

**İSTANBUL TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK ELEKTRONİK FAKÜLTESİ**

**ENDÜSTRİYEL ÜRETİM HATLARINDA PEKİŞTİRMELİ ÖĞRENME İLE  
GÜÇLENDİRİLMİŞ OTOMATİK MALZEME YENİLEME SİSTEMİNİN  
TASARIMI, GELİŞTİRMESİ VE SİMÜLASYONU**

**KONTROL VE OTOMASYON MÜHENDİSLİĞİ TASARIMI II**

**Onur Ulusoy**

**Kontrol ve Otomasyon Mühendisliği**

**Tez Danışmanı: Doç. Dr. Ahmet Onat**

**HAZİRAN 2024**



**İSTANBUL TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK ELEKTRONİK FAKÜLTESİ**

**ENDÜSTRİYEL ÜRETİM HATLARINDA PEKİŞTİRMELİ ÖĞRENME İLE  
GÜÇLENDİRİLMİŞ OTOMATİK MALZEME YENİLEME SİSTEMİNİN  
TASARIMI, GELİŞTİRMESİ VE SİMÜLASYONU**

**KONTROL VE OTOMASYON MÜHENDİSLİĞİ TASARIMI II**

**Onur Ulusoy  
(030180088)**

**Kontrol ve Otomasyon Mühendisliği**

**Tez Danışmanı: Doç. Dr. Ahmet Onat**

**HAZİRAN 2024**



*Bana sürekli destek verdiğini bildiğim aileme ve  
tez çalışmam boyunca benden destegini ve anlayışını  
esirgemeyen değerli hocam Doç. Dr. Ahmet Onat'a,*





## ÖNSÖZ

Bu tez çalışmasının hazırlanmasında emeği geçen ve desteklerini esirgemeyen herkese teşekkürlerimi sunarım. Tez sürecinde bana yol gösteren ve bilgi birikimlerini paylaşan danışman hocam Prof. Dr. Ahmet Onat'a, tez çalışmalarım sırasında bana yardımcı olan araştırma arkadaşlarına ve sürekli moral ve motivasyon sağlayan aileme minnettarım.

Ayrıca, bu tezin gerçekleştirilmesinde maddi ve manevi desteklerini esirgemeyen İstanbul Teknik Üniversitesi'ne özel teşekkürlerimi sunarım.

Proje sürecinde moral ve motivasyon sağlayan Siemens A.Ş. Türkiye'deki çalışma arkadaşlarım ve mentorlarım Alpaslan Yıldız ve Mustafa Tiftikçi'ye, robotik ve mekatronik konularındaki bilgi ve deneyimlerini benimle paylaştıkları için ayrıca teşekkür ederim.

Bu tez çalışması, endüstriyel üretim süreçlerinde verimliliği artırmayı ve malzeme yenileme operasyonlarını yeniden şekillendirmeyi amaçlayan önemli bir adım olarak görülmektedir. Bu süreçte katkı sağlayan ve destek veren herkese tekrar teşekkür ederim.

Haziran 2024

Onur Ulusoy





## **İÇİNDEKİLER**

	<u>Sayfa</u>
<b>ÖNSÖZ .....</b>	vi
<b>İÇİNDEKİLER .....</b>	ix
<b>KISALTMALAR .....</b>	xi
<b>SEMBOLLER .....</b>	xiii
<b>ŞEKLİ LİSTESİ.....</b>	xv
<b>ÖZET .....</b>	xvii
<b>SUMMARY .....</b>	xix
<b>1. GİRİŞ .....</b>	1
1.1 Projenin Amacı.....	1
1.2 Projeye Genel Bakış .....	2
1.2.1 Yaklaşım .....	2
1.2.2 Yöntem.....	2
1.3 Literatür Araştırması .....	3
1.3.1 Arkaplan ve motivasyon .....	3
1.3.2 Endüstriyel uygulamalarda pekiştirmeli öğrenme .....	4
1.3.3 Sürü zekâsı algoritmaları .....	5
1.3.4 Otomatik malzeme ikmal sistemlerinin tasarımı .....	6
1.3.5 Simülasyon ve test ortamları.....	7
<b>2. PROBLEM AŞAMALARI.....</b>	8
2.1 Ayırık Zaman Yapay Zekâ Algoritmaları .....	8
2.1.1 İlk aşama: tekli komut verilmesi .....	9
2.1.2 İkinci aşama: sıralı komut verilmesi .....	9
2.1.3 Üçüncü aşama: birden fazla robotun iş birliği yapması .....	10
2.2 Sürekli Zaman Entegrasyonu .....	11
2.2.1 Robot tasarımı ve hazırlığı .....	11
2.2.2 Navigasyon ve lokalizasyon.....	11
2.2.3 AI algoritmalarının entegrasyonu .....	11
<b>3. TASARIM VE ALGORİTMA GENEL BAKIŞI.....</b>	12
3.1 Tasarlanan Ortam ve Koşullar.....	12
3.2 Q Learning.....	15
3.2.1 Genel program mantığı akışı.....	15
3.2.2 Aksiyon seçimi.....	16
3.2.3 Keşfetme Oranının Zamanla Azalması .....	17
3.2.4 Ödül / ceza alma mekanizması.....	18
3.2.5 Öğrenme Mekanizması .....	19
<b>4. YAPAY ZEKÂ VE ÖĞRENME MODELİ TASARIMI.....</b>	21
4.1 Birinci Aşama: Tekli komut öğrenimi .....	21
4.1.1 Ödül kuralları .....	21
4.1.1.1 İlk etap ödül kuralları ve sonuçları .....	21
4.1.1.2 İkinci etap ödül kuralları ve sonuçları .....	23

4.1.1.3 Üçüncü etap ödül kuralları ve sonuçları	25
4.1.1.4 Dördüncü etap ödül kuralları ve sonuçları	26
4.1.2 Sonuçlar.....	29
4.2 İkinci Aşama: Sıralı Komut Öğrenimi .....	32
4.2.1 Tasarım.....	33
4.2.1.1 Kutu temizleme mekanığı .....	33
4.2.1.2 Kutu temizleme başlangıç koşulları .....	34
4.2.1.3 Öğrenme ağacı yapısı .....	36
4.2.1.4 Ödüllendirme.....	38
4.2.1.5 Kutu temizleme uygulamaları ve sonuçları.....	40
4.2.1.5.1 Örnek 1 .....	40
4.2.1.5.2 Örnek 2 .....	43
4.2.1.5.3 Örnek 3 .....	44
4.2.1.6 Seçilmiş kutular için yol temizleme optimizasyonu .....	45
4.2.1.7 Uygulama .....	46
<b>5. SÜRÜ ZEKASI TASARIMI VE UYGULAMASI .....</b>	<b>49</b>
5.1.1 Tasarım.....	49
5.1.1.1 Ajanların özellikleri	49
5.1.1.2 Ajanların görselleştirmesi	52
5.1.1.3 Alan kısıtlamaları	52
5.1.1.4 Ajanların varsayılan davranışları	54
5.1.1.4 Ajan çağrıma karar için Q-Learning modeli	55
5.1.1.4.1 Kullanılan Q-tablosu yapısı	55
5.1.1.5 Öğrenme ve ödül mekanizmaları	56
5.1.1.6 Ajanların kendine görev atama modeli	60
5.1.1.6.1 Yerel Q-tablosunun yapısı	60
5.1.1.6.2 Q-Değerlerinin başlatılması ve güncellenmesi	61
5.1.1.6.3 Pekiştirme kriterleri	61
5.1.2 Uygulamalar .....	63
5.1.2.1 Uygulama 1	63
5.1.2.2 Uygulama 2	67
<b>6. SONUÇ VE ÖNERİLER .....</b>	<b>71</b>
6.1 Çalışmanın Uygulama Alanı .....	71
<b>KAYNAKLAR.....</b>	<b>73</b>
<b>ÖZGEÇMİŞ .....</b>	<b>75</b>

## **KISALTMALAR**

<b>AI</b>	: Artificial Intelligence
<b>JIT</b>	: Just-in-Time
<b>RL</b>	: Reinforcement Learning
<b>ROS</b>	: Robot Operating System
<b>2D</b>	: Two-Dimensional
<b>3D</b>	: Three-Dimensional
<b>SI</b>	: Swarm Intelligence



## **SEMBOLLER**

$\alpha$	: Öğrenme oranı (Learning rate)
$\gamma$	: İndirim faktörü (Discount factor)
$\epsilon$	: Keşfetme oranı (Exploration rate)
$Q(s, a)$	: Durum-eylem çiftinin değeri (Value of state-action pair)
$r$	: Anlık ödül (Immediate reward)
$s$	: Durum (State)
$a$	: Eylem (Action)



## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1 : Algoritmanın uygulandığı unity ortamının görseli .....	12
Şekil 3.2 : Kutuların ve şeffaf hareket alanlarının detaylı görseli.....	13
Şekil 3.3 : Planlanan yapının kuşbakışı ve yandan temsili görünümü.....	14
Şekil 4.1 : İlk etap ödül sisteminin denendiği diğer simülasyon ortamı.....	23
Şekil 4.2 : Kutunun en yüksek iki pozisyon arasında sıkışma durumu .....	24
Şekil 4.3 : Kutunun 4'lü veya 6'lı çevrimlerde sıkışma durumu .....	25
Şekil 4.4 : Kutunun en üst düzeye ulaşmadan yoluna devam etmesi sorunu ....	27
Şekil 4.5 : Kutunun daha uzun olan yolu tercih etmesi sorunu görselleştirmesi	28
Şekil 4.6 : Test rotası görselleştirmesi .....	29
Şekil 4.7 : Bölüm başına varış başarısı grafiği .....	30
Şekil 4.8 : Bölüm başına katedilen mesafe ve skaler mesafe grafiği.....	30
Şekil 4.9 : Bölüm başına epsilon değeri grafiği .....	31
Şekil 4.10 : Bölüm başına keşfetme ve tecrübe过的 faydalanan miktarı grafiği	31
Şekil 4.11 : Bölüm başına toplam ödül / katedilen mesafe grafiği .....	32
Şekil 4.12 : Bölüm başına imkânsız ve istemeyen hareketler grafiği .....	32
Şekil 4.13 : Üstteki kutuların temizlenme sürecini gösteren bir demo .....	35
Şekil 4.14 : Kutu temizliğinden sonra sistem.....	36
Şekil 5.1 : Ağaç yapısının görselleştirmesi .....	37
Şekil 5.2 : İlk bölümdeki yeniden konumlandırma.....	41
Şekil 5.3 : İkinci bölüm: öğrendikten sonraki gelişmiş strateji .....	42
Şekil 5.4 : Bölüm 1 ve 2: Problemli yeniden konumlandırma ve çözüm.....	43
Şekil 5.5 : Kutuların sistem oturuktan sonra sıralı hareketi .....	45
Şekil 5.6 : Bölüm 1: Çoklu Engeller .....	47
Şekil 5.7 : Bölüm 2: Engelleşicilerin yeniden konumlandırılması.....	47
Şekil 5.8 : Bölüm 3: Yeniden konumlandırmada artan verimlilik .....	48
Şekil 5.9 : Küresel beyin ajanları koordine ediyor .....	51
Şekil 5.10 : Ajan ve harita görselleştirmesi.....	52
Şekil 5.11 : Alan kısıtlamaları nedeniyle ajan görev devri.....	54
Şekil 5.12 : Bölüm 1: Tek ajan operasyonu .....	64
Şekil 5.13 : Bölüm 2: İki ajanın işbirliği .....	65
Şekil 5.14 : Bölüm 3: Tamamen eşzamanlı operasyon .....	66
Şekil 5.15 : Çok ajanlı koordinasyon eğitim sonu .....	69



# **ENDÜSTRİYEL ÜRETİM HATLARINDA PEKİŞTİRMELİ ÖĞRENME İLE GÜÇLENDİRİLMİŞ OTOMATİK MALZEME YENİLEME SİSTEMİNİN TASARIMI, GELİŞTİRMESİ VE SİMÜLASYONU**

## **ÖZET**

Bu çalışma, endüstriyel üretim hatlarında malzeme yenileme süreçlerini iyileştirmek amacıyla, Pekiştirmeli Öğrenme (RL) ile desteklenen, üretim hatlarının özel tasarlanmış depo ortamlarında çalışan kara robotlarına odaklanmaktadır. Ana hedef, Just-in-Time (JIT) üretim ilkeleriyle uyumlu, malzeme yenileme operasyonlarını optimize eden spesifik bir sistem geliştirmektir. Projenin temel bileşeni ve çekirdeği, pekiştirmeli öğrenme tabanlı yapay zekâ ve sürü zekâsı (swarm intelligence) algoritmalarıdır. Bu algoritmalar, üretim süreçlerinden elde edilmesi istenen verime dayalı olarak oluşturulan belirli kuralların robotların malzeme yenileme görevlerini nasıl yerine getireceklerini belirleyen bir yapay zekâ sistemini içermektedir. Yapay zekâ sisteminin verimliliği, robotların görevlerini en etkin ve hızlı şekilde yerine getirmelerini sağlamak üzere tasarlanmıştır.

Bu çalışma, endüstriyel üretim süreçlerinde verimliliği artırarak malzeme yenileme operasyonlarını yeniden şekillendirmeyi hedeflemektedir. Malzeme yönetimi ve üretim süreçlerinde karşılaşılan zorluklar, dinamik düzenleme ihtiyaçları ve geniş alan gereksinimleri dikkate alınarak, 3 boyutlu prizmatik kutu ızgarası tabanlı bir sistem önerilmiştir. Yapay zekâ destekli robotlar aracılığıyla, malzeme kutuları etkin bir şekilde yönetilmekte ve üretim taleplerine dinamik olarak uyum sağlanmaktadır. Bu yöntem, malzeme dağıtımını optimize ederek gereksiz stokların azaltılmasına, maliyetlerin düşürülmESİNE ve üretim süreçlerinin kesintisiz ve verimli bir şekilde yürütülmesine katkı sağlamaktadır. Ayrıca, malzeme dağıtımını ve önceliklendirme süreçlerindeki mevcut zorlukları aşarak, üretim hatlarının daha etkin ve sürekli çalışmasını sağlayacak yenilikçi ve kapsamlı bir çözüm sunmaktadır.

Robotun çalışacağı yerin koşulları ve her bir aşamada spesifik görevi belirlenmiş, yapay zekâ algoritmalarında oldukça ilerlenmiş ve çeşitli simülasyon ortamlarında gerçekleşmiştir. Projenin simülasyon aşamasında elde edilen veriler, robotların farklı koşullar altında nasıl performans göstereceğini değerlendirmek için kullanılmıştır. Proje tamamlanmış ve simülasyonlar aracılığıyla gerçek dünya uygulamaları için sağlam bir zemin hazırlanmıştır. Bu bağlamda, projenin son aşamasında robotlar hazırlanan sanal ortamlarda test edilmiş ve elde edilen sonuçlar, yapay zekâ sisteminin etkinliğini ve doğruluğunu göstermiştir. Danışman öğretim üyesinin önerisi ve yönlendirmesi ile proje kapsamında yapay zekâ (AI) bölümne öncelik verilmiş ve yıl boyunca bu alanda yoğun çalışmalar yapılmıştır. Bu süreçte, AI algoritmalarının geliştirilmesi ve optimizasyonu üzerine odaklanılarak simülasyon ortamlarında detaylı testler gerçekleştirerek sonuçlar doğrultusunda algoritmaların gelişimi sağlanmıştır. Algoritmaların tasarımını ve geliştirilmesi, proje hedeflerine ulaşmada temel bir rol oynamış ve üretim süreçlerine entegre edilecek robotik sistemlerin verimliliğini ve etkinliğini artırma potansiyelini göstermiştir. Bu gelişmeler, projenin ilerleyen aşamalarında robot tasarımları ve entegrasyon çalışmalarının daha sağlam bir temel üzerine oturmasını sağlamıştır. Bu aşamada elde edilen bulgular ve veriler, sonraki dönemlerdeki uygulamalara yön verme açısından değerli bir kaynak teşkil etmektedir.



**DESIGN, DEVELOPMENT, AND SIMULATION OF AN AUTOMATED  
MATERIAL REPLENISHMENT SYSTEM ENHANCED BY  
REINFORCEMENT LEARNING IN INDUSTRIAL PRODUCTION LINES**

**SUMMARY**

This study focuses on black robots operating in specially designed warehouse environments of production lines, supported by Reinforcement Learning (RL), with the aim of improving material replenishment processes in industrial production lines. The main objective is to develop a specific system that optimizes material replenishment operations in accordance with Just-in-Time (JIT) production principles. The core and essential component of the project are the artificial intelligence (AI) and swarm intelligence algorithms based on reinforcement learning. These algorithms include an AI system that determines how the robots will perform material replenishment tasks based on specific rules created to achieve the desired efficiency from production processes. The efficiency of the AI system is designed to ensure that the robots perform their tasks in the most effective and fastest manner.

This study aims to reshape material replenishment operations by increasing efficiency in industrial production processes. Considering the challenges encountered in material management and production processes, the need for dynamic arrangement, and the requirement for large spaces, a system based on a 3-dimensional prismatic box grid is proposed. Through AI-supported robots, material boxes are managed efficiently and dynamically adapted to production demands. This method optimizes material distribution, reducing unnecessary stock, lowering costs, and contributing to the seamless and efficient execution of production processes. Additionally, it offers an innovative and comprehensive solution that overcomes current challenges in material distribution and prioritization processes, ensuring more efficient and continuous operation of production lines.

The conditions of the environment where the robot will operate and its specific tasks at each stage have been determined, significant progress has been made in AI algorithms, and they have been implemented in various simulation environments. The data obtained during the simulation phase of the project have been used to evaluate how the robots would perform under different conditions. The project has been completed, and a solid foundation for real-world applications has been established through simulations. In this context, in the final phase of the project, the robots were tested in prepared virtual environments, and the results demonstrated the effectiveness and accuracy of the AI system.

On the recommendation and guidance of the advisor, priority was given to the AI component within the scope of the project, and intensive work was carried out in this area throughout the year. During this process, the development and optimization of AI algorithms were focused on, detailed tests were conducted in simulation environments, and algorithms were improved based on the results. The design and development of the algorithms played a crucial role in achieving the project's objectives and demonstrated the potential to enhance the efficiency and effectiveness of robotic systems to be integrated into production processes. These advancements have provided a stronger foundation for robot design and integration efforts in the later stages of the project. The findings and data obtained at this stage serve as valuable resources for guiding applications in subsequent periods.

# **1. GİRİŞ**

Endüstriyel üretim hatlarında malzeme yenileme süreçlerinin verimliliğini artırmak ve maliyetleri düşürmek, günümüzün rekabetçi üretim ortamında kritik bir ihtiyaçtır. Bu çalışmada, yapay zekâ ve robotik teknolojiler kullanılarak, bu süreçlerin nasıl optimize edilebileceği araştırılmıştır. Proje, Pekiştirmeli Öğrenme (RL) ve sürü zekası algoritmalarıyla donatılmış kara robotların üretim hatlarında malzeme yenileme görevlerini en etkin şekilde yerine getirmesini hedeflemektedir. Bu kapsamda, detaylı simülasyon çalışmaları yapılmış ve elde edilen bulgular, gelecekteki uygulamalara ışık tutacak değerli veriler sunmuştur.

## **1.1 Projenin Amacı**

Çalışmanın temel amacı, modern üretim hatlarında karşılaşılan malzeme dağıtım ve önceliklendirme süreçlerinin zorluklarını ele almak ve bu süreçleri yapay zekâ ve robotik teknolojilerle optimize ederek yeni çözümler geliştirmektir.

Günümüzde, üretim süreçlerinde malzemenin doğru zamanda doğru yere ulaşılması, özellikle malzeme akışındaki düzensizlikler, talep değişkenlikleri nedeniyle büyük önem taşımaktadır. Bu zorluklar, üretim hattının verimliliğini ve sürekliliğini etkileyebilmekte, maliyetleri ve israfi artırabilmektedir. Geleneksel yöntemler, çoğunlukla insan gücüne veya basit otomasyon sistemlerine dayanmaktadır. Ancak bu sistemler, talep değişikliklerine hızlı bir şekilde uyum sağlayamamakta ve JIT üretim ilkeleriyle tam uyumlu olmamaktadır. Bu durum, üretim sürekliliğini ve esnekliğini olumsuz yönde etkilemektedir.

Bu çalışmada önerilen çözüm, Pekiştirmeli Öğrenme (RL) ile eğitilen bir yapay zekâ sistemi kullanarak üretim hatlarındaki malzeme dağıtımını optimize etmek ve malzeme ihtiyaçlarını öğrenme süreci dahilinde en verimli şekilde karşılayabilmektir. AI destekli sistem, üretim süreçlerindeki talep değişikliklerine dinamik olarak adapte olabilir ve malzeme dağıtım süreçlerini sürekli olarak iyileştirebilir. Tasarlanan robot

sürüsü, üretim alanlarında yüksek verimlilik ve hassasiyetle hareket ederek malzeme akışını kesintisiz ve etkili bir şekilde yönetecektir.

Önerilen sistem, JIT üretim ilkeleriyle tam uyumludur. Malzeme dağıtım süreçlerinin daha etkin yönetimi, gereksiz stokların azaltılmasına, böylece maliyetlerin düşürülmeye katkı sağlar. Ayrıca, üretim süreçlerinin kesintisiz ve verimli bir şekilde yürütülmesini destekleyerek toplam üretim verimliliğinin artmasına yardımcı olur. Bu yaklaşım, malzeme dağıtımını ve önceliklendirme süreçlerindeki mevcut zorlukları aşarak, üretim hatlarının daha etkin ve sürekli çalışmasını sağlayacak yenilikçi ve kapsamlı bir çözüm sunmaktadır.

## **1.2 Projeye Genel Bakış**

### **1.2.1 Yaklaşım**

Malzeme yönetimi ve üretim süreçlerinde karşılaşılan depolama zorluklarına yenilikçi bir çözüm olarak, ham malzemeleri veya yarı mamul ürünleri içerebilen 3 boyutlu prizmatik kutu ızgarası tabanlı bir sistem tasarlanmıştır. Geleneksel depo sistemleri, dinamik düzenleme zorlukları, geniş alan ihtiyacı ve esnek olmayan yapıları nedeniyle hızla değişen üretim taleplerine uyum sağlamada sıkılıkla yetersiz kalmaktadır. Tasarlanan bu yeni sistem, bu sınırlamaların üstesinden gelerek alan kullanımını optimize eder ve üretim ihtiyaçlarına dinamik bir şekilde uyum sağlar. Izgara sisteminin üzerindeki bir düzlemdede hareket eden yapay zekâ destekli robot sürüsü aracılığıyla malzeme kutuları etkin bir şekilde yönetilir. İzlenen yöntemlerle ve tasarlanan algoritmalar dahilinde belirli hareketleri (aksiyonlar) yapabilen robotlar, belirli kurallar dahilinde, istenen zamanlarda ihtiyaç duyulan malzeme kutularını ızgaranın en üst düzlemindeki stratejik konumlara veya gerek duyulmayan eşyaları alt bölmelerde istenen konumlara en hızlı şekilde yerlestirebilir.

### **1.2.2 Yöntem**

Bu proje, farklı aşamaları ile kümülatif olarak ilerleyen bir yöntem dahilinde ele alınmıştır. İlk ana aşamada, yapay zekâ algoritmalarının tasarımını, programlama dilleri kullanılarak kullanılabilir simülasyon ortamlarında gerçekleştirılmıştır. Bu süreç, algoritmaların etkinliğini test etmek ve geliştirmek için kritik öneme sahiptir. İkinci

ana aşama ise, geliştirilen yapay zekâ algoritmalarının gerçek robotlara uygulanmasıdır. Bu kapsamda, farklı simülasyon platformları kullanılarak robotların tasarımları ve entegrasyonunun yapılması planlanmıştır. Bu iki aşama arasındaki geçiş, algoritmaların simülasyondan gerçek dünya uygulamalarına sorunsuz bir şekilde aktarılmasını sağlamak için dikkatle yönetilecektir. Projenin bu bölümlenmiş yapısı hem teorik hem de pratik uygulamaların bir arada ele alınmasını ve optimizasyonun her iki alanda da etkili bir şekilde gerçekleştirilemesini mümkün kılmaktadır.

### **1.3 Literatür Araştırması**

#### **1.3.1 Arkaplan ve motivasyon**

Endüstriyel üretim hatlarında malzeme yenileme süreçlerinin optimize edilmesi, üretim verimliliğinin artırılması ve maliyetlerin düşürülmesi açısından büyük bir öneme sahiptir. Geleneksel yöntemler, sık sık karşılaşılan hatalar ve gecikmeler nedeniyle üretim sürecinde aksamalara yol açabilir. Bu nedenle, yapay zekâ ve otomasyon teknolojilerinin kullanımı, malzeme yenileme süreçlerini daha etkili ve verimli hale getirmek için giderek daha fazla araştırılmaktadır.

Endüstriyel üretim hatlarında malzeme yenileme, üretim sürecinin sürekliliğini sağlamak ve stok yönetimini optimize etmek için kritik bir rol oynar. Malzeme eksiklikleri, üretim hatlarında durmalara ve verimlilik kaybına neden olabilir. Bu nedenle, doğru zamanda ve doğru miktarda malzemenin sağlanması, üretim sürecinin kesintisiz ve verimli bir şekilde devam etmesi için hayatı öneme sahiptir. Aynı zamanda, aşırı stok birikimini önlemek, maliyetleri düşürmek ve depolama alanını verimli kullanmak açısından da önemlidir.

Malzeme yenileme, endüstriyel üretim hatlarının performansını doğrudan etkileyen bir faktördür. Malzeme tedarikindeki herhangi bir aksama, üretim sürecinde kesintilere yol açabilir ve bu da hem maliyetleri artırır hem de müşteri memnuniyetini olumsuz etkiler. Bu nedenle, malzeme yönetiminin optimize edilmesi, endüstriyel süreçlerde verimliliği artırmak için kritik bir öneme sahiptir (Shi et al., 2020). Reinforcement Learning (Pekiştirmeli Öğrenme, RL) ve sürü zekâsı algoritmalarının kullanımı, bu süreçleri daha esnek ve uyumlu hale getirebilir, böylece üretim hatlarının dinamik gereksinimlerine cevap verebilir (Kozlica et al., 2023).

Just-in-Time (JIT) üretim prensipleri, malzeme ve ürünlerin ihtiyaç duyulduğu anda ve sadece ihtiyaç duyulan miktarda tedarik edilmesini amaçlar. Bu yaklaşım, stok maliyetlerini minimize ederken, üretim sürecinin esnekliğini artırır ve müşteri taleplerine hızlı yanıt verme yeteneğini geliştirir. JIT sistemleri, üretim hatlarında gereksiz stokları ortadan kaldırarak, üretim maliyetlerini azaltmayı ve üretim süreçlerini daha etkin hale getirmeyi hedefler (Viharos & Jakab, 2021).

Bu çalışma, Pekiştirmeli Öğrenme (Reinforcement Learning, RL) ve sürü zekâsı algoritmalarını kullanarak, endüstriyel üretim hatlarında malzeme yenileme süreçlerini optimize etmeyi amaçlamaktadır. Bu yenilikçi yaklaşımlar, üretim hatlarının dinamik ve karmaşık gereksinimlerine uyum sağlamak ve JIT ilkeleri ile uyumlu malzeme yönetimi stratejileri geliştirmek için kullanılacaktır.

### **1.3.2 Endüstriyel uygulamalarda pekiştirmeli öğrenme**

Pekiştirmeli öğrenme (Reinforcement Learning, RL), bir ajan (agent) ve çevresi (environment) arasındaki etkileşimi temel olarak, ajanın belirli bir görevi optimize etmesi için ödül (reward) ve ceza (penalty) mekanizmalarını kullanarak öğrenmesini sağlayan bir yapay zekâ yöntemidir. RL, Markov Karar Süreçleri (Markov Decision Processes, MDP) üzerine kuruludur ve ajanın bulunduğu durumları (states), gerçekleştirdiği eylemleri (actions) ve bu eylemler sonucunda aldığı ödülleri içeren bir döngü ile çalışır. Ajan, çevresindeki durumları gözlemleyerek ve bu durumlara uygun eylemler seçerek, en yüksek toplam ödül elde etmeyi hedefler (Nian et al., 2020).

Pekiştirmeli öğrenme, endüstriyel üretim ve imalat süreçlerinde birçok alanda uygulanmaktadır. Bu alanlar arasında üretim planlaması, süreç kontrolü, bakım stratejileri ve malzeme yenileme yer almaktadır. RL, özellikle karmaşık ve dinamik üretim ortamlarında, geleneksel yöntemlerin yetersiz kaldığı durumlarda önemli avantajlar sunmaktadır. Örneğin, RL algoritmaları, üretim hatlarında sıralama ve planlama sorunlarını çözmek için kullanılmıştır ve geleneksel sezgisel yöntemlere göre daha etkili sonuçlar elde edilmiştir (Panzer & Bender, 2021). Ayrıca, RL'nin açıklanabilirliği üzerine yapılan çalışmalar, üretim kontrol stratejilerinin daha anlaşılır ve güvenilir olmasını sağlamaktadır (Kuhnle et al., 2021). Pekiştirmeli öğrenme, malzeme yenileme sistemlerinde de önemli avantajlar sunar. RL tabanlı sistemler, malzeme ihtiyaçlarını anlık olarak belirleyerek, stok yönetimini optimize edebilir ve gereksiz stok birikimini önleyebilir. Bu, üretim hatlarının daha verimli ve kesintisiz

çalışmasını sağlar. Ayrıca, RL algoritmaları, değişen üretim koşullarına hızlı bir şekilde uyum sağlayarak, malzeme yenileme süreçlerinin dinamik olarak optimize edilmesine olanak tanır (Shi et al., 2020). Bu da maliyetlerin düşürülmesi ve verimliliğin artırılması açısından önemli katkılar sağlar.

Bu bağlamda, endüstriyel uygulamalarda pekiştirmeli öğrenmenin kullanımının, üretim süreçlerinde esneklik ve verimlilik sağlamada önemli bir rol oynadığı görülmektedir. RL, sadece malzeme yenileme sistemlerinde değil, genel olarak üretim ve imalat süreçlerinde de devrim niteliğinde yenilikler getirebilecek potansiyele sahiptir.

### **1.3.3 Sürü zekâsı algoritmaları**

Sürü zekâsı ya da swarm Intelligence (SI), doğadaki kolektif davranışları taklit eden ve bireylerin yerel etkileşimleri sonucu ortaya çıkan kompleks sistemleri inceleyen bir yapay zekâ teknigidir. SI, karınca kolonileri, kuş sürüleri, arı kovanları gibi doğal sistemlerin davranışlarını model alarak, merkezi bir kontrol olmaksızın bireylerin basit kurallar ile organize olmasını sağlar. Bu yaklaşım, adaptasyon, ölçeklenebilirlik ve dayanıklılık gibi avantajlar sunar. Swarm Intelligence, çoğunlukla optimizasyon problemlerinin çözümünde, robotların ve insansız araçların kontrolünde, sosyal davranışların tahmin edilmesinde ve telekomünikasyon ağlarının geliştirilmesinde kullanılır (Wanka, 2019).

