

**Yıldız Teknik Üniversitesi**

**Elektrik Elektronik Fakültesi**

**Bilgisayar Mühendisliği**

BLM1022

YAPISAL PROGRAMLAMAYA GİRİŞ

GR:1

MEHMET FATİH AMASYALI

DÖNEM PROJESİ

İsim: Muhammed Taha Güneş

No: 21011017

E.posta: [taha.gunes@std.yildiz.edu.tr](mailto:taha.gunes@std.yildiz.edu.tr)

İçindekiler

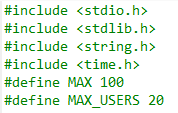
Sayfa 3’te kod açıklamaları yer almaktadır.

Sayfa 8’de çıktı açıklamaları yer almaktadır.

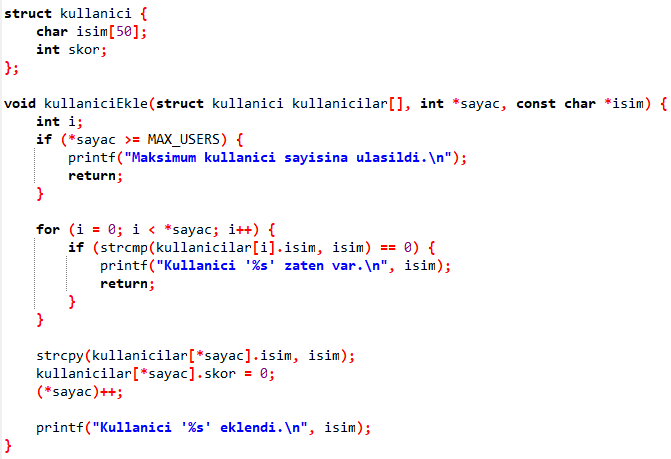
Sayfa 12 ve 13’de program karmaşıklığı yer almaktadır.

Kod Açıklamaları

Klasik kullanacağımız kütüphaneleri ve tanımlamalarımızı başa yazdık.



Kullanıcı isimlerini tutmak için structure yapısı tanımladık ve altındaki fonksiyonla kullanıcı alıp olanları da kontrol ettik. Oyuncu sayısı sınırı da koyduk.



Skor artır fonksiyonunda kullanıcının skorlarını

**>>** oyunun tamamlanma süresi,

**>>** matrisin boyutu,

**>>** matrisin rastgele/dosyadan oluşturulma şekli

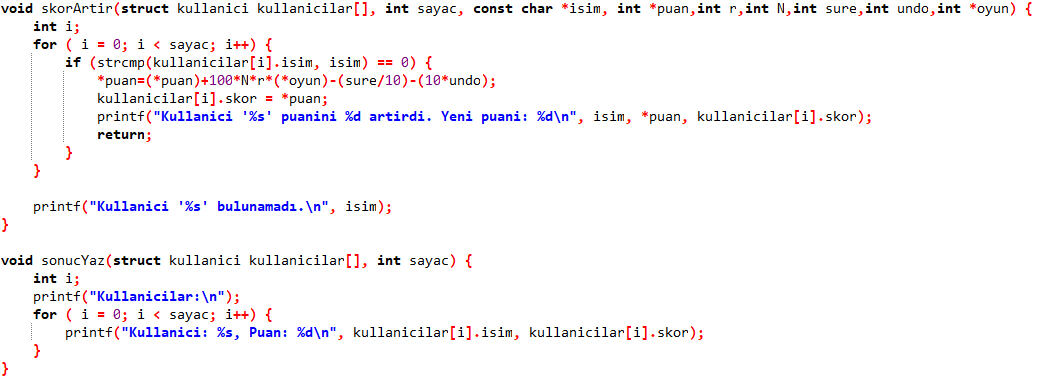
**>>** Yapılan yanlış işlem (UNDO) sayısı

