

Amiral Battı Yapay Zeka Modeli

Erdem Akyüz
TOBB ETÜ Bilgisayar Mühendisliği
IEEE Conference Publishing
e.akyuz@etu.edu.tr

Abstract—Amiral battı oyunu 19. yüzyılın sonlarında ortaya çıkmıştır. Ve günümüzde bile hala oynanan bir oyun olmuştur. Bu makalenin temel amacı amiral battı oyununda yapay zeka ile karar vererek oyunu oynayacak modelin ne tür aşamalar ele alınarak tasarlandığını ve oyuna entegre edildiğini anlatmaktır. Bu model basit algoritmaların üst üste konulmasından elde edilen sonuçlar ile geliştirilmiştir. Makalenin bölümlerinde bunlara tek tek değinilecektir. Ayrıca bu oyunun orijinal halinde olmayan yeni bir mekanik olan gemilerin hareketi eklendiğinde oluşacak yeni oyuna eklenmiş olan yapay zeka modelinden de bahsedilecektir.

Index Terms—Battleships, AI, Amiral Battı, Yapay Zeka

I. GİRİŞ

Amiral battı oyunu 1. Dünya Savaşı'ndan beri oynanmaktadır. Tarihte bu oyun kağıt kalem kullanılarak oynanmaktaydı. 20. yüzyılın sonlarına doğru bilgisayarların ortaya çıkmasıyla beraber bu oyun bilgisayar ortamına da aktarılmıştır. Oyun 2 oyuncu ile sıra sıra hamle yapılarak oynanmaktadır.

Oyunun oynanış biçimi şu şekilde işlemektedir: Oyun başlamadan önce her 2 oyuncu da gemilerini 10x10 boyutunda karelerden oluşmuş bir harita üzerine konumlandırmaktadır. Gemiler yan yana gelebilir Ancak üst üste gelecek şekilde konumlandırılmaz. Gemilerin boyutları aşağıdaki şekildedir.

Adet	Gemi	Kare
1	Amiral (uçak) gemisi	5
1	Kuvazör (savaş) gemisi	4
1	Muhrip	3
1	Hücumbot	3
1	Denizaltı	2

Bu gemiler 10x10 boyutundaki haritaya yerleştirildikten sonra oyunculardan birisi bu koordinatlar içerisinde bir nokta belirler ve karşısındaki kişiye söyler. Eğer bu noktada bir düşman gemisi konumlandırdıysa o kişi gemilerinden birinin vurulduğunu bildirir. Tur bu şekilde diğer kişiye geçer ve o kişide bir nokta söyler. İlk önce kimin diğer oyuncunun tüm gemilerini vurmayı başarsa o kişi kazanan olur.

II. OYUN TASARIMI

Öncelikle bu projenin amacı gereğince modelin çalışabilmesi için bir amiral battı oyunu python dilinde geliştirilmiştir. Ancak oyun mantığı sadece modeli denemek amacıyla kaldırılmıştır. Programda sadece yapay zeka modeli oyunu oynayabilecektir. Amacı ise en az deneme yaparak tüm hedefleri vurmak olacaktır. Gemilerin yönü ve lokasyonu rastgele belirlenecektir.

Literatürde gemilerin yerini belirlemede de yapay zeka modellerinden yararlanılmaktadır. Ancak proje sadece tek bir oyun performansını ölçeceği için bu aşama es geçilmiştir.

