

Yapay Zeka Derin Pekiştirmeli Öğrenme ile Hayatta Kalmayı Öğreniyor

Hüseyin Buğra Taştan^{1*}, Emre Güngör² ve Serel Özmen-Akyol³

^{1,2,3}Bilgisayar Mühendisliği Bölümü, Mühendislik ve Doğa Bilimleri Fakültesi, Kütahya Sağlık Bilimleri Üniversitesi, Türkiye

*(hbugra.tastan@ogr.ksbu.edu.tr)

(Received: 23 November 2024, Accepted: 26 November 2024)

(2nd International Conference on Trends in Advanced Research ICTAR 2024, November 22-23, 2024)

ATIF/REFERENCE: Taştan, H. B., Güngör, E. & Özmen-Akyol, S. (2024). Yapay Zeka Derin Pekiştirmeli Öğrenme ile Hayatta Kalmayı Öğreniyor. *International Journal of Advanced Natural Sciences and Engineering Researches*, 8(10), 294-301.

Özet – Günümüzde, yapay zeka teknolojisinin hızla gelişmesiyle birlikte, yapay zeka ajanlarının gerçek dünya olay ve ortamlarında başarılı bir şekilde işlev görmesi ve hatta hayatta kalabilmesi giderek daha büyük bir önem kazanmaktadır. Yapay zeka ajanlarının çevresel değişimlere hızlı uyum sağlayarak en uygun kararları alması, bu sistemlerin gerçek dünya problemlerine yönelik potansiyelini ortaya koymaktadır. Bu çalışmada derin pekiştirmeli öğrenme (deep reinforcement learning) teknikleri kullanılarak, yapay zeka ajanlarının dinamik, tehlikeli ve değişken çevrelere uyum sağlamaları hedeflenmiştir. Çalışma kapsamında, ajanların doğal afet, engel ve diğer tehditler gibi gerçek dünyadan ilham alınan zorluklarla başa çıkabilmeleri sağlanmıştır. Bu süreçte, ödül ve ceza sistemleri kullanılarak istenen davranışların öğrenilmesi teşvik edilmiştir. Bu amaçla Unity ML-Agents kullanılarak simüle edilen oyun ortamlarında, ajanlar gerçek dünya koşullarına benzer bir şekilde eğitilmiştir. Eğitim sürecinde Proximal Policy Optimization (PPO) gibi modern algoritmalar kullanılarak, öğrenmenin hızlanması ve kararlı hale gelmesi sağlanmıştır. Ayrıca ödül şekillendirme, müfredat öğrenmesi ve deneyim tekrarı gibi stratejiler ile ajanların daha etkili öğrenmesini desteklemiştir. Bu yaklaşımlar, ajanların yalnızca sanal ortamlarda değil, gerçek dünyada da işlevsel olabilecek yetenekler geliştirmesini mümkün kılmıştır. Çalışmada, yapay zeka teknolojisinin oyun geliştirme, doğal afet yönetimi ve otonom sistemler gibi çeşitli uygulama alanlarındaki kullanımı sunularak değerlendirilmektedir. Sonuçlar, yapay zekanın güvenilirliği ve dayanıklılığı konusunda önemli adımlar atıldığını göstermektedir. Yapay zeka sınırlarının genişletilerek, insan yaşamını kolaylaştıracak ve birçok sektöre katkı sağlayacak ileri teknolojilerin geliştirilmesine ışık tutması hedeflenmektedir. Böylece, derin pekiştirmeli öğrenme tekniklerinin hem araştırma hem de uygulama alanındaki önemini vurgulayan kapsamlı bir çerçeve sunulmaktadır.

Anahtar Kelimeler – Derin Pekiştirmeli Öğrenme, Yapay Zeka Ajanları, Hayatta Kalma Simülasyonları, Ödül ve Ceza Sistemleri, Unity ML-Agents.

