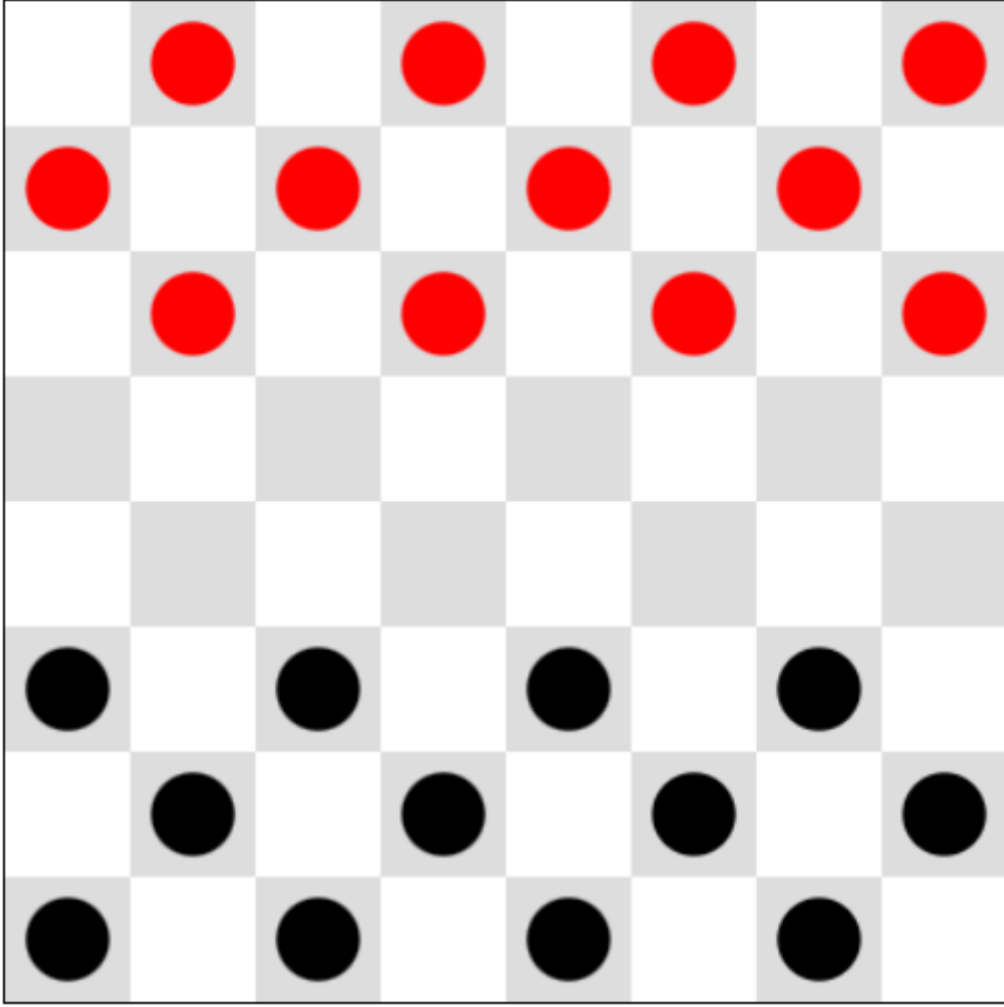


## Checkers (Dama) Oyunu



**Tür:** İki kişilik, deterministik, sıfır toplamlı strateji oyunu

**Neden uygun:**

- Tam bilgiye dayalı bir oyun olduğu için **Minimax** ve **Alpha-Beta Pruning** gibi algoritmaların öğretiminde idealdir.
- Oyun durumu kolayca **durum uzayı** (state space) olarak temsil edilebilir.
- **Heuristik fonksiyon** tanımlamaya elverişlidir (taş sayısı, kral taşlar, merkez kontrolü vb.).
- Gerçek bir örnek olarak **Chinook** yapay zekâsıyla bağ kurulabilir.

