

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

RENKLERİ KULLANARAK YÜKSEK KONTROLLÜ
YAPAY ZEKA OLUŞTURMA

YÜKSEK LİSANS TEZİ

Emre Can ARSLAN

Bilgisayar Mühendisliği Anabilim Dalı

EYLÜL 2023

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

RENKLERİ KULLANARAK YÜKSEK KONTROLLÜ
YAPAY ZEKA OLUŞTURMA

YÜKSEK LİSANS TEZİ

Emre Can ARSLAN

Bilgisayar Mühendisliği Anabilim Dalı

Prof. Dr. Cemil ÖZ

EYLÜL 2023

Emre Can ARSLAN tarafından hazırlanan “Renkleri Kullanarak Yüksek Kontrollü Yapay Zekalar Oluřturma” adlı tez alıřması 27.09.2023 tarihinde ařađıdaki jüri tarafından oy birliđi ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliđi Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiřtir.

Tez Jürisi

Jüri Bařkanı : **Prof.Dr. Cemil ÖZ**
Sakarya Üniversitesi

Jüri Üyesi : **Dr.Öđr.Üyesi Kayhan AYAR**
Sakarya Üniversitesi

Jüri Üyesi : **Dr.Öđr.Üyesi Selman HIZAL**
Sakarya Uygulamalı Bilimler Üniversitesi



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “**Renkleri Kullanarak Yüksek Kontrollü Yapay Zekalar Oluşturma**” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(27/09/2023).





Canım Ailem ve Bütün Öğretmenlerime...



TEŐEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlamasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikle beni yönlendiren değerli danışman hocam Prof. Dr. Cemil ÖZ'e teşekkürlerimi sunarım.

Emre Can ARSLAN



İÇİNDEKİLER

Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	v
TEŞEKKÜR	ix
İÇİNDEKİLER	xi
KISALTMALAR	xiii
TABLO LİSTESİ	xv
ŞEKİL LİSTESİ	xvii
ÖZET	xix
SUMMARY	xxiii
1. GİRİŞ	1
1.1. Tezin Kapsamı	1
1.2. Tezin Amacı	2
2. KAYNAK ARAŞTIRMASI	3
2.1. Yapay Zekâ	3
2.1.1. A* yol bulma algoritması	5
2.2. Renkler	7
2.3. Duygu	11
3. KULLANILAN TEKNOLOJİLER VE YÖNTEM	17
3.1. Kullanılan Yazılımlar	17
3.2. Yöntem	20
3.2.2. Durum makinesi	21
3.2.3. Navigasyon ağı	23
3.2.4. Arayüz	24
3.2.5. Animasyon	25
4. UYGULAMA	31
4.1. Yapay Zekaların Oluşturulması	32
4.2. Duyguların Eklenmesi	32
4.3. Nesne Yönelimli Sistem	34
4.4. Duygu ve Renk Değişimi	38
4.5. Duyguların Modellenmesi	39
4.5.1. Temel duygular	39
4.5.2. İkincil duygular	41
4.6. Duyguların Birleşimi	42
5. SONUÇ	45
KAYNAKLAR	47
ÖZGEÇMİŞ	49



KISALTMALAR

- 2B** : İki boyutlu
3B : Üç boyutlu
NYP : Nesne yönelim programlama





TABLO LİSTESİ

Sayfa

Tablo 4.1. Duygu Birleşimi Tablosu.....	43
--	----





ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Sürü akını algoritması uygulanmış balık sürüsü.....	7
Şekil 2.2. Sıcak Renkler.....	9
Şekil 2.3. Soğuk renkler.....	9
Şekil 2.4. Plutchik'in Duygu Çarkı	12
Şekil 3.1. Durum Makinesi.	22
Şekil 3.2. Navigasyon ağ penceresi	24
Şekil 3.3. Örnek arayüz şeması.....	25
Şekil 3.4. Animasyon denetleyicisi.....	26
Şekil 3.5. Animasyon penceresi.....	26
Şekil 3.6. Animasyon durum penceresi.	27
Şekil 3.7. Animasyon geçiş penceresi.	28
Şekil 3.8. Animatör bileşeni.....	29
Şekil 3.9. Ses kaynağı bileşeni.....	30
Şekil 4.1. Simülasyon ortamı.....	31
Şekil 4.2. Navigasyon ağ ajanının komut dosyası.	32
Şekil 4.3. Renkleri kullanarak yapay zekaların duygu şiddetlerinin ayarlanması.	33
Şekil 4.4. Renkleri kullanarak yapay zekaların duygu şiddetlerinin ayarlanması.	33
Şekil 4.5. Soyutlamayla oluşturulmuş sınıflar.	34
Şekil 4.6. Sınıf ve yapıcı yöntemi.....	35
Şekil 4.7. Arayüz eklenmiş sınıf.....	36
Şekil 4.8. Av ve avcı simülasyonu.....	37
Şekil 4.9. Renk değişimi akış şeması.....	38
Şekil 4.10. Sevinç duygulu yapay zekâ.	40
Şekil 4.11. Üzüntü duygulu yapay zekâ.	41



RENKLERİ KULLANARAK YÜKSEK KONTROLLÜ YAPAY ZEAKALAR OLUŞTURMA

ÖZET

Her ne kadar günümüzde yapay zekâ alanında her geçen gün yeni gelişmelere şahit olsak bile sık sık karmaşık ve gerçekçi ortamlar içeren simülasyonlarla, karmaşık ortamların çeşitli yönlerini anlayan daha akıllı ve daha kapsamlı araçlara olan ihtiyaç giderek artıyor. Simülasyon yapay zekâsı genellikle her simülasyon için özel olarak tasarlandığından, simülasyonlardaki yapay zekâ araçları şu anda geliştiricilerin belirli bir yapay zekâyı hızlı ve verimli bir şekilde oluşturmasına olanak sağlamaya odaklanıyor. Bu yaklaşımla ilgili bir sorun, yapay zekaların sayısı arttıkça bu yapay zekaların birbirleri arasındaki benzerliklerin verimli bir şekilde kullanılamamasına sebep olmaktadır. Renklerin insan üzerindeki duygusal etkisinden ilham alan bu çalışma, temel tanımlamak için kavramsal görüşlere ve eylemlere dayanan kavramsal yapay zekanın renklerin duygusal etkisiyle geliştirilmesini sağlamak için birleşik bir kavramsal çerçevenin kullanımına dayalı üç farklı yaklaşım önermektedir. Yaklaşımlar, simülasyonlar ile örneklendirilmiştir.

Bu tez çalışmasında Unity oyun motorunda renkleri yapay zekalar üzerine renkleri entegre ederek yüksek kontrollü yapay zekalar tasarlanmıştır. Unity oyun motorunun tercih edilme sebebi duyguları simüle etmeyi kolaylaştıran animasyon, seslerin proje içerisinde rahatça kullanılmasını sağlamasıdır. Yapay zekalar tasarlanırken navigasyon ağ ajan bileşeni, çarpıştırıcı bileşeni, animatör bileşeni, ses kaynağı bileşeni, rijit cisim bileşen ve komut dosyası bileşeni kullanıldı. Navigasyon ağ ajan bileşeni, yapay zekaların durum makinesi kullanarak hareket etmesini, ajanların yüklenen renklere göre kaçınmasını veya kovalamasını hızının ve açılma hızının kontrol edilmesini sağlamaktadır. Rijit cisim bileşeni fizik sistemini kullanarak simülasyonun sanal bir ortamda ajanlara kütle eklemesini sağlayarak yer çekimi, sürtünme gibi temel fizik kuralları çerçevesinde yürütülmesini sağlar. Fizik sistemi sayesinde rijit cisim, ajanların eklemleri, hızları ve çarpışmalarını rahatlıkla simüle edebilir. Çarpıştırıcı bileşeni de fizik sistemini kullanarak çalışır. Bu bileşen sayesinde yapay zekalar çarpışmaları test edip birbirlerine kuvvet uygulayabilir veya çarpıştırıcı tetikleyici olarak kullanılarak tetiklenmesi durumunda geliştiricinin tanımladığı ilgili görevlerin yerine getirilmesini sağlar. Animatör bileşeni animasyon denetleyicisini kullanarak tanımlanan avatar üzerinde denetleyicinin içerisinde bulunan animasyonları çalıştırmayı sağlar. Ses kaynağı bileşeni kendisine tanımlanan ses dosyasını bileşen içerisindeki ayarlanabilen koşullar çerçevesinde oynatır oynatılan ses kaynağını ses dinleyici bileşeni olan simülasyon objelerini dinleyip işlem yapılmasını sağlar. Komut dosyası bileşeni yapay zekaların hareketini, yapay zekalara renklerin eklenmesi veya silinmesini, renklere göre diğer bileşenlerle iletişimi sağlayıp yapay zekaların davranışlarının modellenmesini, iki rengin birleşiminin bir davranış karşılığı bulunması durumunda bu renklerin birleşimini sağlayıp yeni davranışların oluşturulmasını sağlar.

Bu tez kapsamında, Unity oyun motorunda kullanılan bileşenlerin özellikleri ve simülasyon objeleri ile yapay zekaların renklerle bağdaştırıldığı simülasyon ortamının tasarımı yapıldı. Simülasyonun amacı doğrultusunda yapay zeka renklerle bağdaştırılıp bağdaştırılan yapay zekalar simülasyon ortamına eklendi. Daha sonrasında ortamdaki yapay zekaların birbiriyle olan etkileşime geçmesi ve bu doğrultuda sahip oldukları renge göre kaçma, kovalama, sevinme, üzülme gibi farklı tepkiler vermesi sağlandı. Dinamik olarak birbirleriyle etkileşim halinde farklı gösteren yapay zekaların duygu durumunun buldukları renkle birlikte değişmesi sağlanıp değişen renk ve duygu koşullarına uyum göstermesi için programlandı. Hareketleri için takip ettikleri ağırlık birbiriyle aynı olması ancak farklı fiziksel veya duygusal özelliklere sahip yapay zekalara farklı yolları kullanma seçeneği sunuldu.

Renkle bağdaştırılmış yapay zekâ simülasyon modelinde, bütün yapay zekalar bir ana sınıftan kalıtım alacak şekilde oluşturuldu. Bu ana sınıf soyut sınıf olarak tasarlanıp simülasyon içerisindeki bütün yapay zekaların sağlaması gereken özellikleri ve davranışları tanımlandı. Ana sınıftan kalıtım alınarak üretilen sekiz sınıfın sekiz farklı rengi temsil edecek şekilde ayarlanması sağlandı. Bu sekiz sınıfın kendi içlerinde kendisinden üretilen yapay zekaların sahip olması gereken özellikler ve davranışların da tanımlandığı şekilde olması ayarlandı. Duyguların ön planda olduğu simülasyon modeli için arayüzleri kullanarak bu sınıfların sağladığı duyguları arayüzün anlaşma koşullarını gerçekleştirerek kendi üzerine eklemesi sağlandı. Bu şekilde yapay zekalar ilk simülasyonda belirli renklere sahip olup bu renkler geliştiricinin kendi belirlediği koşullar içerisinde kavramsal olarak kullanıma hazır getirildi. Diğer bir simülasyonda renklerle birlikte kavramsal olarak duygular da yapay zekalar üzerine tanımlanıp saf halde çalışabilir temel bir sistem oluşturuldu. Duyguların birleşip yeni duygular oluşturduğu bir simülasyon daha tasarlanıp bu simülasyon gelişime açık kavramsal bir simülasyon modeli olarak geliştiricilerin kendi istediği şekilde duyguların birleşimini modellemesi ve yapay zekalar üzerindeki kontrolün artırılması amaçlandı.

Duyguların bulunduğu yapay zekâ simülasyon modelinde sekiz farklı renk ve sekiz farklı rengin temsil ettiği duyguların davranışları tanımlandı. Bunlardan dört tanesi temel duygular olup diğer dört tanesi de ikincil duygular olarak tanımlanmıştır. Temel duygulardan ilki temsilcisi olan sarı renk sevinç duygusunun rengi olarak düşünüldü ve en düşükten en yükseğe sükûnet, sevinç, coşkunluk olmak üzere üç farklı şiddette incelendi. Temel duygulardan ikincisini temsil eden yeşil renk ise korku duygusunun temsilci rengi olarak düşünüldü ve kaygı, korku, dehşet olmak üzere üç şiddette incelendi. Üçüncü temel duygu olan üzüntünün temsilcisi mavi renk düşünüldü dalgınlık, üzüntü, keder olmak üzere üç farklı şiddette incelendi. Dördüncü temel duygu olan öfke ise kırmızı renk kullanılarak temsil edildi ve rahatsızlık, öfke ve hiddet olmak üzere üç farklı şiddette incelendi. Bu dört temel duygunun yanı sıra yine dört tane olmak üzere ikincil duygular da tanımlandı. Bu duygular temel duygular kadar şiddetli olmasa da yapay zekalar üzerinden etkili olabilecek duygular olarak kabul edildi. İkincil duygulardan ilki olan güvenin temsilci rengi fıstık yeşili olarak seçildi ve bu duygu kabulleniş, güven, hayranlık olmak üzere üç farklı şiddette incelendi. İkincil duygulardan ikincisi olan şaşkınlığın temsilci rengi okyanus mavisi olarak seçil ve bu duygu dikkat dağılması, şaşkınlık, hayret olmak üzere üç farklı şiddette incelendi. İkincil duygulardan üçüncüsü olan tiksindenin temsilci rengi olarak mor renk seçildi ve bu duygu sıkıntı, tikslenme, iğrenme olmak üzere üç farklı şiddette incelendi. İkincil duygulardan dördüncü ve sonuncusu olan beklentinin temsilci rengi olarak turuncu renk seçildi ve bu duygu ilgi, beklenti, ihtiyat olmak üzere üç farklı şiddette incelendi. Duygular ve renkler senkronize bir biçimde yapay

zekalara işlendikten sonra renkler sayesinde rahatça gözlemlenebilen ve kontrol edilebilen simülasyonlar, üzerinden test yapılabilir hale getirildi.