**>>** kullanıcının oynadığı oyun sayısı

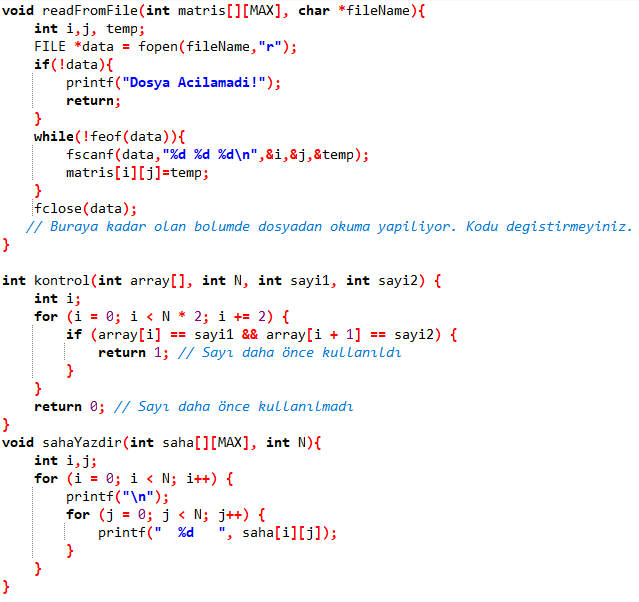
**Gibi parametrelere göre**

**Cozdugunuzmatrisbuyuklugu\*r\*Oynadiginizoyunsayisi\*100-(sure/10)-(10\*gerialmasayisi) puan verdik.**

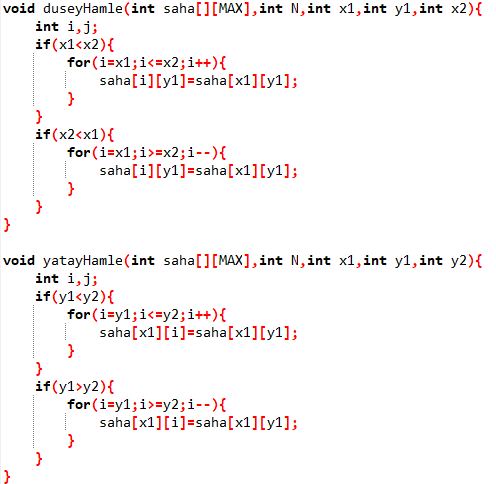
**Alttındaki fonksiyonda ise tüm kullanıcıların puan ve isimlerini yazdırdık.**



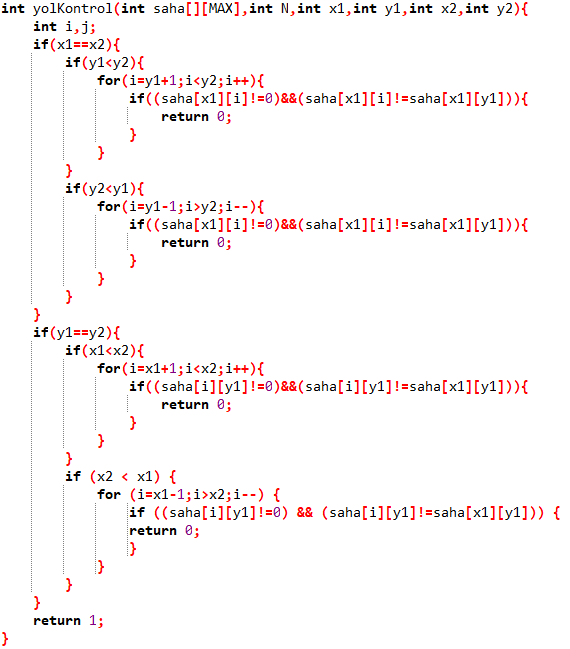
İlk fonksiyonumuz zaten dosya okuma fonksiyonu. Kontrol fonksiyonu ise randomda koordinatların üst üste gelmesini engellemek için koyuldu. Sonraki fonksiyon sahayı yazdırıyor



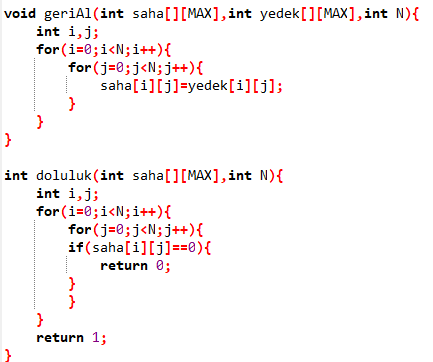
İlk fonksiyonumuz adı üstünde düşey eksende ilerleme sağlıyor. y’ler eşitken x’ler arasında hamleyi sağlıyor. 2.fonksiyon ise yatay eksende hareket sağlıyor. X’ler eşitken y’ler arası hamle yapmaya yarıyor.