I. GİRİŞ

Yapay zeka ve oyun, günümüzde birbirini etkileyen ve geliştiren önemli disiplinler olarak öne çıkmaktadır. Oyunlarda yapay zeka kullanımı, 1950'lerde dama ve satranç gibi basit oyunlarla başlamış ve gelişerek, günümüzde gerçekçi NPC'ler (Non-player character, oyuncu olmayan karakter) oluşturma ve oyun dünyalarını simüle etme işlevlerini üstlenmektedir. Yapay zeka, oyun tasarımında üretken içerik oluşturma gibi yenilikçi uygulamalarla da dikkat çekmektedir [1].

Derin pekiştirmeli öğrenme, oyun ortamlarında karmaşık stratejilerin öğrenilmesi ve uygulanmasında kritik bir rol oynarken, bu süreçte yapay zeka uygulamaları, oyuncu olmayan karakterlerin davranışlarını

daha gerçekçi hale getirmek için kullanılmaktadır. Yapay zeka, oyun tasarımında yalnızca etkileşimli deneyimlerin oluşturulmasında değil, aynı zamanda içerik üretiminde de devrim niteliğinde yenilikler sunarak, geliştiricilerin daha yaratıcı ve dinamik oyunlar tasarlamasına olanak tanımaktadır. Bu bağlamda, derin pekiştirmeli öğrenmenin oyunlardaki uygulamaları, oyuncuların deneyimlerini zenginleştirmekle kalmayıp, aynı zamanda eğitim ve simülasyon gibi alanlarda da etkili bir araç olarak değerlendirilmektedir.

Günümüzde, yapay zeka teknolojilerinin hızla gelişmesiyle birlikte, yapay zeka ajanlarının gerçek dünya ortamlarında başarılı bir şekilde işlev görmesi ve hatta hayatta kalabilmesi giderek daha büyük bir önem kazanmaktadır [2]. Hayatta kalma oyunları genellikle oyuncuların doğal kaynakları toplamalarını, inşa etmelerini ve düşmanlardan kaçınmalarını veya onlarla savaşmalarını gerektirmektedir. Yapay zeka, bu tür oyunlarda oyuncuların karşılaşabileceği düşman davranışlarını, kaynak toplama stratejilerini ve oyun dünyasıyla etkileşimlerini simüle etmek için kullanılabilir.

Güncel bir çalışmada görüntü veya konuşma gibi yüksek boyutlu duyuşal girdiler derin pekiştirmeli öğrenme yöntemleri ile kullanılarak, modele doğrudan ajanları kontrol etmesi öğretilmektedir [3]. Peşin başarılı pekiştirmeli öğrenme uygulamaları, bu tür alanlarda genellikle elle hazırlanmış özelliklerle birleştirilmiş doğrusal değer fonksiyonları veya politika temsillerine dayanması sebebiyle gerçek dünya temsiliinde eksiklik göstermektedir.

Derin pekiştirmeli öğrenme modellerinin daha hızlı ve daha verimli bir şekilde eğitilmesi ve genişletilmesi için asenkron yöntemlerin etkin bir şekilde kullanıldığı başka bir çalışmada, daha karmaşık ve gerçekçi problemler üzerinde daha iyi performans gösteren yapay zeka sistemleri geliştirilmesinin mümkün olduğu ifade edilmiştir [4].

Silver vd. (2016), derin sinir ağları ve ağaç arama algoritması gibi ileri öğrenme tekniklerinin kullanılmasıyla, Go oyununu oynamak için daha etkili bir yaklaşım geliştirilmiştir [5]. Derin sinir ağları, oyun tahtasındaki durumu analiz etmek ve olası hamleleri tahmin etmek için kullanılırken, ağaç araması algoritması, bu tahminleri daha ileri seviyeye taşımak ve olası hamlelerin sonuçlarını değerlendirmek için kullanılmıştır.