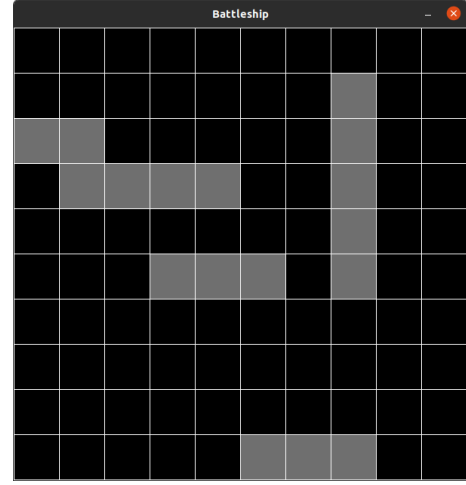


Fig. 1. Projenin görsel tasarımı

III. METODOLOJİ

A. Model seçimi

Oyunumuzu hazırladıktan sonra yapay zeka algoritmasına geçebiliriz. Amiral battı oyunda kullanılabilecek 3 çeşitli yapay zeka modeli bulunmaktadır. Bunlar "rule based expert system", genetik algoritmalar ve yapay sinir ağlarıdır. Bu projede rule based sistemler üzerinden gidilecektir. "Rule based expert systems", kullanıcının yapay zeka modelinin davranışlarını önceden tanımladığı algoritmalar. Bu modeller oyun oyunun çalışma zamanında öğrenme yapmazlar, ancak bulundukları konum ve durumu analiz ederek mantıklı kararlar verebilirler.

B. Oyun Analizi

Amiral battı oyunu 10x10 kareden oluşmaktadır. Yani toplamda 100 tane ateş edilebilen alan bulunmaktadır. Düşman gemilerinin uzunluklarını toplarsak 17 kare olduğu bilinmektedir. O zaman amacımız en az ateş denemesi yaparak 100 kare içinden 17 kareyi bulmamız gerekmektedir.

C. Tamamen Rastgele Model

Öncelikli olarak modelimizi sadece rastgeşe ateş edecek şekilde programlayalım. Model sadece önce ateş ettiği koordinatları hafızasında tutacak ve önceden ateş etmediği koordinatları hedef alacaktır. Beklendiği üzere, rastgele yayılım

ateşi açmanın sonuçları çok kötü sonuçlar veriyor. Tüm gemilerin batmasını sağlamak için karelerin çoğunun vurulması gerektiğinden oyunların tamamlanması uzun zaman alıyor. Bu modelimizi 300 kere denediğimizde ortalama 95.3 turda bitirdiği gözlemlenmiştir. 30000 oyun içinde en az turda biten oyunun 57 turda bittiği gözlemlenmiştir. Bu şekilde oynanan oyunların %50 sinin 96 deneme ateşi yapması gerektiği gözlemlenmiştir. Ve oyunların %99'unun 78 deneme ateşinden daha yapılması gerektiği görülmektedir.