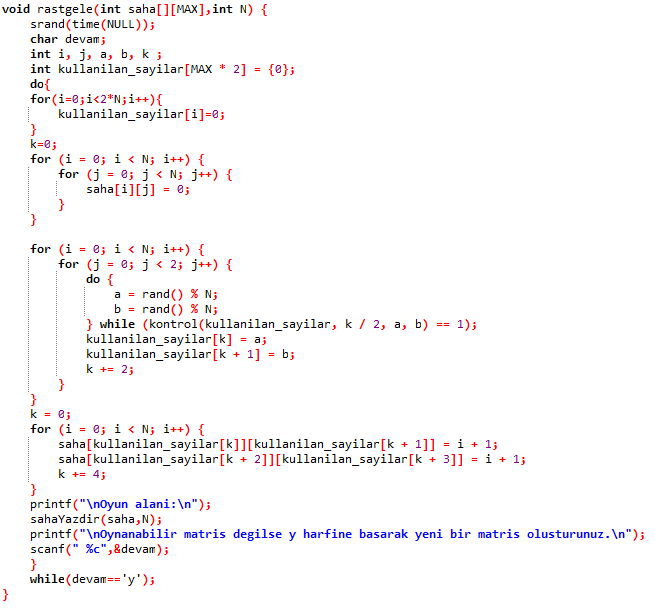
Yolkontrol fonksiyonu baya uzun bir fonksiyon oldu. Girilen koordinatlar arasında eşleştirmek istediğimiz sayıdan başka bir sayı varsa koordinatları yeniden girmemize yarıyor.



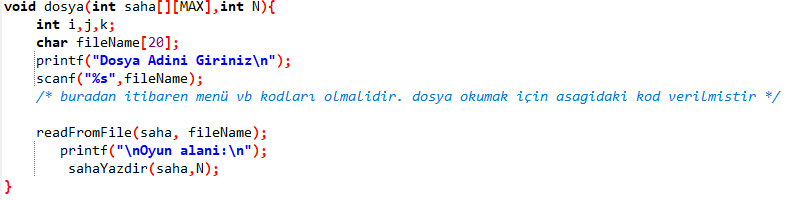
Geri al fonksiyonumuz hamleden hemen önce kopyaladığımı matrisi bize geri getiriyor. Yani hamlenizin önceki anına geçiyorsunuz. Kısacası geri alıyor hamleyi. Doluluk ise matrisin dolup tüm sayıların eşlenip ortada 0 yani boşluk kalıp kalmadığını kontrol ediyor. Bitmişse oyunu bitirmeye yarayacak.



Rastgele fonksiyonumuz rastgele matris üretmeye yarıyor. Sahayı sıfırlayıp random koordinatlar ayarlayarak N tane sayıyı sahaya rastgele diziyor. Kullanılan sayılar dizisi ise randomların üst üste gelmesini engelliyor. y harfine batığınız zaman ise oynanamayacak matrisi yok edip yeniden random matris oluşturuyor.