Genel video oyunları için yapay zeka geliştirilmesinin amaçlandığı bir çalışmada, derin pekiştirmeli öğrenme teknikleri kullanılarak yapay zeka ajanlarının belirli bir görevi gerçekleştirmek için çevreleriyle etkileşimde bulunması ve deneyimlerden öğrenmelerini sağlayan bir makine öğrenimi yaklaşımı sunulmuştur [6].

Bu çalışmanın temel amacı, yapay zeka ajanlarını belirli bir ortamda hayatta kalmaya teşvik etmek ve bu süreçte karşılaşacakları çeşitli zorluklarla başa çıkabilmelerini sağlamaktır. Bu zorluklar, gerçek dünya senaryolarından esinlenmiş olabileceği gibi sanal ortamlarda da simüle edilebilen durumlardır. Bir robotun doğal afetler, engeller veya diğer tehlikelerle karşılaştığı bir ortamda hayatta kalabilme yeteneğinin geliştirilmesi üzerine odaklanılarak derin pekiştirmeli öğrenme algoritmaları ile çalışmanın temeli oluşturulmaktadır. Bu algoritmalar, yapay zeka ajanlarının ortama uyum sağlamak ve belirlenmiş hedefleri başarmak için optimal eylemleri öğrenmelerini sağlamaktadır. Öğrenme süreci genellikle bir ödül sistemiyle desteklenerek; ajanlar belirli görevleri başarıyla tamamladıklarında veya belirli zorlukları aştıklarında ödüllendirilmesi prensibine dayanmaktadır. Bu çalışma sayesinde, oyunlarda karşılaşılan zorluklara karşı hayatta kalma becerisini öğrenebilen bir yapay zeka tasarlanarak literatüre katkı sağlanması amaçlanmaktadır.

II. MATERYAL VE YÖNTEM

A. Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning – DRL)

Derin pekiştirmeli öğrenme (Deep Reinforcement Learning - DRL), derin öğrenme ve pekiştirmeli öğrenmenin birleşimi olarak tanımlanan bir yapay zeka alt alanıdır [7]. Bu yöntem, bir ajan (öğrenen sistem) ile çevresi arasında etkileşim kurarak, belirli hedeflere ulaşmak için en iyi eylemleri öğrenmesini sağlamaktadır.

Pekiştirmeli öğrenme, denetimli ve denetimsiz makine öğrenme yöntemlerinden farklı olarak, bir öğrenen makine olan ajanlar karşılaştığı durumlara tepki vermekte ve buna karşılık sayısal bir ödül almaktadır. Pekiştirmeli öğrenmenin en temel özelliği olan deneme yanılma yöntemi sayesinde ajan elde

edeceği ödülü maksimize etmeyi hedeflemektedir [8]. Pekiştirmeli öğrenme süreci genel adımları şu şekildedir [9].

Gözlemeleme: Model, çevresindeki ortamı gözlemleyerek bu ortam hakkında bilgi toplamaktadır.

Karar Verme: Model, mevcut gözlemlerine dayanarak bir eylem seçmektedir. Bu eylem, belirli bir hedefe ulaşmak için tasarlanmıştır.

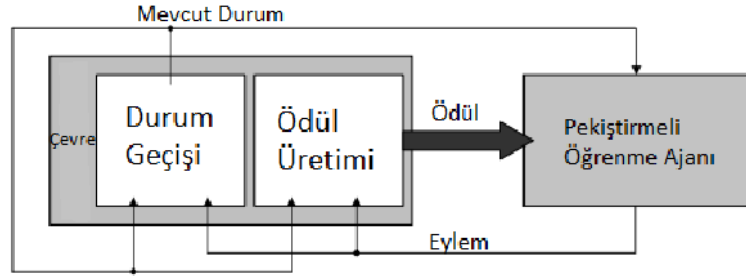
Eylemi Uygulama: Model seçilen eylemi gerçekleştirerek bu eyleme göre çevrede bir değişiklik oluşturmaktadır.

