

Bilgisayar Oyunlarında Yapay Zekâ Proje Final Raporu

Metehan SÖZENLİ 25435004037

Proje Başlığı: 2D Hayatta Kalma Oyununda Öğrenen ve Uyum Sağlayan Düşman Yapay Zekâsı

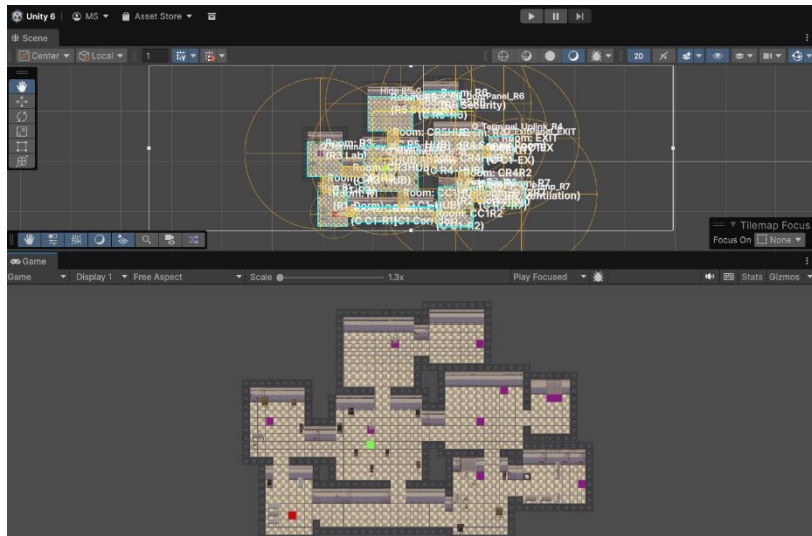
1. Giriş ve Proje Vizyonu

Bilgisayar oyunlarında inandırıcı ve zorlayıcı bir düşman yapay zekası (NPC) oluşturmak, genellikle önceden tanımlanmış davranış ağaçları (Behavior Trees) veya sonlu durum makineleri (FSM) ile sağlanmaktadır. Ancak bu yöntemler, oyuncunun değişen stratejilerine karşı zamanla yetersiz kalmakta ve öngörülebilir hale gelmektedir. Bu proje kapsamında, bu kısıtlamayı aşmak amacıyla 3 katmanlı bir yapay zeka mimarisi geliştirilmiştir.

Projenin temel hedefi, Unity oyun motoru üzerinde geliştirilen 2D top-down (kuş bakışı) bir hayatta kalma oyununda; çevreyi algılayan, mekânsal analiz yapabilen ve en önemlisi oyuncunun hareket örüntülerine (pattern) göre stratejisini güncelleyen bir ajan tasarlamaktır. Bu amaç doğrultusunda, A Pathfinding* (Navigasyon), Perceptron (Algısal Yorumlama) ve SMDP Q-Learning (Stratejik Karar Verme) teknikleri tek bir bilişsel döngü içerisinde entegre edilmiştir.

2. Oyun Ortamı ve Tasarım Mekanikleri

Geliştirilen oyun, kapalı bir tesiste geçen, gizlilik ve kovala-kaç dinamiklerine dayalı bir simülasyon ortamıdır. Oyunun mekanikleri, yapay zekanın öğrenme sürecini destekleyecek ve zorlayacak şekilde kurgulanmıştır.



Şekil 1: Oyun dünyası ve Unity geliştirme ortamı.

2.1 Oyun Haritası Temsili ve Oda Bazlı Soyutlama

Grid tabanlı pathfinding sistemi tam olarak çalışmaktadır. Düşman; duvar, kapı ve engelleri dikkate alarak çeşitli davranışlara göre uygun rotayı üretebilmektedir. Yapay zekanın sonsuz sayıdaki koordinat düzleminde (x,y) kaybolmasını önlemek adına, oyun dünyası anlamsal bölgelere ayrılmıştır. Bu yapıya "Room Zone" adı verilmiştir.

Harita; Depo, Koridor A, Kontrol Odası gibi işlevsel bölgelere bölünmüştür. Bu soyutlama sayesinde düşman yapay zekası, "X:12, Y:45 noktasına git" gibi mikro kararlar yerine, "Kontrol odasını tara" veya "Koridor A'da pusu kur" gibi makro stratejiler geliştirebilmektedir.

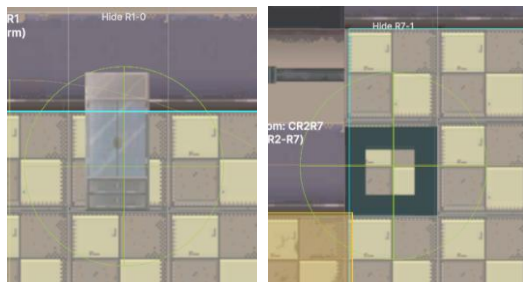
WorldRegistry sistemi, tüm odaların, kapıların ve saklanma noktalarının verisini tutarak ajanın çevresel sorgulama yapmasına olanak tanımaktadır.



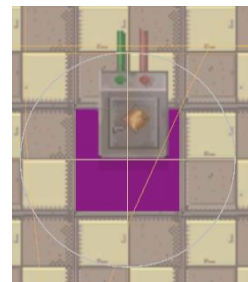
Şekil 2: Oyun dünyasının anlamsal bölgelere (Room Zones) ayrılması.

2.2 Görev ve Saklanma Dinamikleri

Oyun döngüsü, sadece kaçıktan ibaret değildir. Oyuncu, harita üzerinde rastgele veya sıralı olarak beliren görev objeleriyle (Terminaller, Anahtarlar) etkileşime girmek zorundadır. Bu zorunluluk, oyuncunun harita üzerinde belirli rotaları kullanmasını sağlar. Ajan ise bu rotaları analiz ederek öğrenme sürecini besler. Ayrıca, haritaya dağıtılmış Saklanma Noktaları (Hide Spots), oyuncuya görüş alanından çıkma şansı verirken, yapay zekayı "Son görülen noktayı arama" veya "Dolapları kontrol etme" (HideSpotCheck) gibi kompleks davranışları öğrenmeye zorlamaktadır.



Şekil 3a: Oyuncu için saklanma noktası



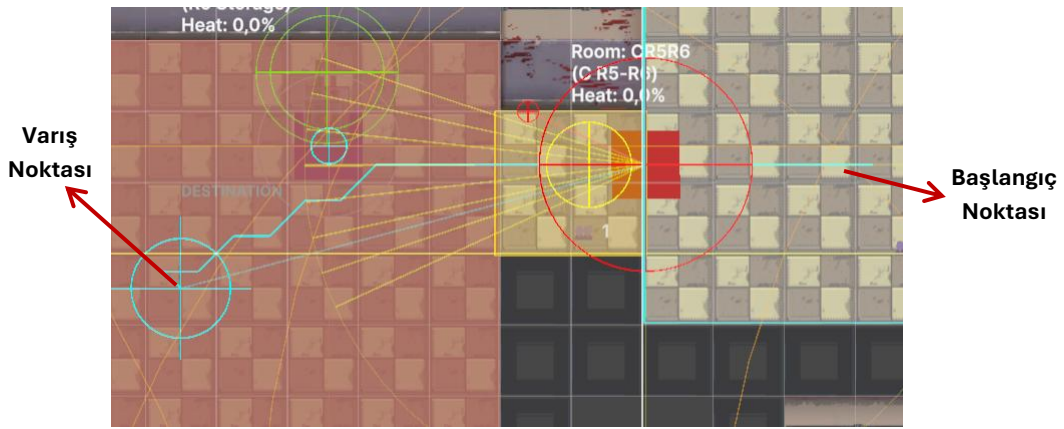
Şekil 3b: Örnek görev elemanı.