Swarm Intelligence, robotik sistemlerde merkezi olmayan kontrol mekanizmaları ile uygulanarak, robotların kendi aralarında işbirliği yapmalarını sağlar. Bu tür sistemlerde, her bir robot basit kurallarla hareket eder ve yerel bilgileri kullanarak kararlar alır. Örneğin, Particle Swarm Optimization (PSO) ve Ant Colony Optimization (ACO) algoritmaları, robotların hareketlerini ve görev dağılımlarını optimize etmek için kullanılır (Chu et al., 2011). Swarm robotikleri, özellikle dinamik ve belirsiz ortamlarda, adaptasyon ve esneklik gerektiren görevlerde büyük avantaj sağlar. Birden fazla robotun birlikte çalışarak karmaşık görevleri yerine getirdiği bu yaklaşım, endüstriyel uygulamalarda da giderek yaygınlaşmaktadır (Schranz et al., 2020).

Swarm Intelligence, endüstriyel uygulamalarda çeşitli başarı hikayelerine sahiptir. Örneğin, robot sürülerinin kendi kendine düzenlenmesi ve malzeme dağıtım gibi görevlerde kullanılması, üretim süreçlerinin verimliliğini artırmaktadır. WSN-MMR

sürü sistemi, kablosuz sensör ağları ile mobil çoklu robotların birleştirilmesi sonucu, karmaşık görevlerin etkin bir şekilde yerine getirilmesini sağlamaktadır (Li & Shen, 2011). Ayrıca, Pareto tabanlı parçacık sürü optimizasyonu, şehir içi tren tarifeleri ve trafik sinyal kontrolü gibi alanlarda uygulanarak, yolcu seyahat süresini ve kullanılan tren stoklarını optimize etmektedir (Yu & Chen, 2016). Swarm Intelligence'in endüstriyel robotik uygulamalarında kullanımı, sistemlerin ölçeklenebilirlik ve dayanıklılık gibi özelliklerini artırarak, daha verimli ve güvenilir çözümler sunmaktadır.

### **1.3.4 Otomatik malzeme ikmal sistemlerinin tasarımı**

Otomatik malzeme yenileme sistemlerinin tasarımı, üretim süreçlerini optimize etmek ve verimliliği artırmak amacıyla dikkatlice planlanmalıdır. Bu sistemlerin mimarisi genellikle üç ana bileşenden oluşur: giriş-çıkış birimleri, taşıma ve depolama sistemleri, ve kontrol yazılımı. Giriş-çıkış birimleri, malzemelerin sisteme alınmasını ve çıkartılmasını sağlar. Taşıma ve depolama sistemleri, malzemelerin üretim hatlarına doğru şekilde ve zamanda taşınmasını ve depolanmasını sağlar. Kontrol yazılımı ise tüm süreci koordine eder ve optimize eder (Li et al., 2023). Otomatik malzeme yenileme sistemlerinin donanım bileşenleri arasında robotik kollar, otomatik yönlendirme araçları (AGV), sensörler ve RFID sistemleri bulunur. Robotik kollar, malzemeleri almak ve bırakmak için kullanılırken, AGV'ler malzemelerin fabrika içinde taşınmasını sağlar. Sensörler ve RFID sistemleri, malzemelerin takibini ve yerleştirilmesini sağlar. Yazılım bileşenleri, genellikle bir gözetim kontrol ve veri toplama (SCADA) sistemi, insan-makine arayüzleri (HMI) ve optimize edilmiş algoritmalarдан oluşur. Bu yazılımlar, sistemin verimli çalışmasını sağlamak için gerçek zamanlı verileri kullanarak kararlar alır (Alnema et al., 2023).

Mevcut üretim hatlarına otomatik malzeme yenileme sistemlerinin entegrasyonu, dikkatli bir planlama ve değerlendirme gerektirir. Bu entegrasyon, mevcut sistemlerin donanım ve yazılım altyapısının incelenmesi ile başlar. Entegrasyon sırasında, malzeme akışının kesintisiz ve verimli olmasını sağlamak için sistemlerin uyumlu çalışması gereklidir. Ayrıca, çalışanların yeni sistemlere adapte olabilmesi için gerekli eğitimlerin verilmesi önemlidir. Bu süreçte, modüler ve esnek tasarımlar tercih edilerek, sistemlerin kolayca güncellenebilir ve genişletilebilir olması sağlanır (Hinrichs et al., 2023).

### **1.3.5 Simülasyon ve test ortamları**

Simülasyon, endüstriyel sistemlerin tasarıımında hayatı bir rol oynamaktadır. Gerçek dünya koşullarında test yapılmadan önce simülasyonlar, sistemlerin performansını değerlendirme ve potansiyel sorunları önceden tespit etme imkanı sunar. Bu, özellikle karmaşık ve yüksek maliyetli sistemlerin geliştirilmesinde büyük avantaj sağlar. Simülasyonlar, fiziksel testlerin yerini alarak maliyetleri düşürür ve tekrarlanabilir test koşulları sağlar. Örneğin, fizik tabanlı simülasyonlar, robotik sistemlerin performansını doğru bir şekilde modelleyerek, kontrol algoritmalarının etkinliğini ve güvenilirliğini artırır (Piliptchak et al., 2019).

3D prizmatik ızgara sistemleri, endüstriyel robotların simülasyon ve test ortamlarında sıkça kullanılan yapılar arasındadır. Bu sistemler, robotların üç boyutlu hareketlerini ve etkileşimlerini detaylı bir şekilde modellemeye olanak tanır. Prizmatik ızgara yapıları, çeşitli malzeme ve bileşenlerin depolanması ve taşınması için optimize edilmiştir. Bu sistemler, robotların hassas konumlandırma ve yönlendirme ihtiyaçlarını karşılayarak, üretim hatlarının verimliliğini artırır. Örneğin, Gazebo gibi açık kaynaklı 3D simülatörler, endüstriyel robot sistemlerinin dinamik ortamlar içinde nasıl performans gösterdiğini değerlendirmek için kullanılmaktadır (Aksu et al., 2018).

Endüstriyel robotlar için simülasyon tabanlı testler, çeşitli uygulama alanlarında büyük başarılar elde etmiştir. Örneğin, Florida International University (FIU) tarafından geliştirilen sanal gerçeklik (VR) tabanlı eğitim prototipi, öğrencilerin endüstriyel robotları güvenli ve düşük maliyetli bir ortamda öğrenmelerini sağlamıştır. Bu sanal öğrenme ortamı, robotların programlanması ve işletilmesi konusunda pratik deneyim sunarak, eğitim sürecini hızlandırmıştır (Peterson et al., 2022). Diğer bir örnek olarak, Gazebo ortamında gerçekleştirilen sanal deneyler, robotların karmaşık görevleri ve işbirlikçi hareketleri simüle ederek, gerçek dünya uygulamaları için gerekli olan kontrol algoritmalarının doğrulanmasını sağlamıştır (Aksu et al., 2018).

## 2. PROBLEM AŞAMALARI

Bu proje, alandaki uzmanlık ve deneyimlerden yararlanarak danışman öğretim üyesiyle titizlikle planlanmış çeşitli aşamalardan oluşmaktadır. Her aşama, proje hedeflerine ulaşmak için sistematik bir yaklaşım sunar ve teorik çalışmaların pratik uygulamalarla entegrasyonunu optimize etmeyi amaçlar. Bu yaklaşıyla, her bir adım, projenin genel başarısına katkıda bulunacak şekilde yapılandırılmıştır.

### 2.1 Ayrık Zaman Yapay Zekâ Algoritmaları

Projenin ilk ve en önemli aşamalarından biri, yapay zekâ algoritmalarının ayrık zamanda tasarlanması ve Unity oyun motoru ortamında eş zamanlı simülasyonunun gerçekleştirilmesidir. Bu ortam ve araçlar, mühendisin yetkinliği ve araçların projedeki etkinliği ve efektifliği göz önünde bulundurularak seçilmiştir. Tasarım ve simülasyon süreci, fonksiyonellik açısından üç ana aşamaya ayrılmıştır:

- **Algoritma tasarımı:** Bu aşamada, yapay zekâ algoritmalarının temel prensipleri belirlenir ve matematiksel modeller oluşturulur. Ayrık zaman dilimlerinde çalışacak şekilde optimize edilen bu algoritmalar, robotların malzeme yenileme süreçlerinde en verimli kararları almalarını sağlamak üzere tasarlanmıştır. Bu süreç, teorik bilgilerin pratik uygulamalarla harmanlandığı titiz bir çalışma gerektirir.
- **Simülasyon ortamının kurulumu:** Algoritmaların performansını değerlendirmek ve doğrulamak amacıyla, Unity oyun motoru kullanılarak detaylı bir simülasyon ortamı oluşturulur. Bu ortam, öngörülen koşullarını en iyi şekilde taklit ederek algoritmaların farklı senaryolarda nasıl performans göstereceğini analiz etmeye olanak tanır. Bu aşamada simülasyon parametreleri dikkatlice ayarlanır ve test edilir.
- **Eş zamanlı simülasyon ve analiz:** Son aşamada, yapay zekâ algoritmaları simülasyon ortamında eş zamanlı olarak çalıştırılır ve performansları detaylı bir şekilde analiz edilir. Bu süreç, algoritmaların etkinliğini ve doğruluğunu değerlendirmek için kritik öneme sahiptir. Simülasyon sonuçları, algoritmaların nasıl performans göstereceğine dair değerli bilgiler sunar ve gerekli iyileştirmelerin yapılmasına olanak tanır.

### **2.1.1 İlk aşama: tekli komut verilmesi**

AI modelinin şu davranışını zamanla öğrenmesi istenir: Tek bir robot belirli bir başlangıç pozisyonundan 3D ızgara alanında hedef konuma tek bir kutuyu en kısa yoldan ve belirlenmiş kurallara göre taşıyacaktır. Eğer seçilen kutunun üzerinde başka bir kutu varsa, öünü açacak şekilde bu aşama için rastgele bir yere bırakılır.

İstenen davranışa uygun kurallar doğrultusunda algoritma oluşturulmuş, iteratif şekilde simülasyonlardaki sonuç ve grafiklerden de yararlanılarak sayısal hesaplamalar optimize edilmiştir. Bu aşamada elde edilen veriler ve algoritmanın kendisi, tasarımin çekirdeğini temsil ettiğinden sonraki aşamalarda daha karmaşık görevler için nasıl geliştirileceğine dair önemli ipuçları sağlamıştır. Bu nedenle, tekli komut verilmesi aşaması, hem robotların temel yetkinliklerinin pekiştirilmesi hem de daha ileri seviye görevler için sağlam bir temel oluşturulması açısından büyük önem taşır. Kurallar ve algoritma ilgili bölümlerde daha detaylı anlatılacaktır.

### **2.1.2 İkinci aşama: sıralı komut verilmesi**

İkinci aşamada, tek bir robotun bir dizi kutuyu yeniden konumlandırmak üzere talimatlandırılması hedeflenir. Bu aşama, robotun daha karmaşık görevleri yerine getirme yeteneğini test etmek ve geliştirmek amacıyla tasarlanmıştır. Robot, belirli bir sıralama ile birden fazla kutuyu taşıırken her adımda en uygun hareket stratejisini zamanla öğrenmek zorundadır. Örneğin, seçilen bir kutunun üstünde başka bir kutu varsa, bu kutu hareket ettirilirken diğer hedef kutuların en kısa yoldan gidebilmesi engellenmeden, yani diğer kutuların yolları önüne konulmadan taşınabilmesi sağlanmalıdır.

Robot, bu süreçte her kutunun en verimli şekilde yeniden yerleştirilmesini sağlamak için çeşitli algoritmalar kullanır. Bu algoritmalar, robotun sadece belirli bir kutuyu taşımakla kalmayıp tüm dizi içindeki diğer kutuların da en uygun şekilde yerleştirilmesini hedefler. Bu sayede, robotun daha karmaşık senaryolarla başa çıkabilme kapasitesi artırılır ve aynı zamanda süreçlerin optimizasyonu sağlanır.

Bu aşamada, robotun çevresel faktörlere karşı hızlı ve doğru tepkiler verebilme yeteneği de geliştirilir. Robot, her bir adımda yeniden konumlandırma işlemlerini gerçekleştirirken tüm süreci değerlendirecek ve en verimli yol haritasını oluşturacaktır. Böylelikle, sıralı komut verilmesi aşaması, robotun sadece bireysel görevleri değil aynı zamanda ardışık ve birbirine bağlı görevleri de başarıyla

yönetebilme yeteneğini güçlendirir. Bu aşamanın başarılı bir şekilde tamamlanması, robotun daha karmaşık üretim hattı senaryolarında etkili bir şekilde çalışmasını sağlayacak önemli bir adımı temsil eder.

### **2.1.3 Üçüncü aşama: birden fazla robotun iş birliği yapması**

Üçüncü aşama, robotlar arasında sürü zekâsı davranışının mantığını ve uygulamasını keşfederek projemizin daha karmaşık ve iş birliğine dayalı yapay zekâ işlevselliklerine doğru önemli bir ilerleme kaydetmesini sağlamaktadır. Bu aşama, robotların birlikte çalışarak daha büyük ve karmaşık görevleri nasıl yerine getirebileceklerini öğrenmelerini hedefler.

Tek bir robotun daha büyük bir harita üzerinde komutları yerine getirmede yetersiz kalabileceği düşüncesiyle sürü zekâsı tasarımını ve uygulamasını hayatı geçirmeye karar verdik. Bu geçiş, çoklu robotların kullanımını gerektirmekte olup, bu robotların seri komutları birlikte çalışarak yerine getirmeleri, verimlilik ve kapsama alanını artırmaktadır.

Bu aşamada, birden fazla robot, birinci ve ikinci aşamalarda öğrenilen görevleri iş birliği içinde gerçekleştirecektir. Her robot, diğer robotlarla koordine olarak en uygun hareket stratejilerini zamanla öğrenerek görevlerini yerine getirecektir. Bu iş birliği, robotların birbirlerinin konumlarını ve hareketlerini dikkate alarak en verimli yolu bulmalarını ve görevlerini engellemeden tamamlamalarını sağlayacaktır.

Robotlar arasındaki bu iş birliği, sadece görevlerin daha hızlı ve etkili bir şekilde tamamlanmasını sağlamakla kalmayıp, aynı zamanda sistemin genel performansını ve adaptasyon yeteneğini de artıracaktır. Bu aşama, robotların daha karmaşık üretim hattı senaryolarında birlikte çalışarak nasıl başarılı olabileceklerini göstermesi açısından kritik öneme sahiptir. Sürü zekâsı algoritmaları ve öğrenme süreçleri, tek bir robottaki bir hatada tolerans sağlayarak esnek bir üretim hattı oluşturmasına olanak tanıyacaktır.

Sonuç olarak, üçüncü aşamada birden fazla robot, birinci ve ikinci aşamalarda gerçekleştirilen işleri iş birliği içinde yaparak projede önemli bir ilerleme sağlayacak ve daha karmaşık görevleri yerine getirmek için gerekli olan öğrenme ve adaptasyon becerilerini geliştirecektir.

## **2.2 Sürekli Zaman Entegrasyonu**

Bilgisayar ortamında gerçekleşmiş, test edilmiş ve sonucu bilinen algoritmaları sürekli ortamdaki robotlara yüklemek için çeşitli çalışmalar üç aşama halinde gerçekleştirilecektir.

### **2.2.1 Robot tasarıımı ve hazırlığı**

Belirlenen çevre ve çalışma şartlarına en uygun robot birimi, tüm mekanikleriyle birlikte Gazebo ortamında yaratılacaktır. Gerekli sensör, aktüatör ve kontrolör bağlantıları çeşitli Gazebo eklentileri (pluginleri) ile yapılacak ve gerekli ROS Node'ları yazılarak çalışmaya hazır hale getirilecektir. Bu aşamada, robotun tasarıımı ve hazırlanması sırasında mekanik ve yazılımsal bileşenlerin entegrasyonu üzerinde çalışılacaktır. Robotun hareket kabiliyeti, çevre algılama yetenekleri ve kontrol sistemleri dikkatlice oluşturulacaktır.

### **2.2.2 Navigasyon ve lokalizasyon**

Platformda yürüyen kara robottu, kutuları almak için her gridde belirlenen x-y pozisyonlarına gelmelidir. Bu hareket için aracın lokalizasyon ve haritalama işlemleri yapılacak, daha sonra ise tam o pozisyona gelmesi için navigasyon ve pozisyon kontrol yaklaşımları kullanılacaktır. Robotun doğru pozisyonlara ulaşabilmesi için haritalama algoritmaları ve lokalizasyon teknikleri geliştirilecek, ardından navigasyon stratejileri uygulanacaktır.

### **2.2.3 AI algoritmalarının entegrasyonu**

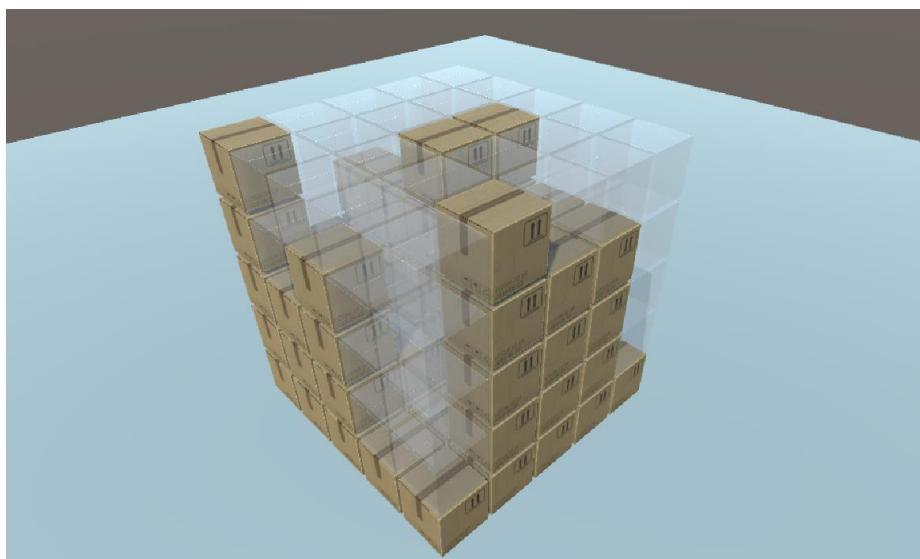
AI algoritmaları, Gazebo ortamında da test edilmek üzere sisteme ve robotlara entegre edilecektir. Bu entegrasyon sürecinde, algoritmaların robotun donanımı ve yazılımı ile uyumlu çalışması sağlanacaktır. AI algoritmaları, robotların görevlerini yerine getirme sürecinde karşılaşabilecekleri farklı senaryolara uyum sağlamak üzere tasarlanacak ve test edilecektir. Sistemin performansını değerlendirmek ve optimize etmek amacıyla simülasyon sonuçları analiz edilecektir.

### **3. TASARIM VE ALGORİTMA GENEL BAKIŞI**

Danışman öğretim üyesinin önerisi ve yönlendirmesi ile proje kapsamında yapay zekâ (AI) bölümne öncelik verilmiş ve yıl boyunca bu alanda yoğun çalışmalar yürütülmüştür. Bu süreçte, AI algoritmalarının geliştirilmesi ve optimizasyonu üzerine odaklanılarak, simülasyon ortamlarında detaylı testler gerçekleştirilmiştir. Algoritmaların tasarımı ve geliştirilmesi, proje hedeflerine ulaşmada temel bir rol oynamış ve üretim süreçlerine entegre edilecek robotik sistemlerin verimliliğini ve etkinliğini artırma potansiyelini göstermiştir.

#### **3.1 Tasarlanan Ortam ve Koşullar**

Projede kullanılan algoritmaların tasarımı ve test süreçleri, belirli varsayımlar ve ihtiyaçlar göz önünde bulundurularak gerçekleştirilmiştir. Bu süreçlerin hem daha görselleştirilebilir hem de hata ayıklama (debug) işlemlerinin hızlandırılabilir olması amacıyla, Unity oyun motoru kullanılmış ve C# dili tercih edilmiştir. Unity'nin esnek ve güçlü yapısı, algoritmaların doğruluğunu ve etkinliğini değerlendirmek için mükemmel bir platform sunar. Bu sayede, algoritmaların performansı gerçekçi ve dinamik bir ortamda test edilerek, pratik uygulamaların daha etkin ve güvenilir bir şekilde geliştirilmesi sağlanmıştır. Şekil 3.1'de bu ortam görülebilmektedir.



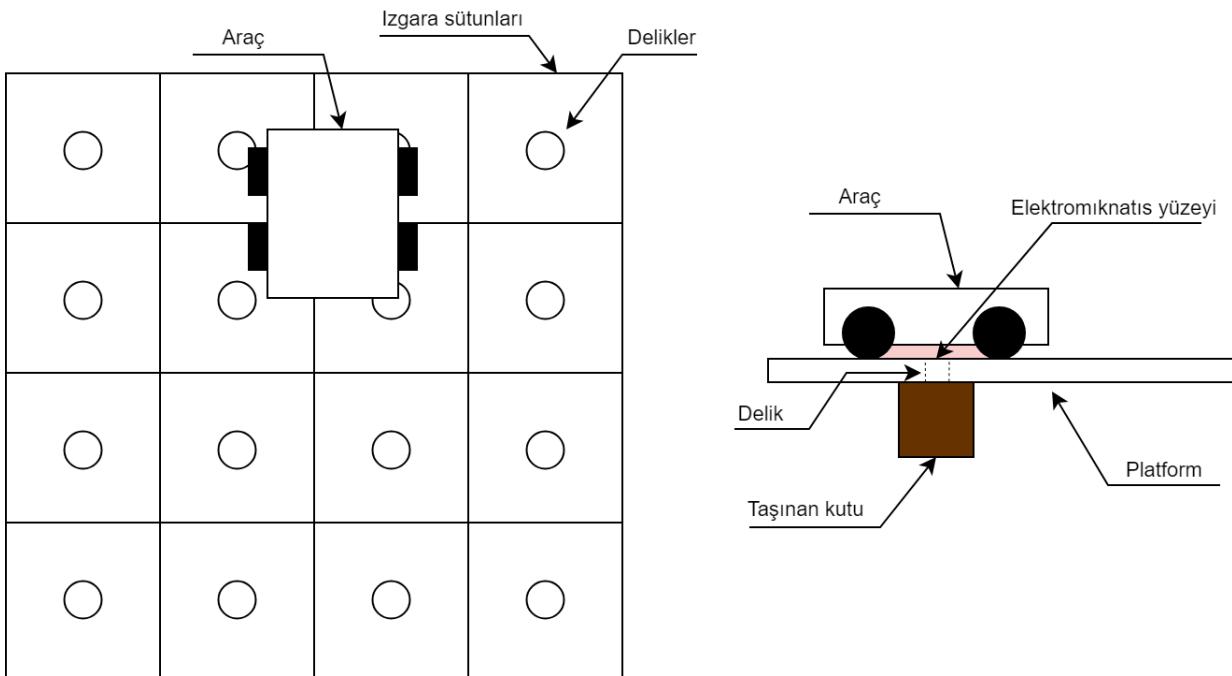
**Şekil 3.1 :** Algoritmanın uygulandığı unity ortamının görseli

Hazırlanan program ile her başlatıldığında verilen nxn boyutunda, her seferinde rastgele doluluk oranına sahip bir kutu ızgara sistemi yaratılmaktadır. Bu ızgara sistemi, algoritmanın kutular üzerindeki hareketlerini ve etkileşimlerini gerçekçi bir şekilde simüle etmekte, bu sayede algoritmaların pratik uygulanabilirliği üzerine fikir vermektedir.



**Şekil 3.2 :** Kutuların ve şeffaf hareket alanlarının detaylı görseli

Robotun gerçek hayatı dört tekerlekli bir kara aracı olarak tasarlanması ve kutuların üzerinde yer alan bir platformda bulunması planlanmıştır. Bu düzenleme, aracın etkin bir şekilde 2D hareket edebilmesi ve aynı zamanda kutuları taşıyabilmesi için optimize edilmiştir. Platformdaki her bir kutu sütununun üstünde, robotun kolunun kutuyu yukarı çeker bilmesi için küçük delikler yerleştirilmiştir. Bu delikler, aynı zamanda aracın hareketini engellemeden maksimum erişim sağlamak üzere tasarlanmıştır. Robot, kol mekanizması aracılığıyla kutuyu yukarı çekerken, kutu en üst noktaya ulaştığında elektromagnit yardımıyla tutacak ve kol ayrıldıktan sonra manyetik olarak bağlı kalan kutu, platform üzerinde sürüklenebilecektir. Planlanan yapının kuşbakışı ve yandan temsili görünümü Şekil 3.3'te gösterilmiştir.



**Şekil 3.3 :** Planlanan yapının kuşbakışı ve yandan temsili görünümü

Ancak, bu proje kapsamında kolun veya manyetik mekanizmanın gerçek hayatı nasıl tasarlanacağına dair çalışmalar yapılmamıştır. Bu unsurlar, sisteme gerçekçilik katmak ve konseptin anlaşılabilirliğini artırmak amacıyla düşünülmüş detaylardır. Projenin temel odak noktası, yapay zekâ algoritmalarının geliştirilmesi ve robotik hareketlerin simülasyon ortamında etkin bir şekilde gerçekleştirilmesidir.

Ancak bu detaylar, kutunun taşınırkenki hareket dinamiklerinin etkin bir şekilde yönetilmesi için projenin başlangıcında dikkatle planlanması gereken önemli unsurlardır. Belirtilen konfigürasyonda, aracın hareketleri esas olarak 2 boyutlu düzlemede gerçekleşeceğin şekilde tasarlanmıştır: sağa, sola, yukarı ve aşağı doğru hareketler ile birlikte, üçüncü eksende kutunun yükseltilmesi ve alçaltılması işlemleri için kurallar tanımlanmıştır. Bu, aracın hem yatay hem de dikey eksenlerde etkin bir şekilde manevra yapabilmesini sağlar ve kutuların yer değiştirme süreçlerinde rol oynar.

Detayların önemli olmasının sebebi ise, araç tarafından gerçekleştirilecek bu hareketler, pekiştirmeli öğrenme (RL) algoritmalarında tanımlanan aksiyonlarla uyumlu bir şekilde entegre edilmiştir. RL algoritmaları, aracın bu hareketlerini kullanarak kutuları izgaradaki hedef konumlarına en etkin şekilde taşıyacak stratejiler

geliştirecek ve bu süreçte kazanılan deneyimleri kullanarak sürekli olarak iyileştirmeler yapacaktır. Bu yaklaşım, hem gerçekçi bir simülasyon ortamı sağlamakta hem de aracın hareket kabiliyetini ve malzeme taşıma süreçlerinin verimliliğini maksimize etmek amacıyla kritik bir rol oynamaktadır.

### **3.2 Q Learning**

Q Learning algoritması, pekiştirmeli öğrenme yöntemlerinden biridir ve ajanların belirli bir ortamda ödül kazanarak optimal aksiyonları öğrenmesini sağlar. Bu algoritma, ajanların uzun vadeli ödülleri maksimize etmeyi hedefleyen kararlar almasını ve bu doğrultuda stratejiler geliştirmesini sağlar. Q Learning, her bir durum ve aksiyon çifti için bir değer belirleyerek ajanların en iyi hareket tarzını zamanla öğrenmesine olanak tanır. Projenin, sisteme karar verdirme aşamasının büyük bölümünde, altında Q Learning mantığı bulunan algoritmalar en başından itibaren tasarlanmış ve uygulanmıştır. Burada bahsedilen ajanlar, Q Learning algoritmalarının özneleridir. Örneğin 1. aşamadaki ajan, herhangi bir kutuyu taşımakta olan robottur.

#### **3.2.1 Genel program mantığı akışı**

Projedeki adımlarda geliştirilen Q Learning algoritmaları, belirli bir program akış mantığı çerçevesinde çalışmaktadır. Bu akış, algoritmanın başlangıcından eğitimin tamamlanmasına kadar olan süreci kapsamaktadır. Veritabanı olarak problem adımlarında Q Tablosundan başka, gereksinime uygun, benzer mantıkta yapılar da kullanılmıştır ancak program akışı anlatılırken temsili olarak Q Tablosu olarak bahsedilecektir. Durumlar ve aksiyonlar 1. aşamaya uygun olarak anlatılacak, 2. ve 3. aşamadaki spesifikasyonlar buradaki referanslarla beraber daha sonraki bölümlerde detaylandırılacaktır.

- **Başlangıç aşaması:** Algoritma başladığında, öncelikle Q Tablosu başlangıç değerleriyle (sıfır) başlatılır. Bu tablo, algoritmanın her durum (1. aşama için ajanın taşıdığı kutunun ayrik zamanda pozisyon vektörü) ve altı adet aksiyon (yine 1. aşama için x-z düzleminde sağ-sol-yukarı-aşağı ve y ekseninde kutuya çekme ya da indirme hareketleri) için bir değer atamasını içerir ve eğitim sürecinde sürekli olarak güncellenir.

- **Güncelleme döngüsü:** Algoritma, belirlenen zaman aralıklarında (frame) güncellenir. Her güncelleme, belirli bir zamanlayıcıya dayalı olarak gerçekleşir ve bu zamanlayıcı, simülasyonun animasyon süresine uygun olarak ayarlanır. Döngü içerisinde belirlenen adım sayısı kadar aşağıda maddelenen adımlar takip edilir.
  - **Aksiyon seçimi ve uygulama:** Ajan, kendi mevcut durumuna (1. aşamada taşıdığı kutunun hangi pozisyonda olduğuna) dayanarak bir aksiyon seçer ve bu eylemi gerçekleştirmeye çalışır. Bu süreç, ajanın seçtiği aksiyon doğrultusunda kutuyu hedefine doğru ilerletmesi ile devam eder.
  - **Ödül hesaplama ve öğrenme:** Her hareketten sonra ajan, gerçekleştirdiği eyleme göre bir ödül alır. Bu ödül, Q Learning yöntemine göre hesaplanır ve Q Tablosunda ilgili parametreleri güncelleyerek ajanın öğrenme sürecine katkıda bulunur.

Daha sonra belirlenen adım sayısına ulaşıldıysa (kutu ulaştırılamadıysa) ya da kutu belirlenen hedefe ulaştırıldıysa aşağıdaki adım takip edilir.

