

Ödev 1 – TicTacToe

Ön Hazırlık

- Başlangıç kodunu bu [bağlantıdan](#) indirin ve zip dosyasını açın.
- Proje dizinine girdikten sonra, bu proje için gerekli olan pygame Python paketini yüklemek için aşağıdaki komutu çalıştırın:

```
pip3 install pygame
```

Tanımlamalar

- Bu projede iki ana dosya bulunmaktadır: `oyun.py` ve `ajan.py`.
- `ajan.py` dosyası oyunun oynanması ve en iyi hamlelerin yapılması için gerekli tüm mantığı içerir.
- `oyun.py` dosyası ise sizin için önceden oluşturulmuştur ve oyunun grafiksel arayüzünü çalıştırmak için gerekli kodları içerir. Bu dosya üzerinde herhangi bir değişiklik yapmanıza gerek yoktur.
- `ajan.py` dosyasındaki tüm gerekli fonksiyonları tamamladıktan sonra, aşağıdaki komutu çalıştırarak yapay zekâya karşı oynayabilirsiniz:

```
python oyun.py
```

- `ajan.py` dosyasını açıp sağlanan kodları inceleyin.
- Tahtadaki olası hamleleri temsil etmek için X, O ve EMPTY olmak üzere üç değişken tanımlanmıştır.
- `initial_state` fonksiyonu, oyunun başlangıç tahtasını döndürür. Bu problemde tahta, üç liste içeren bir liste olarak temsil edilmiştir (üç satırı temsil eder). Her iç liste, X, O veya EMPTY değerlerinden birini içeren üç değerden oluşur. Devamında, sizin tamamlamanız gereken fonksiyonlar bulunmaktadır!

Gereklilikler

- Aşağıdaki fonksiyonları tamamlayın: `player`, `actions`, `result`, `winner`, `terminal`, `utility` ve `minimax`.

`player(board)`

- Bir tahta durumu almalı ve sıradaki oyuncuyu (X veya O) döndürmelidir.
- İlk hamlede X oynar, ardından oyuncular dönüşümlü olarak oynar.
- Eğer girilen tahta terminal (oyun bitmiş) durumdaysa, herhangi bir değer döndürülebilir.

`actions(board)`

- Belirtilen tahtada yapılabilecek tüm olası hamleleri bir küme (set) olarak döndürmelidir.
- Her hamle, (i, j) şeklinde bir demet (tuple) olmalıdır. Burada i, satır numarasını (0, 1, 2), j ise sütun numarasını (0, 1, 2) gösterir.
- Boş olan hücreler, olası hamlelerdir.
- Eğer girilen tahta terminal durumdaysa, herhangi bir değer döndürülebilir.

`result(board, action)`

- Belirtilen tahtada, verilen hamle oynandığında ortaya çıkacak yeni tahta durumunu döndürmelidir (orijinal tahtayı değiştirmeden).
- Eğer girilen hamle geçerli değilse, bir hata (exception) fırlatılmalıdır.
- Yeni döndürülen tahta, belirtilen hamlenin uygulanmış halini göstermelidir.

Önemli Not: Orijinal tahta değiştirilmemelidir! Çünkü Minimax algoritması, hesaplamaları sırasında birçok farklı tahta durumunu değerlendirecektir. Bu yüzden önce tahtanın derin bir kopyasını (deep copy) almanız gerekebilir.

winner(board)

- Verilen tahtada kazananı belirlemelidir.
- Eğer X oyuncusu kazanmışsa, X döndürmelidir. Eğer O kazanmışsa, O döndürmelidir.
- Kazanma durumu, bir oyuncunun yatay, dikey veya çapraz olarak üçlü bir sıra yapmasıdır.
- Hiçbir kazanan yoksa (oyun devam ediyorsa veya berabere bitti ise) None döndürmelidir.

terminal(board)

- Oyun bitmişse True, devam ediyorsa False döndürmelidir.
- Oyun, biri kazandığında veya tüm hücreler dolmasına rağmen kimse kazanmadığında biter.

utility(board)

- Terminal (bitmiş) bir tahta almalı ve tahtanın değerlendirme değerini döndürmelidir.
- Eğer X kazanmışsa 1, O kazanmışsa -1, oyun berabere bitmişse 0 döndürmelidir.
- `utility()` fonksiyonunun yalnızca oyun bitmişse çağrılacağını varsayabilirsiniz.

minimax(board)

- Verilen tahtada, sıradaki oyuncu için en iyi hamleyi döndürmelidir.
- Döndürülen hamle, tahtada oynanabilecek en iyi hamle (i, j) olmalıdır. Eğer birden fazla hamle eşit derecede iyi ise, herhangi biri döndürülebilir.
- Eğer tahta terminal durumdaysa, None döndürmelidir.

İpuçları

- Tüm bu fonksiyonlar için giriş tahtasının geçerli olduğu (3x3 boyutunda, X, O veya EMPTY değerlerinden oluştuğu) varsayılabilir. Fonksiyonların isimlerini veya parametre sayılarını değiştirmeyin.
- Tüm fonksiyonlar doğru şekilde uygulandığında, aşağıdaki komut ile yapay zekâya karşı oynayabilirsiniz:

```
python oyun.py
```

- Eğer iki taraf da optimal oynarsa, oyun her zaman berabere bitecektir! Ama eğer optimal oynamazsanız, yapay zekâ sizi yenebilir.
- `ajan.py` dosyasına yardımcı fonksiyonlar ekleyebilirsiniz, ancak var olan fonksiyon ve değişken adlarıyla çakışmadıklarından emin olun.
- Alfa-beta budama (alpha-beta pruning) kullanımı zorunlu değildir, ancak yapay zekânın daha verimli çalışmasını sağlayabilir.

Teslim

- Dönem boyunca verilen tüm ödevler için final haftasına kadar süreniz bulunmaktadır.
- Ödevinizi tamamladıktan sonra ilgili dosyaları `AdSoyad_Numara_OdevNo` ismini taşıyan bir klasörde sıkıştırarak öğretim üyesinin size ilettiği e-posta adresine gönderiniz.