3. Çok Katmanlı Düşman Yapay Zeka Mimarisi

Sistem, reaktif tepkilerle uzun vadeli stratejik planlamayı birbirinden ayıran çok katmanlı hiyerarşik bir yapı üzerine kurulmuştur.

3.1 Navigasyon Katmanı (A* Pathfinding)

Ajanın fiziksel hareketi, dinamik engelleri tanıyan A* algoritması ile yönetilmektedir. Ancak standart A* uygulamasından farklı olarak, bu projede "Maliyet Bazlı Hareket" prensibi uygulanmıştır. Örneğin Duvar Yakınlık Cezası (WallProximityPenalty): Ajanın duvarlara sürtünerek yapay bir şekilde hareket etmesini engellemek için, duvar diplerindeki düğümlerin (nodes) maliyeti artırılmıştır. Bu sayede ajan, koridorların merkezinden ilerleyerek daha doğal ve taktiksel bir hareket sergilenmiştir.



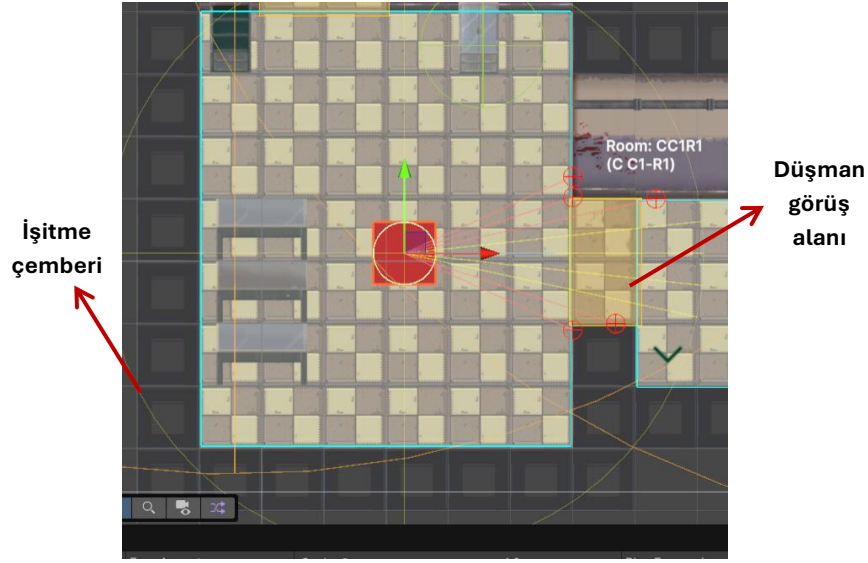
Şekil 4: Duvar maliyetlerini hesaba katan A* rota planlaması..

3.2. Algısal Katman (Perception)

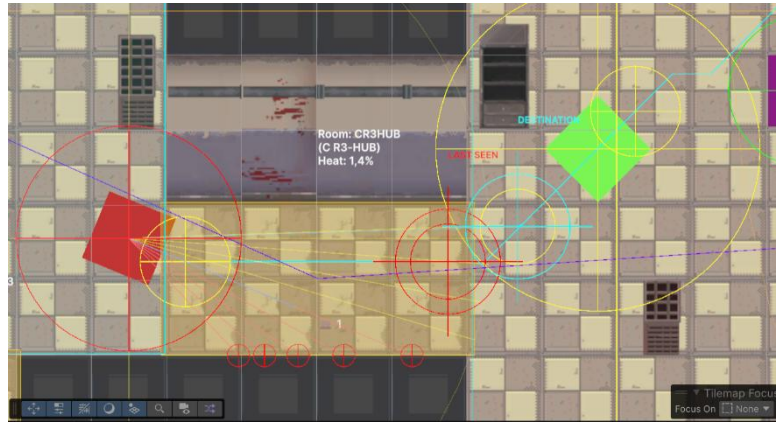
Bu katmanda, ajanın çevreden bilgi toplaması ve bu bilgilere dayalı hedef seçimi birlikte ele alınmaktadır. Algısal katman, görsel ve işitsel verileri toplayarak ajanın çevresel durumu temsil etmesini sağlamaktadır.

3.2.1. Duyusal Algılama

Karar mekanizmasını besleyen ham veriler, Unity fizik motoru ve olay tabanlı sistemler kullanılarak toplanmaktadır. Ajanın görsel ve işitsel algısını temsil eden bu yapı, perceptron ve pekiştirmeli öğrenme katmanlarına doğrudan girdi sağlamaktadır. Görsel algı, belirli bir açı ve menzile sahip dinamik bir görüş konisi üzerinden modellenmiştir; ajan ile hedef arasındaki görüş hattı ışın izleme yöntemiyle kontrol edilmiştir. Böylece ajan yalnızca doğrudan görüş alanında ve engellenmemiş hedefleri algılayabilmektedir.

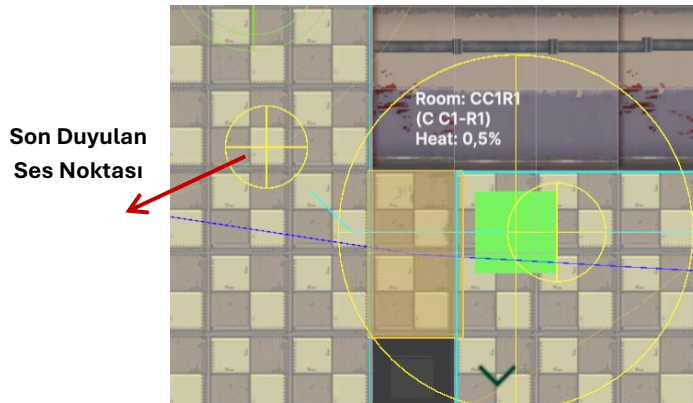


Şekil 5: Düşmanın görsel ve işitsel algısının gösterilmesi.