Geri Bildirim Alınması: Model, geri bildirim almaktadır. Bu geri bildirim, seçilen eylemin ne kadar başarılı veya başarısız olduğunu belirtmektedir.

Ödül veya Cezalandırma: Model, aldığı geri bildirim göre bir ödül veya ceza almaktadır. Eylem başarılıysa, bir ödül, başarısızsa bir ceza almaktadır.

Öğrenme: Model, aldığı geri bildirim dayanarak öğrenmektedir. Başarılı eylemleri teşvik eden davranışları öğrenirken, başarısız eylemlerden kaçınmayı öğrenmektedir.

Yeniden Gözlemeleme ve Yeniden Karar Verme: Model, güncel gözlemlerine dayanarak yeni bir eylem seçerek süreç tekrarlanmaktadır.



Şekil 1. Derin pekiştirmeli öğrenme modeli [8]

B. Makine Öğrenimi Ajanları (Machine Learning (ML) Agents)

ML Agents, Unity'nin yapay zeka ve makine öğrenimi modellerini eğitmek için geliştirilmiş bir araç setidir. Bu araç seti, Unity oyun motorunda oyun nesnelere kontrol etmek için kullanılan yapay zeka algoritmalarını ve modellerini eğitmek için bir dizi araç ve kütüphane sunmaktadır.

ML Agents, karmaşık oyun senaryolarında yapay zeka davranışlarını modelleme ve eğitme sürecini kolaylaştırmaktadır. Bu sürecin ilk adımı, yapay zeka modelini eğitmek için bir çevre tanımlamaktır. Bu çevre, oyun sahnesindeki nesnelere, hedeflere ve etkileşimlere içermektedir. Sonrasında oluşturulan çevrede yapay zeka modelini eğitmek için kullanılacak verilerin toplanması gerekmektedir. Bu genellikle insan ya da önceden belirlenmiş stratejilerle oluşturulmuş verileri içermektedir. Toplanan veriler kullanılarak yapay zeka modeli eğitilmektedir. ML Agents, çeşitli makine öğrenimi tekniklerini destekleyerek eğitim sürecini kolaylaştırmaktadır. Eğitilen yapay zeka modeli, Unity oyun motoruna entegre edilip gerçek zamanlı olarak oyun sahnesinde kullanılmaktadır.

C. Oyun Ortamının Hazırlanması

Çalışma kapsamında hazırlanacak dijital oyun Unity oyun motorunda yapay zeka tabanlı olarak geliştirilmiştir. Bu amaçla öncelikle temel oyun dünyası oluşturularak oyunun geçeceği dünyayı temsil eden bir arazi ya da ortam eklenip sahne oluşturulmaktadır. Çalışmada oyun dünyasını daha gerçekçi hale getirmek için gece gündüz döngüsü gibi dinamik ışıklandırma efektleri eklenmiştir.

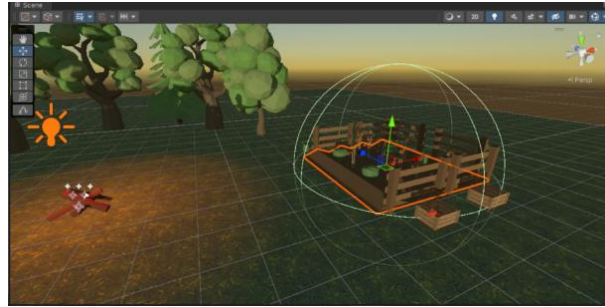


Şekil 2. Oyun ortamı oluşturma

Yapay zeka ile öğrenmenin gerçekleştirilebilmesi için temel unsurların da tanımlanması gerekmektedir. Oyunda yer alacak düşman karakterler tasarlanarak oluşturulmalıdır. Bu karakterler oyuna göre, oyuncuya karşı saldırabilir ya da onlardan kaçabilir şekilde tasarlanmaktadır. Düşmanların hareket ve davranışlarını belirleyen temel kodlar yazılarak yapay zeka algoritmaları kullanılmaktadır. Oyun içinde düşman yapay zekasını etkinleştirecek ve onların oyuncularla etkileşimini sağlayacak kodlar entegre edilmektedir.