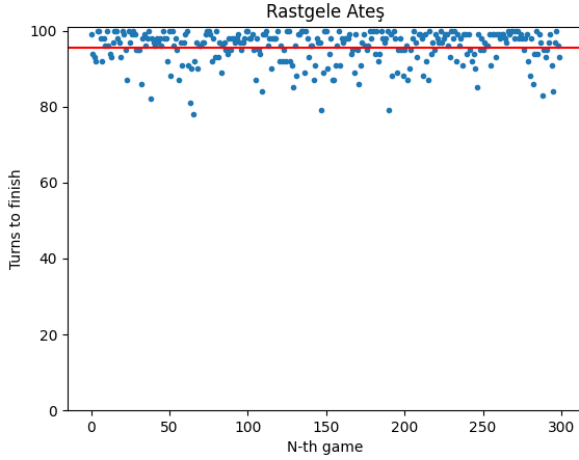


Fig. 2. Rastgele koordinat seçimi

D. Hunt Target Metodu

Elimizde olan rastgele modele yeni bir fonksiyon ekleyelim. Bu fonksiyon ateş ettiği nokta eğer bir gemi ise etrafındaki koordinatları olası bir hedef olarak görüp hafızasına kaydetсин ("Target" moduna girecektir). Ve bir sonraki hamlesini yaparken bu noktalardan faydalansın. Kısacası vurulan noktanın altındaki, üstündeki, sağındaki ve solundaki koordinatı bir listeye ekleyeceğiz ve sonraki tur listede eleman olup olmadığını kontrol ederek hamle yapmasını sağlayacağız. Eğer vurulan nokta bir köşe noktası ise hangi köşe olduğuna bakılarak bazı koordinatlar bu duruma dahil edilmeyecektir. Vurulan koordinatın etrafındaki noktalar bir "stack" yapısında tutulacaktır. Eğer bir stack yapısında bir eleman bulunuyorsa bu eleman "stack"den çıkarılacak ve o noktaya ateş edilecektir. Bu şekilde eğer vurulan geminin koordinatlarının hemen yanında bir başka gemi bulunuyorsa o geminin tüm elemanlarının vurulacağı garanti edilmektedir. Hunt Target metodu kullandığımızda ortalama bitirme tur sayısının 66.6'ya düştüğü görülmektedir. Aynı zamanda bu metod normal bir insanın bu oyunu oynamasıyla elde edebileceği sonuçlardır. Tabi şunu da belirtmek gerekir ki bu metodun vurduğu bir hedef olmadığı takdirde rastgele koordinata ateş edecektir ("Hunt" moduna girecektir). 30000 oyun içinde en az turda biten oyunun 24 turda bittiği gözlemlenmiştir. Her ne kadar bu metod rastgele ateş etme metoduna göre önemli ölçüde iyi sonuçlar versede vurduğu noktaları körü körüne araması gerektiği için en uygun model olarak kesinlikle değerlendirilemez.

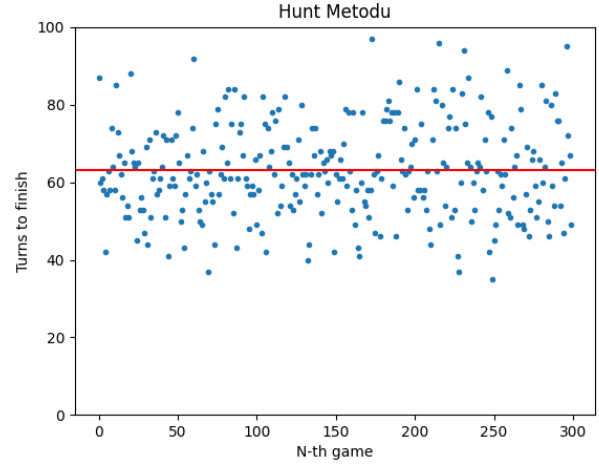


Fig. 3. Hunt Metodu

E. Hunt Target Metodu ve Capraz Koordinat Testi

Hunt Target metodunda bahsedildiği üzere bu metodun vurduğu bir hedef olmadığı sürece rastgele koordinat seçimi ile arama yapacaktır. Hiçbir bilgi olmadan yapılan bu deneme ateşlerini kaldırmayı başarabilirsek bitirme tur sayımızı azaltabiliriz. Bu nedenle rastgele koordinat denemek yerine 100 karelik haritamızı 2 ye böleceğiz. Bu şekilde 100 birimlik bir alanda rastgele ateş etmek yerine 50 birimlik alanda ateş edeceğiz. Bunu yapma nedenimiz ise 1 kare boşluk bırakılarak haritanın yarısını vurduğumuzda haritadaki tüm gemilerin en az 1 parçasını vurduğumuzu garantilemiş oluruz. Hunt metodumuz ise tüm gemilerin 1 parçasını bildiği için gemilerin diğer parçalarını rahatlıkla bulabilir hale gelecektir.

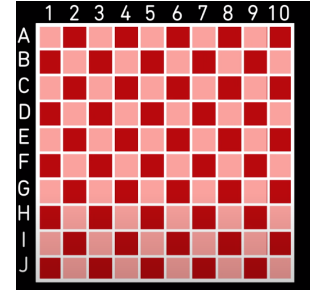


Fig. 4. Çapraz Koordinat Örnek

Peki bu daha iyi hale getirebilir mi? Bunun için gemilerin boyutlarına bakılması gereklidir. 2 kareden birine bakmak yerine haritada mevcut tüm parçaları vurulmamış en uzun gemiyi aranırsa bu sayı daha da azaltmış olunur. Bu şekilde çapraz arama metodunun içindeki her 2 karenin belirlendiği bölüme oyundaki o anda tüm parçaları hala vurulmamış en uzun geminin boyutu yazılırsa elde edilecek sonuç aşağıdaki gibi olacaktır. Sonuç olarak bu metodun entegre edilmesiyle ortalama bitirme sayımızı 60.2'ye indirmiş bulunmaktayız. Bu figür 4'e bakarak incelenebilir. 30000 oyun içinde en az turda biten oyunun 25 turda bittiği gözlemlenmiştir. Ancak bu model vurulan noktaların köşe olmadığı sürece 4 bir tarafını ateş

