

## YAPAY ZEKA PROJE ÖDEVİ

### Tic-Tac-Toe Minimax

18011602 Canberk Güllüoğlu

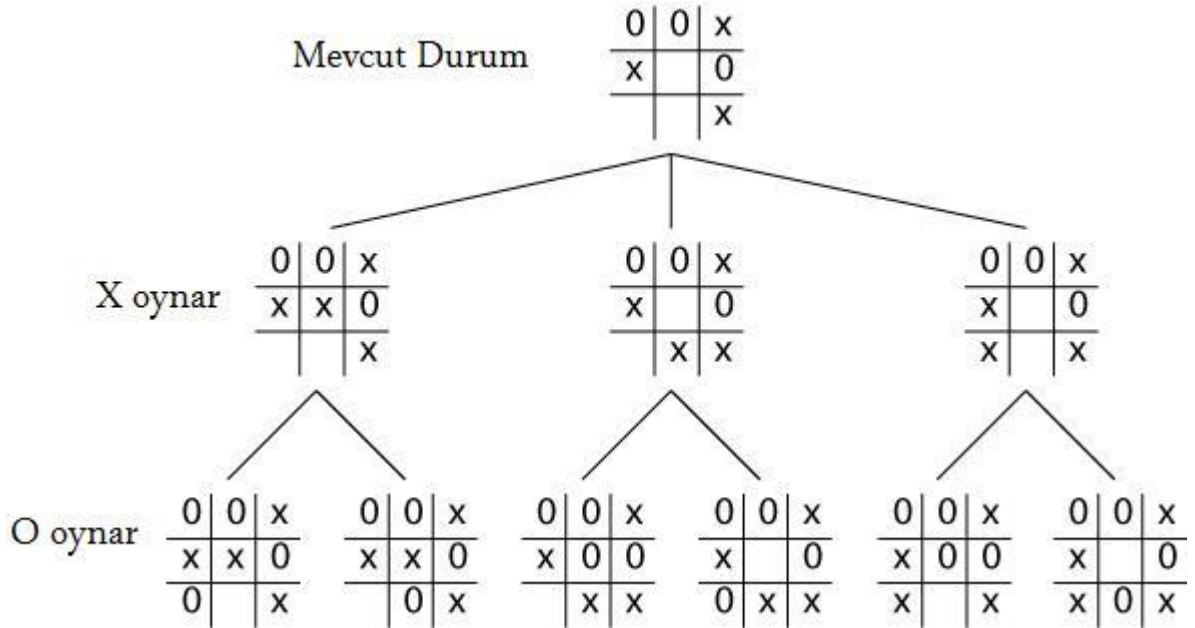
14011052 Rahmi Cemre Gül

#### Konunun Tanıtımı

Tic-tac-toe iki oyuncunun 3e 3 bir tahta üzerinde oynadığı bir oyundur. İlk oyuncu seçebileceği 9 kareden birine "X" atayarak oyuna başlar. İkinci oyuncu ise boş karelere "O" yerleştirir ve oyun karşılıklı devam eder. Kazanmak için oyuncuları temsil eden sembollerin 3'ünün yan yana ya da çapraz olarak dizilmesi gerekmektedir. Bu oyunda bilgisayar için en iyi hamleyi bulan algoritma Minimax Algoritmasıdır.

#### Minimax Algoritması

Minimax Algoritması, tahtanın bulunduğu pozisyonun ardından oluşabilecek pozisyonları bir ağaç yapısında özyinelemeli olarak değerlendirir. Bu değerlendirmede oyun iki oyuncuyla oynandığı için iki taraf için de değerlendirme yapılmalıdır. Hamleyi oynayacak oyuncu için maximum, rakip için ise minimum bulunur. Bu durum derinliğe göre özyinelemeli şekilde devam eder. En sonda karar verilirken, skoru en yüksek olan hamle oynanır. Örnek bir pozisyon için minimax ağacı, hamle sırası X'te:



#### Geliştirme Sürecinde Yaşananlar/Yorumlar:

Projenin 3x3 değil de 5x5 tahta üzerinde oynanması planlandığı için hesaplanması gereken hamle sayısı oldukça arttı. Bu da projemizin test edilmesinde bir takım zorluklara yol açtı. Örneğin, 3x3 tahta için derinlik 3 iken, hesaplanması gereken hamle sayısı  $9*8*7$  ihtimal değerlendirilirken 5x5 tahtada  $25*24*23$  ihtimal değerlendiriliyor. Derinlik arttıkça bu hesap gittikçe daha da zorlaşıyor.