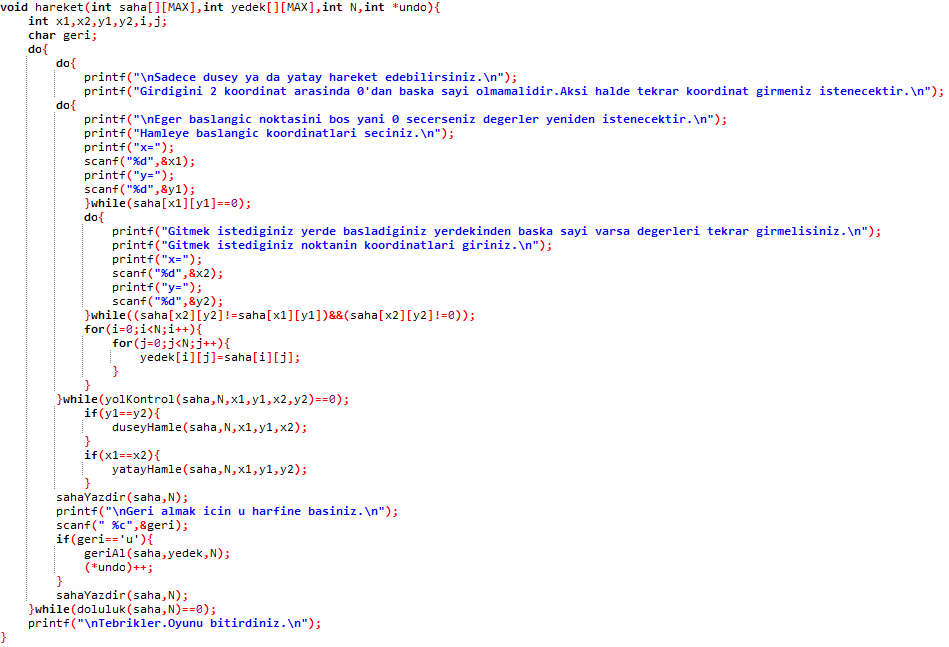


Dışardan verilen dosyanın açılmasını sağlamaktadır.



Hareket fonksiyonumuzda baya uzun bir fonksiyon. İçerisinde birçok başka fonksiyonda barındırıyor. Koordinatları giriyorsunuz ve hareket mümkünse o hareketi gerçekleştiriyor.

İçinde birçok kontrol de var. Girilen koordinat arasında farklı sayı olmamasını kontrol ediyor. Boşluktan başlamanızı engelliyor. Başlangıçta girdiğiniz koordinattaki sayının eşi veya boşluk 2.koordinatta yoksa bir daha girmenizi istiyor. İçerisinde hamle öncesi matrisi kopyalıyor. İsteğe bağlı olarak geri hamle yapmayı mümkün kılıyor. En sonda matris dolmuşsa fonksiyonu bitiriyor



Main fonksiyonumuz içerisinde menümüzü oluşturduk. Kuralları söyledik. Zamanı ölçtük.

Gerekli fonkisyonları da kullandık.

metin, ekran görüntüsü, yazı tipi içeren bir resim

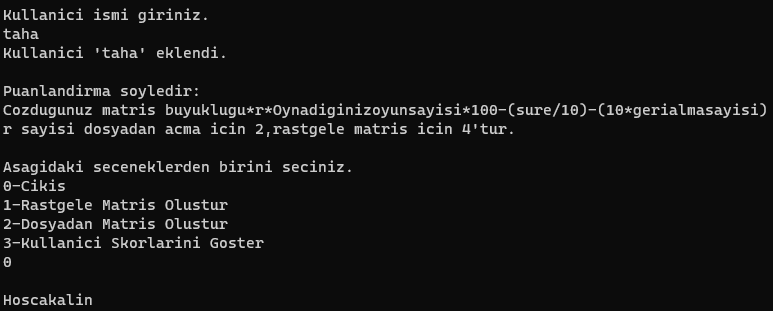
Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