ettiği için fazladan tur oynamış olmakta. Bir sonraki model bu sorunun önüne geçmeyi amaçlamakta ve modele tamamen yeni bir bakış açısı getirmektedir.

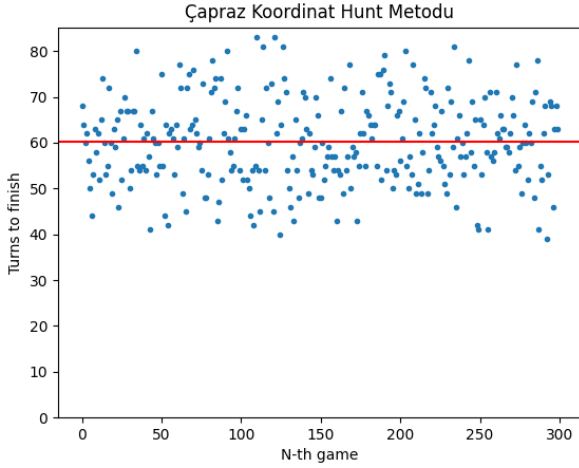


Fig. 5. Çapraz Koordinat Hunt Metodu

F. Olasılık Haritası Metodu

Önceki metodlarda oyunda yararlanılan 2 durum vardı; ateş edilen nokta vuruldu mu, yoksa ıskalandı mı. Bunun yanında oyunun temel kurallarında bulunan bir gemi tamamen battığında bunu rakibe bildirilmesi gerektiği. Aynı zamanda hangi geminin battığı da bildirilmesi gerekmektedir. Bu faktörleri değerlendirdiğimizde oyun haritasında hangi gemilerin battığını ve hangi gemilerin hala yüzdüğünü anlayabiliriz. Bu bilgi de bize oyun alanında "hangi uzunlukta" gemilerin olduğunu bildirir.

Yeni algoritmamız harita üzerindeki tüm koordinatların oyunun her turunda olası olarak bir gemi bulundurduğunun olasılık haritasını hesaplayacaktır. Her bir yeni ateş denemesi yapıldıktan sonra bu olasılık haritası vurulan noktada gemi olup olmamasına göre tekrar hesaplanacaktır.

Bu algoritma öncelikli olarak haritanın ortasını hedef alacaktır. Daha doğru olarak Amiral gemisinin bir köşesinin bulunma olasılığının yüksek olduğu koordinatı hedef alacaktır.

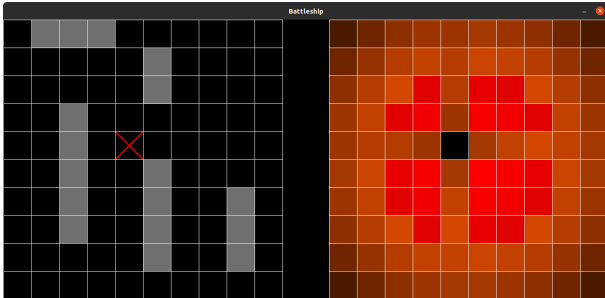


Fig. 6. Amiral Gemisi Hedefi

Amiral gemisinin uzunluğu göz önüne alındığında ateş edilen ilk nokta o geminin sınırları gereğince bulunamayacağı bir sınır çizgisi oluşturacaktır. Bu şekilde en uzun gemi

aranırken her bir ateş edildiğinde eğer o koordinatta bir gemi yoksa ve ıskaladıysak o noktanın olasılığı 0'a indirelecektir. Ve harita üzerindeki her nokta için bir iterasyon yapılır. Bu iterasyonun adımları şu şekildedir (vurulan noktaya (x,y) diyelim):