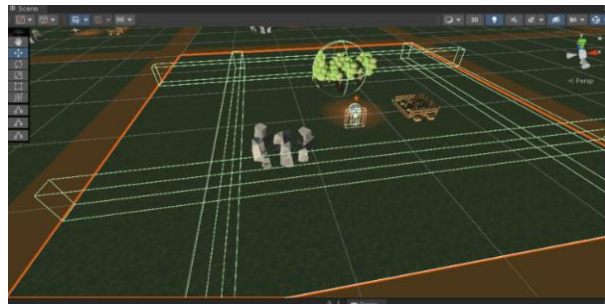
Modeli besleyecek kaynak ve yapıların da tanımlanması gerekmektedir. Oyuncunun kaynakları toplayabileceği ve bunları kullanarak yapılar inşa edebileceği bir sistem oluşturulmaktadır. Bu kaynaklar oyuna göre odun, taş, besin gibi materyalleri içerebilmektedir. Oyuncunun kaynaklarını kullanarak yapılar inşa etmesini ve bu yapılar aracılığıyla kaynakları işlemesini sağlayacak mekanikler tasarlanarak kodlanmıştır.

Ayrıca oyunda düşman ve oyuncu-yapı etkileşimleri de düşünülerek uygun yapay zeka kodları oluşturulmaktadır. Bu sayede düşmanların oyuncunun inşa ettiği yapıları hedef alması ya da onları yok etmeye çalışması, bu durumlarda oyuncunun savunması gibi durumlar için öğrenme adımları eklenmektedir.



Şekil 3. Oyun ortamı

Ödül şekillendirme (reward shaping), pekiştirmeli öğrenme alanında kullanılan ve bir ajanın öğrenme sürecini hızlandırmak için ödül fonksiyonunun modifiye edilmesini ifade eden bir tekniktir [10]. Bu teknik, ajanın istenen davranışları öğrenmesine yardımcı olmak amacıyla ek ödüller sağlayarak, öğrenme sürecini daha etkili hale getirmektedir. Bu teknik, ajanın istenen davranışları daha hızlı öğrenmesini sağlayabilir, ancak dikkatli uygulanmalıdır çünkü yanlış bir şekilde kullanıldığında ajanın performansını olumsuz olarak etkileyebilmektedir.



Şekil 4.Ödül Ceza Sistemi

Curriculum Learning, makine öğrenimi ve yapay zeka alanında, bir modelin eğitim sürecinde daha basit görevlerden başlayarak giderek daha karmaşık görevlere geçiş yapmasını ifade etmektedir [11]. Bu yaklaşım, insanların genellikle temel kavramları öğrenerek başlayıp zamanla daha karmaşık konulara geçmesini temel alan insan öğrenme süreçlerinden esinlenmiştir. Bu, ajanın daha hızlı ve daha kararlı bir şekilde öğrenmesini sağlamaktadır.

Çalışmada bu amaçla hazırlanan örnek kod aşağıda yer almaktadır.

```
private void OnTriggerEnter(Collider other) {
    if(other.gameObject.tag=="odun") {
        AddReward(10f); EndEpisode();
    } if(other.gameObject.tag=="wall") {
        AddReward(-5f); EndEpisode();
    }
}
```

Proximal Policy Optimization (PPO), pekiştirmeli öğrenme alanında kullanılan bir optimizasyon algoritmasıdır. PPO, özellikle derin pekiştirmeli öğrenme uygulamalarında yaygın olarak tercih edilmektedir [12]. Bu algoritma, on-policy (politika tabanlı) bir yöntemdir ve hem kesikli hem de sürekli eylem alanlarıyla çalışabilmektedir. Ayrıca genellikle büyük ölçekli eğitim setleri üzerinde etkili bir şekilde çalışmaktadır. Stabil ve hızlı bir şekilde ajanları eğitmek için tasarlanan Policy Gradient metodlarının bir türü olan PPO çalışmaya dahil edilmiştir.