CREATING HIGHLY CONTROLLED ARTIFICIAL INTELLIGENCES USING COLORS

SUMMARY

Although we are witnessing new developments in the field of artificial intelligence every day, the need for smarter and more comprehensive agents that understand various aspects of complex environments is increasing, with simulations often involving complex and realistic environments. Because simulation AI is often custom-tailored for each simulation, AI tools in simulations currently focus on enabling developers to quickly and efficiently create a specific AI. A problem with this approach is that as the number of artificial intelligences increases, the similarities between these artificial intelligences cannot be used efficiently. Inspired by the emotional impact of colors on humans, this study proposes three different approaches based on the use of a unified conceptual framework to enable the development of conceptual artificial intelligence based on conceptual views and actions to define the core with the emotional impact of colors. Approaches are exemplified by simulations.

In this thesis, highly controlled artificial intelligences are designed by integrating colors on artificial intelligences in Unity game engine. The reason why Unity game engine is preferred is that animation, which makes it easy to simulate emotions, provides comfortable use of sounds in the project. While designing artificial intelligences, navigation network agent component, collider component, animator component, sound source component, rigid body component and script component were used. The navigation network agent component enables the AIs to move using a state machine, to control the speed and angular speed of agents to evade or chase according to the loaded colors. By using the rigid body component physics system, it enables the simulation to add mass to the agents in a virtual environment, allowing it to be carried out within the framework of basic physics rules such as gravity and friction. Thanks to the physics system, the rigid body can easily simulate the joints, velocities and collisions of the agents. The collider component also works using the physics system. Thanks to this component, artificial intelligences can test collisions and apply force to each other, or by using the collider as a trigger, to perform the corresponding tasks defined by the developer if triggered. The animator component uses the animation controller to run the animations inside the controller on the defined avatar. The audio source component plays the sound file defined to it within the framework of the adjustable conditions within the component, listens to the simulation objects that are the sound listener component of the played sound source and provides processing. The script component enables the movement of artificial intelligences, adding or deleting colors to artificial intelligences, communicating with other components according to colors and modeling the behavior of artificial intelligences.

In this thesis, the features of the components used in the Unity game engine and the simulation environment in which the simulation objects and artificial intelligence are associated with colors were designed. In line with the purpose of the simulation, artificial intelligence, which is associated with artificial intelligence colors, has been

added to the simulation environment. Afterwards, it was ensured that the artificial intelligences in the environment interacted with each other and gave different reactions such as escaping, chasing, rejoicing, and upset depending on the color they had. It has been programmed to adapt the emotional state of artificial intelligences, which dynamically differ in interaction with each other, to change with the color they are in, and adapt to the changing color and emotional conditions. The network they follow for their movements is the same as each other, but artificial intelligences with different physical or emotional characteristics were given the option to use different paths.

In the color-associated artificial intelligence simulation model, all artificial intelligences were created to inherit from a master class. This main class was designed as an abstract class, and the features and behaviors that all artificial intelligences in the simulation should provide were defined. Eight classes inherited from the main class have been adjusted to represent eight different colors. These eight classes have been adjusted to have the characteristics and behaviors of the artificial intelligence produced by themselves as defined. By using the interfaces for the simulation model where emotions are at the forefront, it was ensured that the emotions provided by these classes were added to itself by realizing the agreement conditions of the interface. In this way, artificial intelligences have certain colors in the first simulation, and these colors are conceptually ready for use within the conditions determined by the developer. In another simulation, emotions were conceptually defined on artificial intelligence along with colors and a basic system that could work in pure form was created. Another simulation was designed, in which emotions combine and create new emotions, and this simulation, as a conceptual simulation model open to development, was intended for developers to model the combination of emotions as they wish, and to increase the control over artificial intelligence.

In the artificial intelligence simulation model with emotions, eight different colors and the behaviors of emotions represented by eight different colors were defined. Four of these are basic emotions and the other four are defined as secondary emotions. The color yellow, which is the representative of the first of the basic emotions, was considered as the color of the feeling of joy and was examined in three different intensities, from the lowest to the highest, calmness, joy and enthusiasm. The green color, which represents the second of the basic emotions, was considered as the representative color of the emotion of fear and was examined in three intensities as anxiety, fear and horror. The color blue was considered as the representative of the third basic emotion, sadness, and it was examined in three different intensities as absentmindedness, sadness and grief. The fourth basic emotion, anger, was represented using the color red and was analyzed in three different intensities: discomfort, anger and rage. In addition to these four basic emotions, four secondary emotions were also defined. Although these emotions were not as severe as basic emotions, they were accepted as emotions that could be effective through artificial intelligence. The representative color of trust, which is the first of the secondary emotions, was chosen as pistachio green and this emotion was examined in three different intensities as acceptance, trust and admiration. The representative color of surprise, the second of the secondary emotions, was chosen as ocean blue, and this emotion was examined in three different intensities: distraction, surprise, and surprise. The color purple was chosen as the representative color of disgust, which is the third of the secondary emotions, and this emotion was examined in three different intensities: distress, disgust and disgust. The color orange was chosen as the representative color of expectation, the fourth and last of the secondary emotions, and

this emotion was examined in three different intensities: interest, expectation, and caution. After emotions and colors were processed into artificial intelligence in a synchronized manner, simulations that could be observed and controlled easily thanks to colors were made testable on them.





1. GİRİŞ

Yapay zekâ geniş bir kavram olduğundan, bu çalışmanın kapsamını tanımlayarak başlamak önemlidir. Bir simülasyon birebir gerçekleştirme ve bağlam olmak üzere iki ana yönü olduğu düşünülebilir. Simülasyonun birebir gerçekleştirilmesi, sistemin karşılaştığı gerçek zorlukları ve çözülmesi gereken sorunları tanımlayan kurallar ve hedefler gibi unsurları içerir. Öte yandan bağlam, bu sorunların ortaya çıktığı ortamı oluşturan yapay zekalar ve olay örgüsü gibi tüm öğeleri kapsar. Bu çalışma simülasyon yapay zekasına, yani yapay zekanın duygusundaki veya davranışındaki sorunları çözmekle ilgilenen yapay zekaya odaklanmaktadır. Tersine, bağlam, simülasyondaki bütün karakterlerin olay örgüsünü iletirmek için bir dizi eylem gerçekleştirilmesini sağlamak veya diğer yapay zekaların seçimlerine tepki vermek gibi bağlama özgü görevlerle ilgilenir. Bu nedenle, tasarımın kapsamı bu çalışmada simülasyon yapay zekâsı yönü ile sınırlıdır.

1.1. Tezin Kapsamı

Tez kapsamında renklerle yüksek kontrollü yapay zekalar elde edilmesi amaçlandığından simülasyonun ne olduğunu yapay zekaların nasıl simüle edilebileceğini belirlemek gerekir. Simülasyonlar insanların belirli sonuçlar çıkarmak için kullanmasını amaçlar ve onların bilişsel becerilerine ve fiziksel yeteneklerine odaklanır. Bir simülasyon ne kadar zengin ve karmaşıksa, bu simülasyonun incelenmesi aynı ölçüde bilgi ve yetenek gerektirir. Bu nedenle, karmaşık bir simülasyon için gerçekten akıllı ve tamamen otonom bir aracı oluşturmak, tüm insan zekasının büyük bir bölümünü kopyalamak kadar zor olabilir. Öte yandan, yapay zekâ genellikle her simülasyon için bağımsız olarak tasarlanır. Bu, gelişimi tek bir simülasyon projesinin kapsamı ile sınırlı olduğundan, tamamen sağlam bir yapay zekâ oluşturmayı zorlaştırır. Her simülasyon benzersiz olsa da türlerine bağlı olarak birtakım kavramları paylaşabilirler. Türler, simülasyonları kullanıcıların onlarla etkileşim biçimlerine ve kurallarına göre sınıflandırmak için kullanılır. Kavramsal düzeyde, aynı türdeki simülasyonlar tipik olarak aynı kavramlara dayalı benzerlikler

içerir. Bu benzerlikler daha sonra, problem örneğinden bağımsız olarak temel davranışın tanımlanabileceği ve uygulanabileceği ortak problemleri içerir.

1.2. Tezin Amacı

Tezin amacı yapay zekaların rahatça kontrol edilip gözlemlenebildiği bir simülasyon ortamı oluşturmaktır, yapay zekalar için tasarlanan simülasyonda karakterler davranış, konum tahmini ve navigasyon gibi sorunlarla karşılaşır. Her an, bir yapay zekanın durumu değerlendirmesi ve en uygun davranışı göstermesi, bir hedef belirleyip o hedefe ulaşmak için en iyi rotayı bulması gerekir. Bu problemler, yapay zekanın hızı, engeller ve varış konumu gibi veriler kullanılarak kavramsal düzeyde gerekçelendirilebilir. Bu kavramlar, birçok yapay zekâ simülasyonunda ortaktır ve yorumlarının ayrıntıları ne olursa olsun etkili davranışı tanımlamak için yeterlidir. Bu tür çözümler, belirli gezinme sorunları için zaten mevcuttur ve birçok simülasyonda kullanılmaktadır. Dahası, bir simülasyondan elde edilen deneyim aynı türdeki başka bir simülasyonda genellikle zahmetsizce kullanılabilir. Yapay zekâ simülasyonlarında üretilmiş bir çalışma, çoğu durumda yeni bir yapay zekâ simülasyonunda sıfırdan başlanılan bir simülasyondan daha iyi performans göstermelidir ve hatta yeni simülasyonda yeni başlanmış bir simülasyondan daha iyi performans gösterebilir, bu da bunun mümkün olduğunu gösterir. Bir simülasyon için öğrenilen davranışı, benzer kavramları içeren başka bir simülasyonda, ikincisinin ayrıntılarını bilmeden iyi performans göstermek için uygulamak. Açıkçası, ayrıntılar keşfedildiğinde, temel kavramsal davranışı daha da geliştirmek ve hatta onu geçersiz kılmak için kullanılabilir. Bu nedenle, simülasyona özgü örnekler yerine kavramsal sorunları belirleyerek ve hedefleyerek simülasyonlardaki yapay zekâları renkleri kullanarak daha kuvvetli ve daha kontrol edilebilir oluşturmak mümkün olabilir. Bu sayede geliştiricilerin oluşturulan bu yapay zekâları benzer sistemlerde kullanarak onları sınırlamaya iten proje kısıtlamalarını ortadan kaldırması ve onların sürekli, daha kapsamlı ve daha rahat gözlemlenebilir bir tasarım sürecine sahip olmasını sağlamak amaçlanmıştır.

2. KAYNAK ARAŐTIRMASI

Bu alıŐmayla ilgili yayınlanmış araŐtırma makalelerini bulmak iin Google Akademik aracılıđıyla bir literatür taraması yapıldı. Renkler, yapay zekâ, psikoloji ve simülasyon gibi terimlerin kullanıldıđı bir ön arama, ilgili arama kriterlerini belirledi. Daha sonra aŐađıdakilerle üç kapsamlı arama gerekleŐtirildi:

1. (Yapay zekâ veya AI) ve (renkler)
2. (Renkler) ve (psikoloji)
3. (Yapay zekâ veya AI) ve (simülasyon)

Bu alıŐmaya dahil edilecek makalelerin seimini daraltmak iin koleksiyon, yapay zekâ ve renkler terimlerinin yanı sıra belirli simülasyon alıŐmaları ve önde gelen araŐtırmacılar alıntılar iin arandı.

2.1. Yapay Zekâ

Yapay zekayı tanımlamak her ne kadar kolay görünsede bir o kadar da zordur. Para olarak incelendiđinde yapay kelimesinin anlamı dünyada dođal olarak var olanın insanın kopyalaması sonucu elde edilmiş olan bir kopyasıdır. Zekâ ise, farklı yollarla tanımlanabilir. Bu tanımlardan bazıları Őunlardır: soyutlama, mantık yürütme, anlama, kendi farkında olma, öğrenme, duygusal bilgi, muhakeme yapabilme, planlama, yaratıcılık, eleŐtirel düşünme ve problem çözme yetenekleri. Daha genel bir anlamda, zekâ, bilgiyi algılama veya çıkarım yapma ve bu bilgiyi bir ortama veya bađlama uygun davranıŐlar sergilemek iin kullanabilme yeteneđi olarak tanımlanabilir [1]. Daha genel olarak, bilgiyi algılama, çıkarım yapma ve onu bir ortam veya bađlam iinde uyarlanabilir davranıŐlara uygulanacak bilgi olarak tutma yeteneđi olarak tanımlanabilir. Zekâ üzerine yapılan alıŐmaların çođunluđu insan üzerinedir ünkü dünya üzerindeki en zeki ve en karmaŐık zekaya sahip varlık insandır ancak insan olmayan hayvanlar üzerinde de birok alıŐma gerekleŐtirilmektedir, ayrıca bazı alıŐmalar bitkilerin de zekâ sahibi olabildiklerini göstermişlerdir.

Bu düşüncelerden yola çıkarak zekâyı herhangi bir canlının bulunduğu fizyolojik veya psikolojik etmenler doğrultusunda içgüdüsel veya kendi iradesiyle verdiği kararlar olarak tanımlayabiliriz. Yapay zekâ oluşturulurken de zekanın bu işlevleri genel veya kapsamlı olarak kopyalanmaya çalışılmıştır.

