

# YAPAY ZEKA DERSİ PROJESİ

SERKAN YAMAÇ 15011908

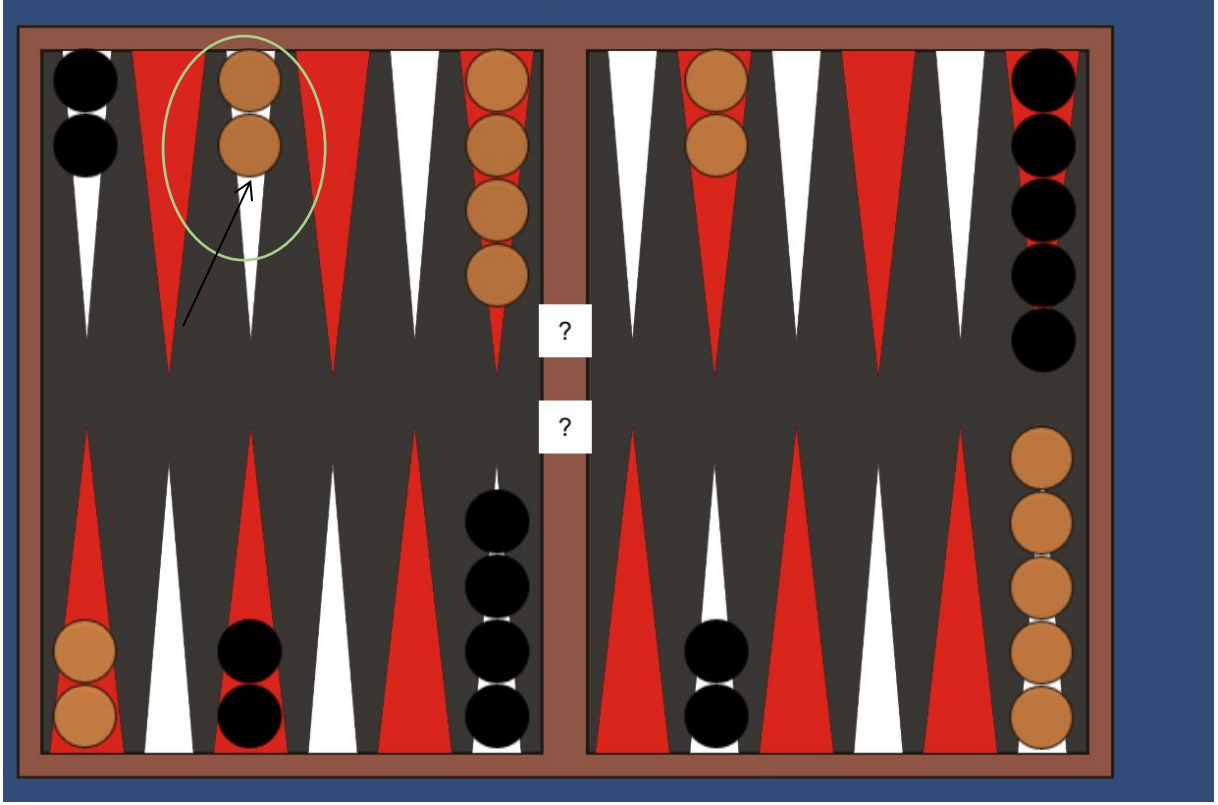
Konu: Expectiminimax Algoritması ile Tavla Oyunu Tasarımı (ComputerVsHuman)

Youtube: [https://youtu.be/0uafyH\\_dD3M](https://youtu.be/0uafyH_dD3M)