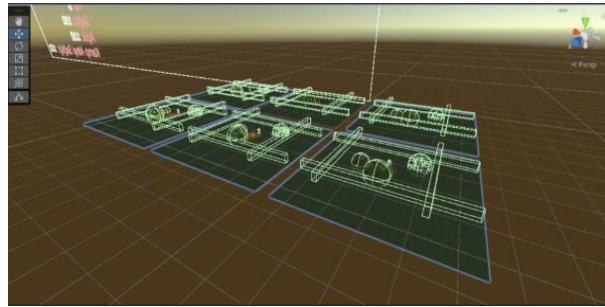
D. Ajanların Eğitimi ve Geliştirilmesi

Ajanların daha hızlı, stabil ve daha verimli bir şekilde eğitilmesini sağlamak için çalışmada bazı gelişmiş eğitim tekniklerinden faydalanılmıştır.

ML-Agents eğitim sürecinde kullanılan birçok hiper parametre bulunmaktadır ve öğrenme hızı, eğitim devri sayısı, ödül şekillendirme faktörleri gibi bu parametrelerin dikkatli bir şekilde ayarlanması önemlidir.

Literatürde ajanların eğitim verimliliğini artırmak için kullanılan birçok teknik mevcuttur. Bunlar arasında deneyim tekrarı, örnek çeşitliliği sağlama, eğitim verilerinin dengelemesi gibi teknikler bulunmaktadır. Bu teknikler, ajanların daha hızlı ve daha stabil bir şekilde öğrenmelerini sağlamaktadır.

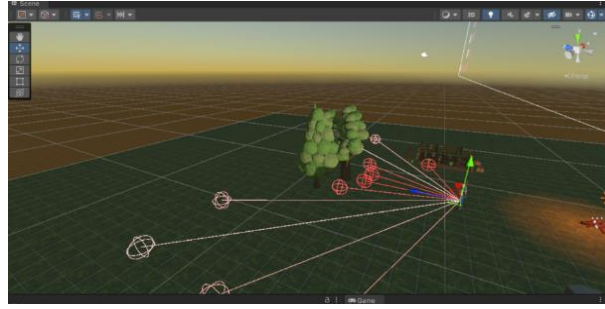
ML-Agents, paralel eğitim modunu desteklemektedir. Bu, birçok ajanın aynı anda eğitilmesini ve eğitim sürecinin hızlandırılmasını sağlamaktadır. Paralel eğitim, özellikle büyük ölçekli eğitim setleri üzerinde çalışırken çok faydalıdır ve eğitim süresini önemli ölçüde azaltmaktadır [13].



Şekil 5. Eğitim örneği

Unity ML-Agents, yapay zeka ajanlarının çevrelerini algılamasına yardımcı olmak için ışın algılama sensörü (Ray Perception Sensor) adında bir algı sensörü sunmaktadır. Bu sensör, ajanın çevresini ışınlarla algılamasına olanak tanımaktadır. Sensör, ajanın önünde bir ışın demeti oluşturur ve bu ışınlar çevredeki nesnelerin konumunu ve türünü belirlemek için kullanılmaktadır. Özellikleri belirlenerek algılama için kullanılan ışınların sayısı ve yayılma açısı ayarlanabilir, ışınların ne kadar uzaklıkta algılama yapacağı belirlenebilir ve sadece belirli nesnelerin algılanması için farklı katmanlar oluşturulabilir. Çalışmada bu sayede bir ajanın önünde duvarlar, engeller veya diğer nesnelere varsa, bu nesnelerin varlığını tespit

edilmektedir. Unity içindeki bir oyun nesnesi olarak kullanılan bu sensör ajanın vücudu üzerinde veya ayrı bir nesne olarak yerleştirilebilmektedir. Şekil 6'da yer aldığı üzere daha sonra ML-Agents kodunda bu sensöre erişilerek çevre bilgileri alınmaktadır.