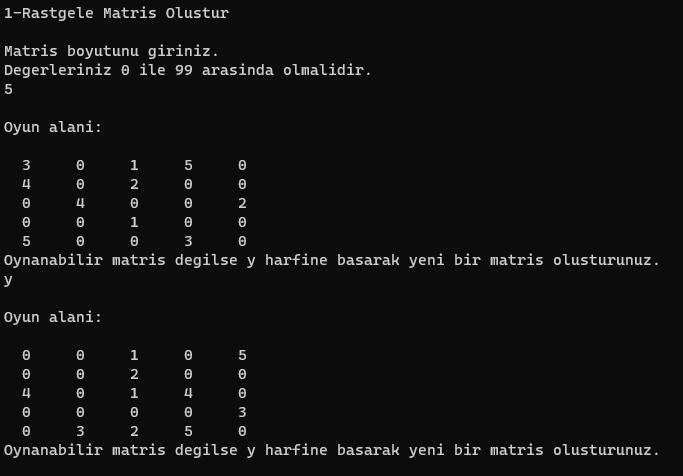
Açıklama otomatik olarak oluşturuldu

Çıktı Açıklamaları

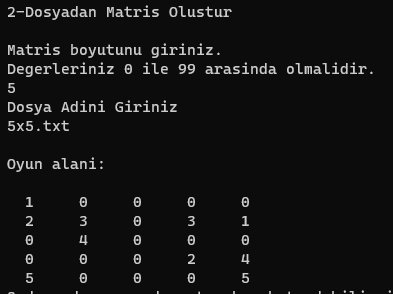
İlk önce programın açılışına bakalım. İlk olarak kullanıcı ismi istiyor. Puanlandırma bilgisi veriyor. Sonra menümüzden bir seçenek seçmemizi istiyor. 0 basarsanız programdan çıkıyor.



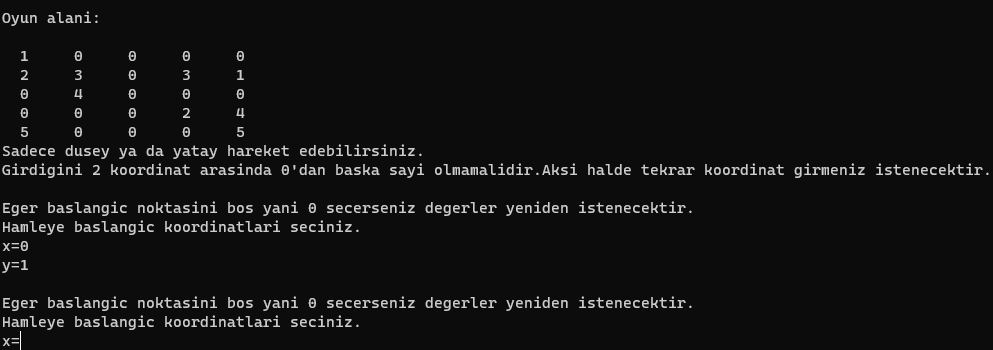
Rastgele seçeneğini seçerseniz size rastgele oyun alanı getirmektedir. Bu matrisin oynanabileceğine kullanıcı karar verecektir. Eğer oynanmayacak gibiyse y harfi ile yeni matris üretebilir. Devam etmek istiyorsa y harici herhangi bir tuşa basabilir.



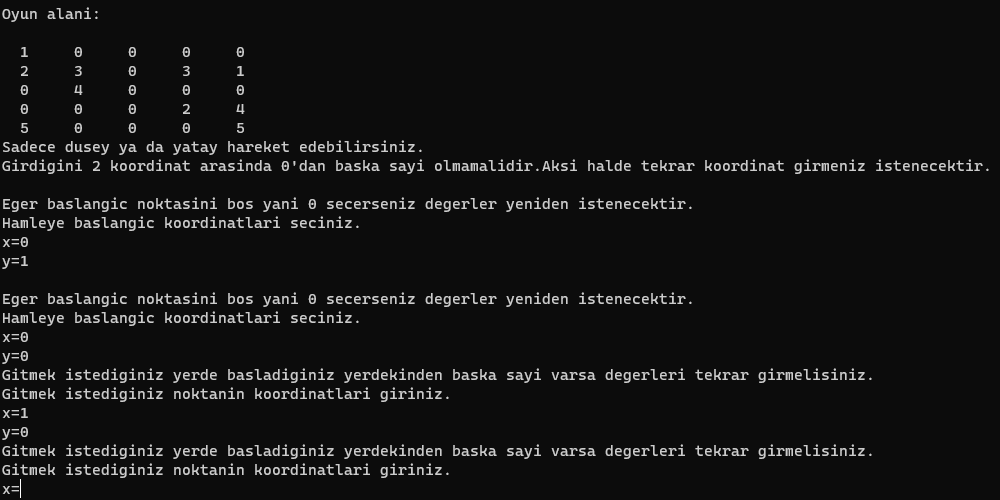
Dosyadan matris oluşturmada boyut bilgisi isteniyor. Sonrasında ise dosya ismi uzantısı ile girince dosyayı açar.