Yapay zekâ günümüzde sağlık, eğitim, askeri, finans, sosyal medya, robotik, oyunlar ve sistem simülasyonu gibi pek çok alanda kullanılmaktadır. Bu çalışmada ise yapay zekanın simülasyon alanında kullanıma değinildi.

Yapay zekalar bazı arařtırmacılar tarafından simülasyonlarda kullanırken duygu açısından iki farklı şekilde değeriendirilmektedir. Bazı arařtırmacılar ve geliřtiriciler duyguların yapay zekalarda önemli bir özellik olabileceğini savunurken, diğeriileri duygusal tepkilerin gereksiz olduğunu düşünürler. Duyguların gereksiz olduğunu savunanlar, yapay zekâ için duygusal tepkiler gereksiz olduğunu savunmuşlardır. Duygusal tepkilerin, karar verme sürecinde objektifliğı etkileyebileceğı ve yapay zekanın işlevselliğini azaltabileceğı düşünölmüş. Ayrıca, duygusal tepkilerin karmaşık bir yapıya sahip olduğunu iddia ve yapay zekâ sistemlerinin bu tepkileri gerçekçi bir şekilde taklit etmesinin zor olduğu da belirtilmiştir. Duyguların gerekli olduğunu savunanlar ise duyguların yapay zekalarda önemli bir rol oynayabileceğini düşünmüşlerdir. Duygusal tepkilerin, insanlarla daha iyi etkileşim kurabilmek, duygusal anlamda destek sağlamak ve insan benzeri davranışları sergilemek gibi avantajları olduğunu öne sürölmüşdür. Bu yaklaşıma göre, duygusal zekâ, yapay zekaya insanların duygusal yanıtlarını daha iyi anlama ve tepki verme yeteneğı kazandırabilir. Yapılan çalışmalardan bahsetmek gerekirse, MIT Media Lab'ın önde gelen arařtırmacılarından biri olan Picard, duygusal zekâ ve insan-makine etkileşimi konularında çalışmalar yapmıştır. Duygusal yapay zekâ alanında öncü çalışmalarıyla tanınır ve duygusal tepkilerin yapay zekada önemli bir rol oynayabileceğini savunan görüşlere sahiptir [2]. Picarda benzer bir görüşe sahip MIT'de Robotik ve yapay zekâ Laboratuvarı'nın kurucusu olan Breazeal, duygusal robotlar ve insan-robot etkileşimi konusunda çalışmalar yapmıştır [3]. Duygusal robotlar aracılığıyla insanlarla daha derin bir bağ kurulmasının ve etkileşimlerin daha anlamlı hale gelmesinin mümkün olduğunu savunmaktadır.

Picard ve Breazeala farklı görüşte olan arařtırmacılarından Minsky, duygusal tepkilerin yapay zekada gereksiz olduğunu düşünen görüşlere sahiptir. Minskye göre, yapay

zekanın amacı sadece bilgi işleme ve zihinsel yeteneklerin taklit edilmesidir, duygusal tepkiler bu süreç içerisinde gerekli değildir [4].

Araştırmacıların görüşlerine ve bu alandaki çalışmalara dayanarak yapay zekalar ihtiyaç ve duruma göre farklı şekillerde kullanılabilir denilmesi mümkündür. Süreç basit bir şekilde gerçekleştirilmek isteniyorsa yapay zekalara duygu ekleyip sürecin daha karmaşık bir hale getirilmemesi gerekir, duygular her ne kadar sistemin gerçeklik payını artırıyor olsa da arka plandaki hesaplamalar kısa sürede gerçekleştirilmek istenen süreçleri çok uzun bir hale getirebilir. Süreç dinamik bir şekilde gerçekleştiriliyorsa, zamanla değişen veya gelişen bir sürecin kurgulanması gerekir. Bu tür süreçler genellikle bir sistem veya olayın çeşitli bileşenlerinin etkileşimi veya birbirleriyle ilişkisi sonucunda ortaya çıkar. Dinamik süreçler, bir veya daha fazla değişkenin zamanla farklı değerler alması ve sistemdeki etkileşimlerin sonuçlarına bağlı olarak durumun değişmesi anlamına gelir. Bu sebeple yapay zekaların duygulara sahip olması sistemi dinamik kılmakta oldukça etkilidir.

2.1.1. A* yol bulma algoritması

Peter Hart, Nils Nilsson ve Bertram Raphael tarafından 1968 yılında Stanford Araştırma Enstitüsü'nde geliştirilmiştir. Algoritma, graf teorisi ve yapay zekâ alanlarında kullanılan etkili bir yol bulma yöntemidir [5]. PeteHart, Nils Nilsson ve Bertram Raphael, araştırmaları sırasında, Dijkstra algoritmasının yol bulma problemlerindeki etkinliğini artırmak için bir geliştirme yapmak istediler. Dijkstra algoritması, en kısa yolu bulmak için genişlik öncelikli bir arama kullanırken, a* algoritması hem maliyeti hem de tahmini maliyeti dikkate alarak daha verimli bir yol bulma yöntemi sunar. A* algoritması, her düğüm için bir tahmini maliyet hesaplar ve bu maliyeti mevcut düğüm maliyetiyle birleştirerek bir öncelik sırası belirler. Bu şekilde, algoritma daha olası yol seçeneklerini önceliklendirir ve en az maliyetli yolun bulunmasını sağlar. A* algoritması, yapay zekâ uygulamalarında ve video oyunlarında sıkça kullanılan bir yol bulma algoritması olmuştur ve yıllar içinde birçok varyasyonu ve iyileştirmesi yapılmıştır, ancak temel fikir ve algoritma yapısı Peter Hart, Nils Nilsson ve Bertram Raphael tarafından geliştirilmiştir.

2.1.2. Sürü akını algoritması

Sürü akını algoritması, Craig Reynolds tarafından 1986 yılında geliştirilmiştir [6]. Reynolds, kuşların, balıkların veya sürü halinde hareket eden diğer organizmaların

davranışını modellemek ve simüle etmek için bir algoritma arayışındaydı. Bireysel organizmaların basit kurallara dayalı etkileşimleri sonucunda ortaya çıkan sürü hareketi ve sürünün kolektif davranışını gözlemlemek istiyordu. Bu amaçla, "boids" adını verdiği sanal organizmaları kullanarak sürü davranışını simüle eden bir algoritma geliştirdi.

Sürü akını, sürü davranışlarını taklit eden bir yapay zekâ tekniğidir. Sürü akını, bir grup varlığın (genellikle kuşlar, balıklar, böcekler gibi) birlikte hareket ederek organize ve senkronize bir şekilde davranmasını sağlar. Bu teknik, sürü halindeki varlıklar arasında koordinasyonu ve etkileşimi modellemek için kullanılır.

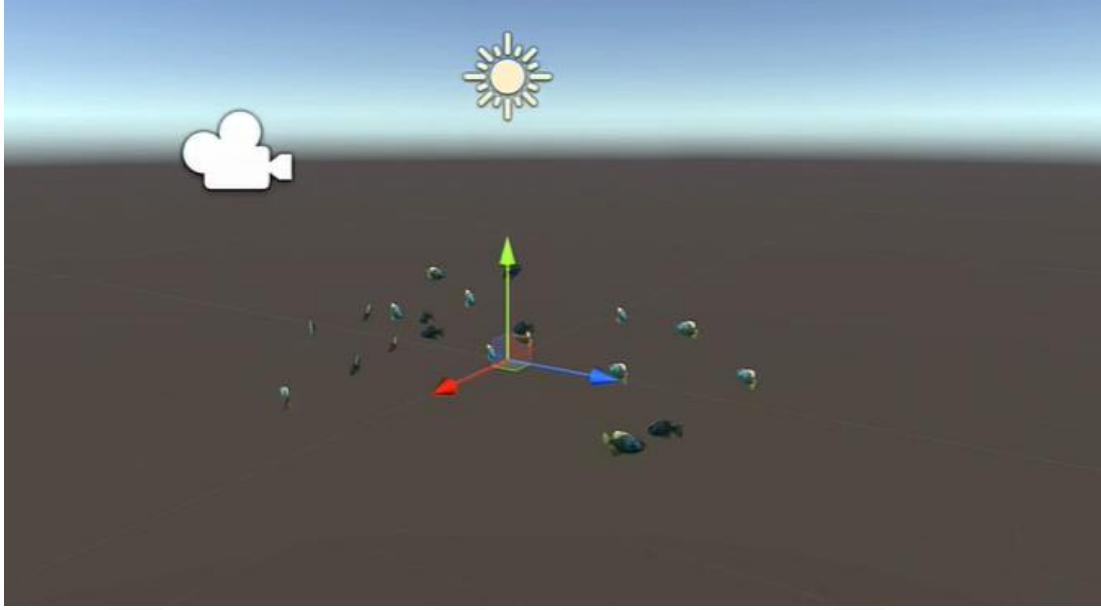
Sürü akını, birçok varlık arasında sürü davranışlarını taklit etmek üzere üç temel bileşenden oluşur: ayrışma, hizalama ve uyum.

Ayrırma: Varlıkların birbirlerine olan yakınlığına dayalı olarak birbirlerinden uzaklaşmalarını sağlar, çarpışmaları önler ve birbirlerine karışmalarını engeller. Her canlı, diğer canlılarla belirli bir mesafe içinde bulunmayı tercih eder.

Uyum: Varlıkların sürüde birbirlerine yakın olmasını ve bir grup olarak hareket etmeyi tercih etmelerini sağlar. Varlıklar, sürünün merkezine doğru hareket ederek birlikte kalırlar.

Hizalama: Varlıkların komşularının hareket yönünü takip etmelerini sağlar ve varlıkların birbirlerinin hareketlerini gözlemleyerek ortak bir yöne doğru hizalanmalarını sağlar. Bu sayede, sürüdeki varlıkların aynı yöne doğru hareket etmeleri ve senkronize bir şekilde davranmaları sağlanır.

Sürü akını algoritması, animasyon, yapay yaşam simülasyonları, trafiğin modellenmesi ve robotik gibi birçok alanda kullanılmıştır. Reynolds'un geliştirdiği bu algoritma, sürü davranışının anlaşılması ve simülasyonu konusunda önemli bir adımdır. Bu algoritmayla oluşturulmuş bir balık sürüsü simülasyonu Şekil 2.1'de gösterilmiştir.



Şekil 2.1 Sürü akını algoritması uygulanmış balık sürüsü

2.2. Renkler

Renk, elektromanyetik spektrumun belirli dalga boylarında yer alan görsel bir algıdır. İnsanlar ve diğer canlılar renkleri gözleri aracılığıyla algırlar. Renk, görsel uyarıların özelliklerinden biri olup ve çevremizdeki dünyayı algılama, iletişim kurma ve duygusal tepkilerimizi ifade etme açısından önemli bir yere sahiptir.

Renklerin önemine bazı açılardan değinildiğinde renklerin en çok algı ve dikkat çekmek için kullanıldığı fark edilir. Renkler, nesnelerin veya sahnelerin dikkat çekici olmasını sağlar. Belirli renkler, gözleri üstüne çekip ilginin odaklanmasını sağlar. Örneğin, parlak ve canlı renklere sahip nesneler genellikle göze çarpar ve hızla fark edilir. Duygusal ve sembolik anlamlar taşıyabilir ve iletişimde güçlü bir araç olarak kullanılabilir.

Renkler duygusal durumları, kişilik özelliklerini veya toplumsal mesajları ifade etmek için de kullanılabilir. Örneğin, kırmızı renk genellikle enerji, tutku veya tehlike gibi duygusal anlamlar taşırken, mavi renk sakinlik, güven ve sadelik gibi anlamlar ifade edebilir.

Bir markanın veya ürünün tanınma derecesini arttırabilir ve hedef kitleyle ilişki kurabilir. Birçok marka incelendiğinde, belirli renkleri logosunda veya ambalajında

kullanarak marka kimliğini ve kişiliğini yansıttığı görülmektedir. Renkler, tüketicilerin markaları hatırlamasını ve duygusal bir bağ kurmasını sağlar.

Tasarım veya görsel sunumlar aracılığıyla görsel hiyerarşi oluşturmada kullanılabilir. Farklı renklerin kullanımı, dikkati belirli alanlara yönlendirerek ve önemli bilgilerin vurgulanmasına yardımcı olabilir. Kültürel olarak belirli anlamlar ve semboller taşıyabilir. Farklı kültürlerde renklerin anlamları ve sembolik değerleri farklılık gösterebilir. Örneğin, beyaz renk bazı kültürlerde saflık ve temizlik sembolüken, başka kültürlerde yas veya ölümlle ilişkilendirilebilir. Renkler, sanat, tasarım, pazarlama, iletişim ve psikoloji gibi birçok alanda önemli bir rol oynar.

İnsanların duygusal tepkilerini etkileyebilir, algılarını şekillendirebilir ve iletişimde güçlü bir araç olarak kullanılabilir. Yapay zekalar da insanların davranışına benzer davranışlar gösterdiğinden ötürü bu çalışmada renklerin yapay zekaların duygularını etkileyebilen ve onlar üzerinde kontrol sağlanmasına yarayan bir araç olarak görülmüştür.