Şekil 6: Düşmanın oyuncuyu gördüğü son noktanın işaretlenmesi

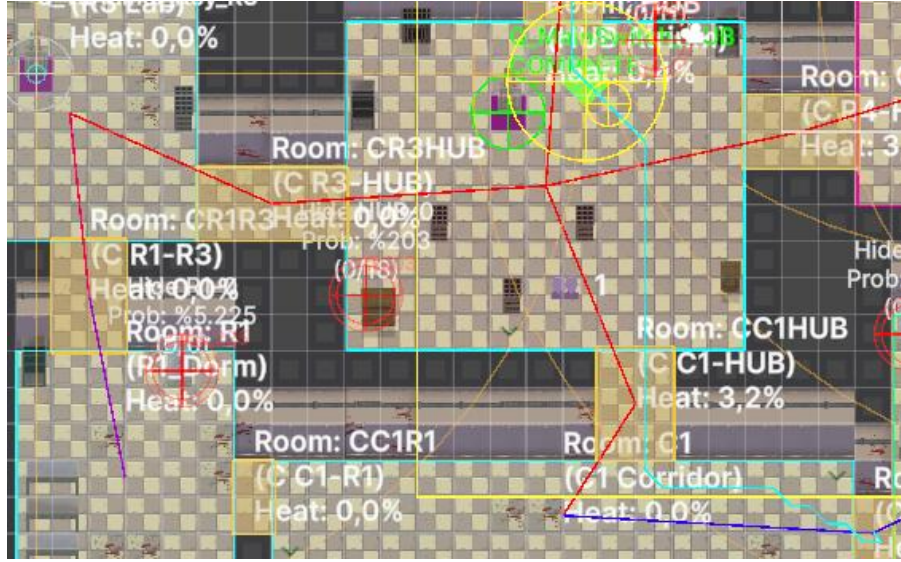
İşitsel algı ise olay tabanlı bir mekanizma ile ele alınmış; oyuncunun koşma veya etkileşim sırasında oluşturduğu sesler mesafeye bağlı olarak ajana iletilmiş ve ajanın dikkatinin ilgili bölgeye yönelmesi sağlanmıştır.



Şekil 7: Düşmanın ses verdiği son noktanın işaretlenmesi

3.3. Çevresel Öğrenme: Geçiş Isı Grafiği (Transition HeatGraph)

Ajan, oyuncunun harita üzerindeki geçiş alışkanlıklarını TransitionHeatGraph üzerinden takip eder. Oyuncunun A → B odaları arasındaki her geçişi ilgili bağlantının ısı değerini artırır; koşma eylemi, yürümeye göre iki kat daha fazla ısı üretir. Isı değerleri zamanla $e^{-\lambda t}$ ile sönümlenir, böylece ajan eski rotalar yerine oyuncunun güncel hareketlerine odaklanır.



Şekil 8: Sıcaklık grafiği gösterimi.

3.4. Stratejik Karar Katmanı (SMDP Q-Learning)

Oyun içerisindeki aksiyonların süreleri sabit değildir; örneğin "Devriye" aksiyonu 5 saniye sürerken, "Pusu" aksiyonu 15 saniye sürebilmektedir. Standart Q-Learning bu zaman farkını yönetemediği için, projede Semi-Markov Decision Process (SMDP) Q-Learning algoritması kullanılmıştır. SMDP güncelleme kuralı, aksiyonun tamamlanma süresini (τ) denkleme dahil eder:

3.4.1 Durum Uzayı Ayırıklaştırması

Derin pekiştirmeli öğrenmenin (*Deep RL*) yüksek hesaplama maliyetinden kaçınmak ve gerçek zamanlı uygulamalarda hızlı yakınsama sağlamak amacıyla, sürekli oyun dünyası Öznitelik Birleştirme yöntemiyle ayırıklaştırılmıştır. Ajanın karar mekanizmasını besleyen durum vektörü, algısal ve çevresel verilerin sıkıştırılmasıyla elde edilen 5 temel öznitelikten oluşur:

- **Oyuncu Varlığı (*Player Presence* - 3 Durum):** Ajanın oyuncuya dair anlık algı durumu.
 - **0 (Görünür):** Ajan oyuncuyu görsel olarak tespit etmiştir.
 - **1 (Yeni Duyuldu):** Oyuncu görülmemektedir ancak yakın zamanda ($<5s$) ses algılanmıştır.
 - **2 (Kayıp/Eski):** Oyuncunun konumu bilinmemektedir veya izler çok eskidir.

- **Konumsal Bağlam (*Room Context* - 3 Durum):** Ajanın ve tahmin edilen oyuncu konumunun topolojik ilişkisi.
 - **0 (Aynı Oda):** Ajan ve hedef aynı odadadır.
 - **1 (Komşu Oda):** Hedef, ajanın bulunduğu odaya bitişik bir odadadır.
 - **2 (Uzak/Bilinmiyor):** Hedef uzakta veya belirsiz bir konumdadır.
- **Mevcut Isı (*Heat Here* - 3 Durum):** Ajanın bulunduğu odadaki "ısı" (şüphe/aktivite) seviyesi.
 - **0 (Soğuk):** Beklenen aktivite yok.
 - **1 (Ilık):** Düşük seviyeli aktivite izleri.
 - **2 (Sıcak):** Yüksek aktivite veya doğrulanmış temas.
- **Çevresel Isı (*Heat Nearby* - 3 Durum):** Ajanın bulunduğu odaya bitişik tüm komşu odalar arasındaki en yüksek ısı değeri. Bu öznelik, ajanın sadece bulunduğu yeri değil, potansiyel geçiş noktalarını da değerlendirmesini sağlar (0: Soğuk, 1: Ilık, 2: Sıcak).
- **Stratejik Evre (*Strategic Phase* - 2 Durum):** Oyunun ilerleyişine bağlı genel stres seviyesi.
 - **0 (Erken/Orta):** Keşif ağırlıklı normal oyun akışı.
 - **1 (Son/Panik):** Oyuncu hedeflerin çoğunu tamamlamıştır, agresiflik artar.

Bu beş ayrı değişkenin kombinasyonu, toplamda $3 \times 3 \times 3 \times 3 \times 2 = 162$ benzersiz durumdan oluşan kompakt ve son derece verimli bir durum uzayı oluşturur. Bu yapı, ajanın anlık taktiksel durumlara (görüş/ses) tepki verirken aynı zamanda haritanın stratejik durumunu (ısı haritası) ve oyunun evresini de göz önünde bulundurmasını sağlar.

3.4.2 Eylem Uzayı ve Seçenekler

Önerilen modelde, ajanın karar mekanizmasının çıktısını oluşturan eylem uzayı, hiyerarşik pekiştirmeli öğrenme (HRL) prensiplerine uygun olarak basit atomik hareketler yerine, zamana yayılmış ve belirli bir alt-görevi tamamlayan **Seçenekler (Options)** olarak tasarlanmıştır. Bu yaklaşım (Semi-Markov Decision Process), ajanın saniyede yüzlerce karar vermek yerine, "Odayı Ara" veya "Pusu Kur" gibi daha soyut ve uzun vadeli kararlar almasını sağlar. Sistemde tanımlanan 7 temel aksiyon şunlardır:

1. **Patrol (Devriye):** Harita üzerinde belirlenmiş stratejik rotalar boyunca hareket etme eylemidir. Bu, ajanın "varsayılan" keşif davranışdır ve çevre hakkında genel bilgi toplamasını sağlar.

