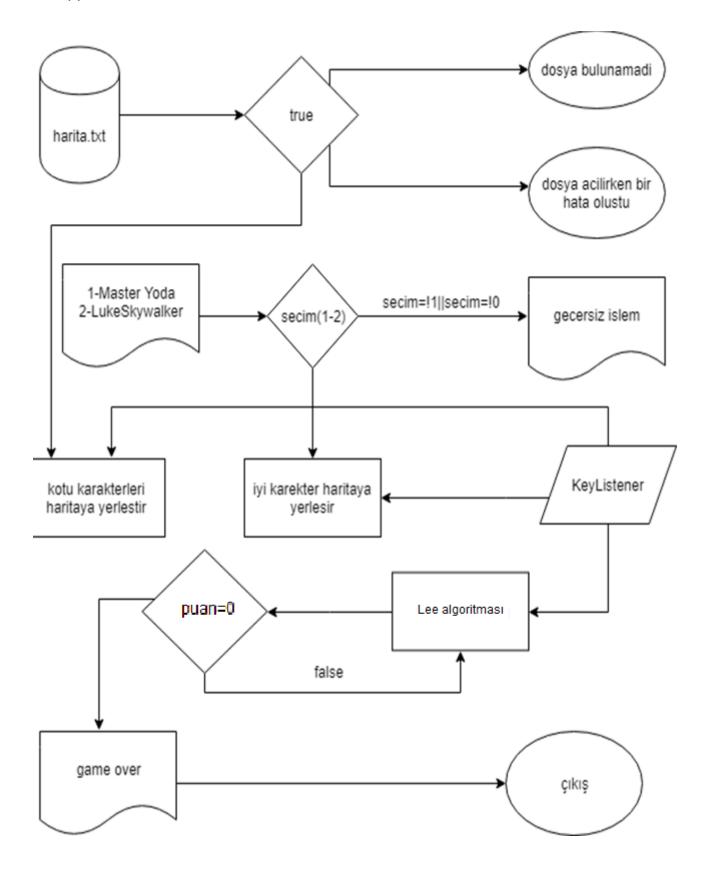
IV. KAYNAKÇA

[1]

 $\underline{https://www.techiedelight.com/lee-algorithm-shortest-path-} \underline{in-a-maze/}$

[2]

https://stackoverflow.com/questions/16366448/maze-solving-with-breadth-first-search



Ek-2 UML Sınıf Diyagramı