## Konular:

- Oyun ağacı (Game Tree) analizi
  - Minimax ve Alpha-Beta Pruning algoritmaları
  - Heuristik fonksiyon tasarımı
  - Nash dengesi ve optimal strateji
  - Sıfır toplamli oyunlarda denge durumu (beraberlik çözümü)
- 



### 1. Oyunun Tanımı

- **Oyuncu sayısı:** 2
  - **Tahta:** 8×8 kare (sadece koyu kareler kullanılır)
  - **Başlangıç:** Her oyuncunun 12 taşı vardır.
  - **Amaç:** Rakibin tüm taşlarını yemek veya hareket edemez hale getirmektir.
  - **Hareket:**
    - Normal taşlar ileri çapraz bir kareye gider.
    - Rakip taş bir kare ötede ve onun arkasındaki kare boşsa “üzerinden atlayarak” o taşı yer.
    - Bir taş karşı sıraya ulaşırsa “**kral**” olur ve artık **geri** de hareket edebilir.
- 



### 2. Oyun Kuramı Perspektifi

- **Deterministik** → Zar veya şans faktörü yok.
- **Sıralı (Sequential)** → Oyuncular sırayla hamle yapar.
- **Sıfır toplamli (Zero-sum)** → Bir oyuncunun kazancı diğerinin kaybıdır.
- **Tam bilgi (Perfect information)** → Her iki oyuncu da tahtayı ve tüm olasılıkları bilir.

Bu nedenle checkers, **Zermelo’nun teoremi** kapsamına girer: “Tam bilgiye sahip sıralı sıfır-toplamli bir oyunda, bir oyuncunun kazanma veya en azından berabere kalma stratejisi vardır.”

---

### 🌳 3. Oyun Ağacı (Game Tree)

Her hamlede çok sayıda olası durum ortaya çıkar.

- Ortalama **branching factor** (hamle sayısı):  $\approx 8-10$
- Ortalama **derinlik** (hamle uzunluğu): 50–100
- Tüm olası oyunların sayısı  $\sim 10^{40}$  (yani inanılmaz büyük).

Bu nedenle **tam arama (full search)** mümkün değildir.

Bu noktada **Minimax** ve **Alpha-Beta Pruning** devreye girer.

---

### 🎮 4. Minimax ve Alpha-Beta Uygulaması

**Minimax algoritması:**

Her oyuncu (MAX ve MIN) sırayla hamle yapar.

- MAX (örneğin kırmızı taşlar) kazancı maksimize etmeye çalışır.
- MIN (örneğin siyah taşlar) kazancı minimize etmeye çalışır.

Oyun ağacındaki **yaprak düğümler** (terminal durumlar) şu şekilde değerlendirilir:

- $+\infty \rightarrow$  MAX kazanır
- $-\infty \rightarrow$  MIN kazanır
- $0 \rightarrow$  Berabere

Fakat her pozisyonu hesaplamak mümkün olmadığından, **bir derinlik limiti** konulur (örneğin 6 hamle ileriye kadar bakılır) ve:

- Her pozisyona bir **heuristic (tahmini) değer** atanır.

## 🧠 Heuristik Fonksiyon Örneği:

Değer = (Taş sayısı farkı) + (Kral taş sayısı × 3) + (merkez kontrolü × 0.5)

Böylece algoritma, sadece taş sayısını değil, tahtadaki stratejik konumu da hesaba katar.

## ⚡ Alpha-Beta Pruning:

Minimax aramasında, kesin sonucu etkilemeyecek dallar elenir → böylece çok daha hızlı sonuç alınır. Bu teknik, checkers gibi karmaşık oyunlarda hesaplama süresini düşürür.

---

## 🤖 5. Yapay Zekâ Başarısı

- **1994'te** Jonathan Schaeffer ve ekibi tarafından geliştirilen **Chinook** adlı program, checkers'ta **dünya şampiyonunu yenen ilk yapay zekâ** olmuştur.
- **2007'de** Chinook, oyunu tamamen **çözmüştür**:

“Her iki taraf mükemmel oynarsa oyun **beraberlikle** biter.”

Bu, Zermelo teoreminin pratik bir doğrulamasıdır.

---

## 🧩 6. Oyun Kuramı Analizi

**Oyun tipi**     Deterministik, sıfır toplamı, iki oyunculu

**Strateji türü**     Saf (deterministik) stratejiler yeterlidir

**Nash dengesi**     Her iki oyuncu mükemmel oynadığında denge “beraberliktir”

**Utility (ödül) fonksiyonu**     +1 (kazanma), 0 (beraberlik), -1 (kaybetme)

**Optimal strateji**     Minimax veya onun yaklaşık formu (ör. alpha-beta)

---