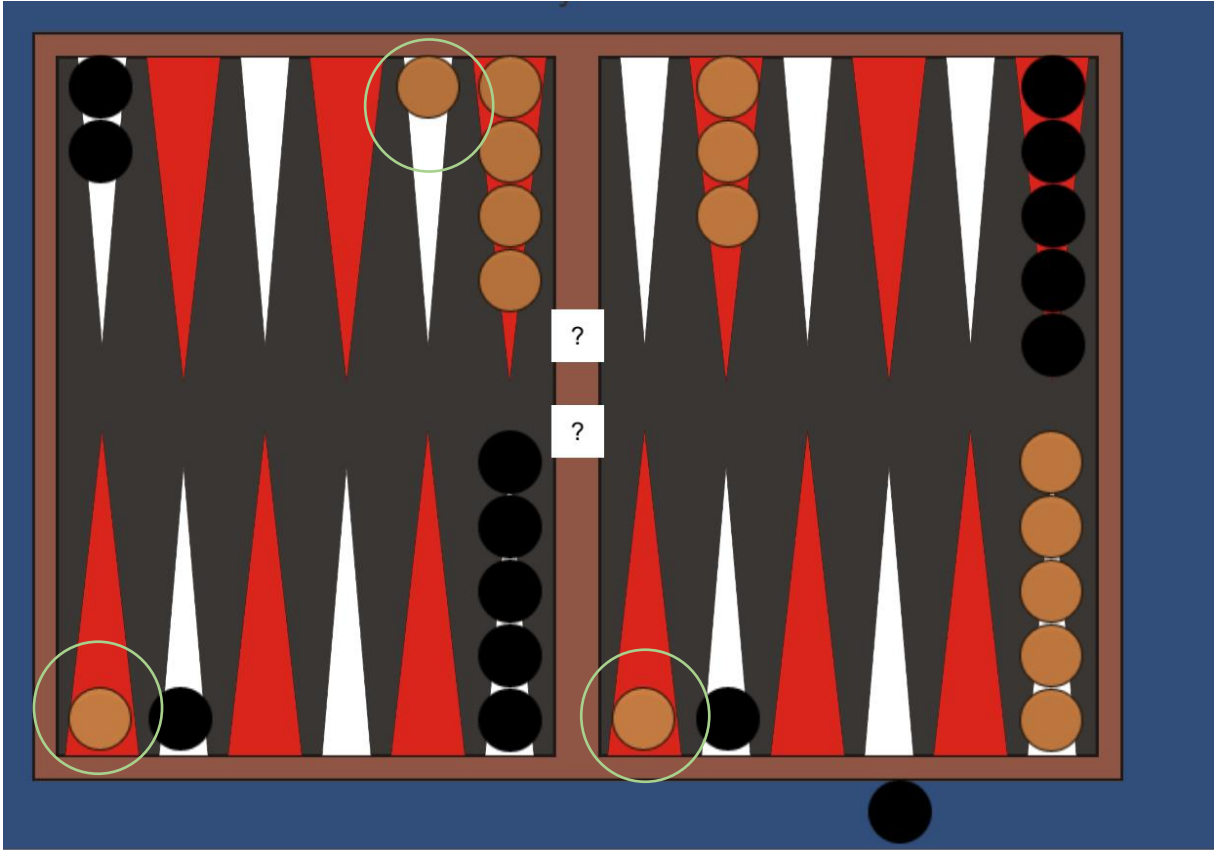
Proje kapsamında öncelikle tavla uygulamasını Unity C# platformunu kullanarak oyun mekaniğini geliştirdim. Daha sonrasında ise oyuna Expectiminimax algoritmasını entegre ederek bilgisayarın bana karşı oynamasına olanak sağladım.

Expectiminimax algoritmasından bahsedecek olursak, bu algoritma sonucunun oyuncunun becerisine, zar atma gibi şans durumlarına bağlı olan iki oyunculu, zero-sum oyunlar yani tavla gibi oyunlarda kullanılan minimax algoritmasının bir varyasyonudur. Bu algoritmada başlıca max-chance-min gibi düğümler bulunmaktadır. Kapsamlı bir ağaçtır fakat eksik bilgi içerir. Derinlik faktörü önemlidir. Derinlik arttıkça yapay zeka daha verimli daha doğru fakat **cok daha yavaş** kararlar verir. Derinlik yükseldikçe ortaya çıkan kombinasyonel hesaplamalar artacağından ağaçta uç noktalara ulaşmak çok zaman alır. Ben derinlik olarak 3 tercih ettim. Bu kombinasyonel sorunları alpha-beta pruning yapısında çözmek mümkündür.

Algoritma kapsamında bilgisayarın hareketleri için bazı durum değerlendirme fonksiyonları mevcuttur. Ben proje kapsamında kendime göre en ideal olarak her iki oyuncunun; kapı sayısı(bir stackte bulunan taş sayısının 1 den büyük olma durumu), açık taşların sayısını(negatif etkileyecek şekilde), kırık taşların sayısı ve koleksiyonda bulunan taşların sayısı şeklinde belirli katsayılar belirleyerek gerçekleştirdim. Bu verilerin katsayıları ile oynadığımızda bilgisayarın oyun tarzını değiştirebiliyoruz. Örneğin humanPlayer olan bizim kırık taş sayısı bilgisayar için yüksek öneme sahip bir değer olarak ayarlandığında bilgisayar kararlarını daha çok rakibin taşlarını kırmaktan yana tercih ediyor. Değerlendirme fonksiyonlarında sağlamış olduğum dengeden sonra örneğin bilgisayar kendisine oyun başında gelen 5-3,6-1 gibi zar ikililerinde kapılar alarak başlıyor. Aşağıdaki şekilde görebileceğiniz üzere 5-3 zarına karşılık hamlesi:



Aşağıda şekilde görebileceğiniz üzere 6-1 zarına karşılık(rakip oyuncuyu kırma kriterinin değeri arttığında yapmış olduğu hamle):



Fakat bu deęerlendirme fonksiyonlarını bir an nce tař toplayabilecek durumlara gre ayarladığımızda ise bilgisayar kapı alma durumlarına ok dikkat etmeden kendi blgesine en hızlı řekilde ulařmaya alıřıyor. Proje kapsamında bu deęerlendirme fonksiyonunun en stabil halini ayarlamak beni biraz uęrařtırdı. (Yařadığım en byk sorun budur.) Bilgisayarın en mantıklı kararları alabilmesi iin deęerlendirme fonksiyonun ve ierisinde ki kriterlerin doęru bir řekilde olduęunu dřnmekteyim.

Algoritmanın beni yenebildięi durumlar oldu. Bunu grmek bařarılı bir algoritma geliřtirdiğimi bana gsterdi. Daha ncede bahsetmiř olduęum aęa derinlięinin artması ok daha bařarılı bir bilgisayar oyuncusu ortaya ıkaracaktır. Fakat burada bir karar vermek gerekirse zaman ve oyun keyfi aısından 3 derinlięinin ideal olduęunu dřnmekteyim.