- **Yeni bölüm başlatma:** Her bölümün tamamlanmasının ardından, algoritma yeni bir bölüme geçer. Yeni bölümde ajanın ve kutunun konumu sıfırlanır ve ızgara kutu yapısı ilk haline döner. Eğer eğitimin sonuna gelindiyse, algoritma gerekli sonlandırma işlemlerini gerçekleştirir.

### 3.2.2 Aksiyon seçimi

Q Learning algoritmasında aksiyon seçimi, keşfetme (exploration) ve deneyimleri kullanma (exploitation) prensiplerine dayanır. Bu süreç, ajanın karar verme mekanizmasının temelini oluşturur ve öğrenme performansını doğrudan etkiler.

- **Keşfetme (exploration) ve keşfetme oranı:** Keşfetme esnasında ajan, yeni hareketler keşfetmeyi amaçlar. Bu nedenle algoritma keşfetme modundayken rastgele aksiyon seçimi etkindir. Bu süreç, ajanın bilinmeyen veya daha az tercih edilen eylemleri denemesi ile sonuçlanır. Keşfetme ya da deneyimleri kullanma, epsilon değerine bağlı bir koşul ile yönetilir; rastgele bir sayı,

epsilon değerinden küçükse, ajan keşfetme modunda hareket eder. Her bölümde keşfetme oranı yani epsilon değeri azalır (decay), bu da zamanla ajanın daha az keşfetmesi ve daha fazla bilinen eylemleri kullanmasını sağlar. Bu düşüş, ajanın deneyim kazandıkça daha öngörelebilir ve optimize edilmiş kararlar almasına, böylece seçimlerde bir yakınsamaya olanak tanır.

- **Deneyimleri kullanma (exploitation):** Ajan, daha önce öğrendiği bilgileri kullanarak bulunduğu durum için en iyi bilinen eylemi gerçekleştirir. Eğer rastgele seçilen sayı epsilon değerinden büyükse, ajan deneyim kullanma moduna geçer. Bu durumda, Q Tablosu incelenir ve mevcut duruma en yüksek Q değerine sahip olan eylem seçilir. Deneyimleri kullanma, ajanın daha önceki deneyimlerinden yararlanmasını ve öğrenilen en iyi hareketi yapmasını sağlar.

Aksiyon seçimi süreci, ajanın hem yeni stratejiler keşfetmesine hem de ögrendiklerini uygulayarak hedefine en etkili şekilde ulaşmasına olanak tanır. Epsilonun azalmasıyla birlikte, ajanın davranışları zamanla keşfetmeden tecrübeye doğru kayar. Bu dengeli yaklaşım, ajanın hem yeni ve potansiyel olarak daha iyi çözümleri keşfetmesine hem de mevcut bilgisini etkili bir şekilde kullanmasına imkân tanır. Bu süreç, ajanın karar verme yeteneğinin sürekli olarak gelişmesini sağlar ve algoritmanın genel başarısını artırır.

### 3.2.3 Keşfetme Oranının Zamanla Azalması

Epsilon değeri, Q Learning algoritmasının keşfetme ve deneyimleri kullanma dengesini yöneten önemli bir parametredir. Epsilon değeri, başlangıçta belirlenen bir değerden başlar ve her bölümle birlikte azalmaya başlar. Bu azalma süreci, iki farklı yöntemle ele alınmıştır:

- **Doğrusal Azalma Yöntemi:** Bu yöntemde, epsilon değeri her bölümde belirli bir oranla çarparak azaltılır. Epsilonun minimum bir değere ulaşmasını garanti etmek için, belirlenen minimum epsilon değeriyle karşılaştırılır ve bu iki değerden daha büyük olanı seçilir.
- **Sıcaklık Fonksiyonu ile Azalma Yöntemi:** Bu alternatif yaklaşımada, mevcut durum ve aksiyona bağlı bir sıcaklık fonksiyonu kullanılarak epsilonun

azaltılması denemiştir. Bu yöntem, algoritmanın daha karmaşık senaryolarda farklı eylemleri denemesini sağlamayı amaçlar. Ancak, bu yöntemin uygulanması sırasında dikkate değer bir performans değişimi gözlemlenmemiştir.

Sonuç olarak, daha basit ve öngörlülebilir olan ilk yöntemle devam edilmesine karar verilmiştir. Doğrusal azalma yöntemi, algoritmanın zamanla daha az keşfetmesini ve öğrendiği bilgileri daha fazla kullanmasını sağlayarak keşfetme ve deneyimlerden faydalanan dengesini etkin bir şekilde yönetir. Bu yaklaşım, algoritmanın genel performansını ve karar alma sürecinin etkinliğini artırırken, aynı zamanda basitlik ve anlaşılırlık açısından da avantaj sağlar.

### **3.2.4 Ödül / ceza alma mekanizması**

Tasarlanan algoritmalarla ödül/ceza alma mekanizmaları, ajanın öğrenme sürecinin temelini oluşturur. Bu mekanizmalar, ajanın her hareketini değerlendirek ona bir ödül veya ceza verir. Ödülün miktarı, ajanın gerçekleştirdiği hareketin ve o hareketin sonucunun verimliliğine göre belirlenir.

**Ödül Kurallarının Belirlenmesi:** Bu süreç, aslında Q Learning problemi tanımlamakla eşdeğerdir. Ajanın hareketlerini yönlendirmek ve belirli hedeflere ulaşmasını sağlamak için ödül kuralları dikkatlice belirlenir. Proje kapsamında, her bir aşamadaki algoritamlar için bu kurallar, belirlenen problemlere göre deneysel ve iteratif bir yaklaşımla oluşturulmuştur. Her bir kural, zamanla Q tablosunu şekillendirir.

**Kuralların Değerlendirilmesi ve İterasyonu:** Her bir kural eklendikten sonra, ajanın performansı ve Q Tablosundaki değişiklikler incelenir. Bu, ajanın belirli bir davranışını sergileyip sergilemediğini, o harekete yaklaşıp yaklaşmadığını ve öğrenme sürecinin etkinliğini değerlendirmek için kullanılır. Kurallar, ajanın istenen davranışlara doğru ilerlemesini sağlamak amacıyla üst üste eklenir ve sürekli olarak güncellenir.

**Sonuçların İzlenmesi ve İyileştirilmesi:** Her iterasyonda ajanın davranışları ve Q tablosundaki değişiklikler gözlemlenir. Bu, ajanın öğrenme sürecinin etkinliğini

artırmak ve daha iyi sonuçlar elde etmek için algoritmadaki gerekli ayarlamaların yapılmasını mümkün kılar.

Ödül/ceza alma mekanizması, ajanın öğrenme sürecinin temelini oluşturur ve onun daha etkili ve verimli hareket etmesini sağlar. Bu süreç, ajanın sürekli olarak gelişmesine ve belirlenen hedeflere ulaşmasına olanak tanır.

### 3.2.5 Öğrenme Mekanizması

Algoritmada öğrenme süreci, ajanın deneyimlerine dayanarak Q-tablosundaki Q-değerlerini güncellemesiyle gerçekleşir. Q-değeri, belirli bir durumda belirli bir aksiyon alınan beklenen faydasını temsil eder ve hem anlık hem de gelecekteki ödülleri hesaba katarak öğrenme aşamasında şekillenir.

Q Learning güncelleme kuralının denklem 3.1'de ifade edilmiştir.

$$Q(s, a) \leftarrow (1 - \alpha) \times Q(s, a) + \alpha \times (r + \gamma \max(Q(s', a'))) \quad (3.1)$$

#### Denklemenin açıklaması:

- **Eski Q-değeri ( $Q(s, a)$ ):** Eski durumda ( $s$ ) alınan aksiyon ( $a$ ) için Q-değерidir ve ajanın bu güncellemeden önceki bilgisini temsil eder.
- **Öğrenme oranı ( $\alpha$ ):**
  - $\alpha$ , yeni bilginin güncellenmiş Q-değeri üzerindeki etkisini belirler.
  - Yüksek bir  $\alpha$ , algoritmanın yeni bilgiye daha fazla öncelik vermesini sağlarken, düşük bir  $\alpha$ , eski verilere daha fazla ağırlık verir.
- **Anlık ödül ( $r$ ):**
  - Aksiyon alındıktan sonra hemen sonra elde edilen kazançtır.
  - Aksiyonun doğrudan sonucunu yansıtır.
- **İndirim faktörü ( $\gamma$ ):**
  - $\gamma$ , 0 ile 1 arasında bir değerdir ve yeni durumdaki aksiyonlar içinde en yüksek Q-değерinin şuanki durumdaki Q değerine etkisinin bir göstergesidir.
  - Yüksek bir  $\gamma$ , ajanın gelecekteki ödülleri daha fazla önemsemesini sağlayarak modelin daha ileri görüşlü olmasını sağlar.

- **Maksimum gelecekteki Q-değeri ( $\max(Q(s', a'))$ ):**
  - Yeni durumdaki olası tüm aksiyonlar için en yüksek Q-değerini temsil eder.
  - Yeni durumdan itibaren elde edilebilecek optimal gelecekteki ödüllerin bir tahminidir.
- **Anlık ve gelecekteki ödüllerin birleştirilmesi:**
  - İfade  $(r + \gamma \max(Q(s', a')))$ , anlık ödülü gelecekteki ödüllerle indirimli şekilde birleştirir.
  - Bu toplam, mevcut durumda alınan aksiyonun toplam beklenen faydasını temsil eder.
- **Q-değerinının güncellenmesi:**
  - Nihai güncellenmiş Q-değeri, eski değerle bu yeni bilginin bir karışımıdır ve öğrenme oranı  $\alpha$  tarafından modüle edilir.

### Gamma ( $\gamma$ ) parametresinin önemi

- $\gamma$  parametresi, ajanın "görüşünü" artırarak, sadece anlık sonuçlara odaklanmak yerine uzun vadeli sonuçları da göz önünde bulundurmasını sağlar.
- Gelecekteki ödüllerin önemli olduğu karmaşık karar verme görevlerinde, daha yüksek bir  $\gamma$ , ajanın daha stratejik planlama yapmasına ve eylemlerinin uzun vadeli faydalarını daha fazla göz önünde bulundurmasına yardımcı olur.

Özetle, Q Learning'deki öğrenme mekanizması, özellikle anlık ve gelecekteki ödülleri dengeli bir şekilde değerlendiren  $\gamma$  parametresi, ajanın optimal davranışları öğrenme yeteneği için öneme sahiptir. Bu mekanizma, verilen görevin özel zorluklarına ve hedeflerine uygun şekilde ince ayarlanarak etkili ve verimli öğrenmeyi sağlar.

### Özel problemlere göre parametre ayarlaması

- Q Learning algoritmasındaki bu öğrenme mekanizması, projedeki her bir spesifik probleme göre optimize edilmiştir. Örneğin 1. aşamada, kutuların kurallara bağlı hareket ettirilmesi gibi belirli bir hareket problemi için parametreler deneysel ve iteratif bir yaklaşımla ayarlanmıştır.
- Bu optimizasyon, ajanın sadece anlık ödülleri (hedefe ulaşma gibi) değil, aynı zamanda eylemlerinin geniş bağlamını da hesaba katmasını sağlar.

## **4. YAPAY ZEKÂ VE ÖĞRENME MODELİ TASARIMI**

Birinci, ikinci ve üçüncü aşamalarda ele alınan problemlerin her biri için, bahsedilen öğrenme mantığının nasıl kullanıldığı, nelerin neden gerektiği ve nasıl bir yöntemle uygulandığı alt başlıklar halinde aşamaya göre incelenecaktır. Her bir aşamanın özgün gereksinimleri ve bu gereksinimlerin yapay zekâ ve öğrenme modeli tasarımına olan etkisi ayrıntılı bir şekilde ele alınacaktır. Ayrıca, kullanılan yöntemlerin algoritmalar üzerindeki etkisi ve performansı değerlendirilecek, bu sayede projenin genel başarısına katkıda bulunan temel unsurlar vurgulanacaktır.

### **4.1 Birinci Aşama: Tekli komut öğrenimi**

İlk aşamada robotun amacı tek bir kutunun belirlenen kurallara uyarak 3D bir ızgara içinde belirli bir noktaya en kısa sürede ulaşmasını sağlamayı öğrenmesidir. Bu aşamada kullanılan Q-tablosu, durumların aksiyonlarla eşleştirilmesini içerir; burada her bir durum, ızgara içindeki bir pozisyonu temsil ederken her aksiyon, kutunun yapabileceği bir hareketi ifade eder.

Bu ortamda tek bir kutuyu taşıyan roboutn birincil amacı, kutuyu en yüksek düzleme çıkarmak, hedeflenen sütuna taşımak ve indirmek için gerekli adımları atmaktır. Bu hedefe ulaşmak ve performansı en üst düzeye çıkarmak için ödül sistemini kademeli olarak dengeleyip ince ayar yapılmıştır. Bu düzenlemeler, istenen sonuca uygun davranışları teşvik ederken sapmaları caydırma amacıyla titizlikle uygulanmıştır. Her bir değişikliğin mantığını ve bunların genel hedefe ulaşmaya nasıl katkıda bulunduğuunu açıklayan detaylı açıklamalar, belgenin ilerleyen bölümlerinde sunulacaktır.

#### **4.1.1 Ödül kuralları**

Ödül sisteminin ve kurallarının adım adım nasıl geliştirildiği açıklanmıştır.

##### **4.1.1.1 İlk etap ödül kuralları ve sonuçları**

İlk aşamada, kutunun en kısa yolu izlemesini teşvik etmek amacıyla ödül ve ceza kuralları belirlenmiştir. Bu kurallar, ajanın hareketlerini ve kararlarını etkileyerek zaman içinde kutuyu hedefe en etkili şekilde ulaştırmayı öğrenmeyi sağlamayı hedefler.

- **Her adım için küçük bir ceza:** Kutu, her bir birim hareket ettiğinde, küçük bir miktar ceza alır. Bu, her hareketin bir maliyeti olduğu fikrini yansıtır ve kutuyu hedefe en az adımda ulaştırmayı teşvik eder. Bu yaklaşım, ajanın gereksiz hareketlerden kaçınmasını ve daha doğrudan yollar aramasını sağlar.

Matematiksel olarak bu ceza Denklem 4.1'deki gibi ifade edilir:

$$r_s = -0.05 \quad (4.1)$$

- **Hedefe ulaşma ödülü:** Eğer kutu hedefine başarıyla ulaşırsa, bu başarıyı teşvik etmek için önemli bir ödül alır. Bu ödül, ajanın kutuları hedefe ulaştırma motivasyonunu artırır ve doğru yolu bulmasını teşvik eder. Matematiksel olarak bu ödül Denklem 4.2'deki gibi ifade edilir:

$$r_h = 1.0 \quad (4.1)$$

- **Hareket edememe durumunda ceza:** Kutu, herhangi bir engel veya kısıtlama nedeniyle hareket edemediğinde, bu durum için daha büyük bir ceza alır. Bu ceza, kutuların hareket edememe durumlarını yapacak hareketleri önlemeye çalışmasını sağlar ve özellikle kutunun karşılaşacağı engelleri aşmak için alternatif yollar aramasını teşvik eder. Matematiksel olarak bu ceza Denklem 4.3'teki gibi ifade edilir:

$$r_e = -0.5 \quad (4.3)$$

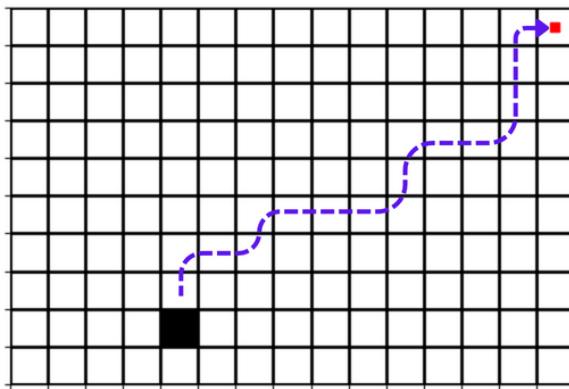
Bu ödül ve ceza sistemine göre, bir adımda alınan toplam ödül  $r$ , denklem 4.4'deki gibi hesaplanır:

$$r = r_s + r_h + r_e \quad (4.4)$$

Bu ilk kurallar, kutunun hedefe en etkili ve verimli yoldan ulaşmasını sağlamak için tasarlanmıştır. Ajanın her hareketinin bir maliyeti olduğunu ve hedefe ulaşmanın önemli bir ödül getirdiğini vurgulayarak ajanın karar verme sürecini etkili bir şekilde yönlendirirler.

**İlk etap ödül kuralları sonucunda**, seçilen kutu hedef konuma, bir bölüm adedinden sonra, optimize bir rotaya oturtulabilmektedir ancak daha önce belirtilen kutunun önce yukarı çıkma, daha sonra en üst düzlemede ilerleme ve hedefin olduğu sütunda aşağı indirilme kurallarına uymaz.

Ek olarak, projenin ilk aşamalarında ilk etap kurallarını denemek ve QLearning'in derinlemesine anlaşılabilmesi için geliştirilen Şekil 4.1'de görülen 2 boyutlu başka bir simülasyon projesinde de kuralların sonuç verdiği görülmüştü. Matplotlib kütüphanesi ile oluşturulan bu simülasyonda da amaç bir kareden diğerine en kısa mesafe ile gitmekti.



**Şekil 4.1 :** İlk etap ödül sisteminin denendiği diğer simülasyon ortamı

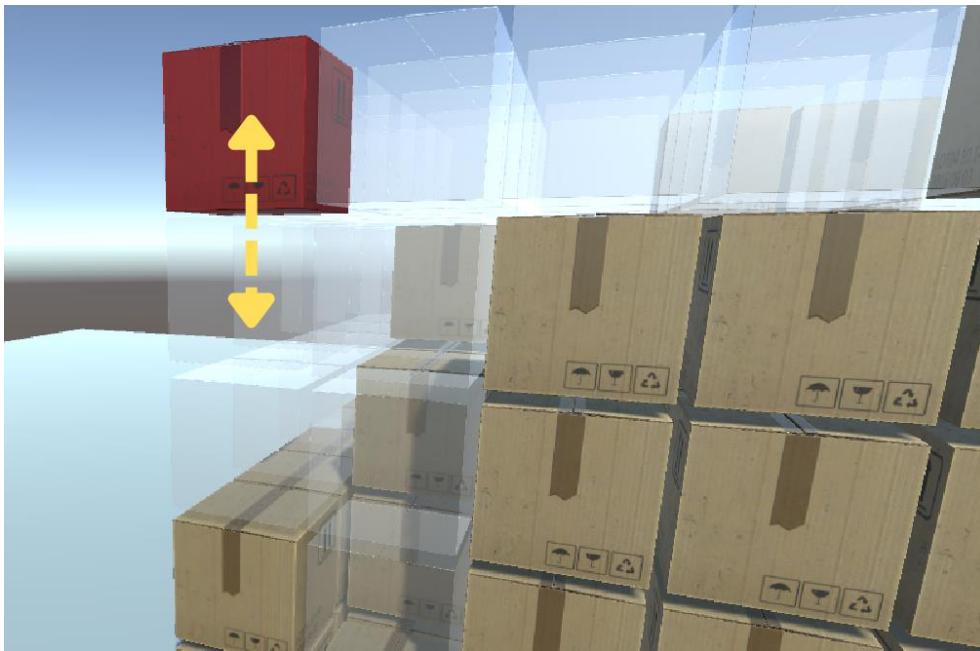
#### 4.1.1.2 İkinci etap ödül kuralları ve sonuçları

İkinci aşamada, kutunun hareketlerini tam olarak istenen şekilde yönlendirmek için özel ödül kuralları belirlenmiştir.

- **En Üst Düzleme Yükselme Ödülü:** Ajanın, kutunun bulunduğu sütunda en yukarıya çıkışmasını teşvik etmek için özel bir ödül kuralı konmuştur. Eğer ajan, y koordinatında yukarı doğru hareket ederken x ve z koordinatlarını başlangıç durumunun koordinatları ile aynı tutarsa, bu başarılı yükselme hareketi için bir ödül alır. Matematiksel olarak bu ödül Denklem 4.5'teki gibi ifade edilir:

$$r_u = 1.25 \quad (4.5)$$

**Yukarı çıkma ve sıkışma sorunu:** Bu ödül kuralı, ajanın kutuyu yukarı çıkarmasını başarıyla teşvik etmiş olsa da, kutunun genellikle en yüksek iki pozisyon arasında sıkışıp kalmasına ve ilerleyememesine neden olmuştur. Şekil 4.2'de bu durumun görselleştirilmesi yer almaktadır.



**Şekil 4.2 :** Kutunun en yüksek iki pozisyon arasında sıkışma durumu

**Aşağı hareket eğilimi ve gamma katsayısının etkisi:** Kutu en üst sütunda olduğunda, Q Tablosunda aşağı hareket eyleminin değerinin yüksek olduğu gözlemlenmiştir. Bu durum, herhangi bir özel teşvik olmadan nasıl meydana geldiği üzerine detaylı bir araştırma gerektirmiştir. Sonuçta, gamma katsayısının etkisi ile ajanın ilerideki ödülleri hesaba katması ve bu yüzden aşağı yönde hareket etme eylemini tercih etmesi anlaşılmıştır.

- **Aşağı hareketin cezalandırılması:** Yukarıda bahsedilen etkiyi bastırmak ve ayrıca kutunun en üst düzleme çıkabildikten sonra 2 boyutlu hareket yapmasını garanti etmek için kutu aşağıya doğru hareket ettiğinde, ancak hedefin x ve z koordinatlarına ulaşmadığında, yani hedefin bulunmadığı başka bir sütunda indirme işlemi yapıldığında ceza uygulanmıştır. Bu yaklaşım, ajanın gereksiz yere aşağı inmesini engellemekte ve onu en üst düzlemede istenen hareketleri yapmaya yönlendirmektedir. Matematiksel olarak bu ceza Denklem 4.6'daki gibi ifade edilir:

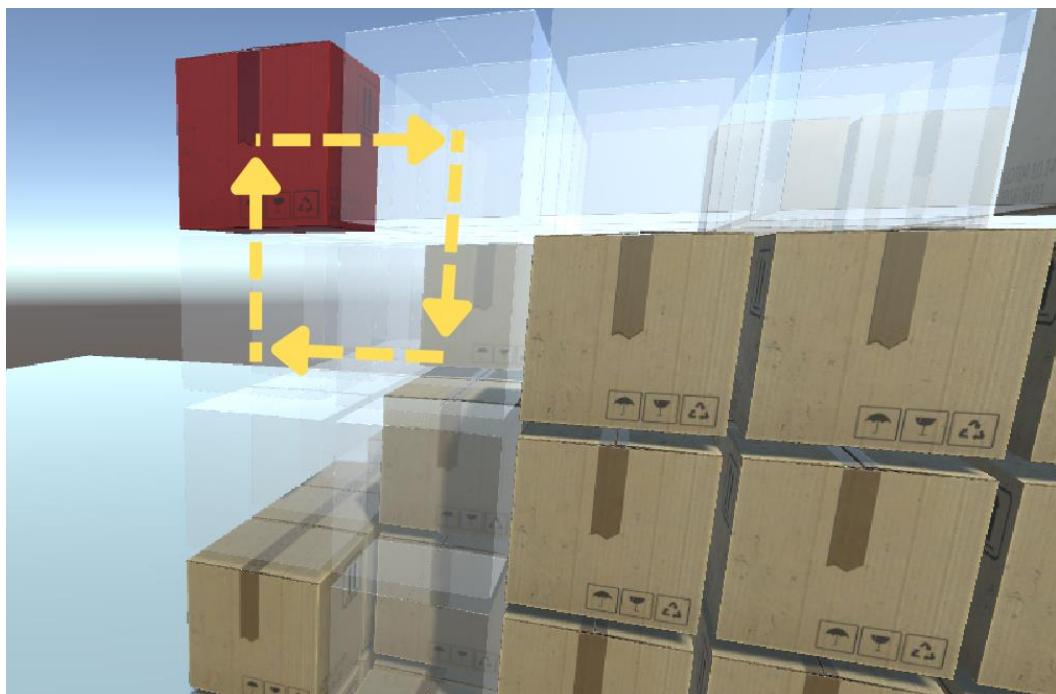
$$r_d = -2.25 \quad (4.6)$$

Bu yeni ödül ve ceza sistemine göre, bir adımda alınan toplam ödül  $r$ , önceki denklemlerle birleştirilerek Denklem 4.7'deki gibi hesaplanır:

$$r = r_s + r_h + r_e + r_u + r_d \quad (4.7)$$

**Bu ödül kurallarının uygulanmasının ardından** kutunun en üst iki blok arasında sıkışması önlense de ajanın farklı bir sorunla karşılaştığı gözlemlenmiştir. Ajan, kutuyu en üst düzeye çıkardıktan sonraörneğin sola hareket edip, daha sonra aşağı indirerek ve sağa hareket ettiğinde ilk pozisyonuna geri dönmesi sonucu, kutu bunun gibi 4'lü veya 6'lı çevrimlerde sıkışmıştır. Bu durum, kutunun hedefe ulaşma stratejisi üzerinde olumsuz bir etki yaratmış ve ilerleme sürecini engellemiştir.

Şekil 4.3'de kutunun bu çevrim içinde nasıl hareket ettiği ve sıkışma durumunun nasıl meydana geldiği görselleştirilmiştir.



**Şekil 4.3 : Kutunun 4'lü veya 6'lı çevrimlerde sıkışma durumu**

#### **4.1.1.3 Üçüncü etap ödül kuralları ve sonuçları**

Üçüncü etapta, kutunun döngüsel ve istenmeyen hareketlerini önlemek amacıyla yeni ödül kuralları eklenmiştir. Bu kurallar, kutunun hedefe daha etkin ve doğrudan bir şekilde ulaşmasını sağlamak için tasarlanmıştır.

- **Döngüsel hareket cezası:** Kutunun daha önce ziyaret ettiği durumları tekrar ziyaret etmesini önlemek için son hareketlerin bir kaydını tutan bir geçmiş

kayıdı (history buffer) oluşturulmuştur. Eğer kutu, kısa bir süre içinde aynı duruma geri dönerse, bu durumun tekrarlanmasını engellemek amacıyla büyük bir ceza uygulanır. Matematiksel olarak bu ceza Denklem 4.8'deki gibi ifade edilir:

$$r_c = -2.0 \quad (4.8)$$

- **Yanlış yönde gitme cezası:** Kutunun hedef sütununa vardığında aşağı doğru hareket etmesi teşvik edilmek istenmektedir. Eğer kutu, hedef sütunundaki x ve z koordinatlarındayken başka bir yöne hareket ederse, bu hareket için ek bir ceza uygulanır. Matematiksel olarak bu ceza Denklem 4.9'teki gibi ifade edilir:

$$r_w = -0.75 \quad (4.9)$$

- **Hedefe ulaşma ödülünün düzenlenmesi:** Başlangıçta hedefe ulaşma durumunda yüksek bir ödül verilmektedir. Ancak bu, kutunun başlangıç pozisyonuna yakın hedeflere doğru daha kısa yolları tercih etmesine ve yukarı çıkma stratejisini göz ardı etmesine neden olmuştur. Bu durumu düzeltmek için, hedefe ulaşma ödülü daha düşük bir değere indirilmiştir. Matematiksel olarak bu ödül Denklem 4.10'deki gibi ifade edilir:

$$r_h = 0.2 \quad (4.10)$$

Bu yeni ödül ve ceza sisteme göre, bir adımda alınan toplam ödül  $r$ , önceki denklemlerle birleştirilerek Denklem 4.11'deki gibi hesaplanır:

$$r = r_s + r_h + r_e + r_u + r_d + r_c + r_w \quad (4.11)$$

**Bu değişiklikler**, ajanın öğrenme sürecini ve karar verme mekanizmasını önemli ölçüde iyileştirmiştir. Seçilen kutu artık daha az döngüsel hareket yapmakta ve hedefe daha doğrudan yolları tercih etmektedir. Ödül kurallarındaki bu iteratif iyileştirmeler, genel performansı ve algoritmanın etkinliğini artırmıştır.

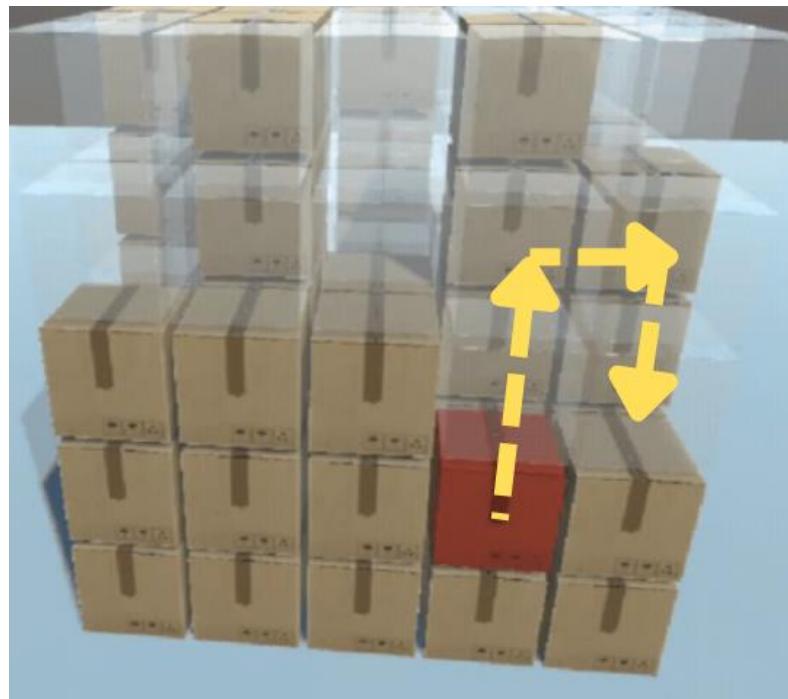
#### 4.1.1.4 Dördüncü etap ödül kuralları ve sonuçları

Tüm hataları görebilecek kadar gözlem yapıldıktan sonra sürecin performansını en iyije çıkarmak için ödüllerde birkaç değişiklik yapılmıştır.

- **Döngüsel hareket cezası değişikliği:** Daha önce ziyaret edilen kutulara tekrar gitmeyi önleyen durum tamponunun boyutu 6'dan 8'e çıkarılmıştır. Bu ayarlama, 8 boyutundaki döngülerde sıkışma gözlemlerine yanıt olarak

yapılmıştır ve gelecekte ızgara boyutlarına göre dinamik olarak tampon boyutunu ayarlayacak bir algoritma geliştirme olasılığını öngörmektedir.

- **Yükselme ödülü değişikliği:** Nadir durumlarda, kutunun en yüksek düzleme çıkmadan daha alt bir düzlemi kullanarak hedefe ulaştığı görülmüştür. Şekil 4.4'te bu soruna ilişkin görselleştirme yapılmıştır.