Projenin tahta boyutu ve derinlik seçme özelliklerinin dışında bir de yapay zekaya karşı yapay zeka oynayabiliyor ve bu özellik oyunun ortasında seçilik deaktif edilebiliyor. Bu da minimax algoritmasına tahtanın verilmesi ile sağlanıyor.

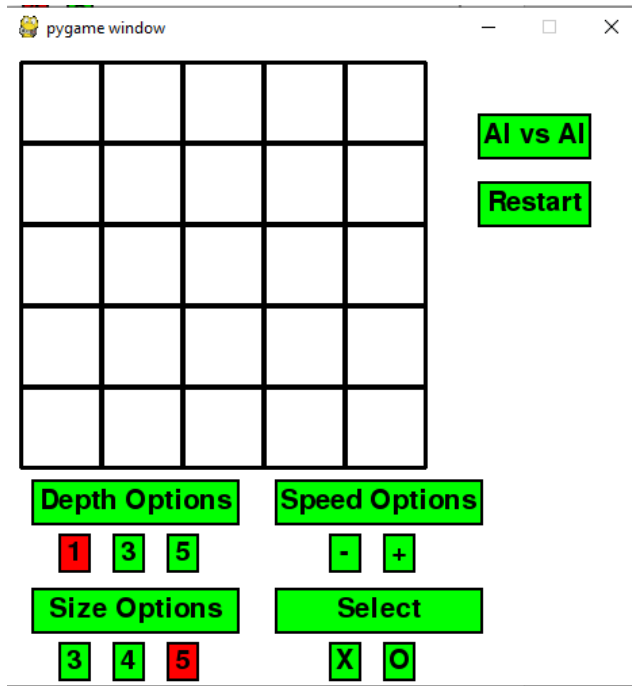
Uygulanması bizim açımızdan zor olan özelliklerden biri ise programın çalışma hızını belirleyen fonksiyondu. Öncelikle bu özelliği sleep() fonksiyonu ile sağlamaya çalıştık ancak sleep sırasında programın arayüz fonksiyonlarının kullanılmadığını farkettilik. Ardından pygame librarysinin sağladığı Clock fonksiyonu ile bu özelliği uygulayabildik.

Program da bulunmasını istediğimiz özelliklerden biri de 5x5 tahtada 5 yerine 4 karenin yanyana gelmesiyle kazanmanın sağlanmasıydı. Bu özelliğin kurgulanmasını başarıyla gerçekleştirek de minimax algoritması skor tutarken karşı tarafın da en iyi hamleyi yapacağını varsaydığı için bir takım zorluklar yaşadık, bunun önüne geçmek için ağaçta ilerlediğimiz sırada, her hamle için kazanan varyasyonların sayısı tutuldu ve eğer hamlelerin skoru aynı ise o hamleden sonra en fazla kazanma sayısı hangi varyasyondaysa o seçildi. Buradaki amacımız algoritma yenildiğini anladığında bile oyunu uzatmaya çalışmasını sağlamaktı. Ancak 5x5'te 4 kare kazanma özelliğini istediğimiz sonuçlara ulaşamadığımız için eklemedik.