- Henüz ateş edilmemiş tüm koordinatların olasılık değerini 1 yap.
- If: Eğer (x,y) noktasına ateş edilmiş ve ıskalanmış ise bu noktanın olasılığı 0 yapılır.
- Else: Eğer (x,y) noktasına ateş edilmiş ve bu noktada bir gemi vurulmuşsa ve o gemi henüz batmadıysa:
 - If: (x,y) noktasının dikey ve yatay tarafı kontrol edilir (bu tarafaki diğer noktaya (x+1,y) diyelim) eğer (x+1,y) harita sınırları içerisindeyse ve (x+1,y)'ye henüz ateş edilmediyse: (Aynısını (x-1,y), (x,y+1), (x,y-1) için de uygula)
 - * If: (x,y) noktasının seçilmiş olan yönündeki (x+1,y) noktasının tersi yönündeki noktaya ((x-1,y) diyelim) ateş edildiyse ve (x-1,y) noktasında bir gemi varsa ve bu gemi batmayan bir gemiyse:
 - (x+1,y) noktasının olasılık değeri 15 puan arttır.
 - * Else:
 - (x+1,y) noktasının olasılık değeri 10 puan arttır.
- Ateş edilmiş ve geminin vurulmadığı tüm koordinatların olasılığını 0 yap.

Bu işlemler yapıldıktan sonra harita üzerinde her bir koordinat üzerinde gemi bulunma olasılığı edilmiş olur. Ardından en yüksek olasılığa sahip koordinatı seçip ateş ederiz bir sonraki tur bu ateş edilmiş noktanın yarattığı etkiler ile tekrardan bir olasılık haritası hesaplanır ve döngü bu şekilde devam eder.

