**EGE ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY ZEKA YÖNTEMLERİ ve UYGULAMALARI**

**2019-2020 BAHAR YARIYILI**

**PROJE 2**

**YAPAY ZEKA / MAKİNE ÖĞRENMESİ / DERİN ÖĞRENME / ENİYİLEME PROJESİ**

Mehmet Enes KAPLAN

05160000386

**İçindekiler**

1. Proje Konusu ve Tanımı\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3
2. Ön Çalışma\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3
3. Kullanılan Ortam, Yöntem ve Kütüphaneler\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3
4. Önerilen Yöntem\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4
5. Deneysel Çalışmalar\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4
6. **Projenin Konusu ve Tanımı**

Bir insana karşı mantıklı hamleler yaparak satranç oynayan program.

1. **Ön Çalışma**

**2.1.a. Videolar**

# -Google's self-learning AI AlphaZero

# <https://www.youtube.com/watch?v=0g9SlVdv1PY>

# 

# -Minimax kullanarak hızlandırılmış satranç oyunu programlama

# <https://www.youtube.com/watch?v=DZfv0YgLJ2Q>

# 2.1.b. Yazılı

# -Taşların puanları ve konumlarına göre aldıkları ekstra puanların önemi <https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/>

# -Bitboard gösterimi bu teknik kullanılarak minimax ile çok daha derine aramalar yapılabilir.

# <https://www.chessprogramming.org/Bitboards>

# -Satranç programlamada deep learning nasıl kullanılır.

# <https://towardsdatascience.com/predicting-professional-players-chess-moves-with-deep-learning-9de6e305109e>

# 

# -Adım adım bitboard ile satranç programlama <https://www.gamedev.net/articles/programming/artificial-intelligence/chess-programming-part-i-getting-started-r1014/>

# -Basit yapay zekalı satranç programlama

# <https://byanofsky.com/2017/07/06/building-a-simple-chess-ai/>

**2.2. Neler Öğrendim**

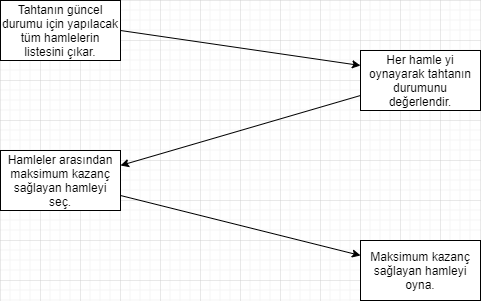
Tahta oyunlarının bitboard gösterimi ile daha iyi temsil edilmesi, bitboard kullanarak satranç da taşlar nasıl temsil edilir, satranç deep learning kullanarak nasıl programlanır, minimax için satranç taşları nasıl değerlendilir ve puanlandırılır.

1. **Kullanılan Ortam, Yöntem ve Kütüphaneler**

Visual Studio Code ortamı kullanılarak Python programlama dili ile proje gerçekleştirildi. Problem Minimax(Alpha-Beta pruning) yöntemi kullanılarak çözüldü ve herhangi bir kütüphane kullanılmadı.

**4. Önerilen Yöntem**

Minimax yöntemi ile birlikte her hamlenin bize olan maliyetini hesaplayarak en optimum hamle seçilerek maximum kazanç elde edildi. Rakip oyuncu hamle yaptıktan sonra o an geçerli olan tüm hamleler değerlendirilir ve her birinin bir kazanç puanı olur. Algoritmamız kazanç puanlarını değerlendirerek kendisi için en uygun hamleyi seçip onu oynayacaktır.



Algoritma en basit anlamıyla sözlü olarak şekilde verilmiştir.

**5. Deneysel Çalışmalar**

Bitboard gösterimi ile bir takım yöntemler denendi. Fakat kısıtından dolayı yapılan çalışmalar askıya alında. Fakat eğer bitboard gösterimi kullanılarak problem çözülseydi minimax ağaç derinliği 4 olduğunda dahi çok hızlı bir şekilde en iyi hamleyi bulacağı düşünülmektedir.