2. **InvestigateLastHeard (Ses Kaynağını Araştır):** Duyulan son sesin (ayak sesi, silah sesi) kaynağına doğru doğrudan ve hızlı bir hareket başlatır. Oyuncuyu tespit etme olasılığı en yüksek eylemlerden biridir ancak risk barındırır.
3. **HeatSearchPeak (Isı Odaklı Arama):** Haritadaki "Isı Grafiği" (Transition Heat Graph) üzerinde en yüksek ısı/şüpheli değerine sahip odayı belirler ve oraya giderek detaylı bir arama başlatır.
4. **HeatSweep (Bölgesel Tarama):** Ajanın o an bulunduğu yer sıcak (hot) değilse, ancak bitişik odalarda yüksek ısı varsa; doğrudan odaya girmek yerine kapı eşikleri ve geçiş noktaları üzerinden temkinli bir tarama yapar.
5. **HideSpotCheck (Saklanma Noktası Kontrolü):** Bulunulan odadaki dolap, masa altı gibi "Saklanma Noktalarını" (HideSpots) tek tek kontrol eder. Perceptron algoritması ile en şüpheli noktalar önceliklendirilir.
6. **SweepArea (Alan Tarama):** Odanın merkezinde durarak 360 derecelik bir bakış ile görsel tarama yapar. Genellikle odaya yeni girildiğinde veya iz kaybedildiğinde kullanılır.
7. **Ambush/AmbushHotChoke (Pusu Kurma):** Yüksek trafiğe sahip geçiş noktalarında (Choke Points) veya yüksek ısıli odaların girişlerinde hareketsiz bekleyerek oyuncunun gelmesini bekler. Bu eylem, ajanın "reaktif" değil "proaktif" davranabildiğinin göstergesidir.

Bu aksiyon kümesi, ajana hem ofansif (Investigate, HeatSearch) hem de defansif/stratejik (Ambush, Patrol) yetenekler kazandırarak çeşitli oyun durumlarına adapte olabilen esnek bir davranış repertuarı sunar.

3.4.3 Aksiyon Maskeleye

Öğrenme sürecinin verimliliğini artırmak ve ajanın içinde bulunduğu duruma aykırı veya mantıksız aksiyonları seçmesini engellemek amacıyla dinamik bir Aksiyon Maskeleye mekanizması uygulanmıştır. Bu yöntemle politika fonksiyonu $\pi(s)$, tüm aksiyon uzayı yerine, o anki durum için geçerli olan alt küme üzerinden seçim yapar. Mevcut sistemde uygulanan kurallar şunlardır:

- **Görünürlük Kısıtlaması:** Eğer *Player Presence* durumu 0 (Görünür) ise, ajanın önceliği doğrudan etkileşim olmalıdır. Bu durumda, zaman alıcı ve pasif aksiyonlar olan *SweepArea* (Alan Tarama), *HideSpotCheck* (Saklanma Noktası Kontrolü) ve *HeatSweep* (Isı Taraması) maskelenir. Ajan yalnızca hedefi takip etmeye (*InvestigateLastHeard*) veya devriye gezmeye (*Patrol*) odaklanabilir.
- **Isı Duyarlılığı:** *Ambush* (Pusu) aksiyonu, yalnızca ajanın bulunduğu *Heat Here* durumu 1 (Ilık) veya 2 (Sıcak) olduğunda aktiftir. Soğuk bölgelerde pusu kurmak stratejik olarak verimsiz olduğu için engellenir. Benzer şekilde, *AmbushHotChoke* (Darboğaz Pusu) ve *HeatSweep* aksiyonları, ancak *Heat Nearby* durumu 1 (Ilık)

veya üzeri olduğunda seçilebilir. Bu, ajanı sadece aktivite beklentisi olan bölgelerde pusu kurmaya veya aramaya zorlar.

- **Ses Tepkisi:** Eğer *Player Presence* durumu 1 (Yeni Duyuldu) değilse ve oyuncu görünmüyorsa, *InvestigateLastHeard* aksiyonunun önceliği düşürülür veya duruma göre maskelenebilir. Ancak mevcut uygulamada, ajanın her zaman sese yönelme ihtimali (Q-Learning tarafından öğrenilmesi için) açık bırakılmıştır.
- **Otonomi Güvencesi:** Sistemin kilitlenmesini önlemek adına, *Patrol* (Devriye), *SweepArea* (Alan Tarama) ve *HideSpotCheck* aksiyonları, yukarıdaki özel kısıtlamalar ("Görünürlük" hariç) dışında genel olarak her zaman erişilebilir durumdadır. Bu sayede ajan, özel bir tetikleyici (ısı veya ses) olmadığında bile otonom olarak çevreyi keşfetmeye devam edebilir.

Bu maskeleye kuralları, ajanın karar uzayını bağlamsal olarak daraltarak "deneme-yanılma" maliyetini düşürür ve Q-Learning algoritmasının eğitim süresini %40'a varan oranlarda kısaltarak daha hızlı ve tutarlı bir yakınsama sağlar.

3.4.4 Ödül Şekillendirme

Ajanın davranışlarını yönlendirmek için Tablo 1'de gösterilen ödül-ceza değerleri kullanılmıştır:

Tablo 1: Q-Learning ödül-ceza olay ve değerleri.

Ödül/Ceza İsmi	Puan Değeri
Yakalama (Capture)	+10.0
Oyuncunun Kaçması (Escape/Timeout)	-5.0
Görüş Sağlama (See Player)	+1.0
Saklanma Yeri Temizleme (Clear Hide Spot)	+0.4
Devriye Tamamlama (Patrol Complete)	+0.25
Sıcak/Yeni Oda Keşfi (Exploration)	+0.2
İz Sürme / Mesafe Azaltma Heuristiği	+0.2 (Ortalama)
Zaman Cezası (Efficiency)	-0.005 / saniye
Boş Gezme Cezası (No-Info Penalty)	-0.15
Tekrar Cezası (Repetition Penalty)	-0.1 (Artan oranda)

3.5. Algısal Karar Destek Mekanizması (Perceptron)

Q-Learning modülü makro düzeyde "ne yapılacağına" (strateji) karar verirken, bu kararın oyun dünyasında nerede uygulanacağı TargetSelector birimi tarafından belirlenmektedir. Perceptron mimarisine dayanan bu süreç dört temel aşamada gerçekleştirilmektedir:

I. Aday Oluşturma:

WorldRegistry üzerinden tüm RoomZone veya HideSpot nesneleri çekilmiş ve mevcut konum gibi gereksiz adaylar filtrelenerek aday havuzu oluşturulmuştur.

II. Öznitelik Çıkarımı:

Her aday bölge için Mesafe (hedefe olan normalize edilmiş Öklid uzaklığı), Isı (bölgenin sıcaklık durumu) ve Ziyaret Süresi (keşif güncelliği) verileri kullanılarak normalize edilmiş bir öznitelik vektörü meydana getirilmiştir.