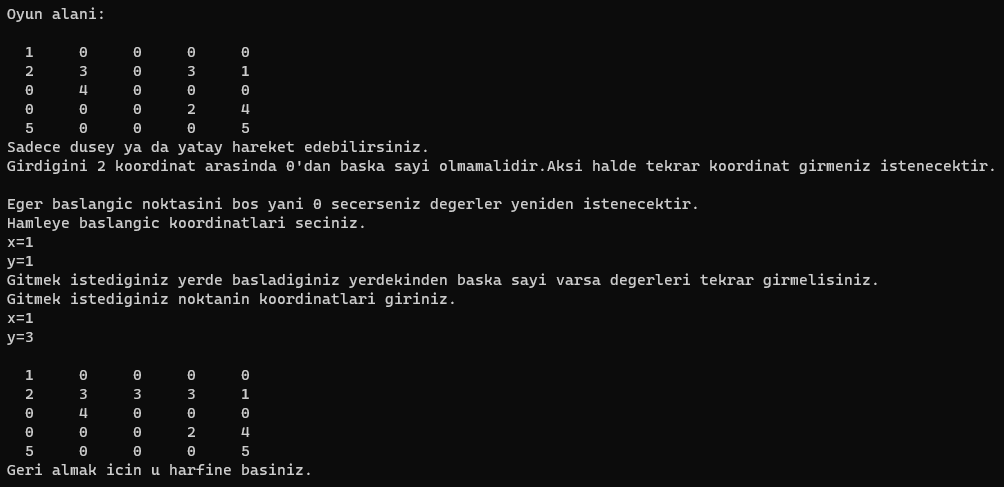
Gördüğünüz gibi başlangıç noktası boş yani 0 olan bir yer seçince tekrardan başlangıç koordinatı girmemizi istiyor.



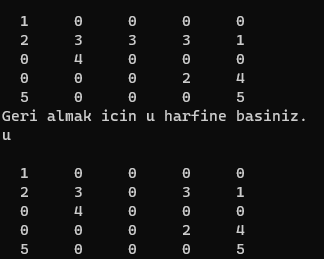
Gittiğiniz yerin koordinatındaki sayı 0 ya da eşleştirdiğimiz sayı değilse tekrardan gidiş koordinatı istiyor.



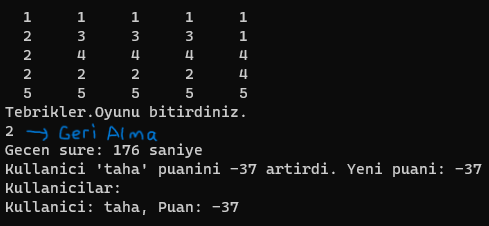
Her şey uygunsa sayıları eşleştiriyor. Hamle yapılıyor.



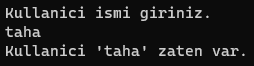
Hamlenizi beğenmediyseniz u harfi ile geri alabilirsiniz. Ama bunun size – puan yanşayacağını unutmayınız.



Tüm tablo dolunca, eşleştirmeler bitince istatistikler ve tebrikler mesajı yazmaktadır.



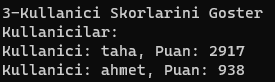
Oyun bittikten sonra tekrar kullanıcı istemektedir. Aynı ismi yazarsanız sizi tanır.



Başka isim yazarsanız onu da oyuncular arasına ekler.



Menüden 3.seçeneğe tıklarsanız scoreboard gelecektir.



Algoritma karmaşıklığı ise en fazla iç içe 2 döngü olduğu için bigO(n^2)’dir.