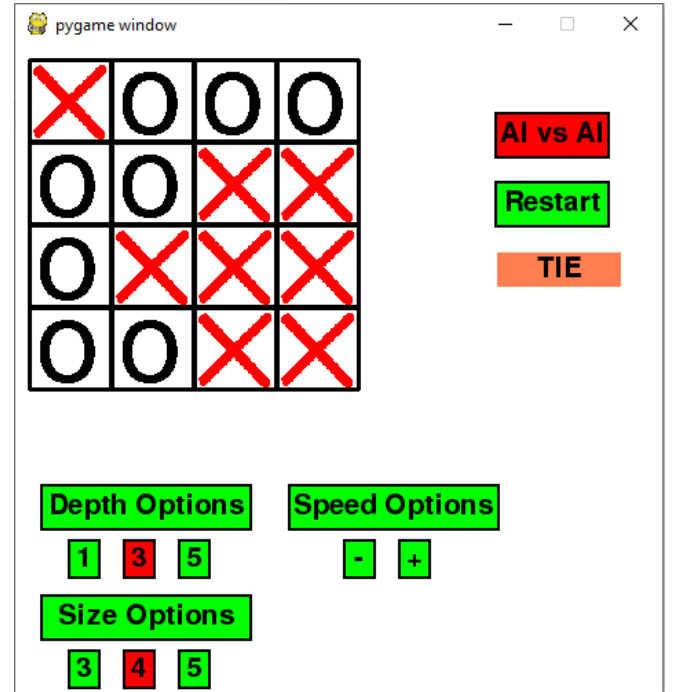
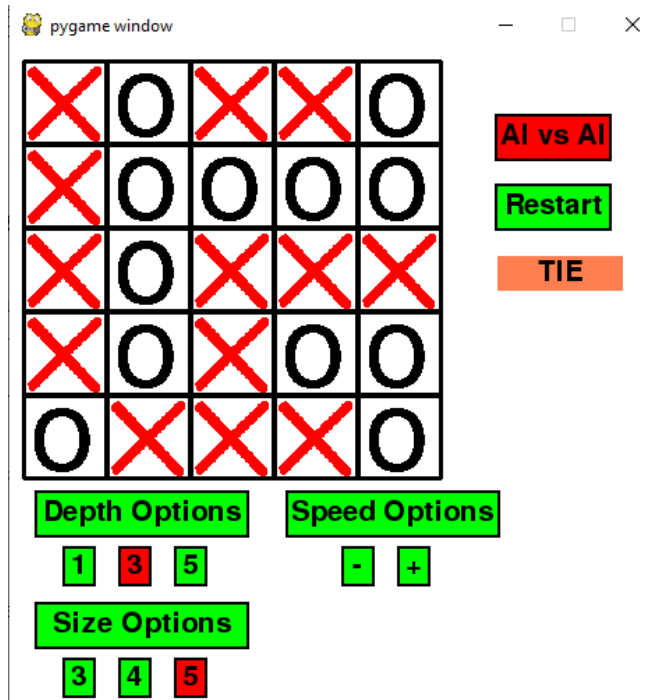
Aynı zamanda skor hesaplanırken skor / derinlik ile oyun kaybedilecekse uzun varyasyon, kazanılacaksa kısa varyasyon seçildi.

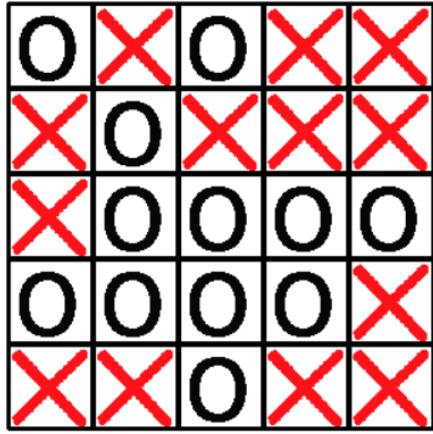
Hesaplanması gereken varyasyon sayısı çok fazla olduğundan algoritmanın çalışmasını hızlandırmak için alfa beta budaması yöntemi kullanıldı. Bu yöntem sayesinde eğer ağacın bir noktasından sonra daha iyi bir skor gelme ihtimali kalmadıysa, o seviyeden sonrasını hesaplamaya gerek kalmadan hamlenin skoru belirlenmiş oldu.

## Program Menü Görüntüsü:



## Programın Çalıştırma Örnekleri:





AI vs AI

Restart

TIE

Depth Options

1 3 5

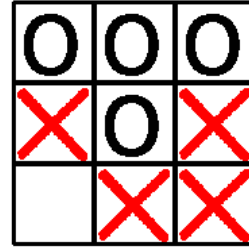
Speed Options

- +

Size Options

3 4 5

pygame window



AI vs AI

Restart

O WINS!

Depth Options

1 3 5

Speed Options

- +

Size Options

3 4 5

10 Deneme İçin Tablo

Derinlik\Tablo Boyutu	3 X-O-TIE	4 X-O-TIE	5 X-O-TIE
1	4-3-3	1-0-9	0-0-10
3	0-0-10	0-0-10	0-0-10
5	0-0-10	0-0-10	0-0-10

3x3, 4x4, 5x5 board için denemeler tabloda gösterilmiştir. Görüldüğü gibi derinlik 3 ve derinlik 5 için algoritma en doğru hamleleri oynayarak oyunu berabere bitirmiştir. Derinlik 1 için 3x3 board üzerinde aynı anda birden fazla kazanç oluşabileceği için algoritma yeterli seviyede hesap yapamamış ve 10 oyundan 7'sinin bir kazananı olmuştur. Diğer boardlar için 3 ve 5 derinliği optimum oynanma için yeterlidir.

Kaynaklar :

<http://bilgisayarkavramlari.sadievrenseker.com/2009/04/29/minimax-agaclari-minimax-tree/>

<https://www.youtube.com/watch?v=l-hh51ncgDI>