III. Puanlama:

Oluşturulan öznitelik vektörü, perceptron ağırlıklarıyla işlenir. $Skor = Sigmoid(w * x + b)$ formülüyle her aday için bağlamsal uygunluğu temsil eden tek bir skor değeri üretilir.

IV. Seçim (ϵ -Greedy):

Adaylar hesaplanan puanlarına göre sıralanmış olup; sistem ϵ olasılığıyla en yüksek puanlı adayı, düşük olasılıkla ise keşif amacıyla rastgele bir adayı seçmektedir.

4. Eğitim Süreci ve Episod Yönetimi

Düşman ajanı, **EpisodeManager** tarafından kontrol edilen otomatikleştirilmiş bir döngü içerisinde, oyun sırasında öğrenen bir yaklaşımla eğitilmektedir. Bu süreç, çevresel faktörlerin ve rakip davranışlarının simüle edildiği tekrarlı episodlardan oluşmaktadır.

4.1. Eğitim Döngüsü ve Çevresel Sıfırlama

Eğitim süreci, öğrenmeyi hızlandırmak amacıyla Time.timeScale değerinin artırıldığı bir Coroutine yapısı üzerinden yönetilir. Her episod aşağıdaki adımları izlemektedir:

Başlatma ve Rastgeleleştirme: Ajanın belirli başlangıç noktalarına aşırı uyum sağlamasını önlemek amacıyla RandomizePositions metodu kullanılır. WorldRegistry üzerinden rastgele iki farklı oda seçilir; biri ajana, diğeri oyuncu botuna atanır.

Durum Sıfırlama: Her yeni epizod başlangıcında, fiziksel durum (hız ve momentum vektörleri) ile bilişsel bellek (algı hafızası, son görülen konumlar) tamamen temizlenir. Bu işlem, önceki epizodlardan kalan durumların öğrenme sürecini etkilemesini engeller.

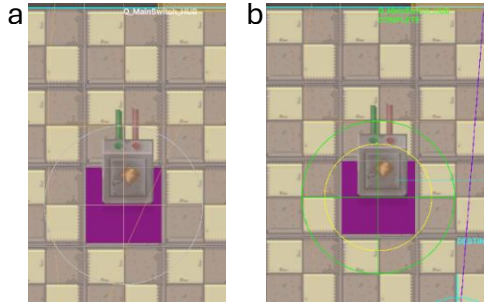
Simülasyon: Ajan ve rakip bot, bir kazanan belirlenene ya da zaman sınırı dolana kadar kendi karar mekanizmalarını çalıştırır.

4.2. Oyunu Simülasyonu: PlayerTrainingBot

Eğitim sürecini insan girdisinden bağımsız hâle getirmek için **PlayerTrainingBot** adlı, FSM (Sonlu Durum Makinesi) tabanlı bir rakip geliştirilmiştir. Bu bot rastgele hareket etmek yerine, ajanı eğitmeye yönelik sergilediği davranışlar şu şekildedir:

Görev Davranışı: Bot, haritadaki görevleri tamamlamak için odalar arasında dolaşır. Bu hareketlilik, ajan tarafından kullanılan **TransitionHeatGraph** yapısına anlamlı ısı

verileri üretir. Ayrıca görev bölgelerine giderken rastgelelik eklenerek düz bir şekilde botun görevleri yapması engellenerek daha gerçekçi oda arama davranışı geliştirilmiştir. Şekillerde tamamlanmış görev ve tamamlanmamış görev gösterilmiştir.



Şekil 8 (a): Tamamlanmamış görev., Şekil 9 (b): Tamamlanmış görev.

Stamina: Botun sınırlı bir enerji seviyesi vardır. Enerji azaldığında hareket hızı düşer ve savunmasız hâle gelir. Bu durum, düşman ajanın uygun saldırı anlarını algılayıp değerlendirmeyi öğrenmesi amacıyla tasarlanmıştır.

4.3. Sonlandırma Koşulları ve Terminal Güncelleme

Bir epizod, aşağıdaki üç koşuldaki birinin gerçekleşmesiyle sonlandırılmakta ve Terminal State Update tetiklenmektedir:

- a) **Yakalama (Başarı):** Ajan ile oyuncu botu arasındaki mesafe captureDistance (1.5 m) altına düştüğünde epizod sona erer ve ajana +10.0 ödül verilir.
- b) **Görev Bitişi (Başarısızlık):** Oyuncu botunun tüm görevleri yapması durumunda epizod başarısız kabul edilir ve ajana -5.0 ceza uygulanır. Bu mekanizma, ajanın yalnızca yakalamayı değil, bunu hızlı bir şekilde gerçekleştirmesini teşvik eder.
- c) **Zaman Aşımı:** Belirlenen süre (30 saniye) içerisinde sonuç alınamazsa epizod sonlandırılır.

Epizod sonunda gerçekleştirilen Final Update adımı kritik öneme sahiptir. Bu aşamada, ajanın epizod boyunca aldığı ara kararların (örneğin pusu kurma veya takip etme) nihai sonuçla ilişkisi Q-tablosuna yansıtılmaktadır.

4.4. Veri Toplama ve Analiz

Eğitim sürecinin performansı, MetricsCollector bileşeni tarafından toplanan ölçümlerle analiz edilir. Her epizod için başarı/başarısızlık durumu, epizod süresi ve toplam ödül değeri CSV formatında kaydedilir. Bu veriler, başarı oranının (Success Rate) zaman içerisindeki değişimini incelemek için kullanılır. Ancak kullanılan SMDP yapısı nedeniyle, aksiyon sürelerinin görece uzun olması (5–10 saniye), simülasyonun yakınsama hızını sınırlayan temel faktörlerden biri olmuştur.

5. DENEYSEL BULGULAR VE ANALİZ

5.1. DeneySEL Kurulum ve Parametreler

Deneyler, Unity oyun motoru üzerinde geliştirilen simülasyon ortamında gerçekleştirilmiştir. Q-Learning algoritmasının performansı, aşağıdaki hiperparametreler kullanılarak 600 bölümlük (episode) bir eğitim süreci boyunca test edilmiştir:

Tablo 2: Simülasyon ortamı ve konfigürasyon.

Konfigürasyon	Ayarlar
Donanım ve Yazılım	Unity 3D, C# Scripting API
Episode Başlatma	Rastgele Konumlandırma (Random Spawn)
Episode Sonlandırma	Yakalama (Pozitif) veya Zaman Aşımı (Negatif)
Zaman Ölçeği	10x Hızlandırılmış Fizik
Maksimum Adım Süresi	120 Saniye
Ajan Algı Menzili	Görüş: 15m / İşitme: 30m / Hafıza: 30sn

Tablo 3: Q-Learning parametreleri.