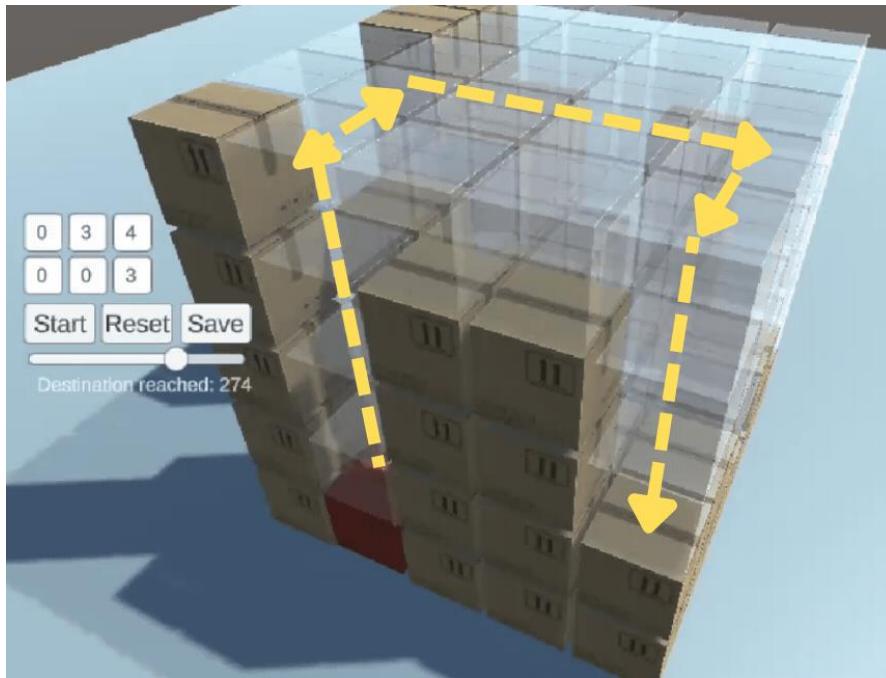


**Şekil 4.4 :** Kutunun en üst düzeye ulaşmadan yoluna devam etmesi sorunu

Bu sorunun önüne geçmek için yükselseme ödülü 1.25'ten 1.75'e artırılmıştır. Ayrıca, bu ayarlama, kutunun yanındaki hedefe kısa yoldan ulaşmasını da engelleyecektir. Matematiksel olarak bu ödül Denklem 4.12'deki gibi ifade edilir:

$$r_u = 1.75 \quad (4.12)$$

- **Her adım için ceza değişikliği:** Her adım için uygulanan cezanın yeni tanıtılan ödüllere kıyasla düşük kalması, bazı durumlarda kutunun uzun yolları tercih etmesine neden olmuştur. Şekil 4.5'te bu durum görselleştirilmiştir.



**Şekil 4.5 :** Kutunun daha uzun olan yolu tercih etmesi sorunu görselleştirmesi

Diger ödüllerle dengeyi sağlamak için, her adım için ceza 0.05'ten 0.22'ye artırılmıştır. Matematiksel olarak bu ceza Denklem 4.13'deki gibi ifade edilir. Bu ayarlama, cezanın genel ödül sistemiyle daha etkili bir şekilde hizalanmasını sağlar ve ajanın daha iyi kararlar almasını teşvik eder.

$$r_s = -0.22 \quad (4.13)$$

Değişikliklerin sonucu:

Bu ayarlamalar ve dengeleme çabalarının ardından, Aşama 1'de büyük hatalar gözlemlenmemiştir. Ödüller ve cezaların ince ayarı, önceki sorunları etkili bir şekilde hafifleterek, ortamda gelişmiş performans ve istikrar sağlamıştır.

#### 4.2.1.1 Tüm ödüllerin özeti

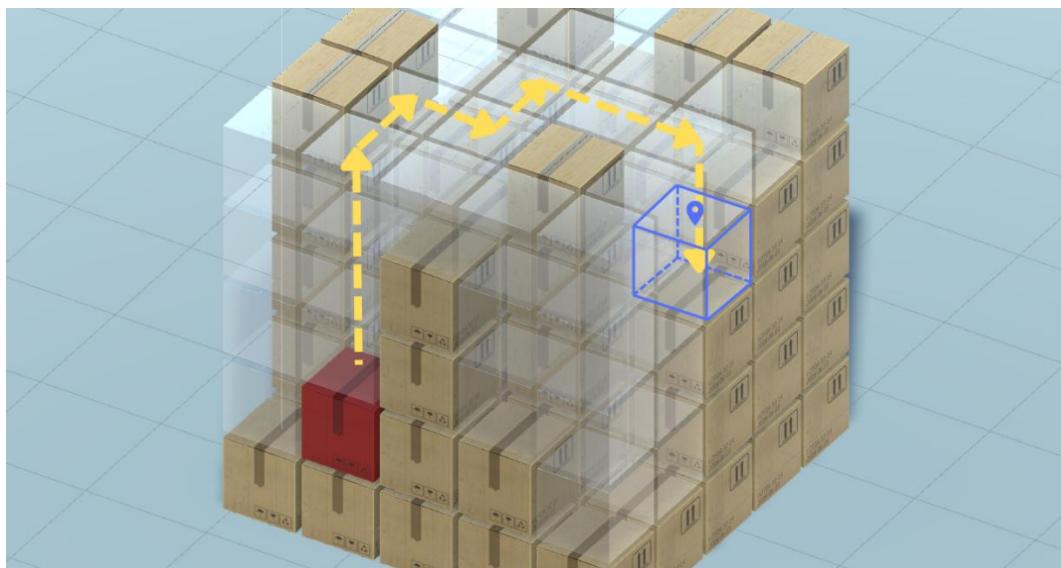
- **Döngüsel hareket cezası ( $r_c$ ):** Kutu, son 6 hareketi içinde ziyaret ettiği bir duruma geri dönerse, bu durumun tekrarlanmasını engellemek için -2.0 puan ödül alır.
- **Yanlış yönde hareket cezası ( $r_w$ ):** Kutu, hedef sütununda olmasına rağmen hedefin dışındaki bir yönü tercih ederse -0.75 puan ödül alır.
- **Hedefe ulaşma ödülü ( $r_h$ ):** Kutu hedefine ulaşırsa, bu başarılı hareket için 0.2 puan ödül alır.

- **Hareket edememe durumunda cezalandırma ( $r_e$ ):** Kutu hareket edemediğinde (engeller veya kısıtlamalar nedeniyle), -0.5 puan ödül alır.
- **Yukarı çıkma ödülü ( $r_u$ ):** Kutu, başlangıç noktasının x ve z koordinatlarıyla hizalanırken yukarı çıkarsa, bu başarılı yükseliş için 1.75 puan ödül alır.
- **Aşağı inme cezası ( $r_d$ ):** Eğer Kutu hedefin x ve z koordinatlarından farklı bir yere inerse -2.25 puan ödül alır.
- **Her adım için küçük bir ceza ( $r_s$ ):** Her hareket için -0.22 puan ödül uygulanır.

Bir adımda alınan toplam ödül  $r$ , önceki denklemelerle birleştirilerek Denklem 4.11'deki gibi hesaplanır.

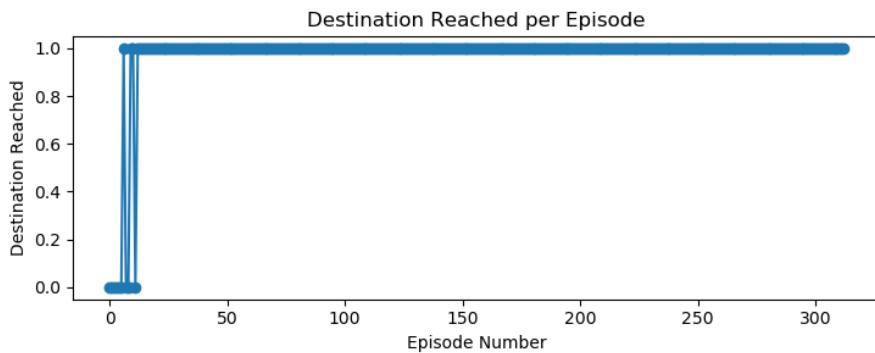
#### 4.1.2 Sonuçlar

Projenin ilk aşaması planlandığı gibi tamamlanmıştır. Optimizasyon sürecinden geçirilmiş algoritma, belirlenen rotada başarılı bir şekilde test edilmiştir. Elde edilen sonuçlar, çeşitli grafikler aracılığıyla görselleştirilerek değerlendirilmiştir. Şekil 4.6'da bir test aşamasında ajanın hareket ettirdiği kutuya ait rota ve hareket stratejisi görselleştirilmiştir.



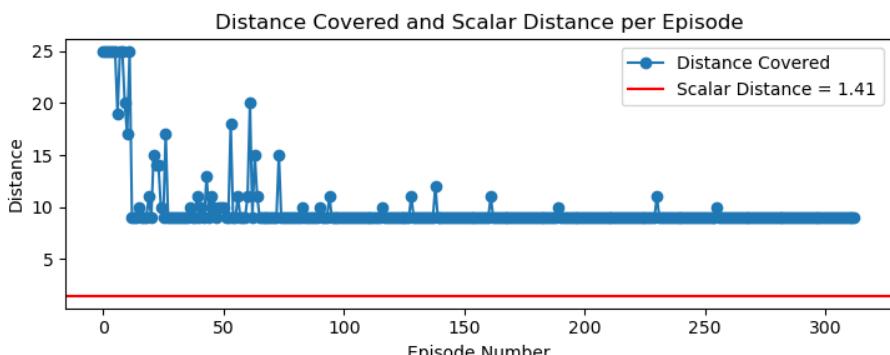
**Şekil 4.6 :** Test rotası görselleştirmesi

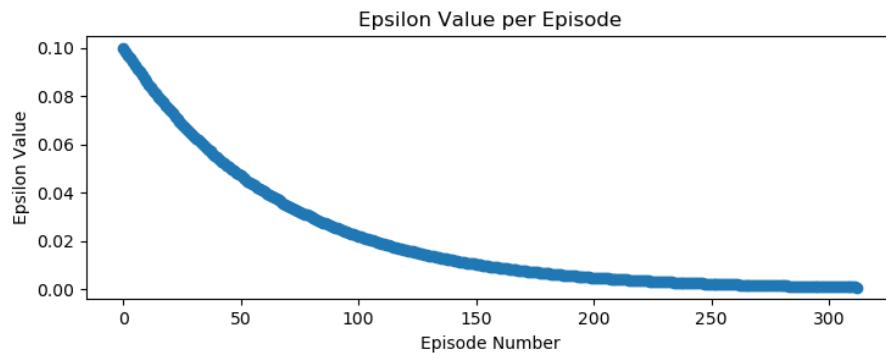
Her bölüm için hedefe ulaşma başarısı, bölüm numarasına karşı çizilen Şekil 4.7'deki grafikle gösterilmektedir. Bu grafik, ajanın hedefe ulaşma yeteneğindeki gelişmeyi ve zaman içindeki başarı oranını yansıtmaktadır. İlk 15 bölümden sonraki her bölümde kutunun hedefe ulaştırıldığı görülmüştür.



Şekil 4.7 : Bölüm başına varış başarısı grafiği

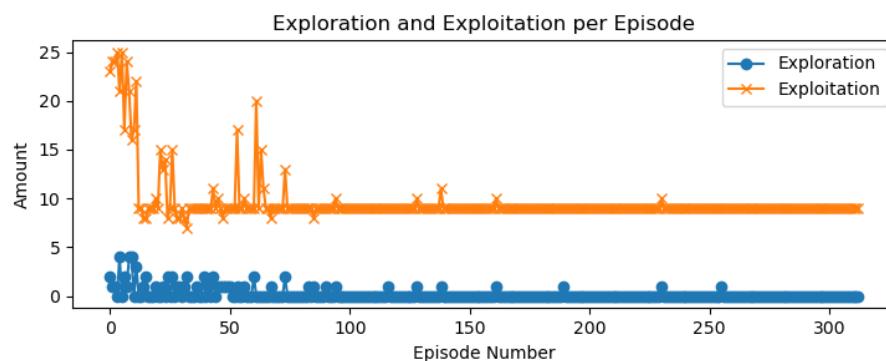
Şekil 4.8'de her bölüm için katedilen mesafe, başlangıç ve hedef pozisyonları arasındaki skaler mesafe ile karşılaştırılarak gösterilmektedir. Bu, ajanın rotasının verimliliğini ve hedefe ulaşmak için ne kadar yol ettiğini gösterir. Belirli bir bölümden sonra kutunun optimal rotaya oturduğu görülmüştür.





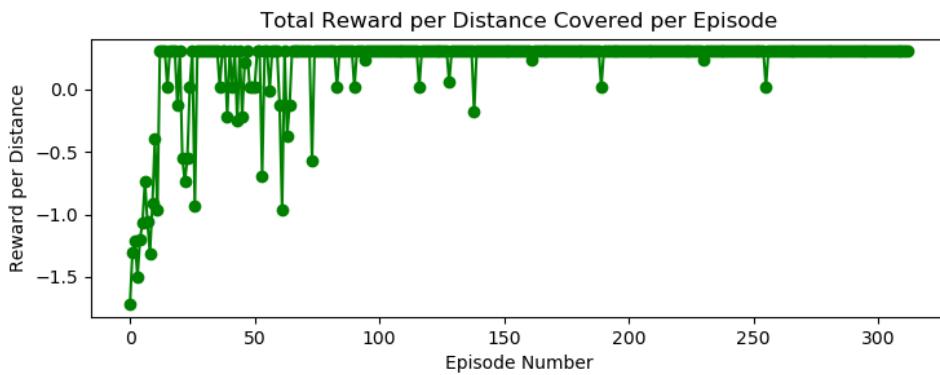
**Şekil 4.9 :** Bölüm başına epsilon değeri grafiği

Şekil 4.10'da her bölümde gerçekleşen keşfetme ve deneyimden faydalananma miktarları karşılaştırımlı olarak gösterilmektedir. Bu grafik, ajanın epsilon değerine bağlı olarak davranışındaki değişiklikleri ve öğrenme sürecindeki evrimi yansıtmaktadır. Başlangıçta ve bazı noktalarda bu parametrelerde dalgalanmalar olmuş, daha sonra belirli değerlere oturmuşlardır.



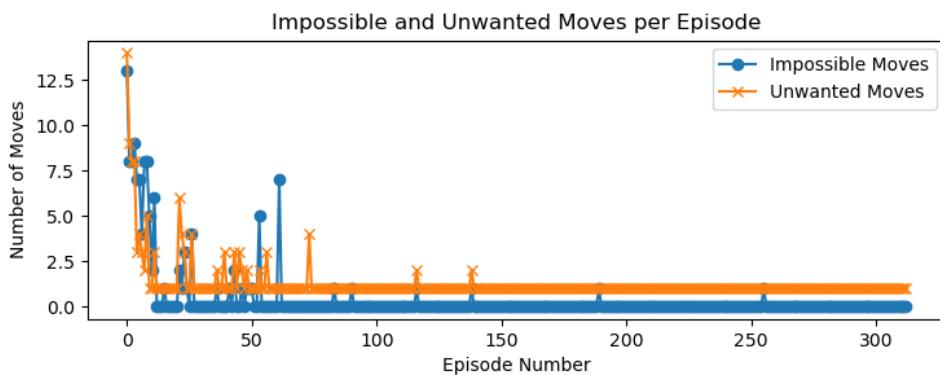
**Şekil 4.10 :** Bölüm başına keşfetme ve tecrübeden faydalananma miktarı grafiği

Şekil 4.11'de her bölümde elde edilen toplam ödül, katedilen mesafe başına düşen oranla gösterilmektedir. Bu, ajanın verimliliğinin ve performansının bölümler arasındaki değişimini gösterir. Bölüm sayısı arttıkça bu parametrenin maksimize edilmesi ve o değerde kalmasının sağlanması algoritmanın başarısını göstermektedir.



**Şekil 4.11 :** Bölüm başına toplam ödül / katedilen mesafe grafiği

Şekil 4.12'de her bölümde yapılan imkânsız ve istenmeyen hareketlerin sayısı gösterilmektedir. Bu grafik, kutunun götürülme sürecinde yapılan hataları görselleştirmektedir. İmkânsız hareketler kutunun yer olmadığı için yapamayacağı manevraları, istenmeyen hareketler ise istenen davranıştan sapan hareketleri temsil etmektedir.



**Şekil 4.12 :** Bölüm başına imkânsız ve istenmeyen hareketler grafiği

## 4.2 İkinci Aşama: Sıralı Komut Öğrenimi

Bu aşamada, algoritmanın 1. aşamada uygulanan tekil komutları sıralı komutlar halinde yürütmemeyi öğrenmesi, bir dizi kutuyu sıralı olarak istenilen konumlarına yönlendirmesi hedeflenmektedir. Her bir komutun öğrenme sürecinin, dizideki diğer komutların uygulanmasını etkileyebileceğini unutulmamalıdır.

2. aşamanın temelinde kutu temizleme mekaniklerinin uygulanması bulunmaktadır ve bu, 1. aşamadan farklı bir öğrenme modeli oluşturma sürecini içerir. Bu süreç, çekirdeğinde kutuları iki ana kriter doğrultusunda yeniden konumlandırmaya odaklanır:

- **Kutunun üzerinde bulunan kutular:** Algoritma, o anki seçilen kutunun hareketi için açık bir yol oluşturmak amacıyla hedef kutunun üzerinde bulunan kutuları temizlemeyi amaçlamaktadır.
- **Kutunun yolunu tıkayan kutular:** Ayrıca algoritma, seçilen kutunun yolunu tıkayan kutulara da odaklanarak kutunun hedefe doğru engelsiz ilerlemesini sağlamaktadır ve 1. aşamada en iyi sonuçların elde edilmesine yardımcı olmaktadır.

## 4.2.1 Tasarım

### 4.2.1.1 Kutu temizleme mekaniği

Kutu temizleme mekanikleri ilk olarak belirlenmiş ve hedef kutunun engelsiz hareketini sağlamak için kutuları yeniden konumlandırma süreci tanımlanmıştır. Bu süreci daha etkili kılmak ve kutuların en verimli şekilde yer değiştirmesini sağlamak amacıyla iki ana adım gerçekleştirilmiştir.

İlk adımda, belirlenen mekanikler çerçevesinde hedef kutunun üzerindeki kutuların pozisyonlarını rastgele değiştirmek için bir yöntem uygulanmıştır. Bu adım, kutu temizleme süreci için başlangıç koşullarını oluşturmayı amaçlamaktadır. Rastgele yeniden konumlandırma yöntemi, sistemin başlangıçta çeşitlilik içeren durumlar oluşturmasına olanak tanır. Bu çeşitlilik, algoritmanın farklı senaryolarla karşılaşmasını ve daha geniş bir yelpazede öğrenmesini sağlar. Rastgelelik, algoritmanın sadece belirli düzenlerde değil, genel olarak farklı durumlarda da etkili olabilmesini hedefler.

Rastgele yeniden konumlandırma adının ardından, kutuların en optimal ve mantıklı şekilde yeniden konumlandırılmasını öğretmek için özel olarak tasarlanmış yeni bir Q-learning mantığı tasarlanmıştır ve entegre edilmiştir. Bu aşamada, algoritmanın kutuları

belirli kurallar ve ödül yapıları çerçevesinde yerlestirmesi sağlanmıştır. Örneğin, kutuların hedef kutunun üzerindeki pozisyonlardan taşınırken en az engelleyici olacakları yeri konumaları hedeflenmiştir. Bu yaklaşım, kutu temizleme sürecinin etkinliğini artırmak için kritik öneme sahiptir.

Yeni Q-learning mantığı, ödül yapılarının tasarılanmasını ve Q-Değerlerinin güncellenmesini içermektedir. Ödül yapıları, algoritmanın istenen davranışları pekiştirmesi ve istenmeyen davranışlardan kaçınması için kritik bir rol oynar. Örneğin, hedef kutunun üzerindeki bir kutunun uygun bir şekilde taşınması ödüllendirilirken, uygun olmayan bir şekilde taşınması cezalandırılır. Bu ödül ve ceza sistemi, algoritmanın en verimli stratejileri geliştirmesine olanak tanır.

Q-Değerlerinin güncellenmesi süreci, algoritmanın her bir adımda aldığı geri bildirimlere göre stratejilerini ayarlamasını sağlar. Bu sürekli geri bildirim ve ayarlama döngüsü, algoritmanın zamanla daha akıllı ve etkili hale gelmesini mümkün kılar. Ayrıca, Q-learning algoritması sayesinde, sistem dinamik değişikliklere hızlı bir şekilde uyum sağlayabilir ve en optimal sonuçları elde edebilir.

Sonuç olarak, kutu temizleme mekanikleri, hedef kutunun engelsiz hareketini sağlamak için dikkatle planlanmış ve uygulanmıştır. Rastgele yeniden konumlandırma ve Q-learning mantığının entegrasyonu ile sistemin performansı ve etkinliği önemli ölçüde artırılmıştır. Bu süreçler, hem teorik hem de pratik açıdan algoritmanın başarısını ve adaptasyon yeteneğini geliştirmiştir.

#### **4.2.1.2 Kutu temizleme başlangıç koşulları**

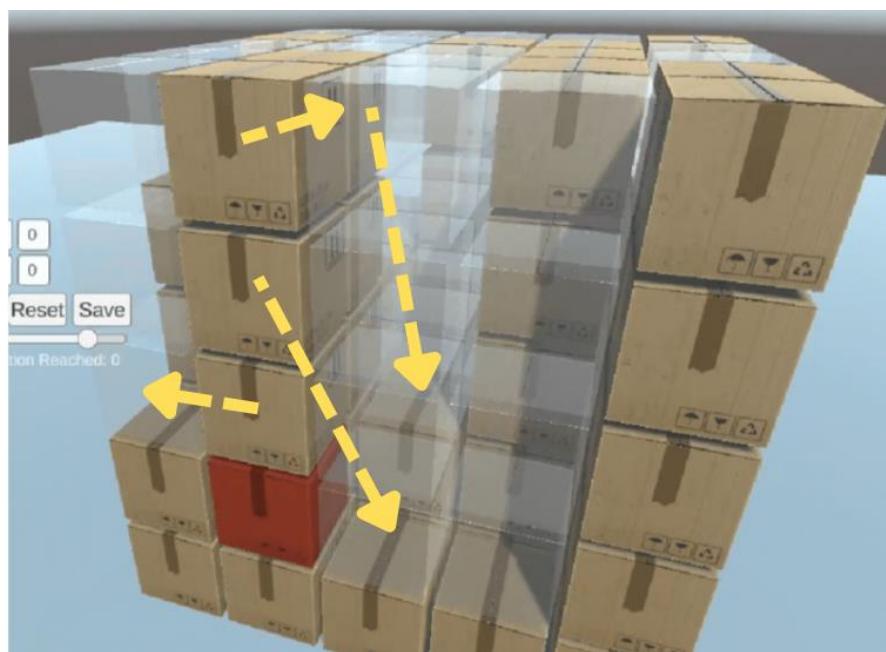
Kutu temizleme mekanikleri, başlangıç koşulu olarak, seçilen kutunun üzerindeki bir kutuyu, 1. aşamadaki kurallar uygulanarak, eğer müsaitse seçilen kutunun hemen yanındaki konuma indirerek yeniden konumlandırmayı içermektedir. Bu yaklaşım, kutuların hareket etmesi gereken minimum mesafeyi ve dolayısıyla gereksiz hareketleri azaltmayı amaçlamaktadır.

Eğer hedeflenen yeniden konumlandırma noktası doluya, o alanı işgal eden kutu da yeniden konumlandırılmalıdır. Bu durum, zincirleme bir yeniden konumlandırma sürecini tetikleyebilir ve bu süreçte her kutu için en uygun yerin bulunması gerekmektedir. Ayrıca, hedeflenen yeniden konumlandırma noktasının üzerindeki

kutular da uygun şekilde taşınmalıdır. Bu, kutuların hareketlerinin engellenmemesi için gereklidir ve sistemin bütünlüğünü korur.

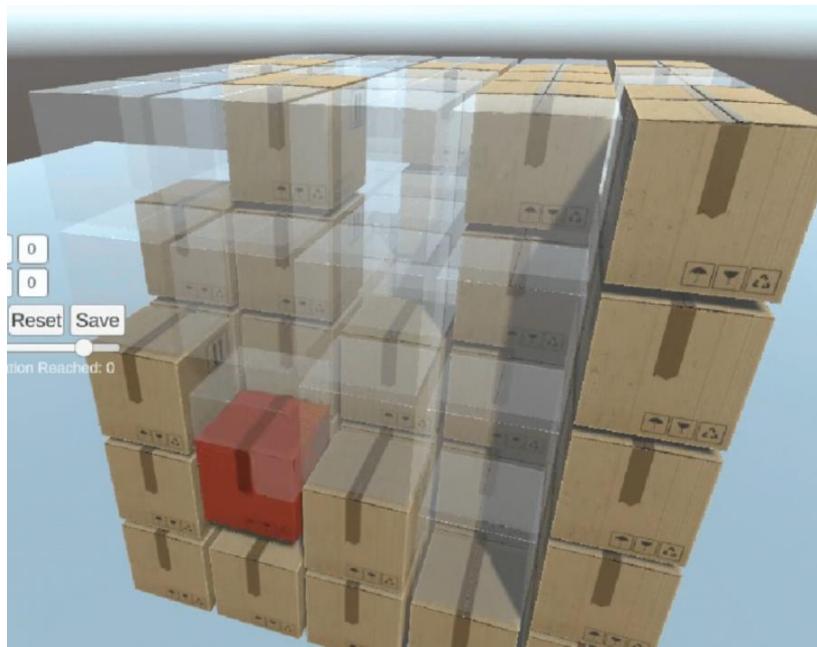
Bu altyapı, mantıklı ve gerçekçi olup, gerçek hayatı senaryoları yansımaktadır. Bu şekilde, algoritma sadece teorik olarak değil, pratikte de uygulanabilir ve işlevsel bir çözüm sunmaktadır. Bu yeniden konumlandırma süreci, kutuların düzenli ve etkili bir şekilde yer değiştirmesini sağlayarak hedef kutunun engelsiz bir şekilde hedefine ulaşmasını mümkün kılar.

Takip eden şekillerde, seçilen kutunun üzerindeki kutuların temizlenme sürecini gösteren bir demoyu sunmaktadır, burada kırmızı kutu seçilen kutudur. Oklar temizlenecek kutuların algoritma tarafından çalışma zamanında belirlenen konumlarıdır.



**Şekil 4.13 :** Üstteki kutuların temizlenme sürecini gösteren bir demo

Uygulamada her bir kutunun yeniden konumlandırılması için 1. aşamanın mantığı ve kuralları uygulanacaktır. Bu sayede, sistemdeki tutarlılık korunacak ve her aşamanın gereksinimleri doğru bir şekilde karşılanacaktır.



**Şekil 4.14 :** Kutu temizliğinden sonra sistem

#### 4.2.1.3 Öğrenme ağacı yapısı

Kutu temizleme mekaniklerine yaklaşımda rastgele hareketlere dayanmak yerine ajanın bu mekanikleri öğrenmesi hedeflenmektedir. Bu hedefe ulaşmak için QTable yerine bir QTree tanıtılmıştır. Bu QTree, önceki bölümde bahsedildiği gibi, kökten uç noktaya kadar olan hareket zincirlerini temsil etmektedir.

QTree'nin kökünde, öncelikle yeniden konumlandırılması gereken kutu yer alır; bu, hedef kutunun üzerindeki kutu olabilir. Çalışma esnasında yeniden konumlandırılan kutular ve sonunda en uç nokta, veri yapılarında bir ağaç yapısına benzer şekilde, zincirleme olarak çocuk düğümler olarak eklenir. Bu yapı, ajanın adım adım kararlar almasını ve her bir kararın sonuçlarını takip edebilmesini, ağacın dalları üzerinde hakimiyet geliştirmesini sağlar. Ağacın her düğümü aşağıdaki bilgileri içerir:

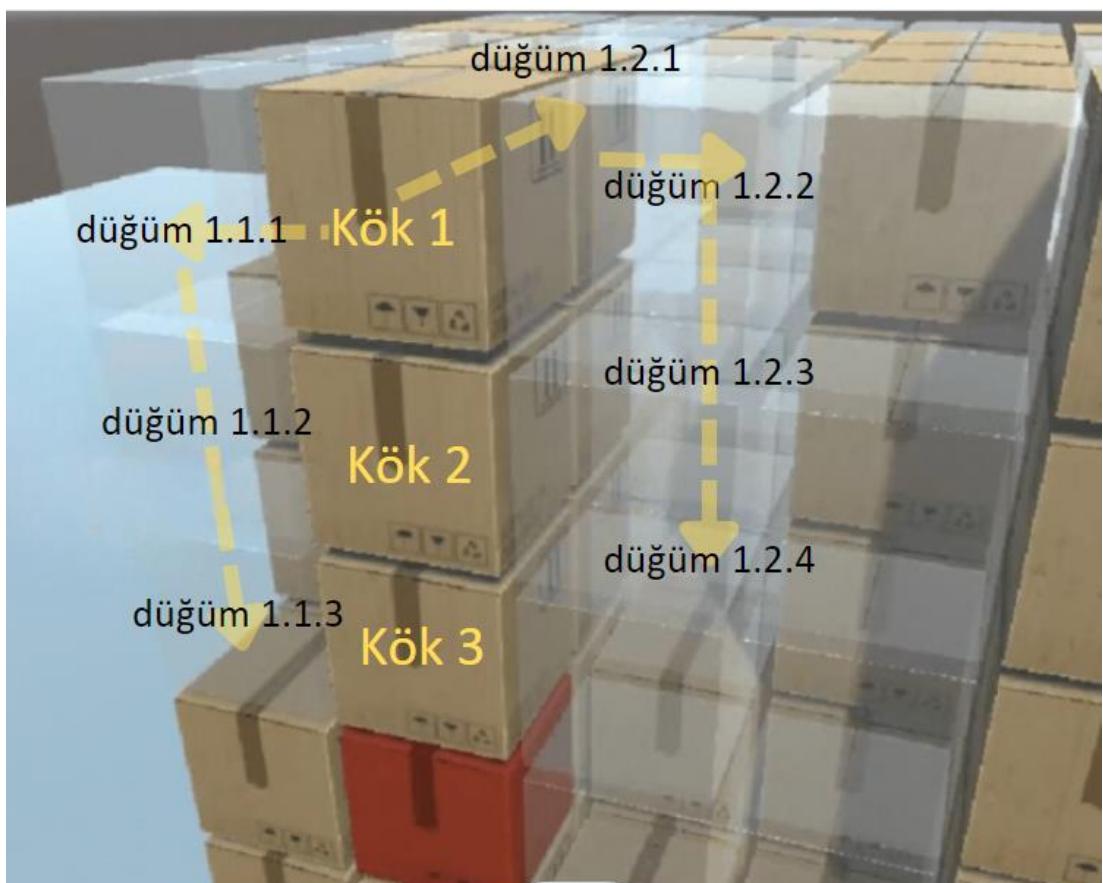
- Pozisyon bilgisi: Kutunun üç boyutlu uzayda ayrık pozisyonu
- Q-Değeri: O pozisyonla ilişkili Q-değeri.
- Çocuklar: O pozisyondan olası aksiyonları temsil eden çocuk düğümler.
- Ebeveyn: Ebeveyn düğüm.