**Ek 1.** Başarının iyileştirilmesi için;

**Adım 1.)** Minimax algoritmasıyla her taş aynı puanda olduğu için hiç bir taşın önceliği olmayacaktır. Dolayısıyla sadece taş kaybını azaltmaya yönelik hamle yapılacaktır.

**Adım 2.)** Her taşın önceliğini değiştirmek. Her taşın önceliğini değiştirerek daha fazla fonksiyonelliğe sahip olan taşlar daha fazla korunup daha ileride ki hamleleri görerek küçük öncelikli bir taşı feda edip rakipten büyük öncelikli bir taş yenilebilecektir.

**Adım 3.)** Taşların konumlara göre ekstra kazanç sağlaması için her taş için tahtada her konuma özel ekstra puanlamalar yapılacaktır. Böylece taşların düşük puan dezavantajı da ortadan kalkacaktır.Örneğin sondan bir önceki satıra kadar gelmiş her piyon bir sonraki adımda ödüllendirileceği için ektra puan alacaktır.

**Ek.2**

|  |  |  |
| --- | --- | --- |
|  |  | **Açıklama** |
| **Makine Öğrenmesi** | Minimax | Minimax ile bir sonraki derinliğin önemi olduğu için ne kadar derine gidersek gidelim hep bir sonraki adımda taşımızı kaybediyor olabilir. Minimax algoritması ile taşı kurtarmak için mantıklı bir hamle yaptığımızı zannederken taşı ne olursa olsun kurtaramıyor da olabiliriz. |
| **Derin Öğrenme** | Convolutional Neural Networks | CNN ile birlikte yukarıda belirtilen derinlik problemi ortadan kalkacaktır. |

**Ek.3** Yararlanılan benzer uygulamalar:

<https://www.freecodecamp.org/news/simple-chess-ai-step-by-step-1d55a9266977/>

<https://towardsdatascience.com/predicting-professional-players-chess-moves-with-deep-learning-9de6e305109e>

<https://www.gamedev.net/articles/programming/artificial-intelligence/chess-programming-part-i-getting-started-r1014/>

<https://erikbern.com/2014/11/29/deep-learning-for-chess.html>

<https://www.youtube.com/watch?v=Xtd4dDKhsC8>

<https://www.youtube.com/watch?v=DZfv0YgLJ2Q>

**Özdeğerlendirme Tablosu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **İstenen Madde** | **Var** | **Açıklama** | **Tahmini Not** |
| **1** | **Kapak Sayfası, Problemin Tanımı, Kullanılan Ortam, Yöntem ve Kütüphaneler (10)** |  | Yapıldı. | 10 |
| **2** | **Araştırma (10)** |  | Yapıldı. | 10 |
| **3** | **Önerilen Yöntem (10)** |  | Yapıldı. | 10 |
| **4** | **Deneysel Çalışmalar** + **Ek 1: Başarım İyileştirme (10)** |  | Yapıldı. | 10 |
| **5** | **Proje Rapor Biçimi, Organizasyonu, Boyutu, Kalitesi, Kaynakça ve atıflar (10)** |  | Yapıldı. | 5 |
| **6** | **Ek 2 ve Ek 3 (10)** |  | Yapıldı. | 8 |
| **7** | **Ara rapor ve demosu (çalışma sürekliliği ve kalitesinin görülmesi açısından) (10)** |  | Yapıldı. | 8 |
| **8** | **Teams üzerinden Demo, Kodlama gösterimi ve Soru Cevap (20) (Sonradan yapılacağı için yaklaşık değer verebilirsiniz)** |  | Yapılması tahmin ediliyor. | 10 |
| **9** | **Özdeğerlendirme Tablosu (10)** |  | Yapıldı. | 10 |
| **100 üzerinden Toplam Not:** | | | | 91 |