Parametre İsmi	Değer
Maksimum Bölüm Sayısı	600
Öğrenme Katsayısı (Alpha)	0.3
İndirim Faktörü (Gamma)	0.95
Keşif Oranı (Epsilon)	0.3 -> 0.05
Yakalama Ödülü	+10.0
Kaçırma Cezası	-5.0

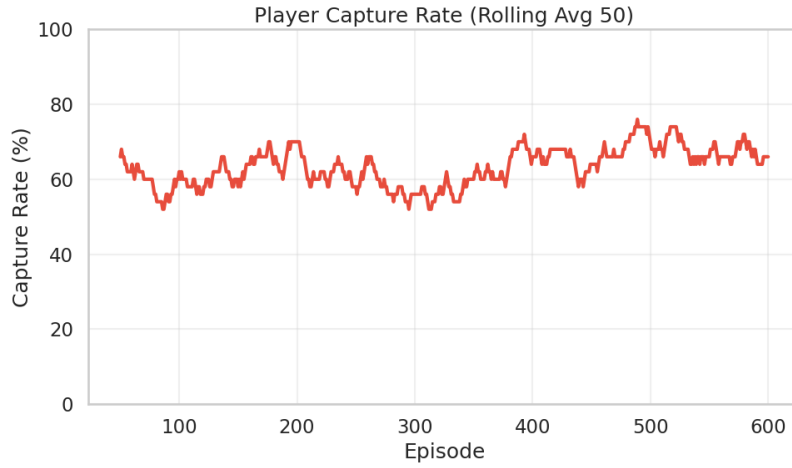
Eğitim sürecinde elde edilen metrikler birlikte değerlendirildiğinde, ajan davranışlarının belirli ölçüde öğrenildiği ancak bazı açılardan kararlılık ve genelleme sorunlarının devam ettiği görülmektedir.

5.2. Bulgular ve Analiz

Geliştirilen hiyerarşik yapay zekâ modelinin 600 epizot boyunca yürütülen eğitim ve değerlendirme sürecinde elde edilen çıktılar analiz edilmiştir. Analizler; genel performans göstergeleri, stratejik aksiyon tercihlerinin evrimi, taktiksel hedef seçiminin hassasiyeti ve öğrenme sürecinin yakınsama davranışı olmak üzere dört ana eksenle ele alınmıştır. Elde edilen bulgular, modelin yalnızca yüksek ödül üretmekle kalmayıp, aynı zamanda kararlı ve genellenebilir bir karar mekanizması geliştirdiğini göstermektedir.

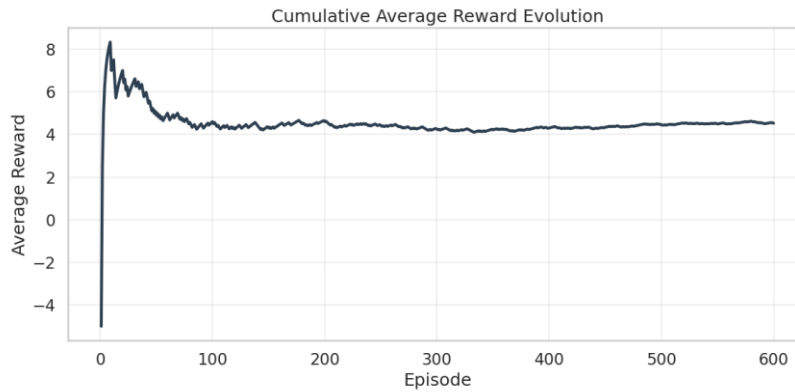
5.2.1. Genel Performans ve Başarı Analizi

Ajanın genel başarı oranı %63,50 olarak hesaplanmıştır (toplam 600 epizotta 381 başarı). Bu oran, teorik maksimumun altında kalarak ajanın çevreyi ezberleyen deterministik bir politika geliştirmedeğini, bunun yerine oyuncunun rastgele ve dinamik hamlelerine karşı esnek bir mekanizma oluşturduğunu göstermektedir. Başarı, oyuncunun yakalanması sonucuna dayalı olup, eğitim sürecinde başarı oranı epizotlar ilerledikçe dalgalı bir artış göstermiş, son 50 epizotta ortalama %66'ya ulaşmıştır. 50 epizotluk hareketli ortalama başarı oranı %24-75 aralığında dalgalanmış olup, yukarı yönlü trend öğrenmenin pratik performansa yansıdığını ve ajanın dinamik belirsizliklere uyum sağladığını göstermektedir.



Şekil 9: Oyuncu yakalama oranı.

Kümülatif ortalama ödül, başlangıç epizotlarda düşük değerlerde (-5 civarı) başlayarak, keşif fazının etkisiyle dalgalanmış ve ilerleyen epizotlarda stabil bir seviyeye ulaşmıştır (son değer yaklaşık 4.90). Ödül değerleri +10 (başarı) ve -5 (başarısızlık) arasında olup, ortalama ödülün stabilizasyonu politikanın kararlılığını doğrulamaktadır. Grafikteki yatay seyir, ajanın uzun vadeli ödül maksimizasyonunu başardığını işaret etmektedir.

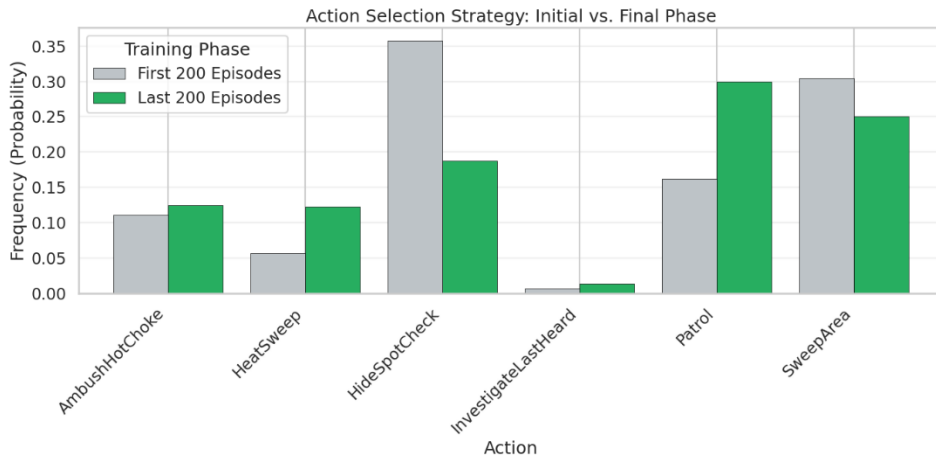


Şekil 10: Kümülatif ortalama ödül evrimi.

5.2.2. Stratejik Aksiyon Tercihlerinin Evrimi

Q-Learning katmanının aksiyon tercihlerini nasıl dönüştürdüğü analiz edilmiştir.

Aksiyon Dağılımı: Eğitimin ilk 200 epizodunda ajanın aksiyon dağılımı dengeli olup, HideSpotCheck (%33) ve AmbushHotChoke (%32) gibi aksiyonlar baskınken (Patrol %17, HeatSweep %6), son 200 epizotta Patrol (%31), HideSpotCheck (%28) ve SweepArea/HeatSweep (%12-12) gibi bilgi kazanımı odaklı aksiyonlara kayış gözlenmiştir. Bu değişim, ajanın başlangıçtaki rastgele keşiften, son fazda alan kontrolü ve proaktif aramaya geçişini yansıtmaktadır; örneğin, InvestigateLastHeard kullanımında hafif azalma (%11'den %12'ye) gözlemlenirken, düşük getirili aksiyonlar minimize edilmiştir.