Şekil 6. Işın algılama sensörü

Natural Locomotion, Unity ML-Agents projesindeki bir bileşendir. Çalışmada kullanılan bu bileşen, ajanların örneğin, bir insanın yürümesi gibi, ajanların da adımlarını belirli bir hız ve doğrultuda atmalarını sağlayarak doğal bir şekilde hareket etmelerine olanak tanımaktadır [14].



Şekil 7. Uygulama ekranı

Çalışmada bir ajanın hazırlanan oyun çevresinde zorlukları yapay zeka destekli öğrenerek hayatta kalma süresinin uzatılması amaçlanmıştır. Q-Learning ve SARSA gibi daha eski yöntemlere göre genellikle daha hızlı ve etkili öğrenen Proximal Policy Optimization (PPO) ve Advantage Actor-Critic (A2C/A3C) gibi algoritmalar kullanılarak, öğrenme süreci hızlandırılmıştır. Ayrıca uygulamada hiper parametre optimizasyonu da yapılmıştır. Öğrenme oranı, bir toplu işteki örnek sayısı (batch size) uygun seçilerek, simülasyon hızı için daha hızlı çalışan simülasyon ortamları kullanılarak ajanın daha fazla deneyim sahibi olması sağlanmıştır.



Şekil 8. En iyi gün ve ödül

Çalışmada ajanın geçmiş deneyimlerini yeniden kullanarak öğrenmesini sağlamak ve öğrenme sürecini hızlandırmak için pekiştirmeli öğrenme alanında kullanılan tekniklerinden biri olan deneyim saklama (experience replay) yönteminden faydalanılmıştır. Bu teknik, bir ajanın her zaman adımında (state-action-reward-next state) elde ettiği deneyimleri saklayarak, bu deneyimlerin daha sonra tekrar kullanılmasını sağlamaktadır [15]. Temel olarak yöntem, ajanın geçmişteki deneyimlerini bir hafıza (replay memory)

içinde depolayarak, bu deneyimlerden öğrenmesini sağlamak amacıyla kullanılmaktadır. Buna göre Şekil 9'da gösterildiği gibi en iyi gün ve ödül olumlu etkilenmiştir.



Şekil 9. Deneyimler sonrası en iyi gün ve ödül

III. SONUÇ VE DEĞERLENDİRME

Çalışmada derin pekiştirmeli öğrenme teknikleri kullanılarak yapay zeka ajanlarının belirli ortamlarda hayatta kalma yeteneklerinin nasıl geliştirilebileceği gösterilmektedir. Bu teknikler, yapay zekanın gerçek dünya senaryolarında daha etkili bir şekilde kullanılmasına yardımcı olmaktadır. Aynı zamanda yapay zeka ajanlarının doğal afetler, engeller veya diğer tehlikeler gibi zorlu durumlarla nasıl başa çıkabileceğini inceleyerek, bu tür senaryolar için model geliştirmenin önemi de vurgulanmaktadır. Bu, gerçek dünya uygulamalarında kullanılabilir önemli bir yetenektir.