Bu hiyerarşik yapı, olası aksiyonların ve ilgili Q-değerlerinin daha düzenli bir şekilde temsil ve takip edilmesini sağlar, bu da ajanın daha verimli öğrenme ve karar verme

süreçlerini kolaylaştırır. QTree, karmaşık problem çözme süreçlerinde bile ajanın mantıklı ve etkili kararlar almasını destekler.

Her QTree, öğrenme ve karar verme süreçleri için düzenli ve erişilebilir olmalarını sağlayarak bir QTrees listesine eklenir. Bu liste, değiştirilecek herhangi bir kutunun yeri için, ajanın tüm olası senaryoları ve bunlara karşılık gelen aksiyonları değerlendirmesine olanak tanır. Ayrıca, QTree yapısı sayesinde, ajanın öğrenme süreci daha esnek ve dinamik hale gelir, böylece yeni durumlara hızlıca adapte olabilir. Bu yapısal yaklaşım, algoritmanın genel performansını ve etkinliğini artırır, böylece sistem daha verimli bir şekilde çalışabilir.

Ağaç yapısının görselleştirilmesiyle ilgili bir örnek şekil 5.1'de verilmiştir. Düğümler çalışma esnasında oluşmaktadır.



**Şekil 5.1 :** Ağaç yapısının görselleştirmesi

Sistemimizde, bir düğümün Q-değerini ve tüm ebeveyn zincirini ağaç benzeri bir şekilde ödüllendirme yöntemi bulunmaktadır. Bu, bir düğüm ödüllendirildiğinde veya cezalandırıldığında, etkilerin ağacın dalında yayılmasını sağlar. Bu yöntem, her bir hareketin genel stratejiye etkisini vurgulayarak ajanın daha bilinçli kararlar almasını sağlar.

Özellikle, bir düğüm ödül veya ceza aldığında, daldaki tüm düğümler benzer şekilde teşvik edilir veya caydırılır. Bu, ajanın sadece bireysel hareketlere odaklanmasını engelleyerek, her aksiyonun geniş kapsamlı sonuçlarını değerlendirmesine olanak tanır. Örneğin, bir kutunun başarılı bir şekilde yeniden konumlandırılması durumunda, sadece o hareket değil, bu harekete yol açan tüm hareketler de ödüllendirilir. Aynı şekilde, olumsuz bir sonuç elde edilirse, bu sonucu doğuran tüm hareketler cezalandırılır.

Bu yaklaşım, olumlu sonuçlara yol açan aksiyonların tüm aksiyon zinciri boyunca güçlendirilmesini sağlarken, olumsuz sonuçlara yol açan aksiyonların caydırılmasını sağlar. Bu sayede, ajanın her aksiyonun sonuçlarını daha geniş bağlamda değerlendirerek daha bilinçli kararlar alması sağlanır. Ajan, geçmiş deneyimlerinden öğrenerek, daha etkili ve verimli stratejiler geliştirir. Bu yöntem, sistemin genel performansını artırırken, öğrenme sürecini de hızlandırır ve optimize eder.

#### 4.2.1.4 Ödüllendirme

- Dallar içindeki her adım için ceza:** Çalışma zamanında oluşturulan her dalın uç düğümü ve tüm dalı (köke doğru azalan belli oranlarda), Projenin 1. adımındaki "her adım için ceza" ile benzer şekilde cezalandırılır. Uç düğüm için ceza  $r_{t,0} = -0.12$  olarak belirlenmiştir ve köke doğru ilerlerken ceza değeri her adımda %10 oranında azalmaktadır. Matematiksel olarak bu ceza Denklem 4.14'teki gibi ifade edilir. Bu strateji, gereksiz aksiyonları caydırarak en hızlı ve en kolay dalları keşfetmeyi teşvik eder. Böylece, ajan gereksiz hareketlerden kaçınarak daha verimli yolları bulmaya yönlendirilir. Bu yöntem, sistemin genel verimliliğini artırarak ajanın optimal çözümler geliştirmesine yardımcı olur.

$$r_{t,n} = r_{t,0} \cdot (0.9^n) \quad (4.14)$$

- **Engelleyen düğümleri cezalandırma:** Tekil komutlar dizisinde olan veya taşınması gereken diğer kutuların yolunu tıkayan herhangi bir kutunun bağlı olduğu dal, bahsi geçen kutu taşınırken, onu engellenmesi durumunda cezalandırılır. Bu durumda, sorumlu düğüm ve tüm dalı benzer şekilde cezalandırılır ve sonraki bölümde kutu taşıma esnasında bir diğer kutunun önünü kapatmayı öngören bu davranış engellenir. Alınan ceza  $r_e$  olarak ifade edilir. Esas kutuları engelleyen bir düğüm için ceza  $r_{e1}$  olarak belirlenmiştir. Ayrıca, daha az önemli olan ve esas kutuların üzerinde açılması için taşınan kutuları engelleyen hareketler için ceza  $r_{e2}$  olarak belirlenmiştir. Matematiksel olarak bu cezalar Denklem 4.15 ve 4.16'da ifade edilir. Bu cezalandırma, engellenmiş yollara yol açan aksiyonları caydırarak ajanın gelecekte benzer durumlardan kaçınmasını teşvik eder. Bu yöntem, ajanın daha stratejik kararlar almasını sağlayarak, hedefe ulaşmasını engelleyen davranışları minimize eder.

$$r_{e1} = -1.25 \quad (4.15)$$

$$r_{e2} = -0.75 \quad (4.16)$$

- **Zaman aşımı cezası:** Kutu temizleme sırasında değiştirilmesi gereken kutu sayısı için bir zaman aşımı cezası uygulanmıştır. Bu ceza, daldı hareket ettirilmesi gereken kutu sayısı 6'ya çıktığında uygulanır ve Q learning'in başlangıcında rastgele seçimin daha yaygın olduğu durumlarda potansiyel zaman kaybını veya sıkışmaları önler. Matematiksel olarak bu ceza Denklem 4.17'de ifade edilir. Şuanda kritik kutu sayısı bahsedildiği gibi statik 6 değeri olarak belirlense de kök kutunun etrafındaki doluluk oranına göre bunu dinamik olarak ayarlayan bir yapı üzerinde çalışılmaktadır. Bu mekanizma, algoritmanın daha hızlı ve verimli bir şekilde öğrenmesini sağlayarak, gereksiz zaman kayıplarının önüne geçer.

$$r_z = -0.3 \quad (4.17)$$

Bu yeni ödül ve ceza sistemine göre, bir adımda alınan toplam ödül  $r$  Denklem 4.18'deki gibi hesaplanır:

$$r = r_{t,n} + r_e + r_z \quad (4.18)$$

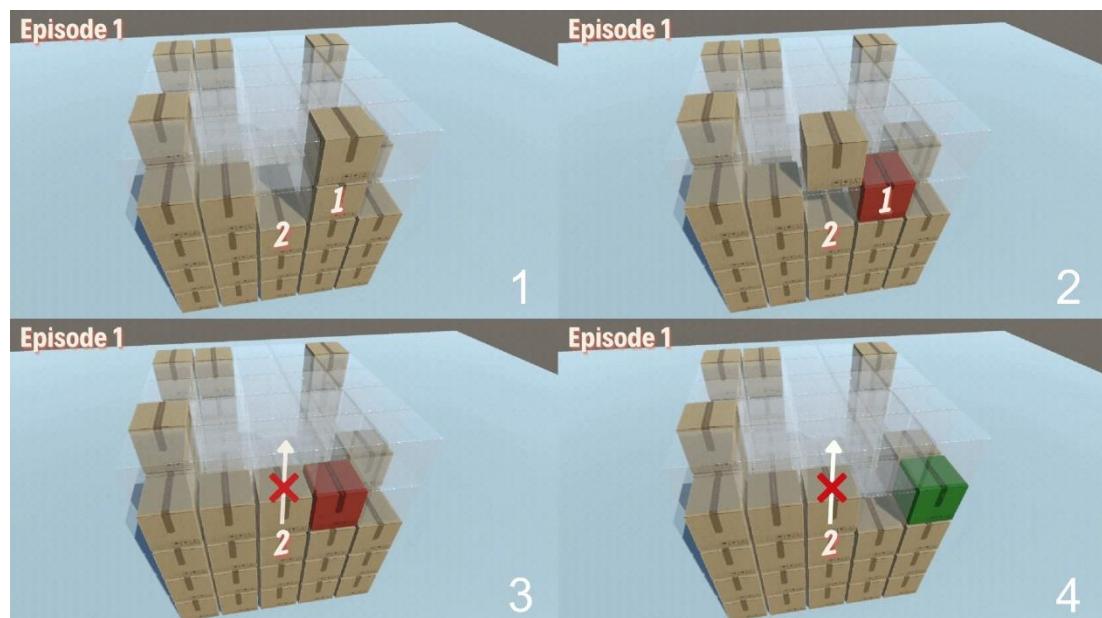
#### 4.2.1.5 Kutu temizleme uygulamaları ve sonuçları

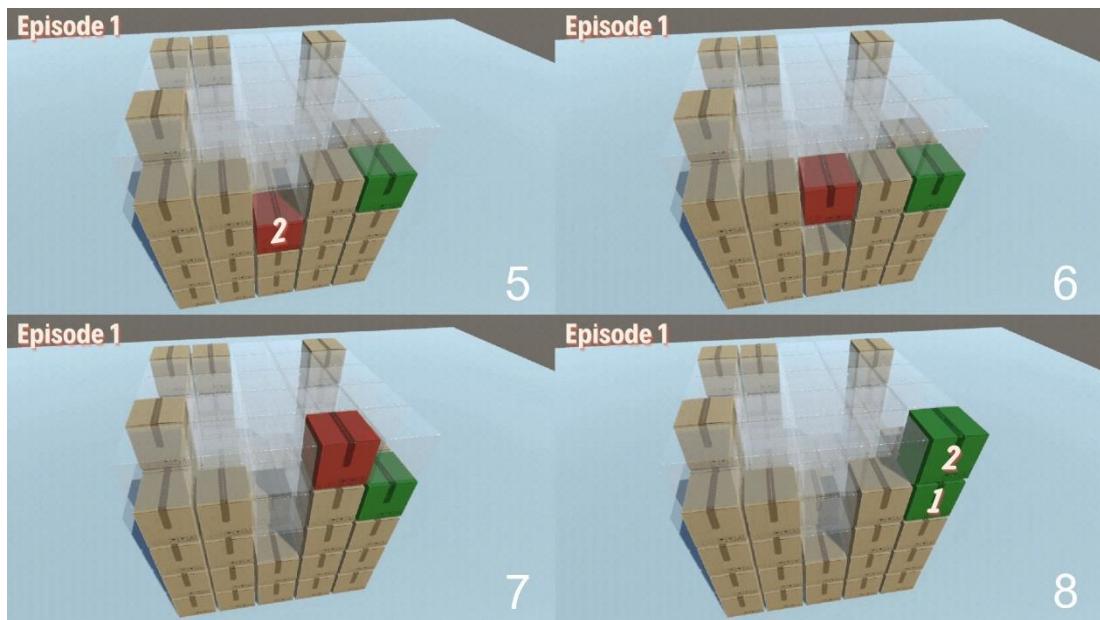
Bu bölümde, algoritmanın farklı senaryolarda nasıl performans gösterdiğini inceleyen örnekler sunulmaktadır. Q-learning algoritmasının, projenin ikinci aşamasının temelindeki kutu temizleme ve yeniden konumlandırma süreçlerinde ajanların nasıl daha etkili stratejiler geliştirdiğini gösteren örnekler, algoritmanın öğrenme yeteneğini ve verimliliğini vurgulamaktadır. Verilen örneklerde, uygulamadaki her bir aşamaya ait resimlerin olduğu kolajlar verilmiştir. Sağ alta sırası yazmaktadır. Resimlerin her birinde ise bilgilendirici işaretler ve yazılar bulunmaktadır. Bu kolajlar, her adımın görsel olarak anlaşılmasını kolaylaştırarak, sürecin nasıl işlediğine dair net bir fikir sunmaktadır.

##### 4.2.1.5.1 Örnek 1

Aşağıdaki örnekte, ajana iki kutuyu sıralı olarak belirli pozisyonlara taşıması emredilmiştir.

Birinci bölümde, ilk kutunun üzerindeki kutu, ikinci kutunun geçiş yolundaki bir konuma taşınarak temizlenir. Bu, ikinci kutunun taşınma sırasında yolunun tıkanması anlamına gelir ve daha sonradan aynı kutunun yeniden konumlandırılması gereklidir. Bu durumda, her iki kutu da nihayetinde gereken yerlere taşınacak olsa da, toplam hareket süresi ve çaba artar, hareket karmaşıklaşır.



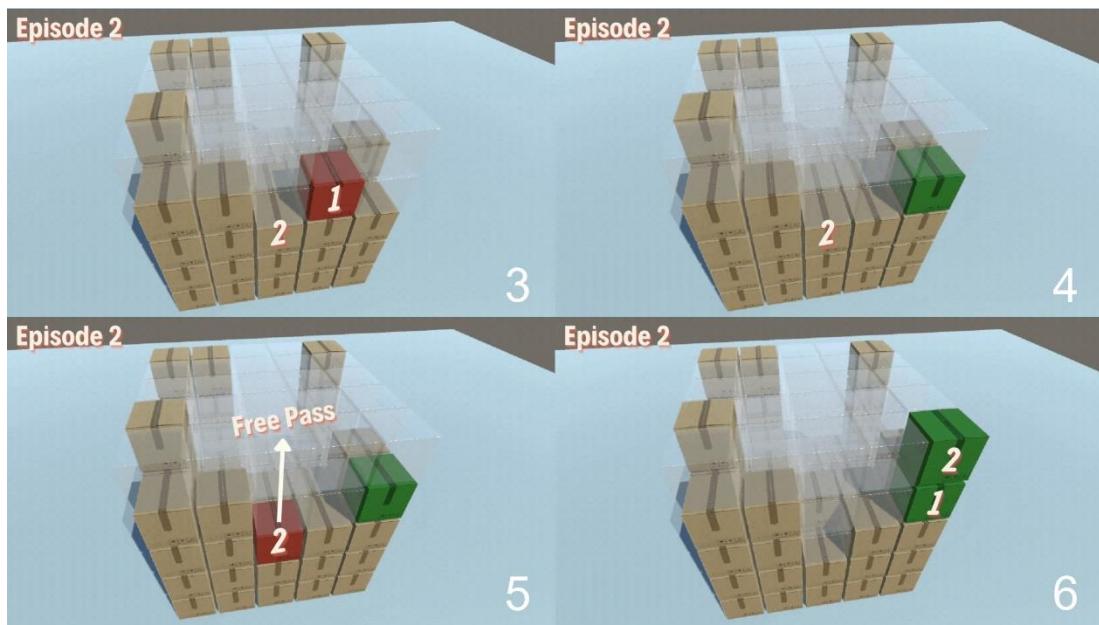


**Şekil 5.2 :** İlk bölümdeki yeniden konumlandırma

Birinci bölümdeki bu süreç, ajan için önemli bir öğrenme deneyimi sunar. Ajan, hareketlerinin sonuçlarını gözlemleyerek, bir sonraki adımda daha stratejik kararlar almayı öğrenir.

İkinci bölümde, ajan birinci bölümdeki bu hatayı fark etmiş, öğrenmiş ve ikinci kutunun yolunu tıkamaktan kaçınmıştır. Bu öğrenme, ajan için kritik bir gelişme olup ikinci kutunun hedefe ulaşması için daha temiz ve hızlı bir yol sağlar. Bu stratejinin uygulanması, toplam hareket süresini ve çabayı önemli ölçüde azaltır.



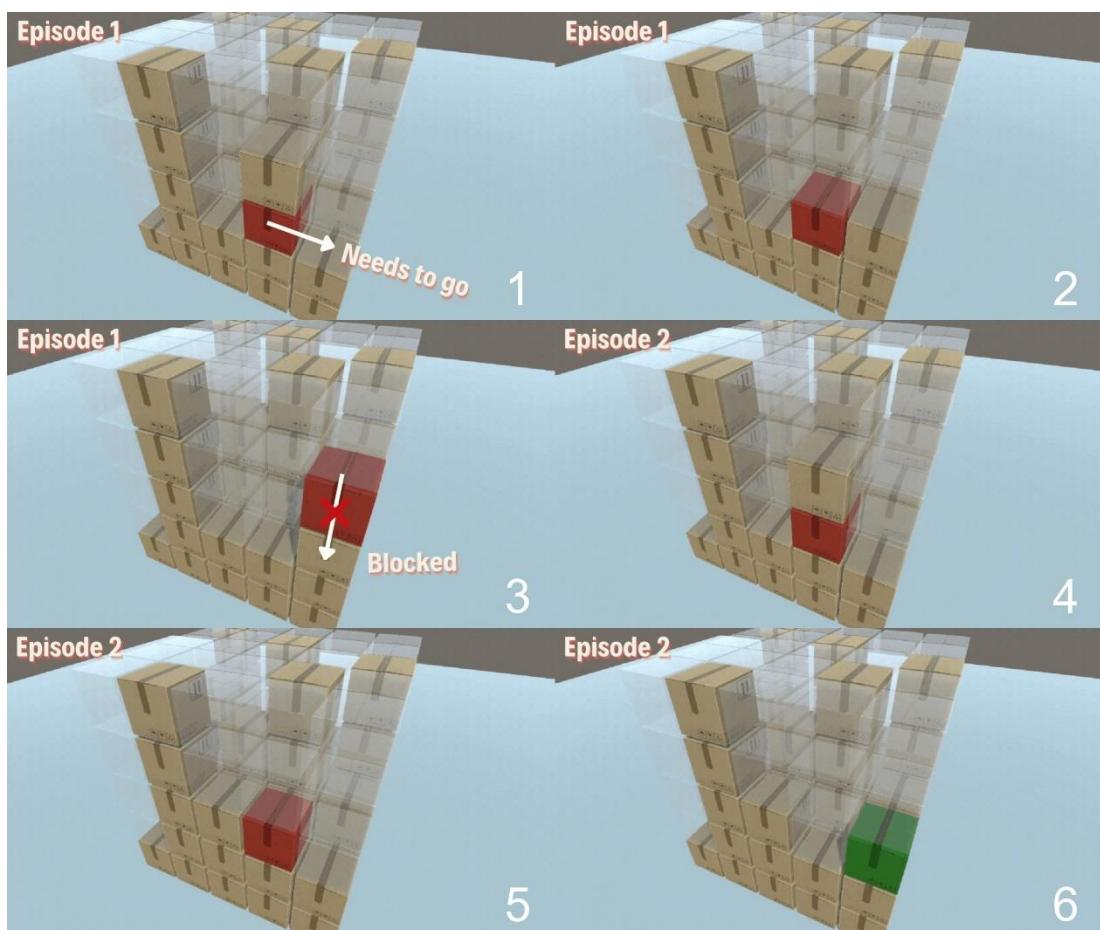


**Şekil 5.3 :** İkinci bölüm: öğrendikten sonraki gelişmiş strateji

Örnekler bu aşamada tasarlanan Q-learning algoritmalarının etkinliğini ve öğrenme sürecinin nasıl geliştiğini göstermektedir. Ajan, her bölümde deneyim kazandıkça, daha verimli ve stratejik kararlar alarak hedefe ulaşmada daha başarılı olur. Bu süreç, algoritmanın genel performansını artırır ve zamanla daha optimal çözümler sunar.

#### 4.2.1.5.2 Örnek 2

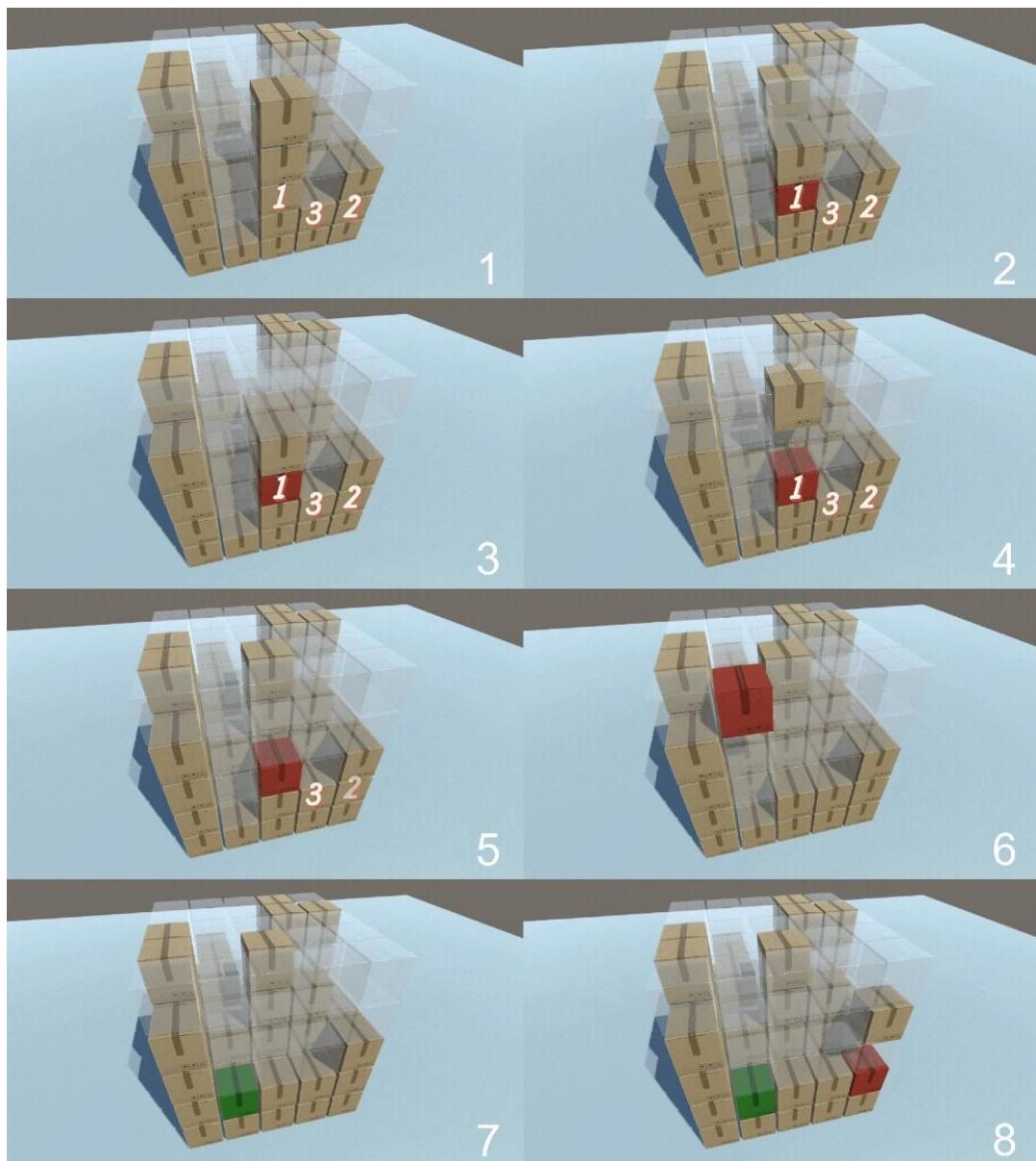
İkinci örnekte, aynı şekilde ajana seçilen kutuyu belirlenen yere götürmesi emredilmiştir. Ne yazık ki, üzerindeki kutu, kutu temizleme esnasında tam olarak hedeflenen kutunun gitmesi gereken yere yerleştirilir. Bu hareket, ciddi şekilde cezalandırılmıştır. Sorun, başlangıçta bölüm 1'de sunulmuş ve ardından bölüm 2'de cezanın etki etmesi sayesinde kutu temizlemede başka bir dalın seçilmesiyle soru çözülmüştür. Bu süreç, ajanın öğrenme ve ayarlama yeteneklerini göstermektedir.

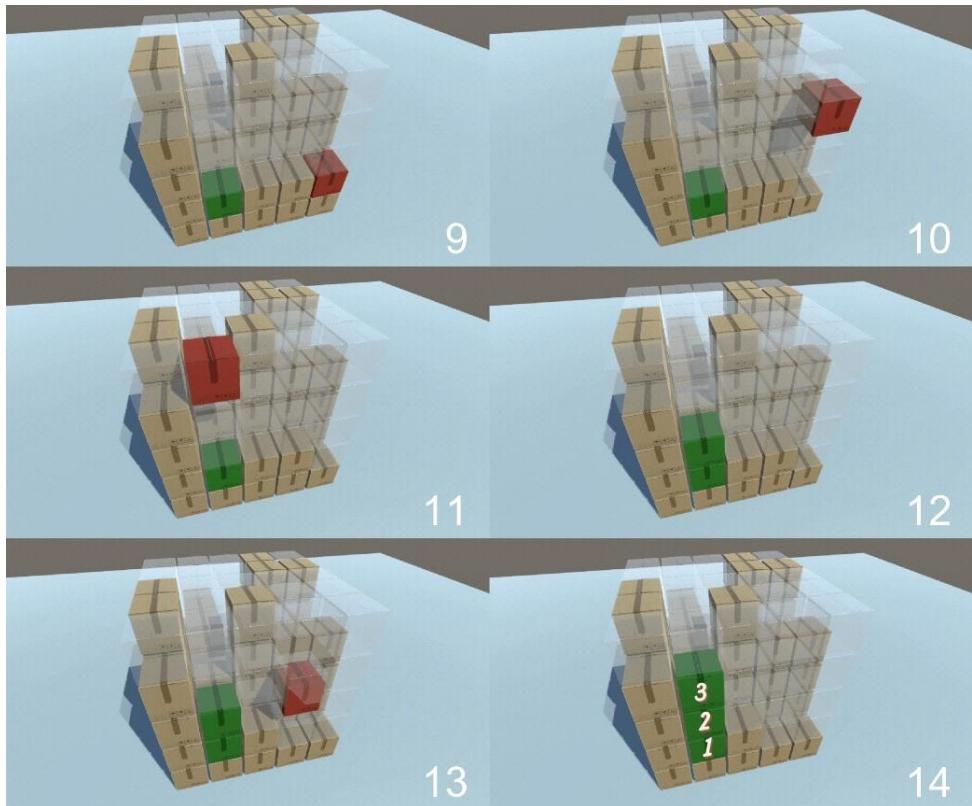


Şekil 5.4 : Bölüm 1 ve 2: Problemlı yeniden konumlandırma ve çözüm

#### 4.2.1.5.3 Örnek 3

1. ve 2. aşamaların kriterlerine göre eğitilen modelde son bölümlerdeki hareketin oturmuş hali şekil 5.5'te gösterilmektedir. Üç kutuya sıralı olarak komut verilmiş ve bu kutular en verimli şekilde hedeflerine taşınmıştır. Bu örnek, algoritmanın öğrenme süreci başarıyla tamamlandığında kutuları en az çaba ile hedeflerine ulaştırdığını ortaya koymaktadır.





**Şekil 5.5 :** Kutuların sistem oturduktan sonra sıralı hareketi

#### 4.2.1.6 Seçilmiş kutular için yol temizleme optimizasyonu

Taşınması başta emredilen (kırmızı) bir kutu bir engelle karşılaşlığında, bu engelin yeniden konumlandırılması, sonraki bölümlerin başında ele alınır. Bu müdahale, 2. aşamada tanıtılan kutu temizleme kurallarına uygun olarak, verimlilik ve zamanlama odaklıdır ve aynı algoritma dahilinde ele alınır. Önemle belirtmek gerekmek ki, bu önceliklendirme yalnızca seçilmiş yani en başında bir yerden bir yere gitmesi emredilmiş kutular için geçerlidir ve engellere neden olan kutuları kapsamaz. Engele sebep olan kutuların da bu kurala dahil edilmesi durumu fazla karmaşık hale getirebileceğinden tercih edilmemiştir, ayrıca bu kutuların alternatif yollar bulabileceği ihtimali daha fazladır yani engel durumunun zararı daha azdır.

Ek olarak, seçilmiş kutunun yolu, 1. aşama stratejilerine göre artık engellenen alanı geçmeyecek şekilde değiştirildiğinde, engelin yeniden konumlandırılması gereksiz hale gelir ve gelecekteki bölümlerde iptal edilebilir. Bu değişiklik, gereksiz

müdahaleler olmadan verimli yol temizliği sağlamak için dinamik bir ayarlamayı yansıtır.

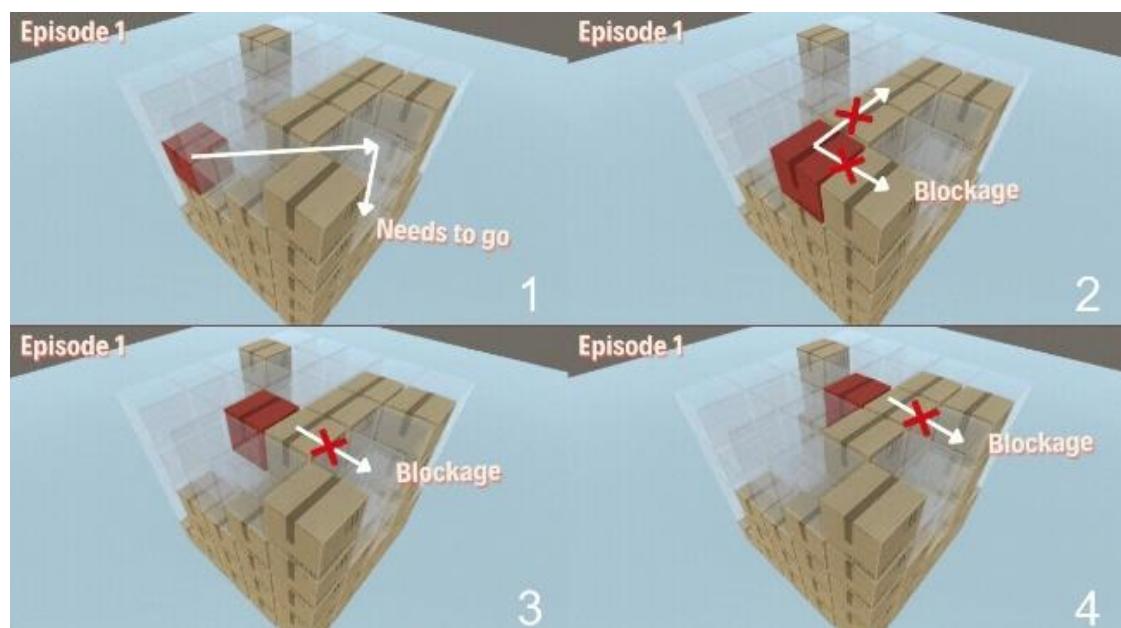
Not edilmelidir ki, mantık benzer olsa da, bu, 2. aşamanın önceki bölümlerinde bahsedilen yerel kutu temizlemeden biraz farklıdır. Yerel kutu temizleme, seçilmiş kutuların üzerindeki kutuları temizlemeyi amaçlıyordu. Burada ise seçilmiş kutuların yolda karşılaştığı büyük engellerin temizlenmesi stratejisi geliştirilmektedir.