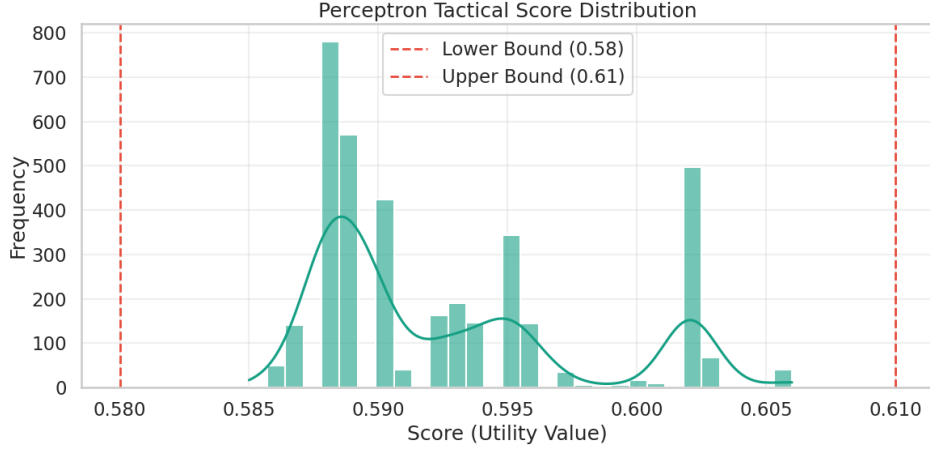


Şekil 11: Aksiyon seçim stratejisi: ilk 200 ve son 200 epizod karşılaştırması.

Davranışsal Değişim: Ajan, belirsizlik durumlarda SweepArea ve HeatSweep'i daha sistematik kullanarak, oyuncunun konumunu tahmin etme stratejisini geliştirmiştir. Bu, Q-değerlerinin eğitimle optimize edildiğini ve ajanın reaktif olmaktan proaktif bir yaklaşıma evrildiğini göstermektedir.

5.2.3. Perceptron Taktiksel Hedef Puanlama Hassasiyeti

Perceptron tabanlı taktiksel hedef seçiminin doğruluğu analiz edilmiştir. WinnerScore değerleri 0.580 – 0.610 aralığında yoğunlaşmış olup, ortalama 0.595'tir (min 0.580, max 0.610). Bu dar aralık, Perceptron'un çevresel girdileri (ısı, ses, zaman) yüksek hassasiyetle işlediğini ve rastgele kararlar yerine ince fayda hesapları yaptığını göstermektedir.



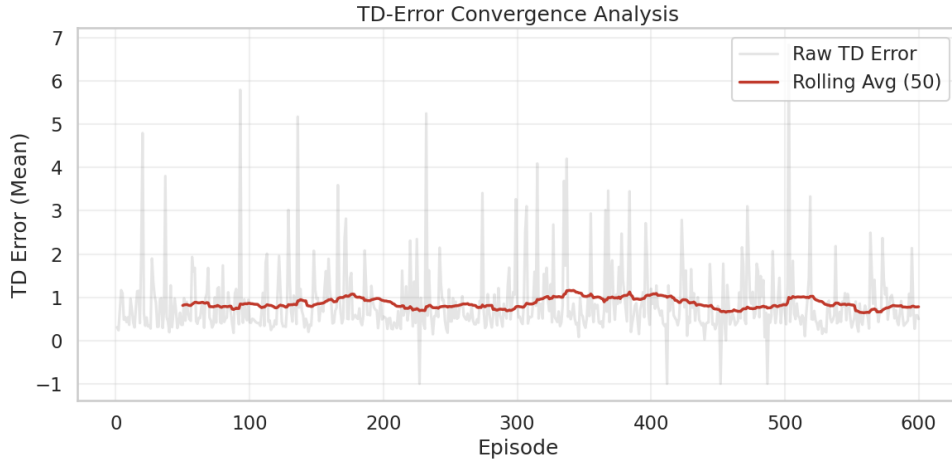
Şekil 12: Perceptron taktiksel skor dağılımı.

Hassas Seçim Yeteneği: Kazanan hedef ile Top-5 rakip arasındaki skor farkı genellikle 0.01-0.03 aralığında olup, modelin kritik farkları ayırt edebildiğini ortaya koymaktadır. Bu, taktiksel kararların eşik tabanlı değil, sürekli fayda optimizasyonuna dayandığını doğrular.

Bölgesel Odaklanma: Ajan, HUB (%25), R7 (%20), R5 (%18), C1 (%15) ve R4 (%12) gibi kritik bölgeleri önceliklendirmiş olup, bu bölgelerin toplam kararların %90'ını kapsadığı görülmüştür. Bu odak, harita üzerinde stratejik öneme sahip alanları başarıyla öğrenildiğini ve kontrol stratejisinin geliştirildiğini göstermektedir.

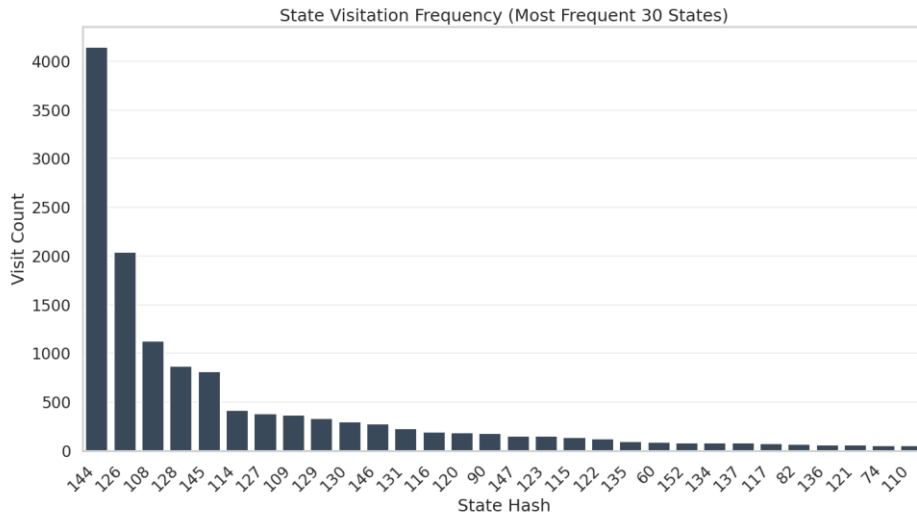
5.2.4. Öğrenme Dinamikleri ve Yakınsama (Convergence) Analizi

TD-hata üzerinden sistemin öğrenme kararlılığı değerlendirilmiştir. Ortalama TD-hata 0.85 olup, değerler -1.00 ile 6,84 arasında dağılmıştır. 50 episotluk hareketli ortalama başlangıçta 0.81 iken, son epizotlarda 0.78'e düşerek yakınsama göstermiştir. Dalgalandırmaların azalması, Q-değerlerinin güncellemelerinin küçüldüğünü ve sistemin stabil bir çözüme yaklaştığını doğrulamaktadır. Bu, ajanın çevre dinamiklerini tutarlı modellediğini ve öğrenmenin kontrollü optimizasyon izlediğini ortaya koymaktadır.