1-kullaniciEkle işlevi: Bu işlev, kullanicilar dizisine bir kullanıcı eklemektedir. Bu işlevin karmaşıklığı O(n) olarak değerlendirilebilir, çünkü mevcut kullanıcıları kontrol etmek ve kullanıcının eklenebilirliğini doğrulamak için döngü kullanılmaktadır.

2-skorArtir işlevi: Bu işlev, bir kullanıcının puanını artırmaktadır. Bu işlevde de kullanicilar dizisi üzerinde döngü kullanıldığı için karmaşıklığı O(n) olarak değerlendirilebilir.

3-sonucYaz işlevi: Bu işlev, kullanicilar dizisindeki kullanıcıları ve puanlarını ekrana yazdırmaktadır. Dizi üzerinde bir döngü kullanıldığı için karmaşıklığı O(n) olarak değerlendirilebilir.

4-readFromFile işlevi: Bu işlev, bir dosyadan matris verilerini okumaktadır. İki döngü kullanarak matrisin tüm elemanlarını okuduğu için karmaşıklığı O(n^2) olarak değerlendirilebilir.

5-kontrol işlevi: Bu işlev, bir dizi içinde belirli bir sayı çiftinin varlığını kontrol etmektedir. Döngü kullanılarak dizinin tamamı kontrol edildiği için karmaşıklığı O(n) olarak değerlendirilebilir.

6-sahaYazdir işlevi: Bu işlev, saha matrisini ekrana yazdırmaktadır. İki döngü kullanarak matrisin tüm elemanlarını yazdığı için karmaşıklığı O(n^2) olarak değerlendirilebilir.

7-duseyHamle işlevi: Bu işlev, oyuncunun bir taşı dikey yönde hareket ettirmesini sağlar. İki döngü kullanarak tahtanın tüm elemanlarını kontrol eder. Bu durumda, karmaşıklık O(n^2) olarak değerlendirilebilir.

8-yatayHamle işlevi: Bu işlev, oyuncunun bir taşı yatay yönde hareket ettirmesini sağlar. İki döngü kullanarak tahtanın tüm elemanlarını kontrol eder. Karmaşıklık O(n^2) olarak değerlendirilebilir.

9-yolKontrol işlevi: Bu işlev, oyuncunun taşının hareket edebilirliğini kontrol eder. İki döngü kullanarak tahtanın tüm elemanlarını kontrol eder. Karmaşıklık O(n^2) olarak değerlendirilebilir.

10-geriAl işlevi: Bu işlev, oyuncunun bir hamlesini geri almasını sağlar. Karmaşıklığı, hareketler dizisini kontrol ettiği için O(m) olarak değerlendirilebilir, burada m hareketlerin sayısını temsil eder.

11-doluluk işlevi: Bu işlev, tahtadaki boş hücreleri kontrol eder. İki döngü kullanarak tahtanın tüm elemanlarını kontrol eder. Karmaşıklık O(n^2) olarak değerlendirilebilir.

12-rastgele işlevi: Bu işlev, rastgele bir hamle seçer. İki döngü kullanarak tahtanın tüm elemanlarını kontrol eder. Karmaşıklık O(n^2) olarak değerlendirilebilir

13-dosya işlevi: Bu işlev, oyun durumunu bir dosyaya kaydeder veya dosyadan yükler. Dosya işlemleri dışında bir döngü içermediği için karmaşıklığı O(1) olarak değerlendirilebilir.

14-hareket işlevi: Bu işlev, kullanıcının bir hamle yapmasını yönetir. İçerdiği işlevlerin karmaşıklıklarına bağlı olarak karmaşıklığı değişebilir. Genel olarak, işlevlerin karmaşıklıklarının toplamı dikkate alınmalıdır

15-main işlevi: Bu işlev, oyunun ana işlevi ve diğer işlevleri çağırır. İşlev çağırma dışında herhangi bir döngü veya tekrarlı işlem içermez. Karmaşıklığı O(1) olarak değerlendirilebilir.