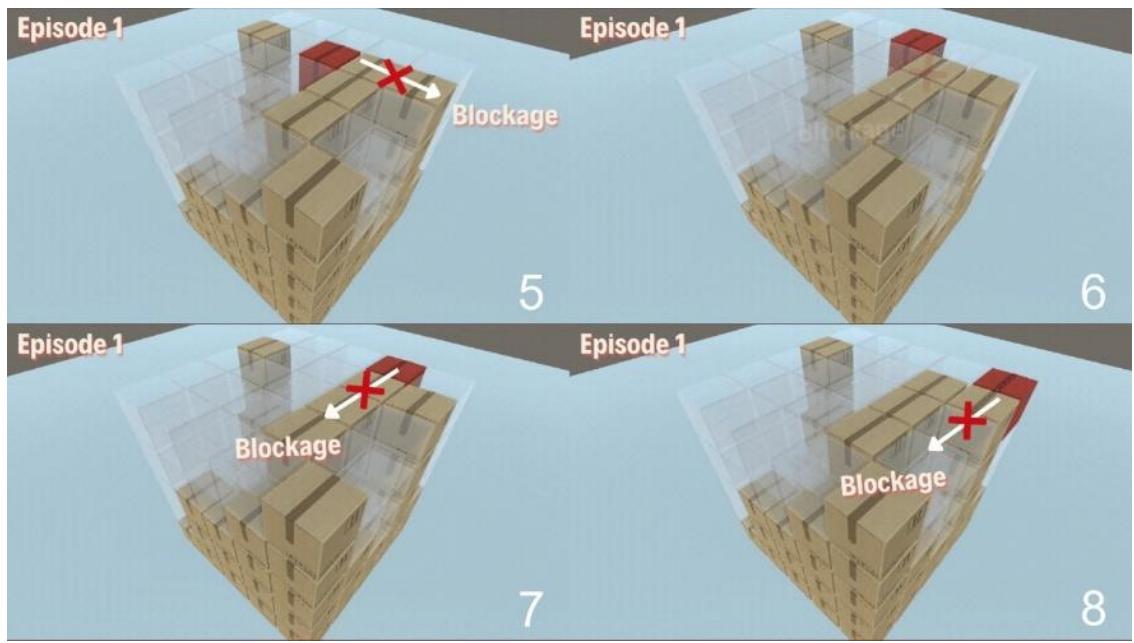
Toplam sonucu geri bildirmek, 1. aşamada hangi yolu tercih ettiğini ve 1. ve 2. aşamalarda toplam verimlilik sonucunun nasıl maksimize edilebileceğini içermektedir ki bu, yerel kutu temizlemede önemli değildi.

#### 4.2.1.7 Uygulama

Aşağıda, ajanın açıklanan konsepti kullanarak bölgeler halinde koşullarla nasıl başa çıktığını detaylı bir şekilde gösteren bir örnek sunulmaktadır.

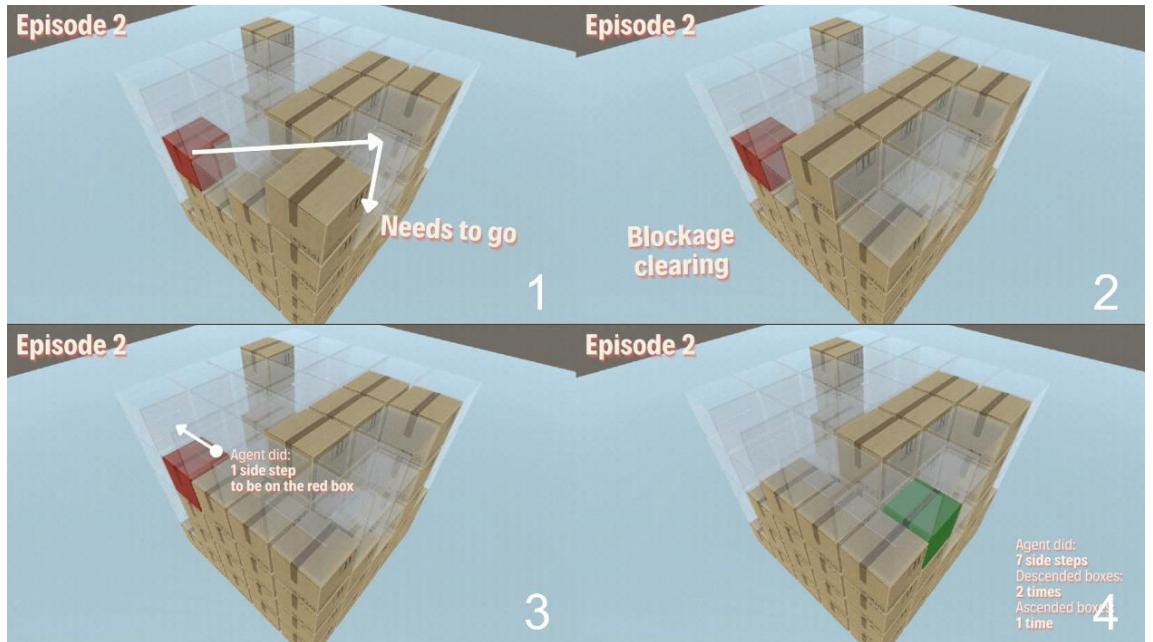
Bölüm 1: Kutu, hedefine ilerlerken fazla engelle karşılaşır ve bu engeller temizlenmeden kutunun alana giriş yapması imkansız olduğu fark edilir. Ajan, bir adım kutuyu almak ve altı adım diğer kutuları yeniden konumlandırmak üzere toplamda yedi yan adım atar.





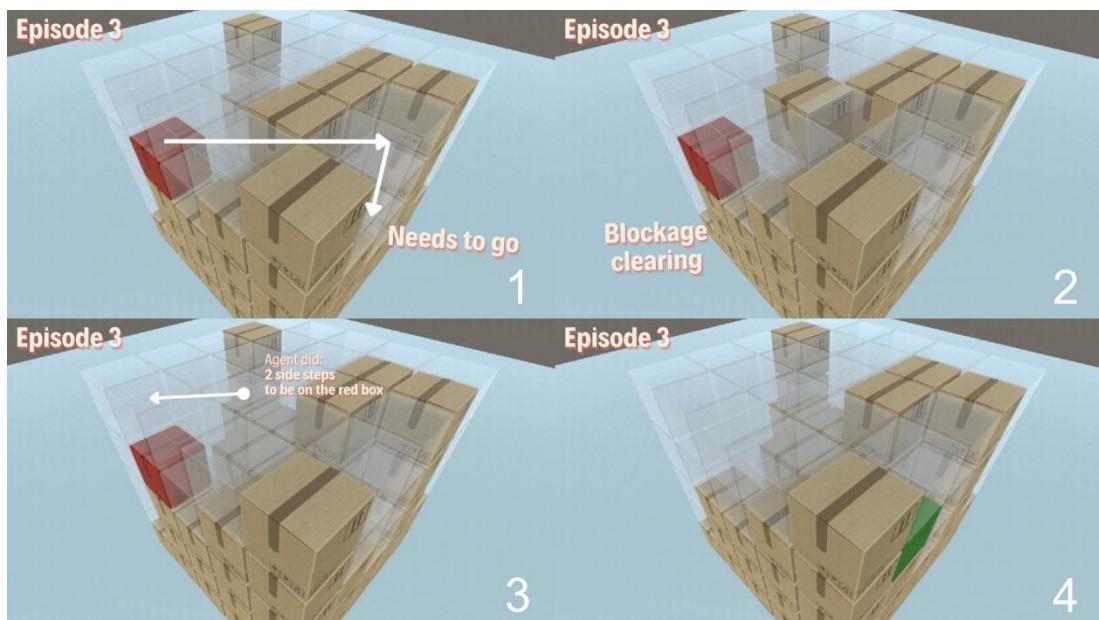
**Şekil 5.6 : Bölüm 1: Çoklu Engeller**

Bölüm 2: Bölümün başlangıcında, bölüm 1'de engelleyici olarak algılanan bir kutu, kutu temizleme mekaniklerine dayalı olarak yeniden konumlandırılır. Bu noktadan sonra kutunun hedefe götürülmesi için yol açılmıştır. Ajan, bir adım kutuyu almak ve altı adım diğer kutuları yeniden konumlandırmak üzere toplamda yedi yan adım atar.



**Şekil 5.7 : Bölüm 2: Engelleyicilerin yeniden konumlandırılması**

Bölüm 3: Seçilen kutunun yolunu temizlemek için daha verimli bir yöntem keşfedilir. Bölüm 2'deki iki yan adım ve bir aşağı adım yerine sadece bir yan adım ve bir aşağı adım gerektiren başka bir bloklayan kutu yer değiştirilecektir. Uygulanan bu desen, 1. aşamanın hedefleriyle iyi uyum sağlar ve bu yöntemi varsayılan temizleme yaklaşımı haline getirir. Kutunun önünü tıkayan birden fazla kutunun olması durumunda seçeneklerin değerlendirileceğini gösterir. Toplamda ajanın yaptığı hareket sayısının tersi ödül olarak yer değiştiren kutuya yansıtılır ve en fazla ödülü olan yol açma metodu seçilir. Ajan, iki adım kutuyu almak ve beş adım diğer kutuları yeniden konumlandırmak üzere toplamda yedi yan adım atar.



**Şekil 5.8 :** Bölüm 3: Yeniden konumlandırma sırasında artan verimlilik

Ajanın yeniden konumlandırma süresi üzerine not: Ajanın kutular üzerinde yeniden konumlanma süresi (toplama süresi) odak noktası değildir, çünkü 3. aşamada komut dizileri birden fazla ajan tarafından eşzamanlı olarak yürütülecektir, bu da bu zaman boşluğunu geçersiz kılar. Bunun yerine, algoritmamız şimdi, birden fazla ajanın bulunduğu senaryolarda daha verimli sonuçlar elde etmek için, 2. Aşamanın kutu temizleme yöntemlerini kullanarak bir ajanın teslimat sırasında atması gereken adımları azaltmaya öncelik vermektedir. Böylece, kutuları yeniden konumlandırma adımları 6'dan 5'e düşürüp üçüncü bölüm daha avantajlı olacaktır.

## **5. SÜRÜ ZEKÂSI TASARIMI VE UYGULAMASI**

Projenin 3. aşaması olarak sistemi daha efektif kılmak için sürü zekâsı tasarlamaya ve uygulamaya karar verilmiştir. Her bir ajan bir birey olarak düşünülmüştür. Tek bir ajanın geniş bir haritada komutları yerine getirmekte yetersiz kalabileceği göz önünde bulundurularak birden fazla ajanın kullanımıyla bu sorunu çözmek istenmiştir. Birden fazla ajan, aynı anda çalışarak hem verimliliği artıracak hem de daha geniş bir izgara alanını yönetebilecektir.

Birden fazla ajanın koordineli çalışması sayesinde, her bir ajanın belirli bir görev veya bölgeye odaklanması mümkün olacak ve böylece genel işlem süresi önemli ölçüde azaltılacaktır. Sürü zekâsı, ajansların birbirleriyle etkileşim halinde çalışmasını, bilgi paylaşmasını ve dinamik olarak görevlere uyum sağlamaşını sağlayarak sistemin esnekliğini ve adaptasyon yeteneğini artıracaktır.

Ayrıca, sürü zekâsı yaklaşımıyla, herhangi bir ajanın başarısız olması durumunda diğer ajanlar devreye girerek görevi sürdürülecektir. Bu, sistemin güvenilirliğini ve dayanıklılığını önemli ölçüde artırarak kesintisiz bir operasyon sağlamaktadır. Ajanlar arasında görev paylaşımı ve işbirliği, her bir ajanın yükünü hafifleterek daha verimli ve etkili bir çalışma ortamı yaratmaktadır.

Sonuç olarak, sürü zekâsının tasarımını ve uygulanmasını, geniş ve karmaşık haritalarda daha etkin bir malzeme taşıma ve yerleştirme süreci sağlayarak, genel sistem performansını ve verimliliğini önemli ölçüde artıracaktır.

### **5.1.1 Tasarım**

Bu bölümde, sürü zekâsı sisteminin tasarımını ve ajanların simülasyon ortamındaki etkileşimleri ve birlikte çalışmayı öğrenmeleri gibi önemli konular ayrıntılı olarak ele alınacaktır. Ajanların yerleşimi, hareket stratejileri ve etkileşim mekanizmaları gibi konulara odaklanılacaktır. Ayrıca, sürü zekâsı yaklaşımının nasıl uygulandığı ve bu uygulamanın sistem performansına etkileri inceleneciktir.

#### **5.1.1.1 Ajanların özelliklerı**

Ajanların sahip olduğu çeşitli özellikler, sistemin genel verimliliğini ve etkinliğini artırmak için tasarlanmıştır. Bu özellikler, ajanların bireysel yeteneklerini maksimize ederken, küresel beyinle uyumlu bir şekilde çalışmasını sağlar:

- **Görüş alanı:** Bu projenin en önemli bileşenlerinden biri, belirli büyüklükteki sınırlı bir alanda iş yapma becerilerini öğrenmeleridir. Bu mikro öğrenme yaklaşımı, bireylerin detaylı ve odaklanmış bir şekilde bilgi edinmesini sağlar. Aynı zamanda, üst akıl olarak tanımlanan merkezi kontrol birimi, küresel beyin, bireyleri belirli görevlere ve konumlara yönlendirir. Bu yapı, yerel düzeyde detaylı bir öğrenme modeli ile küresel düzeyde bir yönlendirme planını birleştirir. Böylece, görevlerin etkin bir şekilde tamamlanması sağlanır ve öğrenme süreci optimize edilir.

Bu sistemin yokluğunda, tüm depoyu kapsayan karmaşık bir öğrenme modeline ihtiyaç duyulurdu. Bu tür geniş kapsamlı bir model, deponun büyülüğu nedeniyle öğrenme sürecini neredeyse imkansız hale getirirdi. Ancak mevcut yapı, hem mikro düzeyde detaylı eğitim hem de makro düzeyde genel yönlendirme sağlayarak verimliliği artırır. Bu sayede, bireyler küçük alanlarda iş yapma becerilerini geliştirirken, üst akıl da genel plan çerçevesinde yönlendirme yaparak karmaşıklığın üstesinden gelinir. Bu iki aşamalı yaklaşım, öğrenme sürecini daha yönetilebilir ve etkili kılar.

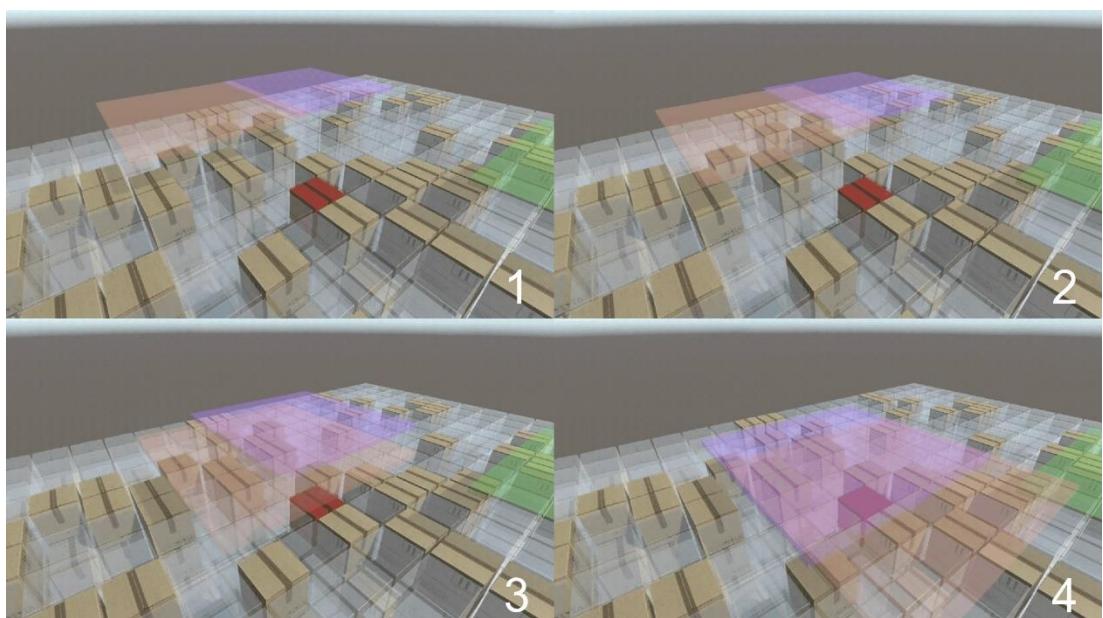
Buna göre, her bir ajanın görüş alanı, kendi renkli düzlemi ile sınırlıdır ve bu nedenle algıları yereldir. Bu yerel algı, ajanların çevrelerini detaylı bir şekilde gözlemleyip, anlık kararlar almasını sağlar. Aynı zamanda, ajanların kendi alanlarına odaklanarak görevlerini daha hızlı ve verimli bir şekilde yerine getirmelerine olanak tanır.

- **Deneyimin bireyler arasında aktarılması:** Tüm ajanlar, 1. ve 2. aşamalardan elde edilen Q tablosu ve Q ağaçlarına erişime sahiptir, bu sayede daha önce bir ajan tarafından öğrenilen herhangi bir ortamda etkili bir şekilde kutu taşıyabilirler. Bu merkezi veri tabanı, her ajanın daha önce edinilmiş bilgilerden yararlanarak hareket etmesini sağlar. Böylece, ajanlar sadece kendi deneyimlerinden değil, tüm sistemin kolektif deneyimlerinden de öğrenirler.
- **Hareketlilik ve görevler:** Ajanlar, kutu taşıyarak veya küresel beyin tarafından belirli alanlara yönlendirilerek serbestçe hareket edebilirler. Bu hareket kabiliyeti, ajanların görüş alanını dinamik şekilde değiştirmelerine olanak sağlar. Görevler arasında geçiş yapabilme yeteneği, ajanların çok yönlülüğünü artırır ve sistemin genel esnekliğini destekler.

- **Sınırlı çalışma alanları:** Küresel beyin, ajanların belirlenen çalışma bölgelerinden ayrılmamalarını sağlamak için sınırlı alanlar tanımlayabilir. Bu, çalışma yoğunluğunun düzlem boyunca eşit olduğu durumlarda kritik öneme sahiptir. Sınırlı çalışma alanları, ajanların çakışma olasılığını azaltır ve belirli bölgelerde yoğunlaşarak daha verimli çalışmasını sağlar.
- **Dinamik tahsis:** Küresel beyin, görevin zorluk veya karmaşıklığına bağlı olarak en yakın ajanları çalışma alanına çağırabilir. Bu özellik, kaynakların optimal kullanımını sağlar. Ajanlar, görevin gerektirdiği yerlerde hızla devreye girerek genel sistem verimliliğini artırır ve işlerin daha hızlı tamamlanmasını sağlar.

Ajanların bu özellikleri, küresel beyin ile koordineli bir şekilde çalışarak sistemin genel performansını optimize eder. Küresel beyin, merkezi bir kontrol mekanizması olarak, ajanların faaliyetlerini gerektiginde yönlendirir ve müdahale eder. Bu aydınlatma, ajanların yerel düzeyde bağımsız ve hızlı kararlar (inisiyatif) alabilmelerine olanak tanırken, küresel beynin de genel strateji ve planlama yapabilmesini sağlar. Bu sayede, sistem hem yerel hem de küresel düzeyde etkili bir şekilde çalışır.

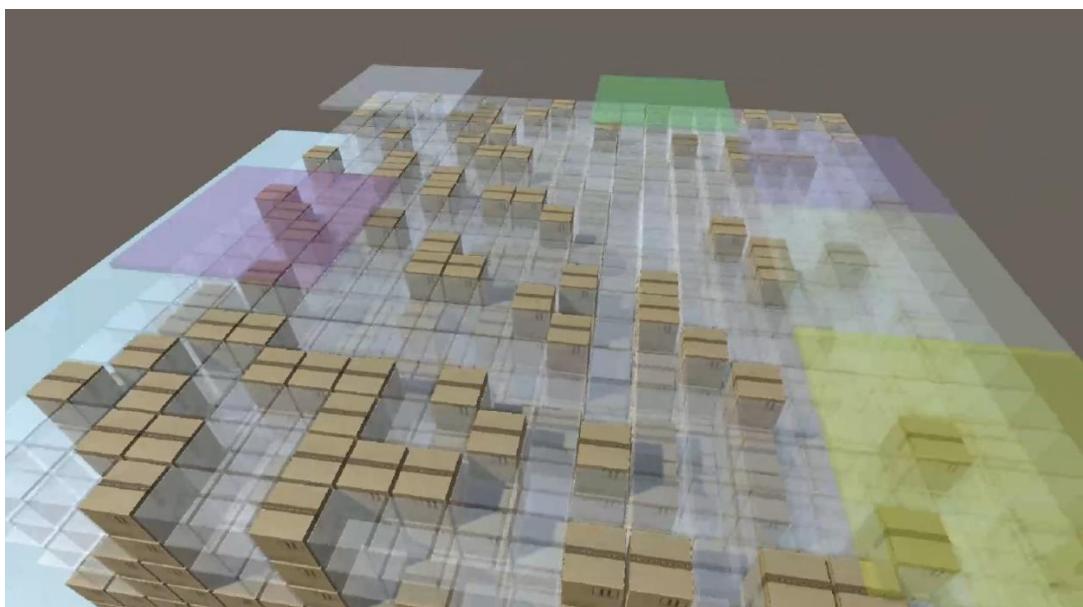
Şekil 5.9, küresel beynin en yakın ajanları çalışma alanına nasıl çağrıdığını göstermektedir. Çağırılan ajanların görüş alanları turuncu ve mor, saydam olarak görselleştirilmektedir.



**Şekil 5.9 : Küresel beyin ajanları koordine ediyor**

### 5.1.1.2 Ajanların görselleştirmesi

Ajanların görüş alanını kutular üzerinde görselleştirmek için simülasyon ortamında renkli düzlemler kullanıldı. Taşıma işlemi sırasında kutuları taşıyan ajanların opaklısı azaltılarak daha net bir görsellik sağlandı. Ayrıca, yol üzerindeki aktiviteleri daha iyi gözlemelemek ve bunun üzerine tasarım yapmak ve simüle etmek amacıyla simülasyon alanı 20x10x20 olarak yapılandırıldı. Ajanların sayısı ve yerleri gerçeğe uygun şekilde rastgele olarak ızgara sistemi üzerinde başlatılmaktadır. Şekil 5.10, ajanlar ve harita görselleştirmelerini gösterilmektedir.



**Şekil 5.10 :** Ajan ve harita görselleştirmesi

### 5.1.1.3 Alan kısıtlamaları

Küresel beyin gereği ajanların hareket alanlarına kısıtlamalar getirebilmektedir. Bu kısıtlamalar, belirli bölgelerde iş yoğunluğunu dengelemek ve ajanların çakışmalarını önlemek için kritik öneme sahiptir.

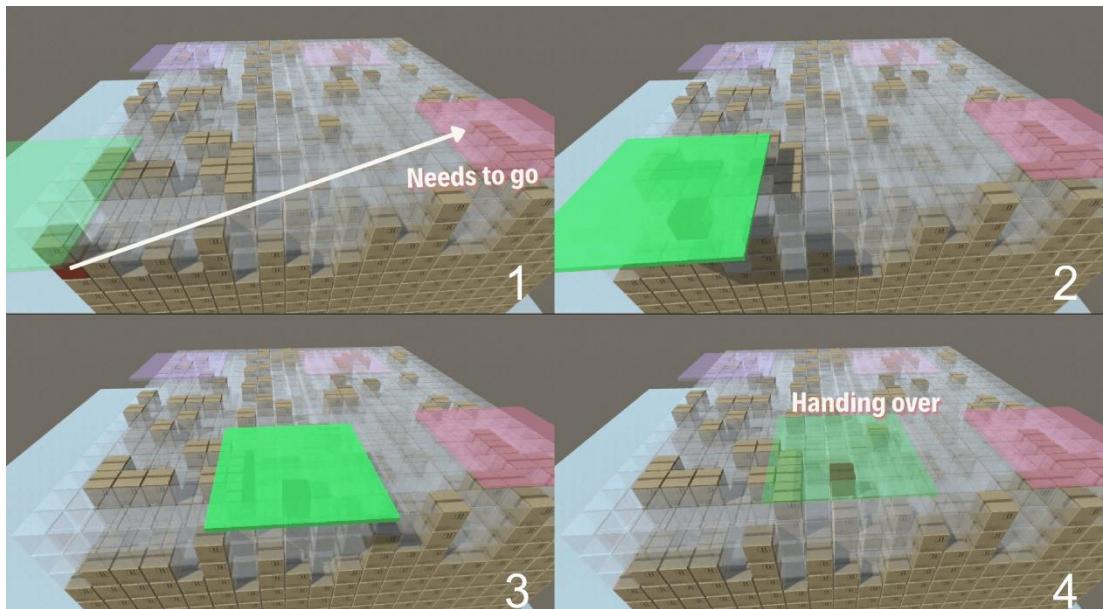
Ajanların bu tür kısıtlamalara ihtiyaç duymalarının birkaç önemli nedeni vardır:

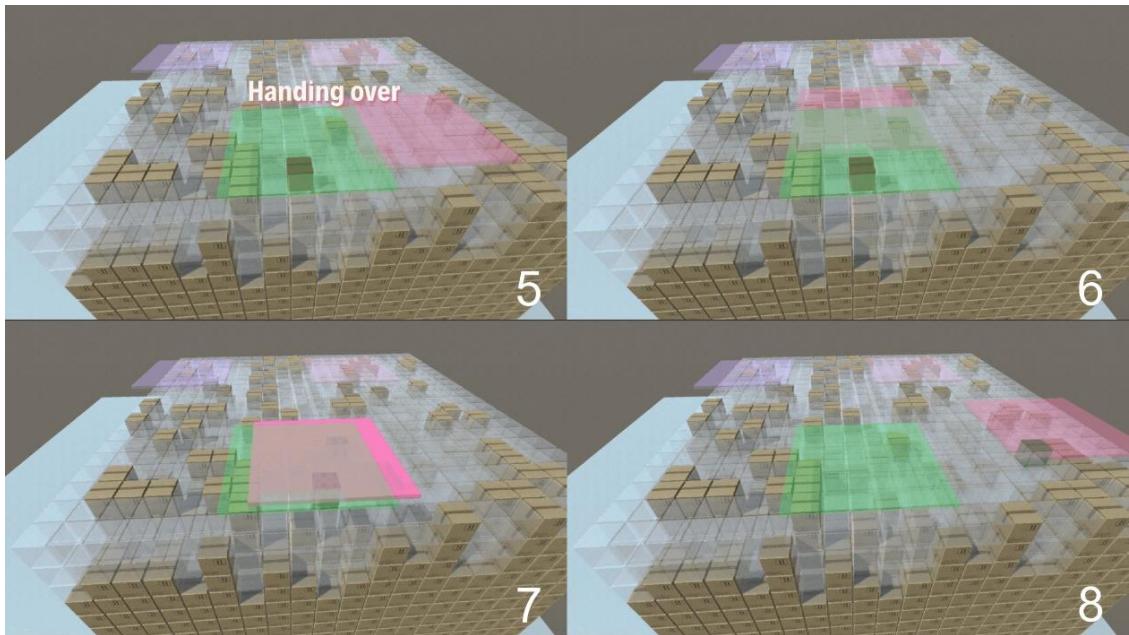
- **Etkili Alan Kullanımı:** Sınırlı çalışma alanları, ajanların belirli bölgelere odaklanması sağlar, böylece her ajanın kendi alanında daha verimli çalışması mümkün olur. Bu, çakışmaları ve gereksiz hareketleri azaltarak genel verimliliği artırır.

- **Koordineli Çalışma:** Ajanların belirli alanlarda çalışması, küresel beyin tarafından daha iyi koordine edilmelerini sağlar. Küresel beyin, her ajanın hareketlerini izleyerek gerekiğinde müdahale edebilir ve kaynakları en verimli şekilde kullanabilir.
- **Görev Paylaşımı:** Sınırlı alanlar içinde çalışan ajanlar, görevleri paylaşarak ve işbirliği yaparak daha karmaşık görevleri daha hızlı tamamlayabilirler. Bu, ajanların birbirlerine yardım etmelerini ve iş yükünü dengelemelerini sağlar.

Küresel beyin ve ajanlar arasındaki bu ayırım, sistemin genel performansını optimize etmek için tasarlanmıştır. Küresel beyin, merkezi bir kontrol mekanizması olarak, genel strateji ve planlama yaparken, ajanlar yerel düzeyde bağımsız ve hızlı kararlar alabilirler. Bu yapı, hem küresel beyinin hem de ajanların güçlü yönlerini kullanarak esnek ve etkili bir sistem oluşturur.

Şekil 5.11'deki senaryoda, küresel beyin ajanlarının hareket alanlarına kısıtlamalar getirmiştir. Sonuç olarak, bir ajan görev sırasında kutusunu başka bir ajana devretmek zorunda kalmıştır. Bu durum, operasyonel çevre içindeki mekansal kısıtlamaları yönetmek için dinamik ve koordineli bir yaklaşımı göstermektedir.





**Şekil 5.11 :** Alan kısıtlamaları nedeniyle ajan görev devri

#### 5.1.1.4 Ajanların varsayılan davranışları

Varsayılan davranışlar, ajanların kendi yetenekleri ve karar verme mekanizmları doğrultusunda hareket etmelerini sağlar. Bu davranışlar şunları içerir:

- **Görüş Alanı:** Ajanlar, kendi renkli düzlemleri ile sınırlı olan yakın çevrelerini gözlemler. Görüş alanı, ajanların çevrelerini hızlı ve etkili bir şekilde değerlendirmelerine yardımcı olur, yani eğer bulunduğu bölgede bir kutu taşınacaksa varsayılan olarak, global beyinden aksine komut gelmedikçe bu görevi üstlenmeye tercih eder.
- **İnisiyatif:** Ajanlar, görüş alanlarındaki herhangi bir kutunun hareketini başlatma yeteneğine sahiptir. Bu yetenek, ajanların görevlerini yerine getirirken proaktif olmalarını sağlar ve görev sürelerini kısaltır. Her bir ajanın kendi inisiyatifiyle hareket edebilmesi, sistemin genel esnekliğini artırır.
- **Takip:** Ajanlar, taşımakla görevlendirildikleri kutuyu takip ederler. Ajanların kutu taşıırken gerçek pozisyonu, taşıdıkları kutunun tam üzerinde bulunur ve hareketin sonunda görüş alanının ortasında kalır. Bu davranış, ajanların her bir görevi bitirdikten sonra hâlâ görev merkezi kalmasına yardımcı olur.
- **Görev Tahsisi:** Bir kutunun taşınması gerekiğinde iki ajanın görüş alanı aynı kutunun üzerinde yer alıyorsa, en yakın olan ajan görevi üstlenir. Bu davranış,

görevlerin hızlı ve etkili bir şekilde dağıtılmasını sağlar ve ajanlar arasındaki çakışmaları en aza indirir.