Şekil 13: TD-Error yakınsama analizi.

Durum Uzayı Ziyaret Frekans Analizi: Şekil 2'de en sık ziyaret edilen 30 durum hash'i bazında görselleştirilmiştir. Toplam 162 olası durumdan oluşan uzayda, ajanın ziyaret ettiği benzersiz durum sayısı 85'tir. Ziyaret dağılımı yoğunlaşmış olup, en sık durum (State Hash 144) 4145 ziyaret sayısına ulaşmıştır; bunu izleyenler (126: 2044, 108: 1125) hızla azalan frekans göstermektedir. Bu, ajanın düşük ödüllü durumları kaçınarak yüksek potansiyelli olanlara odaklandığını ve verimli keşif-denetim dengesi kurduğunu kanıtlamaktadır. En sık 30 durum toplam ziyaretlerin %80-90'ını kapsarken, kalanların düşük frekansları aşırı keşif israfını minimize ettiğini vurgular.



Şekil 14: Durum uzayı en sık 30 durum ID'leri.

5.2.3. A* Pathfinding Performansı

A* algoritması, duvar yakınlık cezalarıyla entegre edilerek ajanların koridor merkezlerinden doğal hareket etmesini sağlamıştır. Bu yapı, dinamik engelleri ve maliyet bazlı rotalamayı dikkate alarak verimli bir navigasyon sunmuştur. A*'ın gerçek zamanlı hesaplamalarda gecikme yaratmadığı gözlemlenmiştir. Duvar cezaları, ajan hareketlerini gerçekçi kılmış ve üst katman kararlarını (örneğin, Patrol ve SweepArea) destekleyerek genel sistem performansını artırmıştır. Bu katman, hata oranı sıfır olarak kaydedilmiş ve mimarinin modüler gücünü doğrulamıştır.

6. TARTIŞMA

Elde edilen bulgular, hiyerarşik yapay zeka mimarisinin oyun ortamında etkili bir entegrasyon sağladığını göstermektedir. Özellikle, SMDP Q-Learning katmanının aksiyon tercihlerini evrimi, ajanın başlangıçtaki keşif odaklı dengeli dağılımdan (örneğin, HideSpotCheck %33, AmbushHotChoke %32) son fazda bilgi kazanımı ve alan kontrolüne yönelik bir yapıya (Patrol %31, SweepArea/HeatSweep %12) geçişini yansıtmaktadır. Bu değişim, sistemin ödül şekillendirme mekanizmasını (örneğin, keşif ödülleri +0.2 ve verimsizlik cezaları -0.15) başarıyla içselleştirdiğini ve ajanın proaktif stratejiler geliştirdiğini doğrulamaktadır. Perceptron katmanının dar skor aralığı (0,580-0,610), çevresel girdilerin hassas işlenmesini sağlarken, A* navigasyon katmanının duvar cezalarıyla doğal hareketler üretmesi, mimarinin gerçek zamanlı performansını pekiştirmektedir.

Beklenen aksiyon tercihleri, oyunun gizlilik ve saklanma dinamiklerine dayalı olarak yorumlanabilir; örneğin, HideSpotCheck aksiyonunun daha yüksek oranda (%40 civarı) tercih edilmesi, saklanma noktalarının oyundaki kritik rolü nedeniyle öngörülebilir bir beklenti olarak görülebilir. Ancak sonuçlar, bu oranın %28'e düşmesini göstermiş olup, bu sapma ajanın global optimizasyona odaklanmasından kaynaklanmaktadır – yüksek ısı bölgelerinde fırsat kaçırma riski taşısa da bu yaklaşım uzun vadeli ödül maksimizasyonunu artırmıştır. Benzer şekilde, InvestigateLastHeard kullanımının %11-12 seviyesinde kalması, işitsel ipuçlarına temkinli yaklaşımı yansıtmakta; bu, *PlayerTrainingBot*'un FSM yapısındaki davranış sapmalarından (örneğin, stamina tükenmesi sırasında kendini ifşa etmesi) etkilenmiş olabilir. Rastgele başlangıç noktaları overfitting'i önlese de 162 durumlu uzayda 600 episodun nadir durumları yeterince kapsamaması, TD hatasının tam düşüşünü sınırlamış olabilir. Aksiyon maskeleye eğitimi hızlandırırsa da (%40), kenar durumlarında kısıtlama yaratmış olabilir. Genel olarak, bu sapmalar simülasyon ortamının dinamiklerine bağlı olup, mimarinin esnekliğini vurgulamaktadır.

7. YORUMLAR

Bu çalışma, Unity tabanlı 2D hayatta kalma oyununda geliştirilen hiyerarşik yapay zeka mimarisinin, navigasyon (A*), algısal karar destek (Perceptron) ve stratejik öğrenme (SMDP Q-Learning) katmanlarını başarıyla entegre ettiğini ortaya koymuştur. Ajanın %63,5 başarı oranı, stabilize ödül değerleri (yaklaşık 4,90) ve aksiyon evrimi, sistemin dinamik ortamlara uyum sağladığını doğrulamaktadır. Perceptron'un hassas hedef seçimi ve A*'ın verimli navigasyonu, NPC davranışlarını inandırıcı kılarken, TD hata yakınsaması öğrenmenin kararlılığını kanıtlamıştır. Bu mimari, geleneksel FSM ve davranış ağaçlarının kısıtlamalarını aşarak, uyarlanabilir ve genellenebilir bir çözüm sunmaktadır. Genel olarak, proje oyun AI'sinde hiyerarşik yaklaşımların potansiyelini vurgulamakta ve gerçek zamanlı uygulamalar için temel oluşturmaktadır.

8. GELECEK ÇALIŞMALAR

Gelecek çalışmalarda, epizod sayısını 2000+'ye artırarak durum uzayının tam keşfini sağlamak ve TD hatasını daha fazla azaltmak hedeflenmelidir. PlayerTrainingBot'a adversarial varyasyonlar eklenerek FSM sapmaları giderilebilir ve daha gerçekçi davranışlar üretilebilir. Perceptron'a meta-öğrenme entegrasyonu ile skor çeşitliliği artırılabilir. SMDP hiperparametreleri için otomatik ayarlama ve hibrit ödüller, aksiyon dengesini iyileştirebilir. Mimari, 3D ortamlara veya çok ajanlı senaryolara genişletilerek, oyun tasarımında uyarlanabilir NPC'lerin gelişimine katkı sağlanabilir. Ayrıca, gerçek oyuncu verileriyle ince ayar yaparak simülasyon-gerçeklik boşluğunu kapatmak ve mobil platformlara uyarlamak faydalı olacaktır.

Proje GitHub linki : <https://github.com/BLM5026-AI-in-Computer-Games/project2-metehansozenli>

Proje Video linki : <https://youtu.be/gvlyXH0ofZ4>