Renkler üzerine pek çok araştırmacı çalışma yapmıştır ama bütün renk teorilerinin temelinde Isaac Newton'un, 17. yüzyılda ışığın ayrıştırılması ve renklerin incelenmesi konusunda yaptığı çalışmalar vardır. Alman yazar ve bilim insanı Goethe, "Renkler Öğretisi" adlı eserinde renklerin doğasını ve etkilerini incelemiştir [7]. Goethe, renklerin algılanmasında duygusal ve psikolojik etkilerin olduğunu savunmuş ve renklerin insan duyguları üzerindeki etkisini açıklamıştır. Amerikalı renk uzmanı ve yazar Birren, "Color Psychology and Color Therapy" gibi birçok eser yazmıştır [8]. Birren, renklerin duygusal tepkileri nasıl etkilediğini ve renk terapisi alanında kullanımlarını araştırmıştır. Renklerin insan zihni ve duygusal sağlık üzerindeki etkilerini vurgulamıştır. İsviçreli psikolog Max Lüscher, "Lüscher Renk Testi" adlı bir psikolojik test geliştirmiştir [9]. Bu test, kişinin renklere verdiği duygusal tepkileri değerlendirerek kişilik özelliklerini ve duygusal durumunu anlamayı amaçlar. Lüscher, renklerin kişilik ve duygusal durum üzerindeki etkilerini araştırmıştır.

Bu eserlerden yola çıkarak renkler kendi başına bir bütün olarak incelenebildiği gibi renklerin insan üzerindeki fizyolojik ve psikolojik etkileri olduğu söylenebilir. Geliştirilen uygulamanın hedefinde de renklerin algı etkisinden faydalanarak yapay zekalara bu etkilerin yerleştirilmesi amaçlandı.

Renkler tercih edilirken ve gruplandırılırken dikkate alınması gereken ilkeler vardır. Bu ilkelerden ilki sıcaklık ilkesidir.

Sıcaklık ilkesine göre renkler, genellikle sıcaklık algısını etkileyebilir. Sıcak renkler (örneğin kırmızı, turuncu, sarı) enerjik, canlı ve hareketli bir hissiyat sağlar. Bu nedenle, enerjik bir ortam yaratmak veya dikkatin çekilmek istenildiği durumlarda sıcak renkler tercih edilebilir. Sıcak renkler Şekil 2.2’de gösterilmiştir.



Şekil 2.2. Sıcak Renkler

Sıcaklık ilkesinin tam zıttı olan ilke soğukluk ilkesidir. Soğukluk ilkesine göre soğuk renkler (örneğin mavi, yeşil, mor) rahatlatıcı, sakinleştirici ve serinlik hissi verir. Bu nedenle, dinlendirici bir ortam yaratmak veya stresi azaltmak istenildiği durumlarda soğuk renkler tercih edilebilir. Soğuk renkler Şekil 2.3’te gösterilmiştir.



Şekil 2.3. Soğuk renkler.

Renk tercihlerinde sadece rengin kendisi değil mekân veya kullanım alanı da tercihleri etkiler. Örneğin, çalışma alanlarında odaklanmayı artırmak için sakin ve nötr renkler tercih edilebilirken, spor salonlarında enerjiyi yükselten ve motivasyon sağlayan renklere yönelmek daha uygundur.

Renklerin birbiriyle uyumu da renk tercihinde oldukça önemlidir. Renk tekerlemeleri ve renk teorileri, renklerin birbiriyle nasıl uyumlu olduğunu anlamak için kullanılabilir. Örneğin, komplementer renkler (örneğin kırmızı ve yeşil) birbirini tamamlayarak görsel olarak dikkat çekici bir etki oluşturabilir.

Renklerin birbiriyle uyumu da önemlidir. Renk tekerlemeleri ve renk teorileri, renklerin birbiriyle nasıl uyumlu olduğunu anlamak için kullanılabilir. Örneğin, komplementer renkler (örneğin kırmızı ve yeşil) birbirini tamamlayarak görsel olarak dikkat çekici bir etki oluşturabilir.

Bütün faktörlerden bağımsız olarak ele alındığında renk tercihi, kişinin kişisel zevklerine ve hislerine de bağlıdır. Herkesin bulunduğu psikolojik durum farklılıklar gösterebildiği gibi renklere de farklı bir tepkisi ve tercihi olabilir. Bu nedenle, kişinin kendisini rahat ve mutlu hissettiği renkleri seçmesi bir tercihtir.

2.3. Duygu

Duygu, insanların içsel deneyimlerini, duygusal durumlarını ve hissettikleri duygusal tepkilerini ifade eden bir kavramdır. Duygular, bireylerin yaşadıkları deneyimler, düşünceler, olaylar veya etkileşimler sonucunda ortaya çıkar.

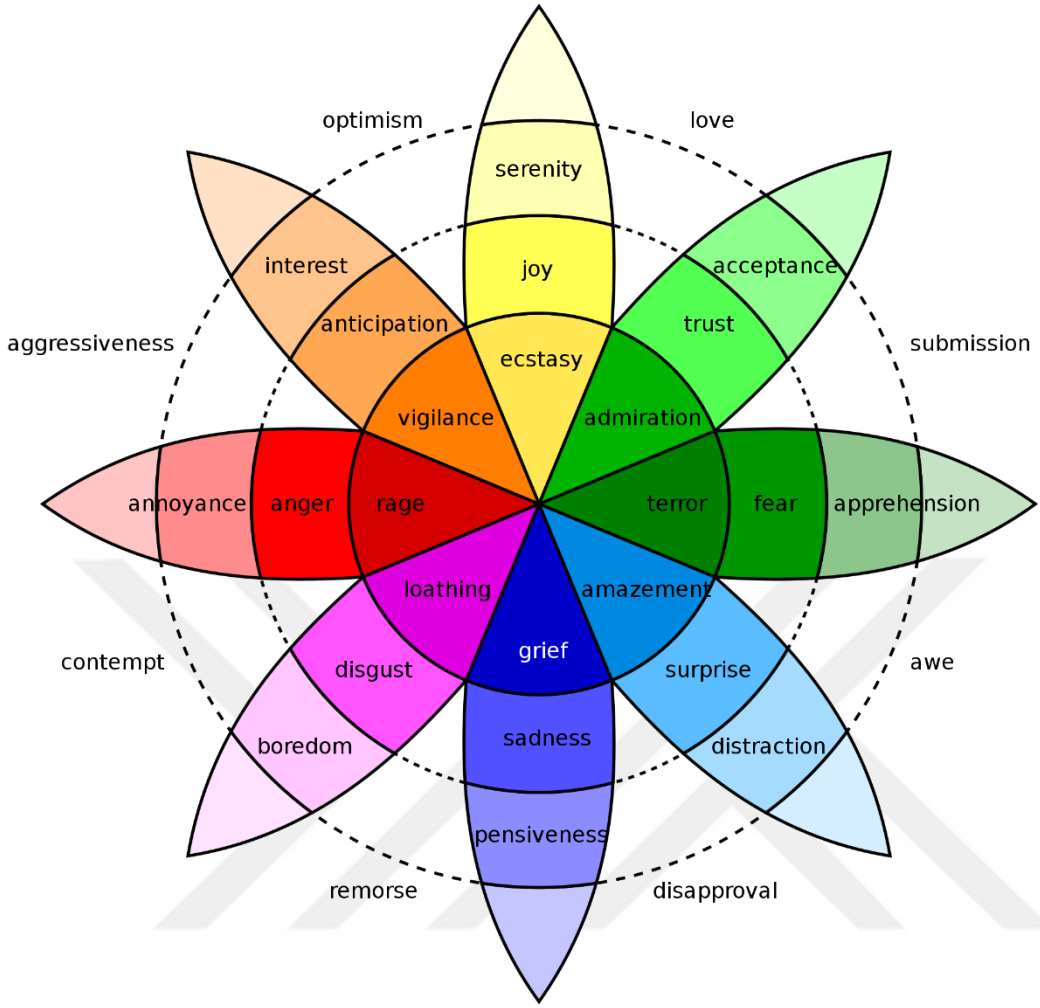
Duygu konusu, birçok araştırmacı tarafından farklı perspektiflerden ele alınmıştır.

Paul Ekman duygusal ifadelerin tanınması ve iletişim üzerindeki etkisini ele almıştır. Ekman duygular ve yüz ifadeleri arasındaki bağlantıyı inceleyen bir psikologdur. Ekman, temel duyguların (örneğin mutluluk, öfke, korku, iğrenme, üzüntü ve şaşkınlık gibi) evrensel olduğunu ve farklı kültürlerde benzer yüz ifadeleriyle ifade edildiğini savunur [10].

Lisa Feldman Barrett, duyguların nasıl oluştuğunu ve beyin-beden ilişkisini tartışmıştır. Barrett, duyguların nasıl oluştuğu ve düşünceler, algılar ve bedensel duyularla nasıl bağlantılı olduğu konusunda çalışmalar yapan bir psikologdur. Barrett, duyguların beynin tahmin süreçlerine dayandığını ve bireyin deneyimlerine ve sosyal ve kültürel faktörlere bağlı olarak şekillendiğini öne sürer [11].

Robert Plutchik "Theories of Emotion" makalesinde duyguların evrimsel temellerini ve temel duyguları işleyen bir teori olan Plutchik'in tekerlek modelini sunmuştur [12]. Bu eserde duyguların temelini oluşturan evrimsel adaptasyonlara ve insan davranışının evrimsel temellerine vurgu yapılmaktadır.

Plutchik, duyguların insanların hayatta kalma ve üreme şanslarını artırmak için evrimsel olarak geliştiğini öne sürmektedir. Bu teori, temel duyguların (örneğin sevinç, üzüntü, öfke, korku gibi) insanların içinde bulunduğu çeşitli durumlara tepki olarak ortaya çıktığını savunur. Plutchik'in duygu çarkı Şekil 2.4'te gösterilmiştir.



Şekil 2.4. Plutchik'in Duygu Çarkı [12].

Plutchik'in duygu çarkı, duyguların karmaşık ve çok yönlü doğasını açıklamaya çalışan bir teorik modeldir. Duygu çarkı duygusal deneyimleri belirli temel duygusal kategorilere ayıran bir yaklaşımdır. Bu kategoriler ve davranış biçimleri farklı şekillerde yorumlanmış olsa da kabul gören yaklaşımlar mevcuttur.

Bu çalışmalardan yola çıkarak renklerin de yapay zekaların üzerindeki etkisi düşünüldüğünde simülasyon çalışmasında kullanılacak modellerden en uygunu Plutchik'in duygu çarkı modelidir. Bu modelin tez kapsamında kullanılması yapay zekaların belirli bir kural çerçevesinde yürütülüp kontrol edilmesinde büyük kolaylık sağlayacaktır. Duygu çarkı modeli çerçevesinde duygular yapay zekalara eklenirken duyguların, davranışlar üzerinden nasıl etkileri olabileceğinin belirlenmesi gerekir.

Temel duygular analiz edilirken farklı kaynaklar incelendi. Duygular konusu geniş bir alan olduğundan dolayı bu kaynaklarda farklı teoriler ve yaklaşımlar görüldü. Yaklaşımlardan ilki Ekman'ın "Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life" kitabındaki yaklaşımdır. Ekman bu eserinde temel duyguların tanınması ve iletişimde duygusal ifadelerin nasıl doğru bir şekilde okunabileceği konularını ele almıştır [13]. Izard "The Psychology of Emotions" adlı eserinde duyguların psikolojik ve biyolojik temellerini inceleyip, temel duyguların analizini de yapmıştır [14]. Tomkins "Affect Imagery Consciousness: The Complete Edition" adlı eserinde duyguların işleyişini ve temel duyguların analizini derinlemesine incelemiştir [15]. Bu eserlerden de faydalanılarak sekiz duygu analiz edilip belirli sonuçlar ortaya çıkarılmıştır.

Sevinç, pozitif duygu ve mutluluk halidir. Bu duygu genellikle başarı, sevindirici bir olay, sevilen bir kişiyle zaman geçirme, olumlu bir haber veya beklenmedik bir mutluluk anında ortaya çıkar. Sevinç, genellikle yüzde gülümseme, neşeli bir tavır, enerjik bir ruh hali ve içsel bir coşkuyla kendini gösterir. Kişi, sevinç anlarında pozitif enerji hisseder, olumlu düşüncelerle dolup taşar ve yaşamdan keyif alır. Sevinç, insanların motivasyonunu artırabilir, ilişkileri güçlendirebilir ve genel yaşam kalitesini yükseltebilir. Sevinç, herkesin yaşadığı farklı deneyimler ve tetikleyicilerle ilişkilendirilebilir. Kişiden kişiye değişebilir, çünkü her canlının sevinç hissettiği şeyler ve etkilendiği faktörler farklı olabilir. Ancak genel olarak sevinç, olumlu bir duygu olarak kabul edilir ve canlıların yaşantısında önemli bir yer tutar.

Öfke, güçlü bir duygu ve içsel bir huzursuzluk halidir. Genellikle bir haksızlık, tehdit, hayal kırıklığı veya öfke uyandıran bir durum karşısında öfke duygusu ortaya çıkar. Canlıların güçsüz hissettiği, sınırlarının ihlal edildiği veya kontrol kaybı yaşadığı durumlarda öfke hissi ortaya çıkabilir. Öfke, bedensel ve zihinsel düzeyde çeşitli belirtilerle kendini gösterebilir. Bunlar arasında artan kalp atış hızı, yükselen kan basıncı, kas gerginliği, nefes alma hızında artış, sinirlilik, hırçınlık, bağırarak konuşma ve saldırganlık eğilimi sayılabilir. Öfke, doğal bir duygudur ve canlıların hayatta kalmasına ve sınırlarını korumasına yardımcı olabilir.

Korku, endişe, rahatsızlık ve tehdit hissidir. Genellikle bir tehlike, bilinmezlik veya gerilimli bir durum karşısında ortaya çıkar. Canlılar korktuğunda, bedensel ve zihinsel olarak tetikte olurlar, kaçma veya savunma tepkileri gösterebilirler. Kalp atışının hızlanması, terleme, titreme, nefes alma hızının artması gibi fiziksel belirtilerle kendini

gösterebilir. Korku, hayatta kalma içgüdüsünün bir parçasıdır ve tehlikeli durumlar karşısında canlının kendisini korumak için uyarı sistemi olarak işlev görür.