Ajanların bu varsayılan davranışları, sistemin genel verimliliğini ve uyumunu artırmak için tasarlanmıştır. Küresel beyin, gerekli gördüğü durumlarda müdahale ederek ajanların hareketlerini koordine eder, ancak bu varsayılan davranışlar, ajanların bağımsız olarak etkili bir şekilde çalışabilmelerini sağlar. Varsayılan davranışlar, ajanların özerkliklerini korurken, küresel beyinle uyum içinde çalışmalarını sağlar. Ajanlar, yerel düzeyde kararlar alarak çevrelerine hızlıca uyum sağlayabilir ve görevlerini yerine getirebilir.

#### **5.1.1.4 Ajan çağrıma kararı için Q-Learning modeli**

Küresel beynin bir görevi yerine getirmek için çağrılmazı gereken optimal ajan sayısını dinamik olarak belirlemek için yeni bir Q-learning algoritması tasarlanmıştır. Karar verme süreci, operasyonların karmaşıklığını ve kaynakların verimli bir şekilde tahsis edilmesi ihtiyacını yansıtan çeşitli faktörler tarafından yönlendirilir.

##### **5.1.1.4.1 Kullanılan Q-tablosu yapısı**

Projenin 1. aşamasındaki benzer şekilde veritabanı olarak bu modelde de Q-tablosu kullanılmıştır. Bu aşamadaki tablonun spesifik özellikleri özetlenmiştir:

- **Durumlar (States):** Q-tablosundaki her durum, görevin merkezinde yer alan yani taşınması gereken kutunun 3D ızgara pozisyonunu temsil eder. Bu, küresel beynin her görevin konumuna özgü geometrik gereksinimleri ve kısıtlamaları değerlendirmesine olanak tanır.
- **Eylemler (Actions):** Her eylem, "ajan ihtiyacını" belirten bir gerçek sayı (floating number) değeri yansıtır. Bu, küresel beynin optimal görev performansı için gerekli gördüğü ajan sayısını ifade eder. Bu sayının yuvarlanmış değeri, kaç ajanın çağrılabacağını belirler. Örneğin, bu değer 1.6 ise, 2 ajan çağrılacaktır.

### **5.1.1.5 Öğrenme ve ödül mekanizmaları**

Öğrenme sürecinde tamamlanan görevlerin sonuçlarına ve verimliliğine dayalı olarak o görev için ajan ihtiyacı değerlerini ayarlar ve aşağıdaki ödül yapısını kullanır:

- **Etkisiz ajan kullanımı için ceza:**

Ajanlar iş yerine çağrılmış etkili bir şekilde kullanılmadığında, yani gerekli kaynakların fazla tahmin edilmesi durumunda, dinamik bir negatif ödül (ceza) değeri ayarlanır. Bu ayarlama, ajanın çalışma süresinin o iş yerindeki toplam ajan çalışma süresine oranına ters orantılı olarak yapılır ve benzer gelecekteki durumlarda kaynak israfını önlemek için olay yerindeki kutuya denk gelen durumda, aksiyon olarak ajan ihtiyacı değerini azaltarak gelecekte uygun sayıda ajan çağrımasına yardımcı olur. Ceza değeri ( $r_i$ ), ajanın katılım oranı normalize edilerek -2 ile 0 arasında ölçeklendirilir. Matematiksel olarak Denklem 5.1'de ifade edilmiştir. Bu ödül, direkt olarak çalışmaya katılma üzerinden performansına göre anlık olarak modeli eğitir.

$$r_i = -2 \times \left( 1 - \frac{\text{Ajanın Katıldığı Program Çevrimi}}{\text{Toplam Program Çevrimi}} \right) \quad (5.1)$$

Uygulama:

Örneğin, diğer ajanlar toplamda 10 program çevrimi boyunca çalışırken, bir ajan sadece 2 program çevrimine katıldıysa Denklem 5.1'e göre cezası şu şekilde hesaplanır:

$$r_i = -2 \times \left( 1 - \frac{2}{10} \right) = -2 \times 0.8 = -1.6$$

Özetle mantık aşağıdaki gibi verilmiştir:

- Bir ajan iş yerine çağrıldığında ve etkili bir şekilde kullanılmadığında, dinamik bir negatif ödül (ceza) alır.
- Bu ceza, ajanın çalışma süresinin toplam ajan çalışma süresine oranına ters orantılı olarak hesaplanır.
- Ceza değeri, ajanın katılım oranı normalize edilerek -2 ile 0 arasında ölçeklendirilir.

- **Çalışma yoğunluğuna dayalı ayarlama:**

- **Çalışma Yoğunluğunun Hesaplanması:** Her görev için ortalama çalışma yoğunluğu, her bir ajanın harcadığı süre içindeki katkısını değerlendirerek hesaplanır. Bu yoğunluk, taşınan toplam kutu sayısının harcanan toplam süreye bölünmesi ve bu değerin görevde yer alan ajanlar arasında ortalamasının alınması ile belirlenir.

Matematiksel olarak bu hesaplama Denklem 4.19'da ifade edilir:

$$Y = \frac{1}{n} \sum_{i=1}^n \left( \frac{k_i}{t_i} \right) \quad (5.2)$$

Bu denklemde:

- $Y$  çalışma yoğunluğunu,
- $n$  görevde yer alan ajan sayısını,
- $k_i$  her bir ajanın taşıdığı kutu sayısını,
- $t_i$  her bir ajanın harcadığı süreyi ifade eder.

Örneğin, toplamda 12 saniyede, 1. ajan 3 kutu taşır ve 2. ajan 5 kutu taşırsa, ortalama çalışma yoğunluğu aşağıdaki gibi hesaplanır:

$$Y = \frac{1}{2} \left( \frac{3}{12} + \frac{5}{12} \right) = \frac{1}{2} (0.25 + 0.4167) = 0.333$$

- **Bölümler Arası Karşılaştırma:** Hesaplanan çalışma yoğunluğu, farklı bölümler arasındaki görev verimliliğini değerlendirmek için kullanılır. Bu karşılaştırmalar, ajanların zaman ve çabalarının ne kadar verimli kullanıldığı ortaya koyar.

Eğer aynı sayıda kutu 12 saniye yerine 9 saniyede taşınabiliyorsa, çalışma yoğunluğu aşağıdaki gibi hesaplanır:

$$Y = \frac{1}{2} \left( \frac{3}{9} + \frac{5}{9} \right) = \frac{1}{2} (0.333 + 0.556) = 0.444$$

Bu yeni yoğunluk değeri, ajanların zamanlarını ve çabalarını daha verimli kullandıklarını gösterir. Yüksek çalışma yoğunluğu, daha kısa sürede daha fazla iş yapıldığını ve böylece görevlerin daha verimli bir şekilde tamamlandığını yansıtır. Bu tür karşılaştırmalar, bölümler arası

performans farklılıklarını belirlemek ve verimliliği artırmak için gerekli iyileştirmeleri yapmak adına önemlidir.

- **Bir Ajan Eklenmesi Durumu:** Üçüncü bir ajan eklenirse ve bu ajan 2 ek kutu taşıyarak toplamda 9 saniyede 3 ajan tarafından 10 kutu taşınırsa, yeni ortalama çalışma yoğunluğu aşağıdaki gibi hesaplanır:

$$Y = \frac{1}{3} \left( \frac{3}{9} + \frac{5}{9} + \frac{2}{9} \right) = \frac{1}{3} (0.333 + 0.556 + 0.222) = \frac{1.111}{3} \approx 0.37$$

Bu yoğunluk, önceki bölümlerin yoğunluklarıyla karşılaştırılarak verimlilik kazançları veya kayıpları belirlenir. Burada, yeni yoğunluk olan 0.37 bir düşüş gösterir. Bu, üçüncü ajanın eklenmesiyle birlikte çalışma yoğunluğunun azaldığını ve ajanların zamanlarının daha az verimli kullanıldığını gösterir. Bu tür analizler, ekip büyülüüğünün ve dağıtımının verimlilik üzerindeki etkilerini anlamak için önemlidir.

- **Ödül Ayarlaması:** Ödüller, bu yoğunluk hesaplamalarına dayalı olarak ayarlanır. Amaç, ajan zamanının ve çabasının verimli kullanıldığını göstermek için optimal bir yoğunluk aralığında kalmaktır. Çalışma yoğunluğu 0.25 ile 0.45 arasındaki optimal aralıkta kaldığında pozitif bir ödül verilir. Bunun tersine, çalışma yoğunluğu bu aralığın dışına çıkarsa, kaynakların verimsiz kullanıldığını veya aşırı yüklenildiğini gösterir ve negatif bir ödül uygulanır.

Ödül hesaplaması Denklem 5.3'te verilmiştir:

$$r_y = \begin{cases} k \times (x - 0.25) & \text{eğer } 0.25 \leq x \leq 0.45 \\ -k \times |x - 0.35| & \text{eğer } x < 0.25 \text{ veya } x > 0.45 \end{cases} \quad (5.3)$$

Burada  $k$ , ödülen büyülüğünü ayarlayan bir katsayıdır ve 1 olarak belirlenmiştir.

### Örnekler:

- Ortalama çalışma yoğunluğu 0.35 ise (optimal aralıktaki):

$$r_y = k \times (0.35 - 0.25) = 0.10$$

- Ortalama çalışma yoğunluğu 0.20 ise (optimal aralık dışında):

$$r_y = -k \times |0.20 - 0.35| = -0.15$$

- Ortalama çalışma yoğunluğu 0.50 ise (optimal aralık dışında):

$$r_y = -k \times |0.50 - 0.35| = -0.15$$

Bu ödül sistemi, düşük yoğunluklu veya aşırı yoğunluklu çalışma koşullarında negatif ödüllerle kaynak israfını ve aşırı yüklenmeyi engellemeye çalışır. Ödüller nispeten düşük tutulur ve yavaş yavaş etki eder, böylece modelin zamanla daha verimli çalışma koşullarını öğrenmesi sağlanır.

Bu yöntem, ajanların dağılımının gerçek performans verilerine dayanarak sürekli optimize edilmesini sağlar ve gelecekteki görevler için en verimli ve etkili stratejileri teşvik eder.

- **Seyahat mesafesi için negatif ödül:**

Bir ajanın iş sahasına ulaşmak için kat etmesi gereken mesafeye orantılı olarak negatif bir ödül (ceza) atanır. Bu ceza, ajanın kat ettiği mesafe  $d$  ile doğru orantılıdır ve  $k$  sabit ceza katsayısı ile çarpılarak hesaplanır. Matematiksel olarak bu ceza Denklem 5.4'te ifade edilir:

$$r_d = -d \times k \quad (5.4)$$

Bu ceza, ajanların gereksiz yere uzun mesafeler boyunca hareket etmesini caydırır, yerel kaynak kullanımını teşvik eder ve zaman gecikmelerini azaltır. Sistemimizde  $k$ , 0.02 olarak belirlenmiştir.

**Örnek:**

İş almak için 10 birim mesafeden gelen bir ajan, aşağıdaki gibi bir ceza alacaktır:

$$r_d = -10 \times 0.02 = -0.2$$

Bu ceza, ajanların daha kısa mesafelerden iş almasını teşvik eder ve sistemin genel verimliliğini artırır.

- **Planlanan iyileştirmeler:**

Bir ajanın mevcut konumuna yakın bekleyen görevleri olup olmadığı gibi ek hususlar, karar verme sürecini daha da iyileştirmek için dahil edilebilir.

Yakınında bekleyen görevleri olan ajanlar, uzak konumlara çağrılmak için daha az uygun olarak işaretlenebilir, bu da genel sistem verimliliğini optimize eder.

Sonuç olrak bu fazda, toplam ödül ( $r$ ) Denklem 5.5'te gösterildiği gibi hesaplanır:

$$r = r_i + r_y + r_d \quad (5.5)$$

### 5.1.1.6 Ajanların kendine görev atama modeli

Ajan çağrıma süreci tamamlandıktan sonra iş akışındaki bir sonraki kritik karar, görev yerindeki iş grubundaki hangi kutunun hangi ajan tarafından ele alınacağının belirlenmesidir. Bu karar verme süreci, her çalışma yerine özgü yerel bir Q-learning modeli eğitmek suretiyle yapılır.

#### 5.1.1.6.1 Yerel Q-tablosunun yapısı

Projenin en önemli bileşenlerinden biri, bireylerin belirli büyüklükte sınırlı bir alanda iş yapma becerilerini öğrenmeleridir. Bu mikro öğrenme yaklaşımı, bireylerin detaylı ve odaklanmış bir şekilde bilgi edinmesini sağlar. Bu çerçevede, yerel Q-tablosu, bireylerin her bir durumda hangi eylemleri seçmeleri gerektiğini belirleyen bir karar verme aracıdır. Q-tablosu, her bir durum-eylem çiftine ilişkin ödül beklentilerini saklar ve bu beklentiler zamanla güncellenerek bireylerin en uygun eylemleri öğrenmelerine yardımcı olur. Yerel Q-tablolarının kullanılması, öğrenme sürecini daha yönetilebilir kılar ve bireylerin küçük, sınırlı alanlarda etkili bir şekilde iş yapabilmesini sağlar. Bu yapı, daha büyük ve karmaşık sistemlerin parçalarına ayrılmasını ve her bir parçanın optimize edilmesini mümkün kılar. Yerel Q tablolarına ilişkin durumlar ve aksiyonlar detaylı olarak verilmiştir:

- Durumlar (States):** Q-tablosundaki her durum, iş yerinde taşınması gereken kutuların kimliklerini temsil eder ve 3D ızgaradaki o anki yerlerini tutar. Önceki aşamalardan farklıdır, 3D ızgaradaki yeri değişse bile kimlikleri aynı kalır, ancak önceki aşamalarda sadece yerleri tutulmaktadır.
- Aksiyonlar (Actions):** Aksiyonlar, her potansiyel ajanın belirli bir kutuyu taşımak için sunduğu teklif değerleri olarak tanımlanır. Bu değerler, ajanların kutuyu taşıma sorumluluğunu üstlenme tekliflerini temsil eder ve mevcut pozisyonları, önceki görevleri ve diğer ilgili kriterleri dikkate alır.

### 5.1.1.6.2 Q-Değerlerinin başlatılması ve güncellenmesi

- **Başlatma:** Başlangıçta, Q-tablosu, her kutuya mesafe olarak en yakın ajanın en yüksek değeri alacak şekilde ayarlanır, çünkü yakınlık, görev tahsisini belirlemede varsayılan birincil faktördür. Bu varsayılan ayar, gerçek görev yürütme sonuçlarına dayalı olarak öğrenilen değerler tarafından geçersiz kılınana kadar devam eder.
- **Öğrenme yoluyla güncellemeler:** Her ajanın bir kutu için sunduğu teklif, görev yürütme sonuçlarına dayalı olarak dinamik bir şekilde güncellenir ve yeni stratejilerin keşfi ile bilinen verimli stratejilerin kullanımı arasında bir denge sağlar. Q-değerleri, her bölümün sonunda çeşitli pekiştirme sinyallerine dayalı olarak ayarlanır.

### 5.1.1.6.3 Pekiştirme kriterleri

Görev ödülleri öğrenildikten sonra her turda en yüksek Q değerine sahip olan ajan görevi alır çünkü en fazla teklifi o yapmıştır.

- **Yakınlık bonusu:** Bir kutunun ajana daha yakın olması durumunda pozitif bir ödül uygulanır, bu da görev'e başlama süresini ve göstereceği çaba en az olan ajanın önceliklendirilmesi demektir. Ödül, ajanın kutuya olan uzaklığını ( $d$ ) ile doğru orantılıdır. Matematiksel olarak bu ödül Denklem 5.6'da ifade edilir:

$$r_p = -k \times d \quad (5.6)$$

Örneğin, ajan kutuya 2 birim uzaklıktaysa ve  $k = 1$  kullanılarak -2 birim ödül alır.

- **Son iş bonusu:** Ajanın kısa süre önce taşıdığı bir kutu üzerinde çalışması durumunda güçlü bir pozitif ödül verilir, bu da ajanların başladıkları işi sürdürmelerini teşvik eder. Ödül, ajanın kutuyu taşıdığı süre ( $t$ ) ile doğru orantılıdır ve  $k = 1.5$  olarak belirlenmiştir. Matematiksel olarak bu ödül Denklem 5.7'de ifade edilir:

$$r_l = t \times k \quad (5.7)$$

Örneğin, ajan son 1 zaman çevrimi içinde aynı kutuyu taşıdıysa 1.5 birim ödül alır.

- **Kolon kısıtlaması:** Ajanın kısa süre önce çalıştığı kolon içerisindeki bir kutu üzerinde çalışması durumunda negatif bir ödül verilir, bu da görev atamalarında mekansal çeşitliliği teşvik ederek gereksiz hareketleri önler. Ödül, ajanın kutuyu taşıdığı süre ( $t$ ) ile doğru orantılıdır. Matematiksel olarak bu ceza Denklem 5.8'de ifade edilir ve  $k = 1.1$  olarak belirlenmiştir.

$$r_k = -t \times k \quad (5.8)$$

Örneğin, aynı kolon içinde çalışırsa 1.1 birim ceza alır.

- **Kısıtlama cezası:** Ajanın hareket alanı dışında bulunan bir kutu üzerinde çalışması durumunda güçlü bir negatif ödül uygulanır. Ajan, küresel beyin tarafından sınırlandırılmış alanın dışındaki bir kutuyu taşımaya çalışırsa -12 puan ceza alır. Matematiksel olarak bu ceza Denklem 5.9'da ifade edilir:

$$r_r = -12 \quad (5.9)$$

- **Boşta kalma ödülü:** Ajan uzun süre boşta kaldıysa her bir kutu için pozitif bir ödül verilir. Ödül, ajanın boşta kaldığı süre ( $t$ ) ile doğru orantılıdır. Matematiksel olarak bu ödül Denklem 5.10'da ifade edilir ve  $k = 0.4$  olarak belirlenmiştir.

$$r_i = t \times k \quad (5.10)$$

Örneğin, ajan 2 çevrimdir boşta ise taşınması gereken her kutu için 0.8 birim ödül alır.

Bu yeni ödül ve ceza sistemine göre, bir adımda alınan toplam ödül  $r$  Denklem 5.11'de ifade edilir:

$$r = r_p + r_l + r_k + r_r + r_i \quad (5.11)$$

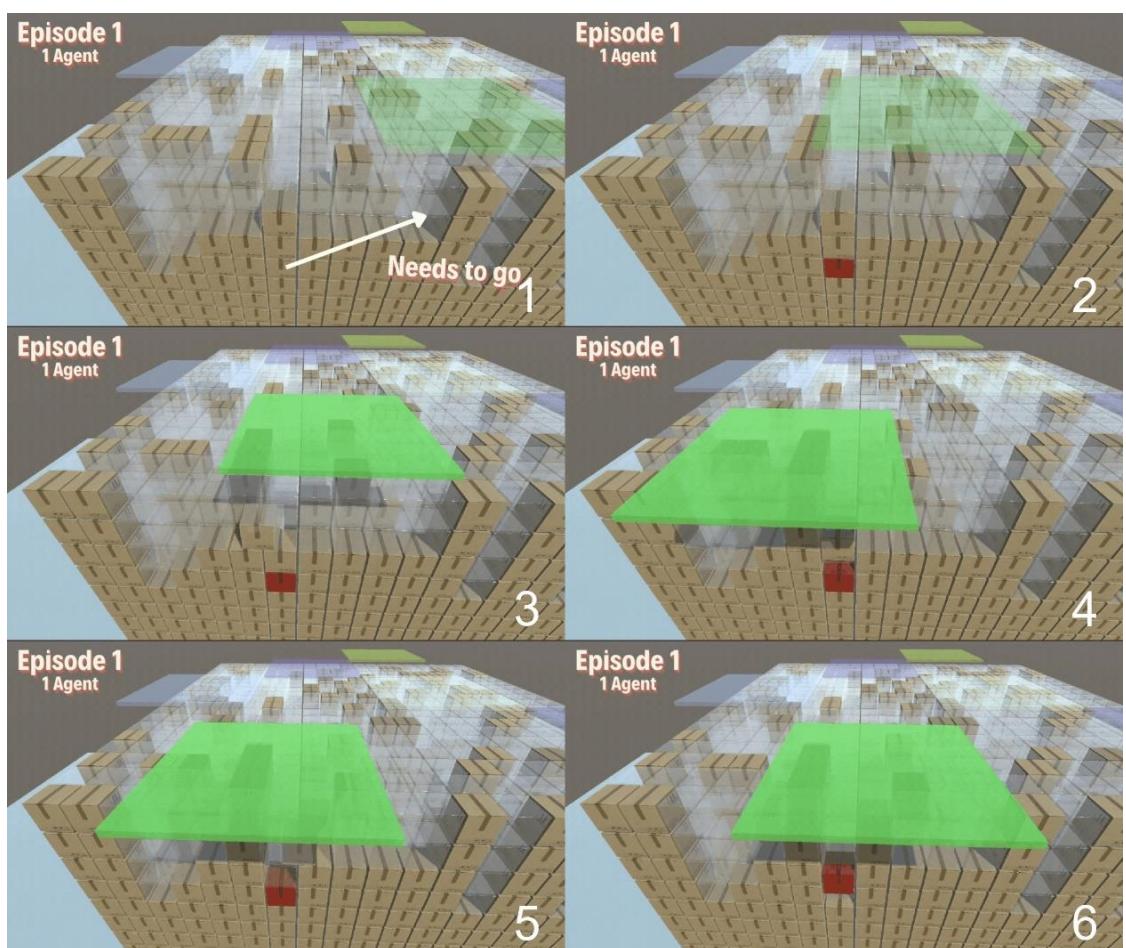
## 5.1.2 Uygulamalar

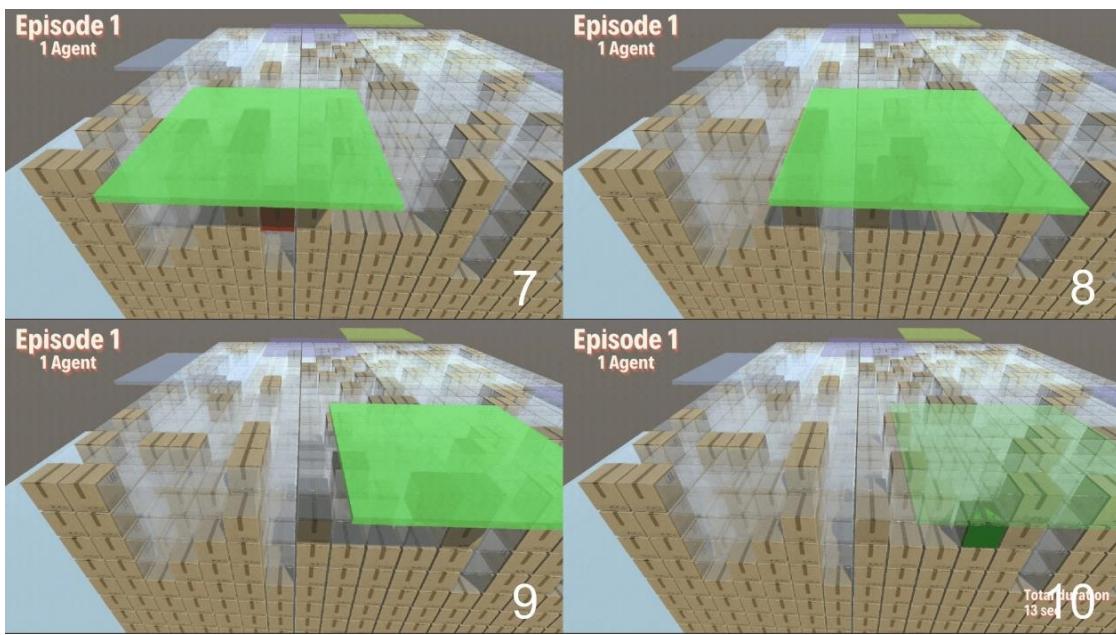
Bu bölümde, küresel beynin iş yerine hangi ajanların geleceğine karar vermesi ve ardından iş yerine gelen ajanların kendi aralarında nasıl iş dağılımı yaptığını gösteren, iki modelin birlikte kullanıldığı uygulamalar incelenecektir. Bu uygulamalar, küresel beynin ve ajanların koordineli bir şekilde çalışmasını ve görevlerin etkin bir şekilde dağıtılmasını ayrıntılı bir şekilde açıklamaktadır.

### 5.1.2.1 Uygulama 1

Bu örnekte ajanların ve küresel beynin üç bölüm boyunca nasıl çalıştığını incelenmektedir.

**Bölüm 1:** Küresel beyin, seçilen kutunun hemen üzerinde hiçbir ajan olmadığı için bir yeşil renkli ajanı çağrırmaya karar verir. Bu tek ajan, başka ajan olmadığı için taşınması gereken kutunun üzerindeki 2 kutuyu temizleyip ardından seçilen kutuyu hedefine taşımakla kendini görevlendirir. Her hareketin 1 saniye sürdüğü varsayıldığında, ajan kutuları temizlemek için 11 saniye ve kutuları almak için 2 saniye harcar.

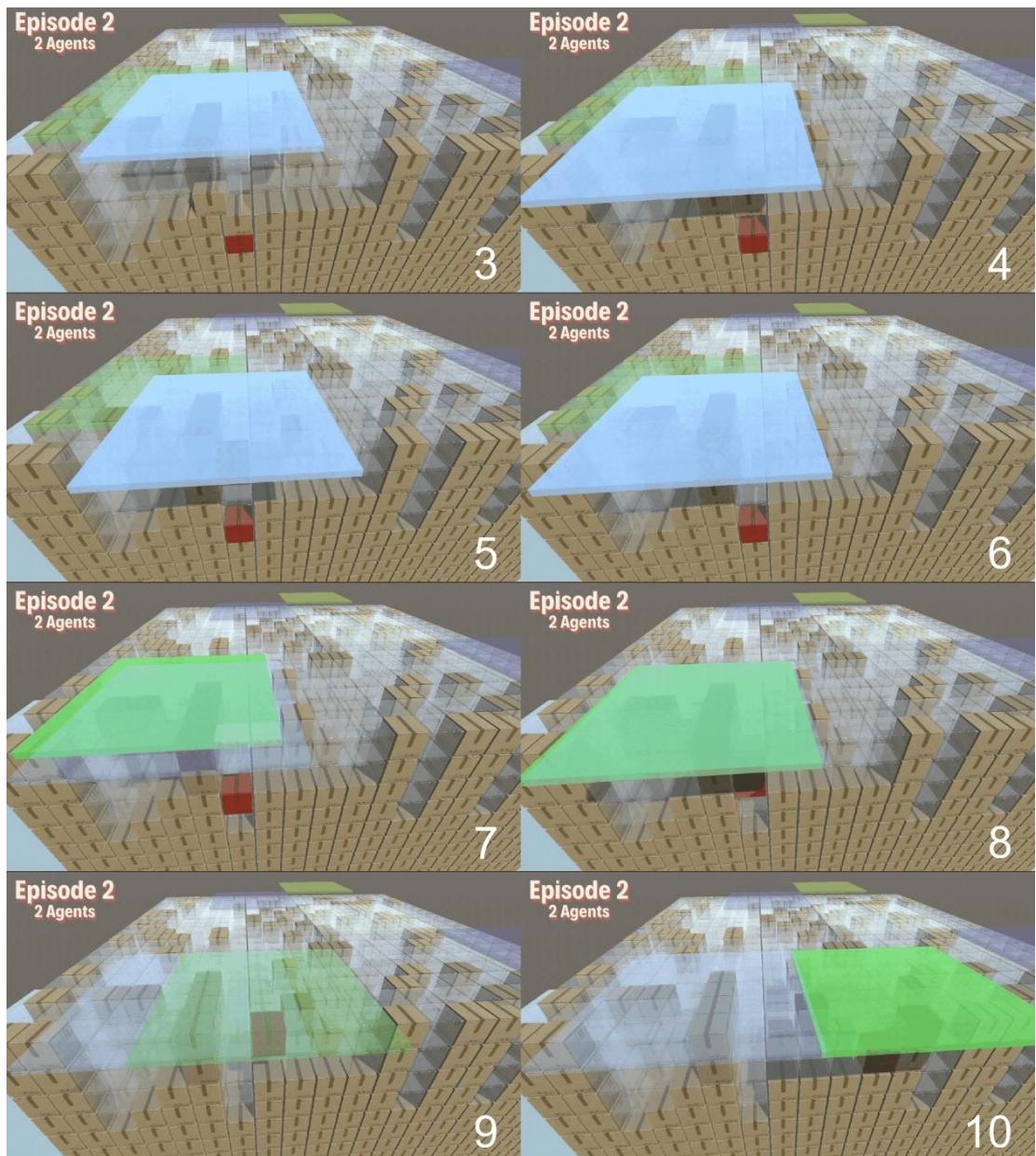




**Şekil 5.12 :** Bölüm 1: Tek ajan operasyonu

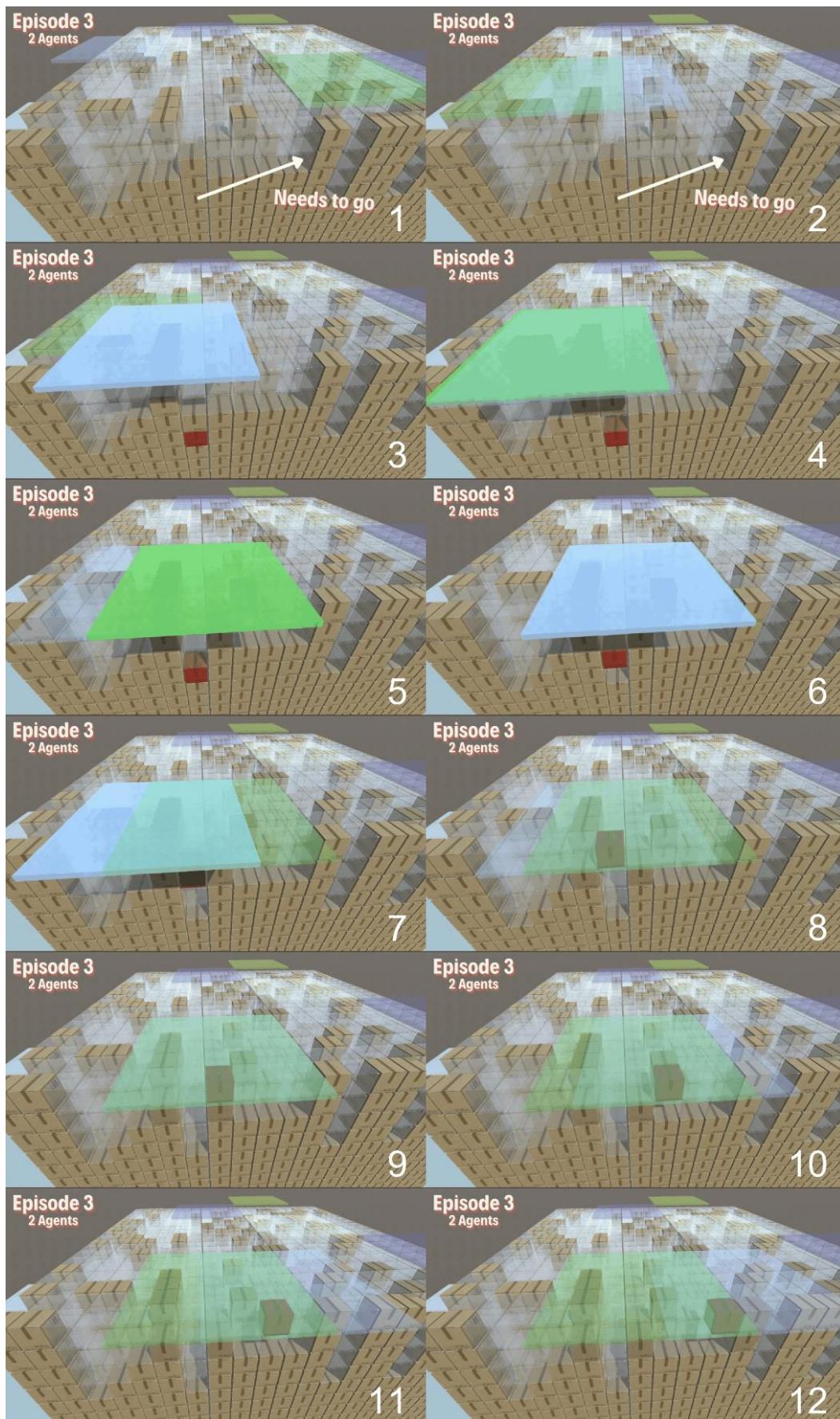
**Bölüm 2:** Küresel beyin, bu görev için 2 ajanın (mavi ve yeşil) daha verimli olabileceğine karar verir. Ajanlar çağrıldıktan sonra ise lokal kararlarda mavi ajan kutuları temizlerken, yeşil seçilen kutuyu taşır. Bu koordinasyon, ajanların kutu hareketi için 12 saniye ve kutulara yaklaşmak için 1 saniye harcamasına neden olur. Kutu temizleme stratejisindeki fark nedeniyle, kutu hareketi için 11 saniye süren ilk bölüme kıyasla 1 saniyelik bir kayıp olur. Böylece, görev süresi toplamda 12 saniye olur.