Derin sinir ağları ve pekiştirmeli öğrenme algoritmaları hibrit bir şekilde kullanıldığı bu çalışmada oluşturulan yapay zeka modelinin eğitim sürecinin nasıl optimize edilebileceği gösterilmektedir. Bu algoritmalar, yapay zeka ajanlarının çevresine uyum sağlamasını ve belirli hedefleri başarmasını sağlamak için en uygun eylemleri öğrenmelerine yardımcı olabilmektedir. Ayrıca çalışmada Unity ML Agents aracılığıyla yapay zeka modellerinin nasıl eğitilebileceği ve gerçek zamanlı simülasyonlarda nasıl kullanılabileceği gösterilmektedir. Bu, özellikle oyun geliştirme alanında yapay zeka teknolojilerinin nasıl entegre edilebileceğini ve geliştirilebileceğini göstermektedir. Kullanılan ödül sistemi ise, yapay zeka ajanlarının istenilen davranışları öğrenmesini teşvik etmektedir. Bu tür teşvik sistemleri, yapay zeka modellerinin belirli görevleri başarıyla tamamlamasını sağlayarak, performanslarını artırabilir ve daha geniş uygulama alanları için adapte edilebilir.

Bu çalışmanın literatürde yapay zeka alanında derin pekiştirmeli öğrenme tekniklerinin pratik uygulamalarını gösteren ve yapay zeka ajanlarının gerçek dünya senaryolarında nasıl geliştirilebileceğini araştıran önemli bir çalışma olarak öne çıkması hedeflenmektedir. Bu sayede, gelecekte yapay zeka teknolojilerinin günlük hayatta ve endüstride daha geniş alanlarda nasıl kullanılabilirliğini anlamamıza yardımcı olabilecektir.

KAYNAKLAR

- [1] K. Hartness, "ROBOCODE: Using Games to Teach Artificial Intelligence," *Journal of Computing Sciences in Colleges*, vol. 19, no. 4, pp. 287–291, 2004.
- [2] S. T. Jadhav, S. M. Pathan, D. S. Revgade, R. H. Chivade, and M. R. Bendre, "AI in Game Development," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 06, no. 05, pp. 7716–7719, 2024.
- [3] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning," *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pp. 135–160, 2020.
- [4] V. Mnih *et al.*, "Asynchronous Methods for Deep Reinforcement Learning," in *Proceedings of 33rd International Conference on Machine Learning*, 2016, vol. 48, pp. 1928–1937.
- [5] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [6] R. R. Torrado, P. Bontrager, J. Togelius, J. Liu, and D. Perez-Liebana, "Deep Reinforcement Learning for General Video Game AI," *IEEE Conference on Computational Intelligence and Games, CIG*, pp. 1–8, 2018.

- [7] P. V. Rao *et al.*, “Deep Reinforcement Learning: Bridging the Gap with Neural Networks,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 15s, pp. 576–586, 2024.
- [8] O. Engin and S. Bünyamin, “Makine Çizelgeleme Problemlerinin Çözümünde Pekiştirmeli Öğrenme Etkisinin Analizi,” *ALKÜ Fen Bilimleri Dergisi*, vol. 6, no. 2, pp. 116–140, 2024.
- [9] H. Guo *et al.*, “Deep Reinforcement Learning for Dynamic Algorithm Selection: A Proof-of-Principle Study on Differential Evolution,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 7, pp. 4247–4259, 2024.
- [10] H. Ma, K. Sima, T. V. Vo, D. Fu, and T. Y. Leong, “Reward Shaping for Reinforcement Learning with An Assistant Reward Agent,” *Proceedings of Machine Learning Research*, vol. 235, pp. 33925–33939, 2024.
- [11] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 41–48.
- [12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 1–12.
- [13] “Training your agents 7 times faster with ML-Agents.” [Online]. Available: <https://unity.com/blog/engine-platform/training-your-agents-7-times-faster-with-ml-agents>.
- [14] C. Yang, K. Yuan, S. Heng, T. Komura, and Z. Li, “Learning Natural Locomotion Behaviors for Humanoid Robots Using Human Bias,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2610–2617, 2020.
- [15] D. Zha, K. H. Lai, K. Zhou, and X. Hu, “Experience Replay Optimization,” in *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019, vol. 2019-Augus, pp. 4243–4249.