Üzüntü, kayıp, acı, hüznün veya hayal kırıklığı gibi durumlarda hissedilir. Canlılar üzgün olduklarında, içsel bir boşluk, hüznün veya mutsuzluk hissi yaşayabilirler. Üzüntü, birçok farklı sebepten kaynaklanabilir, örneğin bir sevilenin kaybı, ayrılık, hayal kırıklığı, başarısızlık veya travmatik bir olay gibi durumlar üzüntüye neden olabilir. Üzüntü, kişinin zihinsel ve fiziksel durumunu etkileyebilir. Depresif hissetmek, enerji eksikliği, iştah değişiklikleri, uyku sorunları, konsantrasyon güçlüğü gibi belirtiler üzüntüyle ilişkilendirilebilir. Ayrıca, umutsuzluk, motivasyon kaybı ve genel olarak yaşamdan zevk alamama gibi duygusal tepkiler de üzümlük durumunda yaşanabilir.

İğrenme, bir şeye veya duruma karşı duyulan yoğun bir hoşnutsuzluk veya itici hissi ifade eder. İğrenme, genellikle kokular, tatlar, görüntüler veya dokunma gibi duysal uyanlarla ilişkilendirilir. İğrenme sayesinde canlılar savunma mekanizması varmış gibi zararlı veya tehlikeli olan şeylerden uzak dururlar. İğrenme, kişisel tercihler, kültürel faktörler ve deneyimler gibi birçok etkene bağlı olarak değişebilir. Bir şeyin iğrenç olarak algılanması farklılıklar gösterebilir. Örneğin, bazı insanlar için belirli bir yiyecek iğrenç olabilirken, başkaları için lezzetli olabilir. İğrenen canlının, vücudunda fiziksel tepkilerin yanı sıra duygusal tepkiler de gerçekleşebilir. Mide bulantısı, tiksinti hissi, isteksizlik, kusma hissi gibi fiziksel belirtiler iğrenmeyle ilişkilendirilebilir. Duygusal olarak ise istifade etme, uzak durma, nefret etme veya rahatsızlık hissetme gibi tepkiler görülebilir.

Şaşkınlık, beklenmedik bir olay veya durum karşısında ortaya çıkan şok veya şaşkınlık hissidir. Şaşkınlık, bir anda meydana gelen bir değişiklik veya sürprizle karşılaşıldığında ortaya çıkar. Bu değişiklik veya sürpriz, olumlu veya olumsuz olabilir. Şaşkınlık hissi genellikle anlık bir durumdur. Bir olayın veya bilginin beklenmedik veya tahmin edilemez olduğunda, canlılar genellikle şaşkınlık hissederler ve bunun sonucunda gösterdikleri tepki de genelde tahmin edilemez olur. Şaşkınlık, kişinin düşünce süreçlerini ve tepkilerini geçici olarak duraklatabilir ve olayı anlamlandırmak için bir süre gerektirebilir. Her canlı şaşkınlık hissini farklı şekillerde deneyimleyebilir ve kişisel tepkileri değişebilir. Şaşkınlık, bir olayın veya durumun anlaşılması ve uygun bir yanıt verme sürecini başlatır. Canlılar, şaşkınlık hissini daha

fazla bilgi edinerek, olayı anlamlandırarak ve farklı tepkiler göstererek başa çıkmaya çalışırlar.

Güven, canlılar arasındaki ilişkilerde önemli bir yere sahiptir. Güven genellikle bir varlığa inanma, bağlılık hissi ve risk alabilme yeteneği olarak tanımlanır. Güven kavramı, bilgi ve yeteneklere dayanan bir inanç sistemi olarak anlaşılabilir. Canlılar, güvendikleri bir varlıkla daha rahat, emniyetli ve huzurlu hissederler. Güven duygusu, karşılıklı anlayış, açıklık, dürüstlük ve tutarlılık gibi faktörlere dayanabilir. Güven, bir ilişkinin veya bir ortamın temel taşlarından biridir. İnsanlar, güvendikleri bir ortamda daha rahat iletişim kurabilir, duygusal bağlar geliştirebilir ve iş birliği yapabilirler. Güven, başkalarına güvenmek ve güvenilmek arasında karşılıklı bir ilişkiyi ifade eder. Ancak, güven duygusu kolay kazanılmaz ve kolay kaybedilir. Yalanlar, aldatma, ihlaller veya hayal kırıklıkları güveni zedeler ve onarılmaya ihtiyaç duyar. Güven, zamanla inşa edilir ve sürdürülmesi önemlidir.

Beklenti duygusu, bir olayın veya durumun gerçekleşmesini beklerken hissedilen bir duygudur. İnsanlar genellikle gelecekte olacak bir olay, deneyim veya sonuç hakkında beklenti duygusu yaşarlar. Beklenti duygusu, umut, heyecan, merak, sabırsızlık veya endişe gibi farklı hisleri içerebilir. Beklenti duygusu, beklentilerin gerçekleşmesini umut ettiğimiz bir gelecek durumuyla ilgilidir. Örneğin, bir bekleyen anne veya baba, çocuğunun doğumunu beklerken heyecan ve umut dolu bir beklenti duygusu yaşayabilir. Beklenti duygusu, bireylerin motivasyonunu artırabilir ve gelecekteki bir olaya hazırlık yapmalarını sağlayabilir. Ancak, bazen beklenen olay gerçekleşmeden önce yaşanan bekleyiş süreci, kişide sabırsızlık veya endişe gibi daha negatif duygulara da neden olabilir. Beklenti duygusu, hayatın birçok alanında rol oynar. Örneğin, bir iş görüşmesi için beklemek, sınav sonuçlarını beklemek, tatil veya etkinlikleri beklemek gibi durumlarda beklenti duygusu hissedilebilir. Her bireyin beklenti duygusu farklı olabilir ve beklentilerin gerçekleşmesiyle birlikte duygu durumları da değişebilir. Beklenti duygusu, insanların geleceğe dair umutlarını, hedeflerini ve isteklerini yansıtan bir duygudur.



3. KULLANILAN TEKNOLOJİLER VE YÖNTEM

3.1. Kullanılan Yazılımlar

Geliştirilen uygulamada simülasyonu gerçekleştirilmesi için gereken 3B ortam Unity oyun motoru kullanılarak tasarlanmıştır. Simülasyon ortamında bulunan yapay zekalar ise Visual Studio 2022 geliştirme ortamında, C# yazılım dili kullanılarak ve nesneye yönelik programlamanın dört ana prensibinden faydalanılarak oluşturulmuştur.

3.1.1. Unity

Unity, oyun geliştirme, sanal gerçeklik, artırılmış gerçeklik ve diğer etkileşimli deneyimlerin oluşturulması için kullanılan bir oyun motorudur. Unity Technologies tarafından geliştirilen ve yaygın olarak kullanılan bir platformdur.

Unity, geliştiricilere grafikler, fizik simülasyonları, ses, animasyon ve oyun mantığı gibi bir dizi aracı bir araya getirerek oyunlar ve etkileşimli içerikler oluşturmayı sağlar. Hem 2B hem de 3B projeler için kullanılabilir ve birden fazla platformda çalışabilme yeteneği sunar. Windows, WebGL, macOS, iOS, Android, Xbox, PlayStation ve daha birçok platformda oyunların son kullanıcıya ulaştırılmasını sağlar.

Unity'nin özellikleri ve kolay kullanımı, geliştiricilere oyunların yanı sıra simülasyonlar, eğitim uygulamaları, sanal turlar, interaktif hikaye anlatımları ve daha fazlasını oluşturma imkanı sunar. Ayrıca, Unity Asset Store üzerinden geliştiriciler hazır modeller, animasyonlar, efektler, ses dosyaları ve diğer kaynakları satın alıp kendi projelerinde kullanabilirler.

Unity'nin diğer oyun motorlarından üstün taraflarından biri oyun geliştirme zamanında geliştiriciye program kodu yazma olanağı vermesidir. Diğer oyun motorlarının ekserisi grafik ile kodu ayırmışken, Unity ile grafik ve kod birlikte çalışmaktadır. Bu çalışma mantığı geliştiriciye esneklik sağlamakta, geliştirme süresini kısaltmaktadır.

Bütün bu avantajlarının yanında Unity 3B'de yazılmış oyunlar düşük ve orta seviye bilgisayarlarda rahatlıkla oynanabilmektedir. Hypercasual alanındaki oyunların pek çoğu da unity oyun motoru kullanılarak üretilmiştir.

Simülasyon alanında kullanımı oldukça yaygındır. Simülasyonlar, gerçek dünya olaylarını veya sistemleri taklit etmek ve anlamak için kullanılan etkileşimli sanal ortamlardır. Unity'nin güçlü grafik motoru, fizik simülasyon yetenekleri ve geniş platform desteği, simülasyon projelerinin geliştirilmesi için ideal bir ortam sağlar.

Unity'nin simülasyon alanında kullanımı çeşitli uygulama alanlarını kapsar. Bu alanlardan bir tanesi eğitim simülasyonlarıdır. Unity, eğitim amaçlı simülasyonlar oluşturmak için yaygın olarak kullanılır. Öğrenciler, sanal ortamlarda pratik yapabilir ve farklı senaryolarda deneyim kazanabilirler. Eğitim simülasyonları; sağlık simülasyonları, uçuş simülatörleri, iş güvenliği eğitimleri ve askeri simülasyonlar gibi çeşitli alanlarda uygulanabilir.

Fizik motoru ve dinamikler sistemi sayesinde mekanik simülasyonlar için de kullanılabilir. Mekanik simülasyonlar; araç simülasyonları, robotik simülasyonlar, makine davranışlarının modellenmesi gibi alanlarda kullanılabilir. Gerçek dünyadaki mekanik sistemlerin simülasyonunu oluşturarak, performans analizleri yapmak ve iyileştirmeler üzerinde çalışmak mümkün olur.

Büyük ölçekli şehir ve trafik simülasyonlarında kullanılabilir. Bu tür simülasyonlar, trafik akışını, kentsel planlamayı, ulaşım analizlerini ve çevresel etkileri modellemek için kullanılabilir. Böylece, gerçek dünya senaryolarını simüle ederek, şehir planlaması ve trafik yönetimi üzerinde çalışmak mümkün olur.

Son zamanlarda Metaverse gibi terimlerle de oldukça yaygınlaşan sanal gerçeklik deneyimleri de Unity oyun motoru üzerinden geliştirilebilir. Sanal gerçeklik kullanıcıları gerçekçi 3D grafikler ve interaktif ortamlar oluşturarak sanal dünyalara taşıyabilir. Sanal gerçeklik, oyunlar, eğitim simülasyonları, mimari görselleştirmeler ve turizm simülasyonları gibi alanlarda uygulanabilir.

Unity'nin simülasyon alanındaki kullanımı, gerçek dünya senaryolarını taklit etmek, deneyim kazanmak, analizler yapmak ve iyileştirmeler üzerinde çalışmak için büyük fırsatlar sunar. Ayrıca, Unity'nin geniş kaynaklara ve büyük bir kullanıcı topluluğuna sahip olması, simülasyon projelerini geliştirenlerin bilgi paylaşımını ve destek almasını mümkün kılar.

3.1.2. Visual Studio

Visual Studio, tüm geliştirme döngüsünü tek bir yerde tamamlamak için kullanılabilen güçlü bir geliştirici aracıdır. Kod yazmak, düzenlemek, hata ayıklamak ve derlemek ve ardından uygulamanın dağıtılması için kullanılan kapsamlı bir tümleşik geliştirme ortamıdır (TGO). Visual Studio, kod düzenleme ve hata ayıklamanın ötesinde, yazılım geliştirme sürecinin her aşamasını geliştirmek için derleyiciler, kod tamamlama araçları, kaynak denetimi, uzantılar ve daha birçok özellik içerir [16].

Visual Studio, geliştiricilere verimli ve işbirliği içinde yüksek kaliteli kod geliştirmek için zengin özelliklere sahip bir geliştirme ortamı sağlar. Visual Studionun sağladığı özellikler:

1. İş yükü tabanlı yükleyici, sadece ihtiyacımız olan özellikleri yüklememizi sağlar.
2. Güçlü kodlama araçları ve özellikleri, uygulamanın tek bir yerde oluşturulması için ihtiyaç olan bütün özellikleri sağlar.
3. Platformlar arası geliştirme, herhangi bir platform için uygulama oluşturabilme.
4. Sürüm kontrolü entegrasyonu, belli bir ekipli kod üzerinde işbirliği yapabilme.

Visual Studio IDE, kodun güvenle yazılmasını ve yönetilmesini sağlayan birçok özellik sağlar. Örneğin, IntelliSense kod önerilerini kullanarak hızlı ve doğru bir şekilde kodlamayı, eylemler öneren ampulleri kullanarak kodda hızlı iyileştirmeler yapılmasını veya ana hatları kullanarak kod bloklarının genişletip/daraltılmasını sağlar.

Visual Studio 2022 simülasyon geliştirme için oldukça ideal bir TGO'dur. Simülasyon endüstrisinde sıkça kullanılan birçok oyun motoru ve yazılım iskeleti ile uyumlu çalışır ve gelişmiş özellikler sunar.

Gelişmiş bir kod düzenleme deneyimi sunar. Otomatik tamamlama, hata ayıklama, kod analizi, hızlı gezinti ve daha pek çok özellik, oyun geliştirme sürecini kolaylaştırır.

Gelişmiş hata ayıklama araçlarına sahiptir. Hata ayıklama sırasında adım adım kod yürütme, hata ayıklama noktaları oluşturma, değişken izleme ve hatta performans profil oluşturma gibi özellikler sunar. Bu, simülasyonlarda karşılaşılan hataları bulmaya ve performans optimizasyonu yapılmasına yardımcı olur.