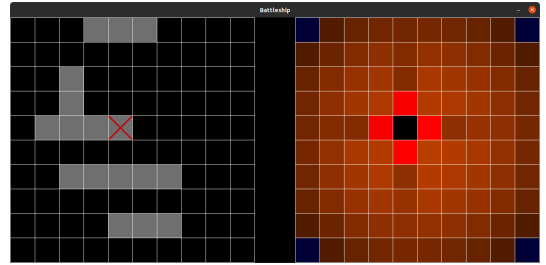


Fig. 7. Vurulan nokta olasılığı

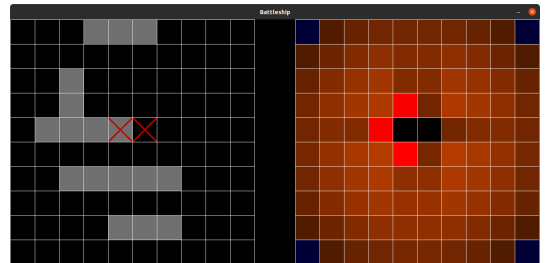


Fig. 8. Vurulan nokta olasılığı

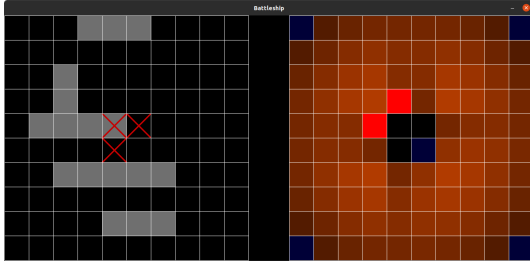


Fig. 9. Vurulan nokta olasılığı

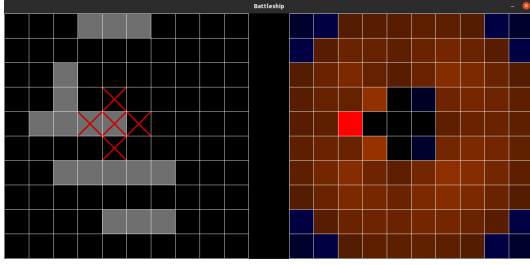


Fig. 10. Vurulan nokta olasılığı

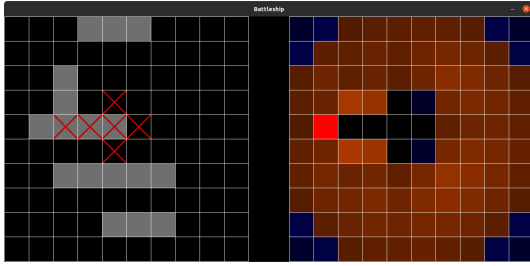


Fig. 11. Vurulan nokta olasılığı

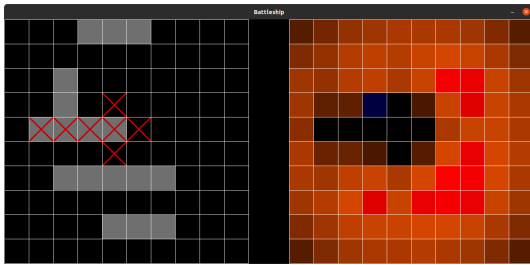


Fig. 12. Vurulan nokta olasılığı

Figür 7-12 arasında görüldüğü üzere bir koordinatta ateş isabetli olursa o noktanın etrafındaki noktaların olasılığı artırılır. Eğer ateş edilen koordinatın etrafındaki değerler yukarıdaki algoritmaya göre 10 veya 15 olarak artırılır. Bu şekilde gemi yönünde bulunan karelerin olasılığı daha yüksek olacaktır. Resimleri sağ tarafındaki olasılık haritasında bir kare ne kadar kırmızıya yakınsa o koordinatın gemi bulundurma olasılığı daha yüksek demektir. 1 geminin tüm parçaları vurulduktan sonra oyunun kuralları gereği algoritma bunu bilebilecek ve hala en az 1 parçası vurulmamış olan en uzun geminin olabileceği koordinatlara ateş edecektir.

Bu algoritmanın performansını incelediğimizde ortalama ateş denemesini 43'e indirdiği görülür. Bu Hunt metoduna göre oldukça iyi bir gelişmedir. 300 oyun içinde en az turda biten oyunun 23 turda bittiği gözlemlenmiştir. En kötü oyunun ise 67 turda bittiği gözlemlenmiştir.

Ancak bu metodun çalışma süresi önceki metodlara kıyasla epey yüksektir. Çapraz koordinatı kullanarak çalıştırdığımız Hunt metodu 300 oyunu sadece 0.15 saniyede simüle edebilirken olasılık haritası metodu için bu süre 78.4 saniye ile çok yüksek seviyelere çıkmıştır. Bunun sebebi her bir ateş denemesi yapılmadan önce tüm haritanın üzerinden geçip tüm koordinatlara bir sayı değeri atandığı içindir. Fakat bu pu projede önemli olan amaç 1 oyun için ortalama performansı yükseltmek olduğundan ve aynı zamanda gerçek bir oyuncuya karşı rakip olarak denendiğinde gerçek oyuncunun düşünme süresinin yanında bu sürenin bir önemi olmayacaktır. Figür 13'te olasılık haritası metodu ile simüle edilmiş 300 oyunun kaç ateş denemesinde bitirilebildiği gösterilmiştir.