**Şekil 5.13 :** Bölüm 2: İki ajanın işbirliği

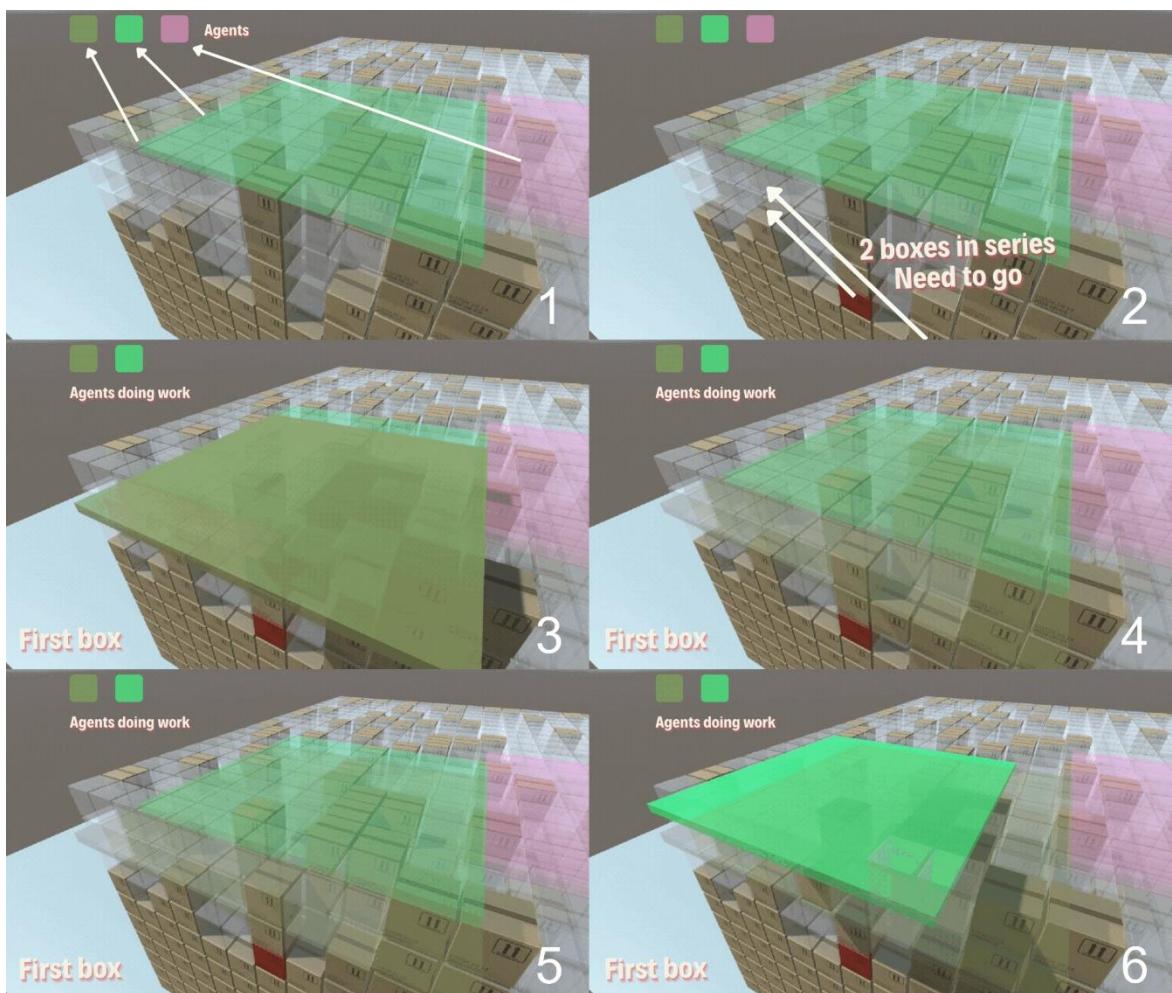
**Bölüm 3:** Ajanlar çağrıldıktan sonra kendi aralarında operasyonların 2 ajanla eşzamanlı olarak yürütülmesine karar verilir, böylece biri çalışırken diğeri beklemez çünkü aynı sütunda taşıma cezası devreye girmiştir. Kırmızı kutunun üzerindeki kutuların her biri farklı ajan tarafından taşınmış, daha sonra kırmızı kutuyu da mavi ajan taşımıştır. Bu senkronizasyon, kutuları almak için seyahat süresini ortadan kaldırarak toplam görev süresini 11 saniyeye düşürür, bu da önceki öğrenme adımdan daha verimli olduğunu kanıtlar.

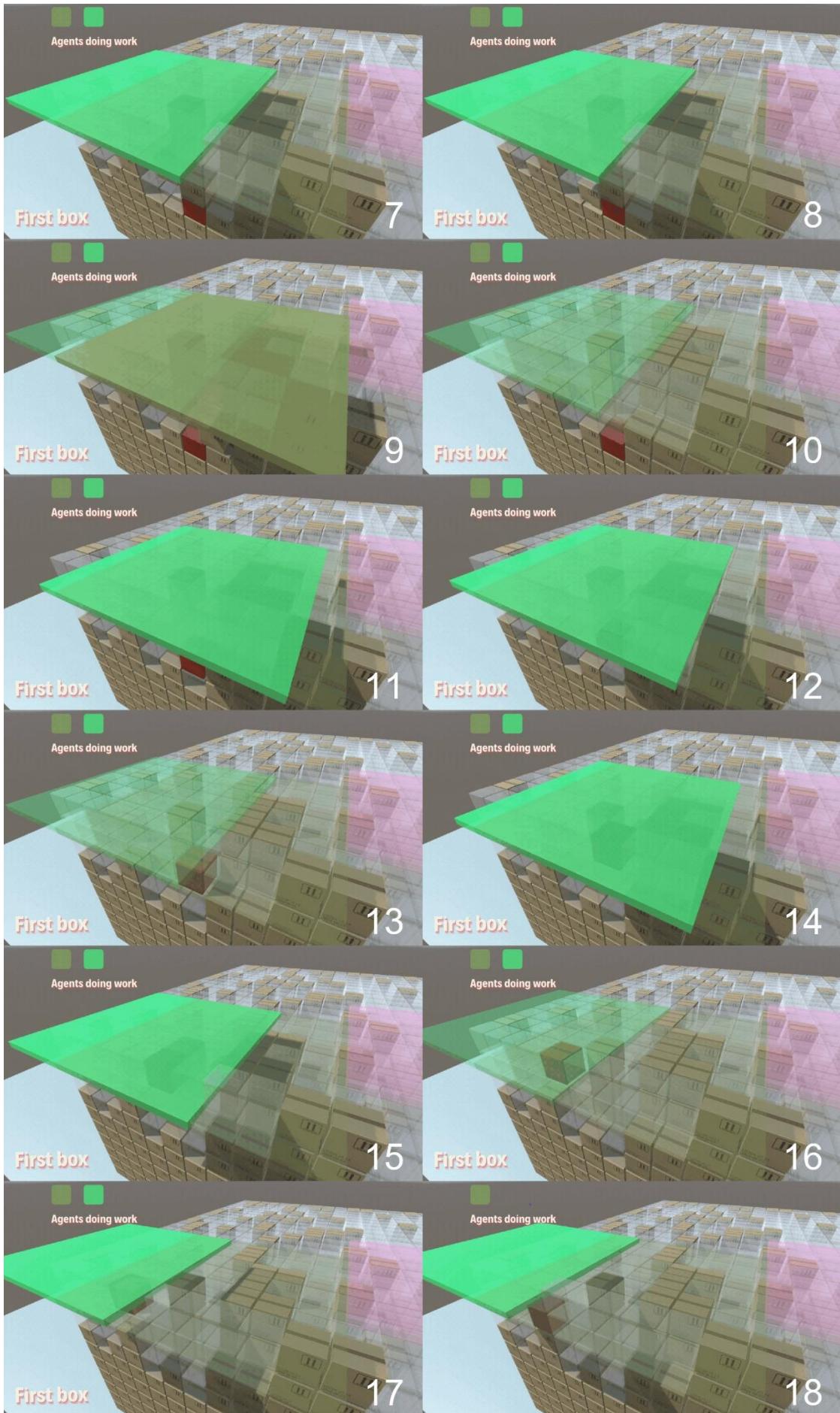


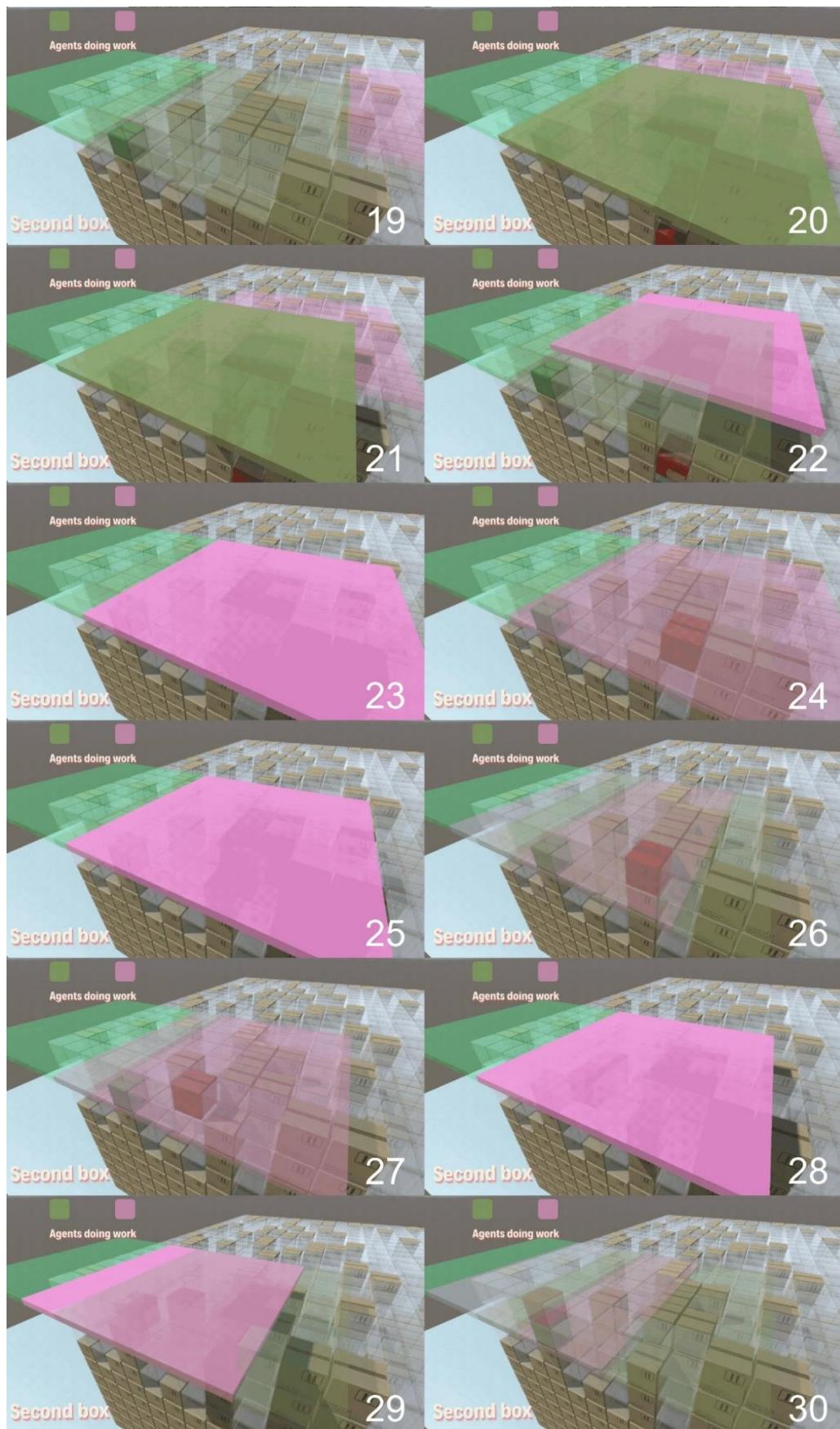
**Şekil 5.14 :** Bölüm 3: Tamamen eşzamanlı operasyon

### 5.1.2.2 Uygulama 2

Seçilen 2 kutunun sıralı olarak hedeflerine taşınması istenmiştir. Şekil 5.15'deki şekilde, eğitim süreci sonunda hareketler oturuktan sonraki koordinasyon görülmektedir. Ajanların görüş alanı, daha önceden de anlatıldığı gibi ızgara yapısının üzerinde opak ya da saydam olabilen düzlemler olarak temsil edilmekte ve opak olmaları aktif katılımlarını göstermektedir. İlk olarak, küresel beyin 1. kutu için 1. ve 2. ajanları (koyu yeşil ve açık yeşil) görevlendirir. İkisi birlikte 1. Kutunun üzerindeki kutuları sırasıyla temizlerler. Daha sonra ise 2. ajan kırmızı kutuyu hedefine taşıır. Ardından, 2. kutu görevi için 1. ve 3. ajanlar (açık yeşil ve pembe) aynı şekilde küresel beyin tarafından iş yerine yönlendirilir. 1. ajan 2. kutunun üzerindeki tek kutuyu temizledikten sonra 3. ajan kutuyu gitmesi istenen hedefe götürür. Bu şekilde istediğimiz kesintisiz ve efektif hareketler zincirini verecek şekilde eğitilen modelleri çalışırken gözlemlemiştir.







**Sekil 5.15 :** Çok ajanlı koordinasyon eğitim sonu



## **6. SONUÇ VE ÖNERİLER**

Bu çalışma, endüstriyel üretim hatlarında malzeme yenileme süreçlerini iyileştirmek amacıyla Pekiştirmeli Öğrenme (RL) tabanlı yapay zekâ ve sürü zekâsı algoritmalarının kullanılabilirliğini göstermiştir. Proje kapsamında, küresel beyin ve ajanların koordineli çalışması ile görevlerin daha verimli ve etkili bir şekilde nasıl gerçekleştirilebileceği detaylı olarak incelenmiştir. Sonuçlar, önerilen sistemin operasyonel verimliliği artırma ve kaynak kullanımını optimize etme potansiyelini açıkça ortaya koymuştur.

Gelecekteki çalışmalar, sistemin daha da iyileştirilmesi ve farklı endüstriyel senaryolara uyarlanabilirliğinin artırılması için şu önerilerde bulunabilir:

**Gerçek zamanlı veri entegrasyonu:** Sistemin performansını artırmak için gerçek zamanlı veri akışı ve geri bildirim mekanizmaları entegre edilmelidir.

**Gelişmiş algoritma optimizasyonu:** Daha karmaşık görevler için algoritmaların optimizasyonu ve yenilikçi öğrenme teknikleri kullanılabilir.

**Fiziksel robot uygulamaları:** Simülasyon ortamlarında elde edilen başarıların gerçek dünya robotik sistemlerine uygulanması ve test edilmesi, sistemin pratikteki etkinliğini gösterecektir.

**Kapsamlı senaryo testleri:** Farklı endüstriyel senaryolarda sistemin performansının değerlendirilmesi ve optimize edilmesi, genel uyumluluğu ve esnekliği artıracaktır.

### **6.1 Çalışmanın Uygulama Alanı**

Bu çalışmanın sonuçları, pek çok endüstriyel uygulama alanında önemli faydalara sağlayabilir. Özellikle, otomotiv, elektronik ve tüketim malları üretimi gibi malzeme yönetiminin kritik olduğu sektörlerde, önerilen sistemin kullanılması, verimlilik ve maliyet tasarrufu açısından büyük avantajlar sunabilir. Ayrıca, depo yönetimi ve lojistik gibi alanlarda da sistemin uygulanabilirliği yüksektir.

Önerilen yapay zekâ ve sürü zekâsı tabanlı sistem, sadece üretim hatlarında değil, aynı zamanda geniş ölçekli lojistik ağlarında ve karmaşık malzeme taşıma süreçlerinde de kullanılabilir. Bu sayede, malzeme akışının daha etkin bir şekilde yönetilmesi ve operasyonel süreçlerin iyileştirilmesi mümkün olacaktır.

Sonuç olarak, bu çalışma, yapay zekâ ve robotik teknolojilerin endüstriyel uygulamalardaki potansiyelini vurgulamakta ve gelecekteki araştırmalar için sağlam bir temel oluşturmaktadır. Bu bağlamda, önerilen sistemin daha geniş bir uygulama yelpazesinde değerlendirilmesi ve sürekli olarak geliştirilmesi büyük önem taşımaktadır.

## KAYNAKLAR

- Shi, D., Fan, W., Xiao, Y., Lin, T., & Xing, C.** (2020). Intelligent scheduling of discrete automated production line via deep reinforcement learning. *International Journal of Production Research*, 58, 3362-3380.
- Kozlica, R., Schafer, G., Hirlaender, S., & Wegenkittl, S.** (2023). A Modular Test Bed for Reinforcement Learning Incorporation into Industrial Applications. ArXiv.
- Viharos, Z., & Jakab, R.** (2021). Reinforcement Learning for Statistical Process Control in Manufacturing. *Measurement*, 182, 109616.
- Nian, R., Liu, J., & Huang, B.** (2020). A review On reinforcement learning: Introduction and applications in industrial process control. *Comput. Chem. Eng.*, 139, 106886.
- Panzer, M., & Bender, B.** (2021). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research*, 60, 4316-4341.
- Kuhnle, A., May, M., Schäfer, L., & Lanza, G.** (2021). Explainable reinforcement learning in production control of job shop manufacturing system. *International Journal of Production Research*, 60, 5812-5834.
- Chu, S., Huang, H.-C., Roddick, J., & Pan, J.-S.** (2011). Overview of Algorithms for Swarm Intelligence. In Proceedings of the International Conference on Swarm Intelligence (pp. 28-41). Springer.
- Li, W., & Shen, W.** (2011). Swarm behavior control of mobile multi-robots with wireless sensor networks. *Journal of Network and Computer Applications*, 34(4), 1398-1407.
- Schranz, M., Umlauft, M., Sende, M., & Elmenreich, W.** (2020). Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI*, 7, 36.
- Wanka, R.** (2019). Swarm intelligence. *it - Information Technology*, 61(3), 157-158.
- Yu, B., & Chen, G.** (2016). Swarm intelligence in mechanical engineering. *Advances in Mechanical Engineering*, 8.
- Li, D., Wei, P., Zhao, C., Yang, S., & Li, Y.** (2023). A Mobile Manipulation System for Automated Replenishment in the Field of Unmanned Retail. 2023 IEEE International Conference on Mechatronics and Automation (ICMA), 644-649.

- Alnema, Y. H., Musaa, A. K., Ali, M. N., & Muhammad, A. O.** (2023). Design and Simulation of an Automated Production Plant and Warehouse Management System. 2023 International Conference on Engineering, Science and Advanced Technology (ICESAT), 101-107.
- Hinrichs, M., Grabmaier, V., Stahl, I., & Schneegass, S.** (2023). Designing a Recommendation System for Spare Parts Replenishment. Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia.
- Pilipchak, P., Aksu, M., Proctor, F., & Michaloski, J.** (2019). Physics-Based Simulation of Agile Robotic Systems. Volume 2B: Advanced Manufacturing.
- Aksu, M., Michaloski, J., & Proctor, F.** (2018). Virtual Experimental Investigation for Industrial Robotics in Gazebo Environment. Volume 2: Advanced Manufacturing.
- Peterson, E., Bogosian, B., & Vassigh, S.** (2022). Evaluating an Immersive Learning Environment for Robotics Training. Training, Education, and Learning Sciences.
- Watkins, C.J.C.H., & Dayan, P.** (1992). Q-Learning. Machine Learning, 8(3-4), 279-292.
- Sutton, R.S., & Barto, A.G.** (2018). Reinforcement Learning: An Introduction. MIT Press. This book provides a comprehensive introduction to the field of reinforcement learning, an area of machine learning concerned with how software agents ought to take actions in an environment to maximize some notion of cumulative reward.
- Becer, E., Ulusoy, O., & Turan, K.** (2023). *Tasarı: Alışveriş merkezi, havaalanı veya depo gibi geniş kapalı alanlarda çalışacak otonom bir temizlik robottu tasarımı* (Master's thesis). İstanbul Teknik Üniversitesi, Makina Fakültesi. Danışman: Prof. Dr. Erdinç Altuğ

## ÖZGEÇMİŞ



**Ad-Soyad** : Onur Ulusoy

**Doğum Tarihi ve Yeri** : 26/03/1999

**E-posta** : ulusoyo18@itu.edu.tr

## ÖĞRENİM DURUMU:

- Lisans** : 2024, İstanbul Teknik Üniversitesi, Elektrik-Elektronik Mühendisliği, Kontrol ve Otomasyon Mühendisliği



IEEE üyeleri olarak bizler bütün dünya üzerinde teknolojilerimizin hayat standartlarını etkilemesindeki önemin farkındayız. Mesleğimize karşı şahsi sorumluluğumuzu kabul ederek, hizmet ettiğimiz toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmayı söz verdiğimiz ve aşağıdaki etik kuralları kabul ettiğimizi ifade ederiz.

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;

- 3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;**
- 4. Her türlü rüşveti reddetmek;**
- 5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;**
- 6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;**
- 7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;**
- 8. Bütün kişilere adilane davranışmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırmalık yapma durumuna girişmemek;**
- 9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;**
- 10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.**

**IEEE Yönetim Kurulu tarafından Ağustos 1990'da  
onaylanmıştır.**

## **ETİK KURALLAR UYUM BEYANI**

**Aşağıda belirtilen Mühendisliğin temel ilkelerini biliyor ve kabul ediyorum.**

<b>İsim:</b> Onur	<b>SOYİSİM:</b> <b>ULUSOY</b>	<b>İMZA/TARIH:</b> 29.05.2024
-------------------	----------------------------------	-------------------------------

Mühendisler; mühendislik mesleğinin doğruluğunu, onurunu ve değerini insanlığın refahının artması için kendi bilgi ve becerilerini kullanarak, dürüst ve tarafsız olarak halka, kendi işverenlerine ve müşterilerine sadakatle hizmet ederek, mühendislik mesleğinin yeteneğini ve prestijini artırmaya çabalayarak, kendi disiplinlerinin mesleki ve teknik birliğini destekleyerek yükseltir ve geliştirirler.

- Mühendisler, mesleki görevlerini yerine getirirken toplumun güvenliğini, sağlığını ve rahatlığını en önde tutacaktır.
- Mühendisler, sadece yetkili oldukları alanlarda hizmet vereceklerdir.

- Mühendisler, sadece objektif ve gerçekçi raporlar düzenleyeceklerdir.
- Mühendisler, mesleki konularda işveren veya müsteri için güvenilir vekil veya yardımcı olarak davranışacaklar ve çıkar tartışmalarından kaçınacaklardır.
- Mühendisler mesleki itibarlarını hizmetlerinin gereğine göre tesis edecekler ve diğer meslektaşlarıyla haksız rekabete girmeyeceklerdir.
- Mühendisler, meslek doğruluğunu, onurunu ve değerini yükseltemek ve geliştirmek için çalışacaklardır.
- Mühendisler, mesleki gelişmelerini kendi kariyerleriyle devam ettirecekler ve kendi kontrolleri altındaki mühendislerin mesleki gelişmeleri için olanaklar sağlayacaklardır.

**Bu raporda herhangi bir kaynaktan alıntı yapılmış kısımlar %15'den az, ve paragraf halinde birebir alıntı sayısının ise sıfır olduğunu beyan ediyorum.**

İsim: Onur	SOYİSİM: <b>ULUSOY</b>	İMZA/TARİH: 29.05.2024
		

**STANDARTLAR VE KISITLAR :** Raporda aşağıda istenilenler doldurularak eklenecektir.

**Bitirme Projesinin hazırlanmasında Standartlar ve Kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.**

- 1. Projenizin tasarım boyutu nedir? Açıklayınız.**  
Endüstriyel üretim hatlarında malzeme yenileme süreçlerini iyileştirmeye yönelik yeni ve yenilikçi bir tasarımdır. Var olan sistemlerin üzerine inşa edilmiş olup, özgün bir yaklaşım ve çözümler sunar.
- 2. Projenizde çözüm ürettiğiniz mühendislik problemini ve çözümünüzü kısaca açıklayınız?**  
Proje, malzeme yenileme süreçlerindeki verimsizlikleri çözmeyi hedefler. Pekiştirmeli öğrenme (RL) ile güçlendirilmiş otomatik bir sistem kullanarak malzeme yenileme operasyonlarını optimize eder ve JIT üretim ilkelerine uyumu artırır.
- 3. Lisans eğitiminiz süresince almış olduğunuz derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?**

Bu proje lisans derslerinin ötesinde olduğundan robotik ve yapay zeka konularındaki çalışmalarım bu projede temel oldu. Özellikle pekiştirmeli

öğrenme, algoritma geliştirme, ve sistem tasarımında edindiğim tecrübeler projenin geliştirilmesinde kritik rol oynadı.

**4. Projenizi gerçekleştirirken kullandığınız modern araçlar/yazılımlar/programlar vb. nelerdir? Hangi amaçlarla kullandığınızı kısaca açıklayınız.**

Simülasyon ortamı olarak Unity oyun motoru, grafikler ve simülasyon için Matplotlib kütüphanesi, robotun gerçeklenmesi için ROS (Robot İşletim Sistemi), Gazebo simülasyon ortamı, C#, Python ve C++ programlama dilleri ve görselleştirme kütüphaneleri kullanıldı. Bu araçlar robotun tasarımını, simülasyonu ve algoritmaların geliştirilmesi için temel teşkil etti.

**5. Ders dışında çeşitli disiplinleri içeren sertifikanız var mı? (Örneğin CUDA, Udemy, Coursera gibi online platformlarda bilgi sahibi olmak) Yok, sertifikaya gerek olduğunu düşünmüyorum.**

**6. Kullandığınız veya dikkate aldığınız mühendislik standartları/normları nelerdir?**

Proje, robotik ve yapay zeka uygulamalarında yaygın olarak kabul gören IEEE ve ISO standartlarına uygun olarak geliştirildi. Özellikle, robot güvenliği ve veri güvenliği ile ilgili standartlar dikkate alındı.

**7. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir?**

- a) Ekonomi: Maliyet etkinliği ve bütçe kısıtlamaları göz önünde bulunduruldu.
- b) Çevre Sorunları: Çevreye minimal etki ve enerji verimliliği hedeflendi.
- c) Sürdürülebilirlik: Uzun vadeli kullanım ve dayanıklılık dikkate alındı.
- d) Üretilebilirlik: Gerçek dünya uygulamaları ve üretim süreçleri gözetildi.
- e) Etik: Veri gizliliği ve kullanıcı güvenliği ön planda tutuldu.
- f) Sağlık: Operatör ve kullanıcı sağlığına zarar vermeyecek tasarım yapıldı.
- g) Güvenlik: Sistem ve operasyonel güvenlik en üst düzeyde sağlandı.
- h) Sosyal ve Toplumsal Sorunlar: Toplumun ihtiyaçları ve bekâltileri göz önünde bulundurularak sosyal uyumlu bir çözüm geliştirildi.

**Proje Ekibi (Yürüttüsü/Lideri):** Onur Ulusoy

**Proje Konusu:** Endüstriyel üretim hatlarında pekiştirmeli öğrenme ile güçlendirilmiş otomatik malzeme yenileme sisteminin tasarımını, geliştirmesini ve simülasyonu

**Proje Danışmanı:**

Doç. Dr. Ahmet Onat

Bu proje ..... *Ahmet Onat* tarafından onaylanmıştır.