Birçok popüler oyun motoru ve yazılım iskeleti ile sorunsuz bir şekilde entegre olur. Özellikle Unity, Unreal Engine ve MonoGame gibi oyun geliştirme platformlarıyla uyumludur. Bu sayede, simülasyon projesinin dosya yapısı ve projenin içindeki kaynaklar kolayca yönetilebilir.

Geniş bir eklenti ekosistemine sahiptir. Simülasyon geliştirme için kullanılabilinecek birçok eklenti mevcuttur. Örneğin, oyun motoru entegrasyonları, grafik arayüzler, simülasyon geliştirme şablonları ve diğer özellikler eklenebilir. Bu, geliştirme sürecinin özelleştirilmesine ve ihtiyaçlara göre uyarlanmasına olanak tanır.

Büyük ve aktif bir kullanıcı topluluğuna sahiptir. Bunun sayesinde topluluk, soru sorulması, sorunları çözmek ve ileri düzey rehberlik sağlamak için kaynaklar sunar. Ayrıca, simülasyon geliştirmeyle ilgili özel forumlar ve kaynaklar da mevcuttur

3.2. Yöntem

Uygulama yapay zekâ içerikli bir sanal ortamda gerçekleştirilecektir. Bu nedenle bu ortam belirli koşulları yapay zekalar için sağlayacak bir simülasyon ortamı olmalıdır. Simülasyon ortamı hazırlandıktan sonra navigasyon ağının hazırlanması ve yapay zekaların taşıyıcısı olan navigasyon ağ ajanlarının bu simülasyon ortamına dahil edilmesi gerekir. Simülasyona dahil edilen ajanlar üzerlerindeki komut dosyası bileşenlerini kullanarak yapay zekaların renklerinin gerektirdiği arayüzleri sağlayarak gerekli davranışları yerine getirir.

3.2.1. Simülasyon

Simülasyon, gerçek dünyadaki bir olayı veya sistemleri taklit eden, onları modelleyen ve analiz etmeye yönelik bir süreçtir. Simülasyonlar, gerçek dünya deneyimlerini taklit etmek veya karmaşık sistemlerin davranışını anlamak için kullanılır. Simülasyonların kullanıldığı birçok alan bulunmaktadır bu alanlardan bazıları; havacılık, tıp, otomotiv, şehir planlaması ve askeri operasyonlardır. Simülasyonlara yapay zekalar ekleyerek yapay zekâ içerikli simülasyon elde edilebilir bunun sonucunda gerçek dünya deneyimleri daha dinamik bir hale getirilir. Yapay zekâ içerikli bir simülasyon, gerçek dünya senaryolarını modellemek ve yapay zekâ algoritmalarını kullanarak bu senaryolarda akıllı davranışlar sergilemek amacıyla yapılır. Örneğin, bir askeri simülasyonda, yapay zekâlar gerçek insanlar gibi düşünme, strateji geliştirme ve kararlar alma yetenekleriyle donatılabilir.

Simülasyonlar, yapay zekâ algoritmalarının performansını değerlendirmek, yeni algoritmaları test etmek, karmaşık senaryoları analiz etmek veya yapay zekâ tabanlı sistemlerin gerçek dünya performansını tahmin etmek için kullanılabilir.

Bir otonom araç simülasyonunda, yapay zekâ algoritmaları, trafikteki diğer araçlara tepki verme, yolculuk planlama, güvenli sürüş becerileri gibi yeteneklerle donatılabilir. Bu simülasyonlar, otonom araçların gerçek dünya ortamlarında nasıl davranacaklarını test etmek, performanslarını ölçmek ve iyileştirmek için kullanılabilir. Yapay zekâ içerikli simülasyonlar, yapay zekanın gerçek dünyada kullanıldığı pek çok alanda önemli bir rol oynar. Bu simülasyonlar, yeni algoritmaların geliştirilmesi, eğitilmesi ve optimize edilmesi için etkili bir araç sağlar.

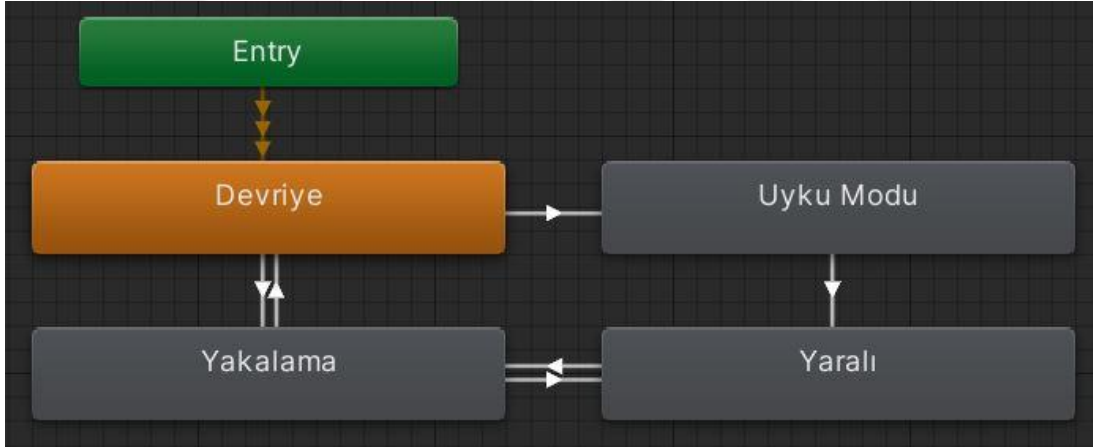
Bu çalışmada da Unity yazılımında oluşturulacak ve kitle simülasyonu olarak bilinen bir davranış ele alınmıştır. Bu durumda, bireysel bazda değil, kitlesel ölçekte araçlarla ilgilenilmesi gerekmektedir. Gruba kendi kimliğini veren hareket, çarpışma, yön ve bir kişiden diğerine hız yayılımı gibi kalabalık simülasyonundan kaynaklanan farklı davranışların gözlemlenmesi planlanmaktadır.

Ortamdaki her bireyi simülasyona dahil etmek ve grup davranışı üretmek karmaşık bir işlem olup ortamdaki bireyler ve nesnelere için çok sayıda karar gerektirir ve bunların tümü basit vektör yöntemleri kullanılarak uygulanabilen davranışlara dayanır. Kalabalık davranışını formüle etmek için birçok farklı yöntem kullanılmıştır. Kalabalıklar kaotik bir şekilde hareket etmezler ve görüldüğünden daha organize dirler. Hareket yönünde görünen akış çizgileri ile kendi kendini organize eden sistemler olabilirler. Kalabalık ne kadar yoğun olursa, türbülansın meydana gelme olasılığı o kadar yüksek olur. Kalabalık davranışını simüle etmenin en basit yollarından biri, Reynolds'un sürü akını algoritmasını uygulamaktır.

3.2.2. Durum makinesi

Durum makinesi, bir sistemin farklı durumlarını ve bu durumlar arasındaki geçişleri tanımlayan bir modelleme yaklaşımıdır. Her durum, sistemin belirli bir zamandaki davranışını temsil eder ve sistem belirli bir olay veya koşul gerçekleştiğinde durumlar arasında geçiş yapar. Bunun yararı, herhangi bir zamanda yalnızca mevcut durumla ilgili olan mantığa ve diğer her şeyi göz ardı ederek yalnızca bunun değişmesine neden olabilecek koşullara odaklanılmasına izin verilmesidir. Durum makinesi nesnenin davranışına bağlı olarak onunla çalışmayı çok daha kolaylaştırabilir.

Örneğin, bir düşman nesnesinin iki durumu olabilir, bir devriye durumu ve bir takip durumu. Düşman devriye gezerken ortalıkta dolaşıyor ama oyuncuyu görünce peşine düşüyor. Bunu bir komut dosyası kullanarak yapmak, tipik olarak, oyuncunun düşmanı görüş alanında olup olmadığını kontrol etmeyi ve ardından yanıt olarak davranışını değiştirmeyi içerir. Düşmanın tek yaptığı, ilerlediği nesneyi değiştirmekse, bunu tek bir senaryoda yapmak çok da zor olmaz ancak, düşman komut dosyasına, oyuncuya uyarıldıklarında zıplamalarını sağlayan küçük bir animasyon veya uyku durumu veya zararlı durum gibi ek durumlar gibi ekstra özellikler eklemeye başlar başlamaz, örneğin, herhangi bir anda düşmanın ne yapması gerektiğini yönetmek zorlaşmaya başlayabilir. Bunun nedeni, başka bir şey yapmasına izin vermeden önce genellikle düşmanın şu anda ne yaptığının kontrol edilmesinin gerekliliğidir. Şekil 3.1’de dört farklı durum içeren durum makinesi örneği verilmiştir.



Şekil 3.1. Durum Makinesi.

Basit bir sistemde bu sorun olmayabilir. Ancak, belirli bir eylemi engelleyebilecek birçok farklı koşul varsa, bir şey yapmaya çalışmadan önce hepsinin kontrol edilmesi gerektiğinden, hızla karmaşık hale gelebilir.

Bir durum makinesi, bir sistemin durumunun nasıl değerlendirileceğini değiştirir. Halihazırda neler olup bittiğini kontrol etmek yerine, her nesnenin yalnızca kendi davranışıyla ve değişmesine neden olabilecek koşullarla ilgilenmesi gerekmektedir. Örneğin, uyku durumunda sistemin hareket etmesine veya oyuncuyu bulmaya çalışmasına gerek yoktur ve düşmanın farklı bir duruma girmesine neden olabilecek tek olay yaralanmadır.

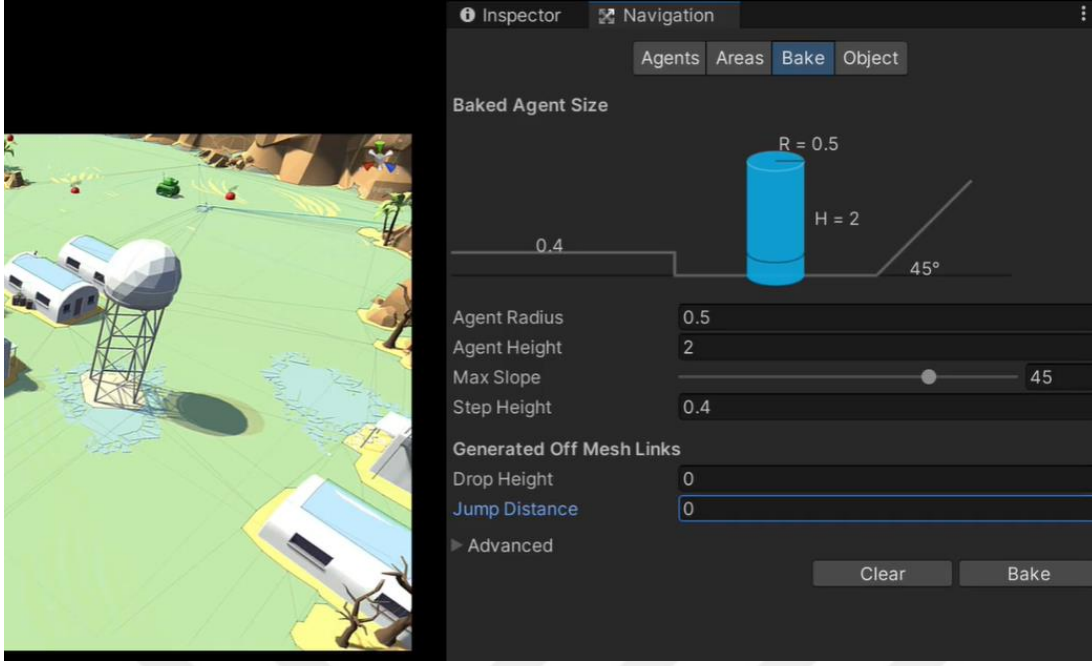
Devriye durumundayken, hasar almak veya oyuncuyu görmek, düşmanın onları kovalamasına neden olur. Son olarak, kovalama durumundayken, düşman oyuncuyu kaybederse devriye durumuna geri dönebilir. Bundan sonra, yeterli zaman geçerse, uyku moduna geri dönebilir.

3.2.3. Navigasyon ağı

Navigasyon ağı, 3B geliştirme sürecinde karakterlerin ve yapay zekanın hareketini kontrol etmek için kullanılan bir tekniktir [17]. Bir sanal dünyanın veya bir sahnenin yüzeyini, yani karakterlerin hareket edebileceği geçerli alanları temsil eden bir veri yapısıdır. Navigasyon ağı bir sanal dünya ortamının statik geometrisini temsil eder. Bu statik geometri, engeller, duvarlar, zeminler gibi karakterlerin çarpması veya geçmesi gereken nesnelere içerir. Navigasyon ağı, sanal dünyadaki geometri üzerinde geçerli geçiş alanlarını belirler ve karakterlerin hareket edebileceği güvenli yolları tanımlar. Bu, karakterlerin doğal hareketlerini taklit etmek için kullanılır. Karakterler, navigasyon ağı üzerinde seyahat ederken engele çarpmadan veya geçmek zorunda oldukları engelleri otomatik olarak atlama gibi işlemleri gerçekleştirebilir. Böylece, yapay zekaya sahip karakterlerin düşmanlardan kaçması, hedeflere doğru ilerlemesi veya belirli bir bölgede dolaşması gibi hareketler akıllıca yönetilebilir. Navigasyon ağı aynı zamanda yapay zeka algoritmaları için de kullanışlıdır. Bir yapay zeka ajanı, Navigasyon ağı üzerinde seyahat ederken hedeflere veya belirli noktalara doğru ilerlemek için navigasyon ağını kullanabilir. Yapay zeka algoritmaları, navigasyon ağı üzerindeki yol verilerini analiz ederek en kısa veya en uygun yolları bulabilir ve bu yolları takip edebilir.