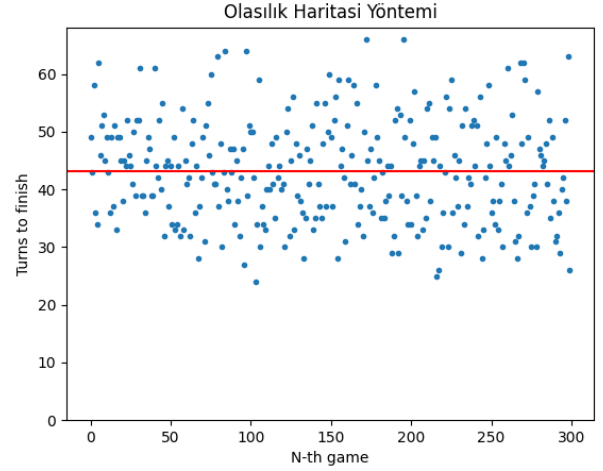


Fig. 13. Olasılık Haritası Metodu

IV. OYUN KURALLARINI DEĞİŞTİRMEK

Buraya kadar konuşulmuş metodların hepsi klasik bir amiral battı oyununun kuralları baz alınarak oluşturulmuş metodolojilerdi. Bu projede ek bir konu olarak bazı kurallarda değişiklikler yapılarak bilinen amiral battı oyununa yeni bir boyut kazandırılmış ve yeni kurallara göre oynanan başka bir model geliştirilmiştir.

A. Yeni Kurallar

Klasik bir amiral battı oyununda gemiler hangi koordinatlara ve hangi yöne (dikey, yatay) konuşlandırıldığı deneme atışları yapılmaya başlanmadan önce belli olur. Ve oyun süresi boyunca hareket etmeden sabit kalırlar. Yeni tasarladığımız bu amiral battı versiyonunda gemiler konuşlandırıldığı yön tarafında hareket etmektedir. Yani bir gemi dikey olarak konuşlandırıldığı takdirde yukarı ve aşağı hareket edecektir. Aynı şekilde yatay konuşlandırılmış bir gemi sağa veya sola gidecektir. Gemiler her bir ateş denemesi yapıldıktan sonra 1 birim ilerleyecektir. Eğer bir geminin ilerleyeceği doğrultadaki

komşu koordinatında başka bir gemi bulunuyorsa veya o nokta harita sınırları dışında yer alıyorsa geminin doğrultusu tam tersi olarak ayarlanıp 1 birim hareketini gerçekleştirecektir. Geminin tüm parçaları vurulmadıkları sürece (vurulan parçalar dahil olmak üzere) hareketine devam ederler. Geminin tüm koordinatları vurulduğu anda gemi olduğu yerde batacağı ve hareketini durduracaktır. Hareket halinde olan diğer gemiler bu gemiye çarparsa doğrultuları ters yön olacak şekilde değiştirilecektir.

B. Yeni Metodoloji

Klasik amiral battı için kullanılan algoritmaların hepsi gemilerin sabit kaldığını baz alarak çalışmaktaydı. Ancak yeni kurallar oyuna entegre ettikten sonra bu varsayımı kullanamamaktayız. Rastgele ateş denemesi yapan metodun bile oyunu bitirebilme garantisi bulunmamaktadır. Çünkü artık 100 deneme sayısı oyun haritasını tamamen taramak için yeterli olmamaktadır. Yine de yapılan simülasyonlar sonucu tamamen rastgele ateş eden bir modelin (önceden ateş ettiği yere tekrar ateş edebilir) 1000 oyundaki ortalama performansı 344.1 olarak hesaplanmıştır. En iyi tur sayısı 115, en kötü tur sayısına sahip simülasyon ise 1146 adımda oyunu bitirmiştir. Figüre 14'te sonuçlar gösterilmiştir. Bizim amacımız bu tur sayısını verilen kurallar kapsamında mümkün olan en düşük seviyeye getirecek optimal modeli bulmaktır.

