

# Altın Toplama Oyunu

Ertuğrul Demir

Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi  
170201052  
[ertugrulbusiness@gmail.com](mailto:ertugrulbusiness@gmail.com)

## Özet

Proje Java dili kullanılarak gerçekleştirilmiştir. Projenin amacı , verilen değerler doğrultusunda kurallara göre bilgisayar tarafından yönetilen botların bir simülasyon gerçekleştirmesidir. Oyun konsol üzerinden yönetilirken çıktılarını da yazı olarak konsola görüntü olarak ise JFrame vasıtasıyla ekrana basmaktadır.

- Harita, iki boyutlu java dizisi üzerinde tutulmuştur.
- Yapılan hareketler program içerisinde tanımlanmış özel nesneler ile tutulmuştur.

Bütün proje Nesne yönelimli programlamadan yararlanılarak geliştirilmiştir.

## 1.Giriş

Oyunun başlaması için konsola bir oyun ismi girilmelidir. Girilen oyun ismini takiben oyunun varsayılan ayarları istenirse değiştirilebilir. Bu değişikliği konsol üzerinden klavye girdileriyle sağlayabilirsiniz. Oyun ayarları seçildikten sonra ismi oyun kendiliğinden bir pencere içerisinde ekranda gözükecektir. Bu pencereyi konsol üzerinden kontrol ederek oyunu yönetebilirsiniz. Oyun bittiğinde oyun sonuçları konsol üzerinde bir tablo şeklinde yazılacaktır.

## 2.Yöntem

Oyun konsol üzerinden yönetilmek ve oyun içerisindeki hareketleri görsel olarak ekrana basılması için tasarlanmıştır.

- Konsol üzerinden hareketleri alabilmesi için Java'nın java içerisindeki scanner'dan yararlanılmıştır.
- Konsol üzerine yazı yazmak için Java standart output kullanılmıştır.
- Ekrana oyun görsellerini yansıtmak için Java Swing kütüphanesinden faydalanılmıştır.
- Görselleri java JFrame sınıfının paint() methodu override edilerek ekrana verilmiştir.

## 3.YALANCI KOD

### 3.1 Yol bulma algoritması

1. Basla
2. Hedef = max\_row,max\_colmn
3. Haritayı dolaş
  - a. For i =0; i< max\_row ;i++;
    - i. For j=0;j<max\_colmn;j++
4. Eğer altın varsa koordinatlarını al
  - a. If(Harita[row]][colmn]=='1')
  - i. new\_row=row
  - ii. new\_colmn=colmn
5. Uzaklığı hesapla
  - a. Distance\_colmn=abs(location-colmn)
  - b. Distance\_row=(abs(location-row))
  - c. Distance=Distance\_colmn+Distance\_row;
6. kazancı hesapla
  - a. gain=Target\_Gold – Distance\*Cost
7. Kazaç eldeki hedeften yüksekse onu yeni hedef yap
  - a. If(max\_gain<gain)
    - i. max\_gain=gain;
    - ii. target\_colmn=new\_colmn
    - iii. target\_row=new\_row
8. Hedefi döndür

#### 4.YAKLAŞIM

Proje deneme yanılma ve bu tecrübeler üzerinden eldeki ürünü geliştirme stratejisini izlemiştir.

Denemeler sırasında karşılaşılan hatalar ilgili teknolojiyi sunan firmaların dökümantasyonları aracılığıyla çözülmeye çalışılmıştır.

Çözülen ve elde edilen kodlar daha sonrasında gerekli isterleri karşılabildiklerinden emin olunmak amacıyla teste tabi tutulmuşlardır.

Kullanılan kodların eksik tarafları üzerine test senaryoları oluşturularak program sınanmıştır.

#### 5.KAZANIMLAR

- Nesne yönelimli programlama
- Programları test etme
- Test senaryosu oluşturma
- Sınıf oluşturma
- Kalıtım
- Kullanıcı arayüzü için Component kullanımı
- Ekranı görsel çıktı verme

#### 6.EKSİKLİKLER

- Projenin zaman yönetimi
- Kullanıcı arayüzü tasarımı
- JFrame üzerine resim çizerken JFrame'in tamamının kullanılması
- Etketif olmayan Kullanıcı etileşimleri

#### 7.SONUÇ

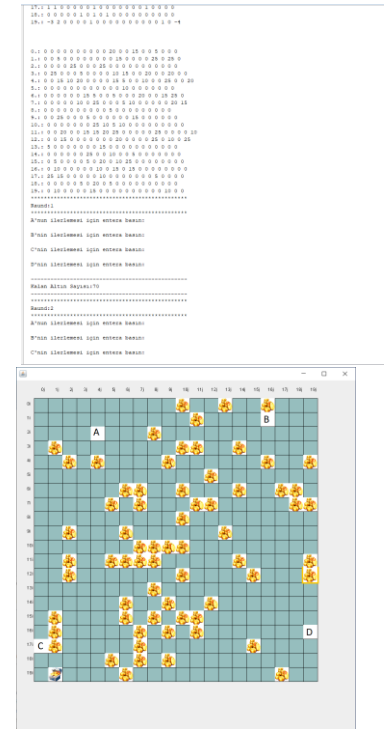
Program konsol üzerinden kontrol edilebilir ve çıktıları konsoldan alınabilir.

Açılan ekran vasıtasıyla oyuncuların yaptıkları hareketler takip edilebilir.

'GAMES' klasörü içerisindeki kayıtlarda tutulan txt dosyaları sayesinde oyuncuların bireysel olarak yaptıkları hareketlere ulaşabilirsiniz.

Projenin eksikleri ise konsol üzerinden idare ediliyor olmasıdır. Tamamen bağımsız kullanıcı arayüzüne sahip değildir.

#### 8.DENEY GÖRSELLERİ



#### 9. Kaynakça

- 1) <https://docs.oracle.com/javase/tutorial/>
- 2) <https://stackoverflow.com/>
- 3) <https://www.tutorialspoint.com/index.htm>
- 4) <https://www.w3schools.com/>