Unity oyun motoru, navigasyon ağı oluşturma ve kullanma konusunda gelişmiş bir desteğe sahiptir. Unity, sahnenin bir kısmını veya tamamını navigasyon ağına dönüştürmek için bir araç seti sunar. Bu araçlar, engelleri ve geçiş alanlarını belirlemek için kullanıcı dostu bir şekilde navigasyon ağı oluşturulmasını sağlar. Ardından, karakterlerin navigasyon ağı üzerinde seyahat etmesi için navigasyon ağ ajanı bileşeni kullanılır. Navigasyon ağ penceresi Şekil 3.2’de gösterilmiştir.

Tez kapsamında navigasyon ağı renkleri kullanarak elde edilen yapay zekaların hareket etmesini sağlamaktadır.



Şekil 3.2. Navigasyon ağ penceresi

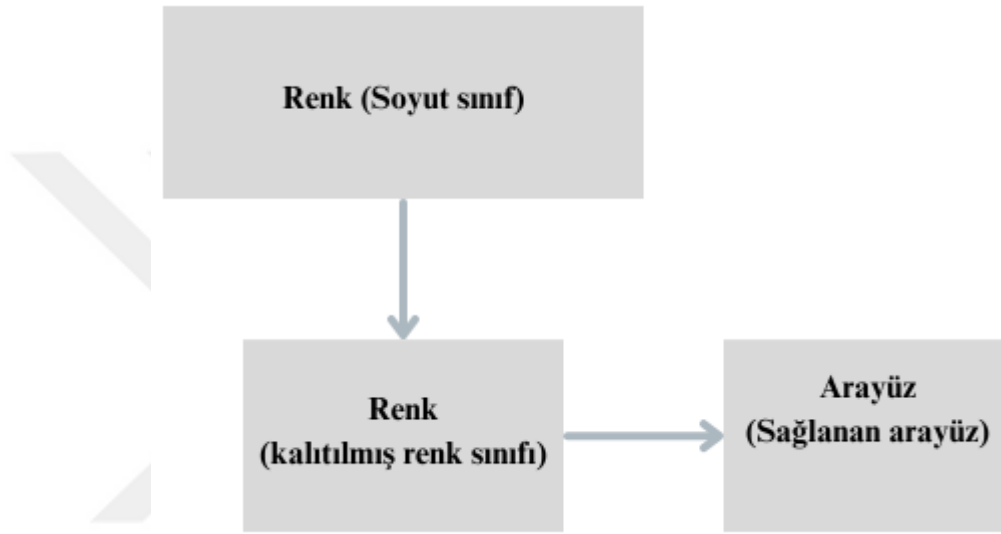
Navigasyon ağı oluşturulduktan sonra ağ üzerinde hareket edecek olan ajanları oluşturulabilir.

Navigasyon ağ ajanı, hedeflerine doğru ilerleyen birbirinden bağımsız karakterler oluşturulmasına olanak sağlar. Navigasyon ağ ajanları, navigasyon ağını kullanarak tasarlanan ortam hakkında akıl yürütür bununla birlikte birbirlerinden ve diğer hareket eden engellerden nasıl kaçacaklarını bilirler.

3.2.4. Arayüz

Arayüz, bir programlama dilinde belirli bir işlevselliği tanımlayan ve bir sınıfın veya bir nesnenin uygulamasını gerektiren bir yapıdır. Bir arayüz, bir veya daha fazla metod, özellik veya olayın bir araya getirildiği soyut bir yapıdır bu açıdan soyut sınıfları andırırlar. Bir sınıf, bir arayüzü uyguladığında, o arayüzde tanımlanan tüm işlevselliği yerine getirmek zorundadır. Bir arayüz, bir sınıfın hangi metotları veya özellikleri içermesi gerektiğini belirler, ancak gerçek bir uygulama sağlamaz. Arayüz, sınıflar arasında bir sözleşme veya kontrat gibi davranır. Bir sınıf, bir arayüzü uygulamak için arayüzde tanımlanan tüm metotları ve özellikleri içermeli ve gerektiğinde bunları uygulamalıdır. Bu uygulama çerçevesinde de renk olarak türetilen sınıflar kendisine duygu olarak tanımlanan arayüzü gerektirdiği koşulları sağlamalıdır.

Arayüzlerin kullanılması, kodun daha modüler, esnek ve genişletilebilir olmasını sağlar. Bir sınıf, birden fazla arayüzü uygulayabilir, böylece farklı işlevselliği olan arayüzleri birleştirebilir. Bu, çoklu mirasın sağladığı esneklik avantajlarını sunar. Nesne yönelimli programlamada önemli bir bileşendir, kodun soyutlamasını, yeniden kullanılabilirliğini ve kodun daha iyi organize edilmesini sağlar. Ayrıca, kodun bağımlılığını azaltır ve farklı sınıflar arasında daha kolay iş birliği yapılmasını sağlar. Bu şekilde renklerin birbirleriyle olan bağlantısı azaltılıp geliştiricinin yapay zekalar üzerindeki kontrolünü artırılır. Şekil 3.3'te taslak arayüz gösterilmiştir.

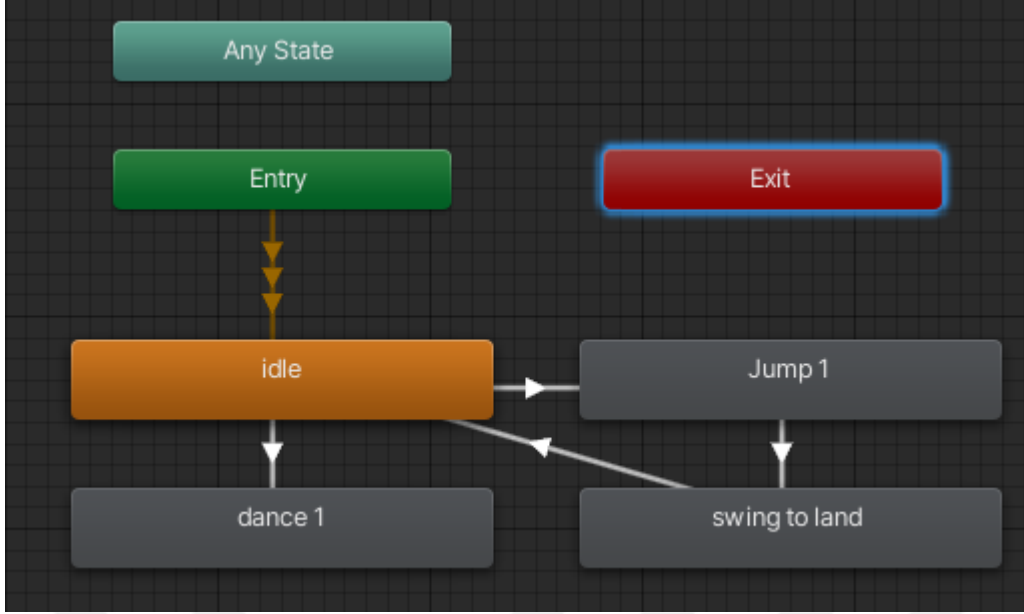


Şekil 3.3. Örnek arayüz şeması.

3.2.5. Animasyon

Unity, güçlü bir animasyon sistemi sunan popüler bir geliştirme ortamıdır. Unity, 2B ve 3B animasyonları destekler ve çeşitli araçlar ve bileşenler sağlar. Unity üzerinde animasyon oluşturulurken ilk adım, animasyon denetleyicisinin oluşturulmasıdır. Animasyon denetleyicisi, animasyon geçişlerini ve durumlarını yöneten bir bileşendir. Proje görünümünde, oluştur denilerek veya sağ tıklayıp çıkan menüden sırasıyla oluştur ve animasyon denetleyicisi seçenekleri seçilerek yeni bir animasyon denetleyicisi oluşturabilir.

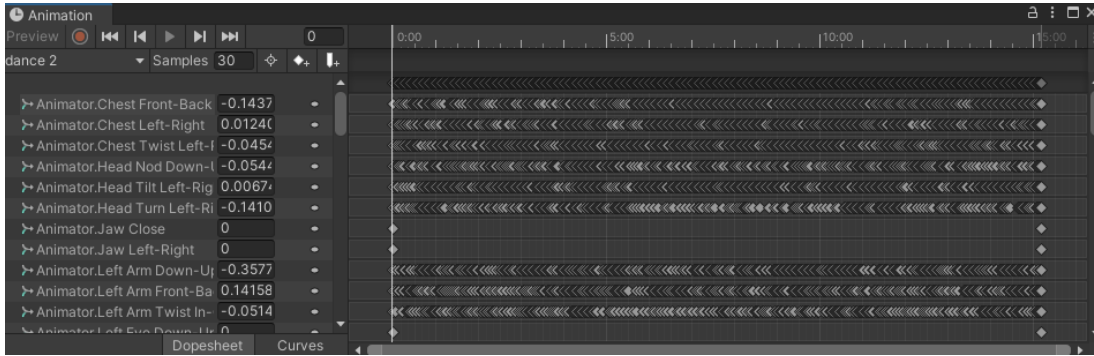
Şekil 3.4'te içeriğinde dört farklı animasyon ve geçişlerini barındıran animasyon denetleyicisinin ekranı gösterilmiştir.



Şekil 3.4. Animasyon denetleyicisi.

Bu adımdan sonra oluşturulan animasyon denetleyicisine animasyonların eklenmesi gerekmektedir. Animasyonlar Unity üzerinden animasyon penceresi kullanılarak oluşturulabilir. Bu pencere üzerinden, farklı durumlar ve geçişler arasında animasyonlar düzenlenebilir.

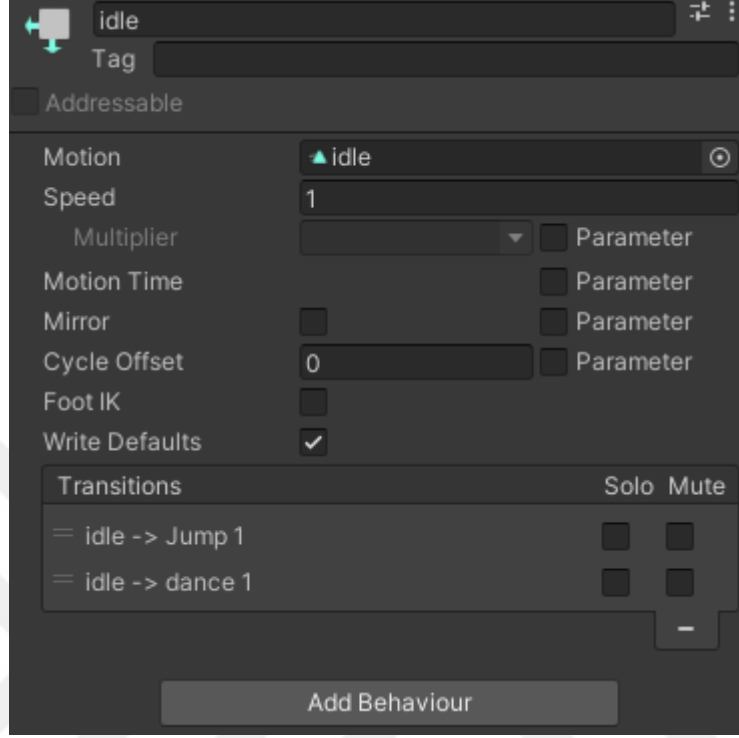
Animasyonları oluşturmak için animasyon penceresi açılıp, bir oyun nesnesi seçilir ve oluştur düğmesi kullanılarak yeni bir animasyon oluşturulur veya mevcut halihazırda mevcut olan bir animasyonu da eklenebilir (Şekil 3.5).



Şekil 3.5. Animasyon penceresi.

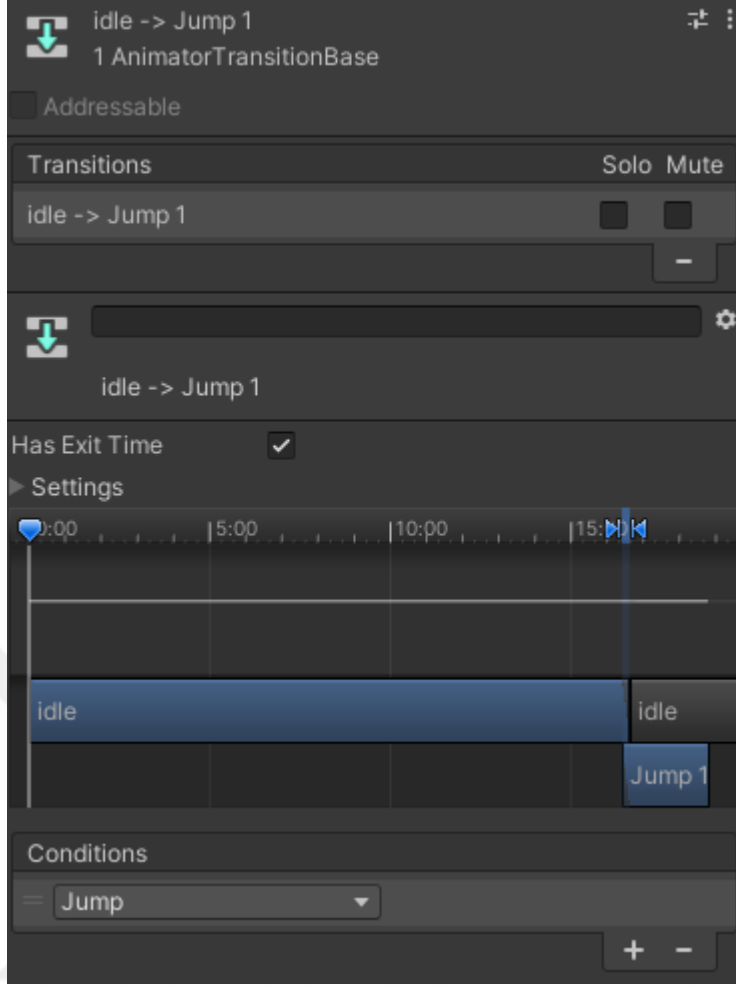
Animasyon denetleyicisini düzenleyerek animasyon geçişleri ve durumları ayarlanabilir. Animasyon denetleyicisi penceresini açılarak, durumlar ve geçişler arasında bağlantılar oluşturabilir ve animasyonların nasıl etkileşimde bulunacağını tanımlanabilir. Animasyon denetleyicisi, durumları ve geçişleri düzenlerken, animasyonlara oynatılacak parametreler veya olaylar eklemek mümkündür.