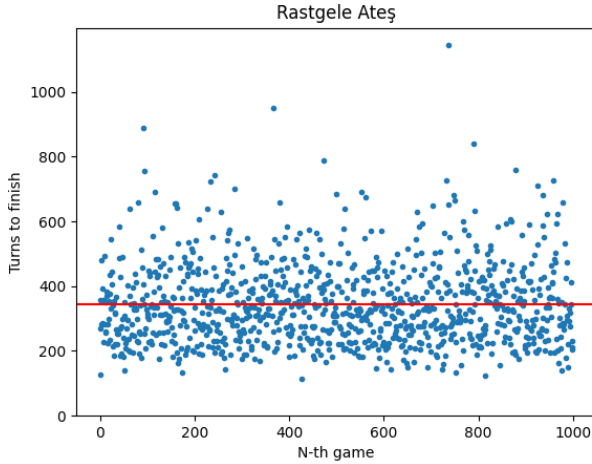


Fig. 14. Rastgele Ateş Metodu

C. Sabit Tarama Metodu

Yeni oyunumuzda eski oyunda varolan bilgilerden vurduğumuz geminin isabetli bir atış olup olmadığından faydalanacağız. Yapılan çeşitli modellerin sonucunda ulaşılan sonuçta model başarımına etki eden en iyi özelliğin vurulan bir noktanın üzerine tekrar ateş etmek olduğu sonucuna ulaşılmıştır. Bu metod "sabit tarama" olarak adlandırılmıştır. Öncelikle oyun alana rastgele olarak ateş açılır. Eğer bir gemiye isabet edersek bir sonraki ateş noktamız da aynı olacaktır. Bu şekilde gemi hareketine devam ettiği süre boyunca o geminin diğer parçaları aynı noktadan geçeceği için üst üste gemileri vurmuş olacaktır. Bunun yanında

vurulan bir nokta (x,y) olduğu takdirde, $(x+1,y)$, $(x-1,y)$, $(x,y+1)$, $(x,y-1)$ noktaları harita dışı bir koordinat olmamak koşuluyla bir stack yapısına kaydedilir. Eğer sabit tarama yapılırken gemiyi vurulmazsa ve bu gemi henüz tam olarak batmamış durumdaysa aynı yere tekrar ateş etmek yerine stack e eklediğimiz noktalara tek tek ateş eder. Ve döngü bu şekilde devam eder. Rastgele ateş eden model 1000 oyunun ortalaması olarak 344 ateş denemesinde oyunu bitirmeyi başarmıştı. Sabit tarama modeli ortalama olarak 118 tur sayısına sahiptir. En kötü tur sayısını 405 bulurken en tur sayısını 55 olarak tamamlamıştır.

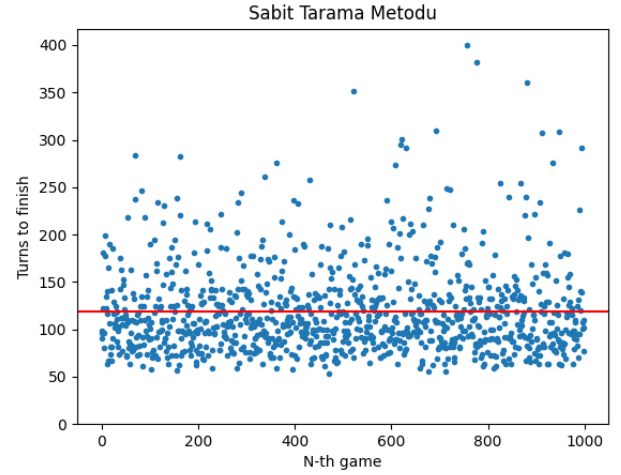


Fig. 15. Sabit Tarama Metodu

Bu metod üzerinde eksik olan faktörlerden birisi vurulan hedeflerin hedef tam olarak batırılmadan başka hedefleri aramaya başlayıp mevcut hedefini unutmasıydı. Bunun üzerine model, başka gemileri rastgele aramak yerine öncelikli olarak tek bir hedefi x ve y koordinatları üzerinde 19 nokta olacak şekilde sınırlandırıp arama yapacağı şekilde güncellendi ancak elde edilen sonuçlar mevcut model ile aynı oldu üstelik metodun çalışma süresi ciddi şekilde artmıştır. Bu yüzden raporda algoritmanın bu yönüne değinilmemiştir.

V. SONUÇ

Bu raporda uzun zamandır oynanmakta olan amiral battı oyununu kendi başına akıllı ve mantıklı kararlar yaparak oynayacak çeşitli modellerden bahsedilmiştir. Aynı zamanda bu modellerin kendi içlerinde karşılaştırması yapılır sonuçlar görseller yardımıyla sunulmuştur. Oyunun klasik versiyonunda mevcut olmayan yeni kurallar koyularak başka yapay zeka modelleri oluşturulmaya çalışılmıştır. Ve başarılı olan model sunulmuştur.