Animasyon durum penceresinden seçilen durumun animasyonu ve hızı değiştirilebilir, yeni geçişler eklenebilir veya kaldırabilir. Şekil 3.6'da bekleme animasyonu durum penceresi gösterilmiştir.



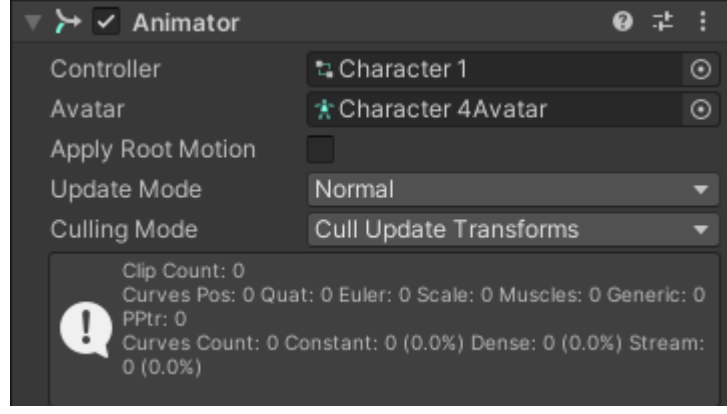
Şekil 3.6. Animasyon durum penceresi.

Animasyon durumlarının geçişi animasyon denetleyicisi tarafından yönetilir. Animasyon durum makineleri, animasyonların geçişlerini, oynatma sürelerini ve diğer parametreleri kontrol etmek için kullanılır. Animasyon durum makineleri animasyonun farklı durumlarını temsil eden durumları içerir, bekleme, dans, zıplama gibi durumlar olabilir. Her durum, bir animasyon klibi veya hareket serisi ile ilişkilendirilir. Durumlar arasındaki geçişler, belirli koşulların sağlanmasıyla gerçekleşir. Karakterin koşma hızı belirli bir eşiğin üzerine çıktığında bekleme durumundan koşma durumuna geçiş yapabilir. Geçişler, belirli bir durumdan başka bir duruma geçiş yapmak için tetikleyiciler veya olaylar tarafından tetiklenebilir. Durum makineleri, animasyon durumlarının oynatma sürelerini, hızlarını veya diğer parametrelerini kontrol etmek için parametreler kullanabilir. Örneğin, bir zıplama animasyonunun yüksekliği veya süresi, durum parametreleri ile ayarlanabilir. Şekil 3.7'de animasyon geçiş kontrol penceresi gösterilmiştir.



Şekil 3.7. Animasyon geçiş penceresi.

Animasyon denetleyicisini bir oyun nesnesine bağlamak için bir animatör bileşenin eklenmesi gerekir. Bir nesne seçilip bileşen ekle düğmesine tıklayarak animatör bileşeni eklenebilir. Animatör bileşeninde, kullanmak istenilen animasyon denetleyicisini atayarak animasyonlar çalıştırılır. Animatör bileşenine, çalıştırılmak istenilen animasyonu başlatmak için gereken parametreler veya olayları tetikleyen kodlar da eklenebilir. Animatör, Unity oyun motorunda yerleşik bir bileşendir. Animasyonların kontrolünü sağlamak ve oyun nesnelerinin animasyon durumlarını yönetmek için kullanılır. Animatör, animasyon denetleyicisini kullanarak animasyon durumlarının ve geçişlerinin yönetimini kolaylaştırır. Şekil 3.8'de animatör bileşeni gösterilmiştir.



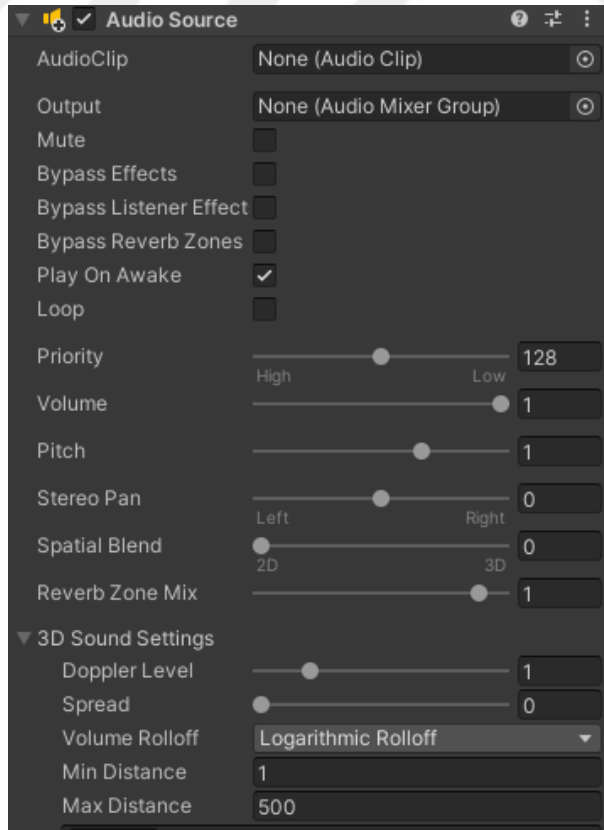
Şekil 3.8. Animatör bileşeni.

3.2.6. Ses

Unity, zengin bir ses sistemi sunan bir geliştirme ortamıdır. Kullanıcılar için uygulamalarda ses efektleri, arka plan müzikleri ve diğer ses unsurlarını yönetmek için çeşitli araçlar ve bileşenler sağlar. Unity üzerinde sesleri oynatmak için öncelikle ses dosyalarının içe aktarılması gerekir. Unity üzerinden ses dosyalarının projelere eklenebilmesi için proje görünümünde ilgili ses dosyası sürekli bırak yapılabilir veya içe aktar düğmesine tıklayarak ses dosyalarını seçilebilir. Desteklenen ses formatları arasında MP3, WAV, OGG ve daha fazlası bulunur. Bu adım tamamlandıktan sonra sonraki adım ses kaynaklarının oluşturulmasıdır. Ses dosyalarını oynatmak için Unity ortamında ses kaynaklarının oluşturulması gerekmektedir. Ses kaynakları, ses dosyalarının oynatılması ve kontrol edilmesi için kullanılan bir bileşendir. Bir nesneye ses kaynağı eklemek için nesne seçilir ve bileşen ekle düğmesine tıklandıktan sonra ses kaynağı bileşeni seçilir. Ses kaynağına, oynatılacak ses dosyasını atanıp diğer ses ayarları da bu pencere üzerinden gerçekleştirilebilir.

Ses kaynakları ile oynatılan ses dosyalarını kontrol edilebilmesi için Unity ortamında bir dizi ses özelliği ve fonksiyonu bulunmaktadır. Ses kaynağının ses seviyesini ayarlanması, sesin tekrarlanması, duraklatılması, durdurulması, 3B ses özelliklerinin kullanılması için ses kaynağı bileşeninin özelliklerini ve fonksiyonlarını kullanılabilir bu şekilde kullanıcıların simülasyonlarda gerçek hayata benzer bir ses efekti yakalaması sağlanır. Unity, ses efektleri ve ayarları için kullanıcılara çeşitli seçenekler de sunar.

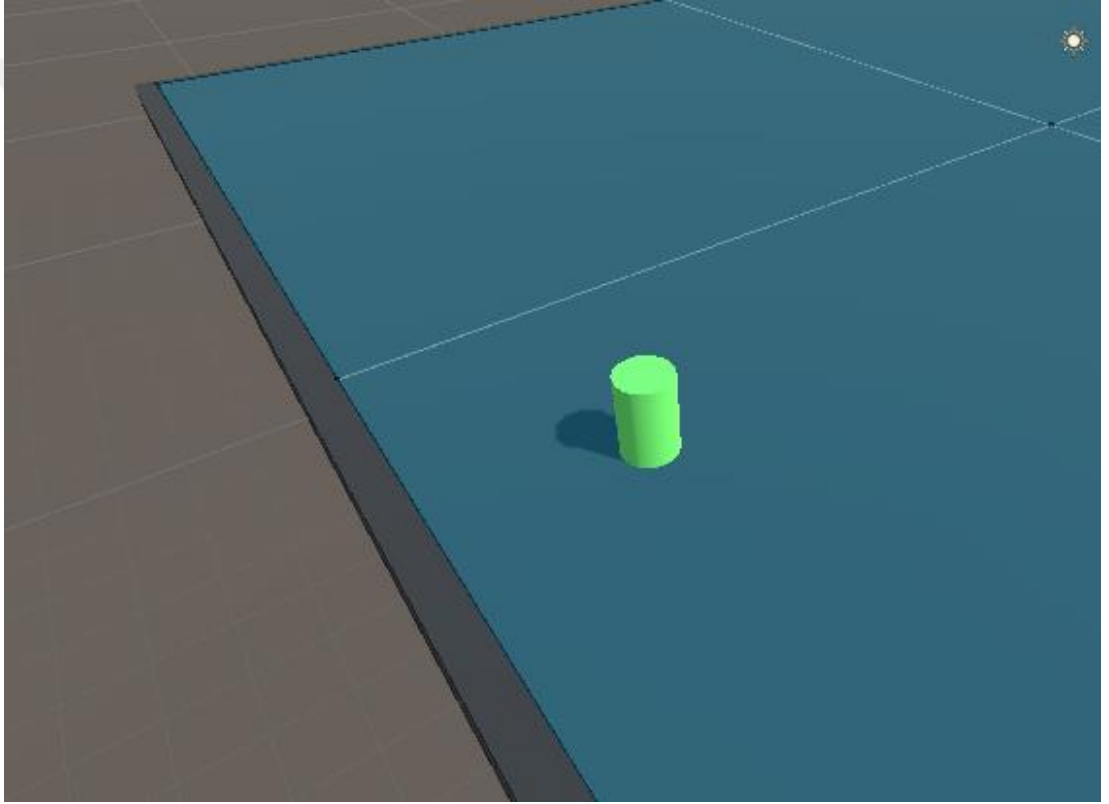
Ses kaynağının 3B ses özelliğini kullanarak sesin konumu ve uzaklığına bağlı olarak nasıl davranacağını ayarlanabilir, bu şekilde yapay zekalar simülasyon içerisinde kendi buldukları durumda bu sesleri yorumlayıp davranışlarını buna göre belirleyebilir. Ayrıca, ses kaynağının efektleri (yankı süresi, ekolayzır vb.) ve ses çıkış ayarları (ses seviyesi, döngü vb.) kullanıcının düzenleyebilmesi mümkün kılınır. Geliştiriciler için seslerin oynatılması ve kontrolü için kodlama yapılması mümkündür. Belirli bir olayın tetiklendiği durumlarda sesin oynatılmasını sağlamak için komut dosyası üzerinden kod eklenebilir. Ayrıca, animasyonlarla, fizik olaylarıyla veya diğer oyun bileşenleriyle sesin senkronize edilmesini sağlamak için de komut dosyası üzerinde kod yazılabilir. Yapay zekaların davranışsal bir ses tepkisi vermesi gerekiyorsa bu komut dosyaları üzerinden bu işlem rahatça gerçekleştirilebilir. Unity ortamında ses sistemi, geliştirilen simülasyonlara atmosfer, gerçeklik ve etkileşim katmanının önemli bir parçasıdır. Ses efektleri, arka plan müzikleri ve diğer ses unsurları simülasyonun gerçekliğini arttırabilir ve kullanıcıların farkındalığını arttırmaya sağlar. Unity ortamı ses sistemi, ses kaynaklarını kontrol etme, ses efektleri uygulama, 3B ses efektleri kullanma ve sesin diğer bileşenlerle senkronizasyonunu yönetme konularında esneklik sağlar. Şekil 3.9'da ses kaynağı bileşeni gösterilmiştir.



Şekil 3.9. Ses kaynağı bileşeni.

4. UYGULAMA

Simülasyon ortamı oluşturulurken Unity3B oyun motoru kullanılmıştır. Simülasyon ortamında karakterlerin yürüyebileceği düzlem oluşturulup navigasyon statik olarak belirlendi. Oluşturulan bu düzlem navigasyon penceresi üzerinden fırınlama işleminden geçirilerek ajanların hareket edebileceği bir düzlem haline getirildi. Simülasyon ortamı Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Simülasyon ortamı.

Sahne tasarımı tamamlandıktan sonra ajan için kameradan çıkan ışınları kullanarak kullanıcının tıkladığı noktaya hareket komutu verilmesini sağlayan komut dosyası oluşturuldu ve ajan için tanımlandı. Oluşturulan komut dosyasının Şekil 4.2’de verilmiştir.

```
if (Input.GetMouseButtonDown(0))
{
    Ray ray = cam.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;

    if (Physics.Raycast(ray, out hit))
    {
        agent.SetDestination(hit.point);
    }
}
```

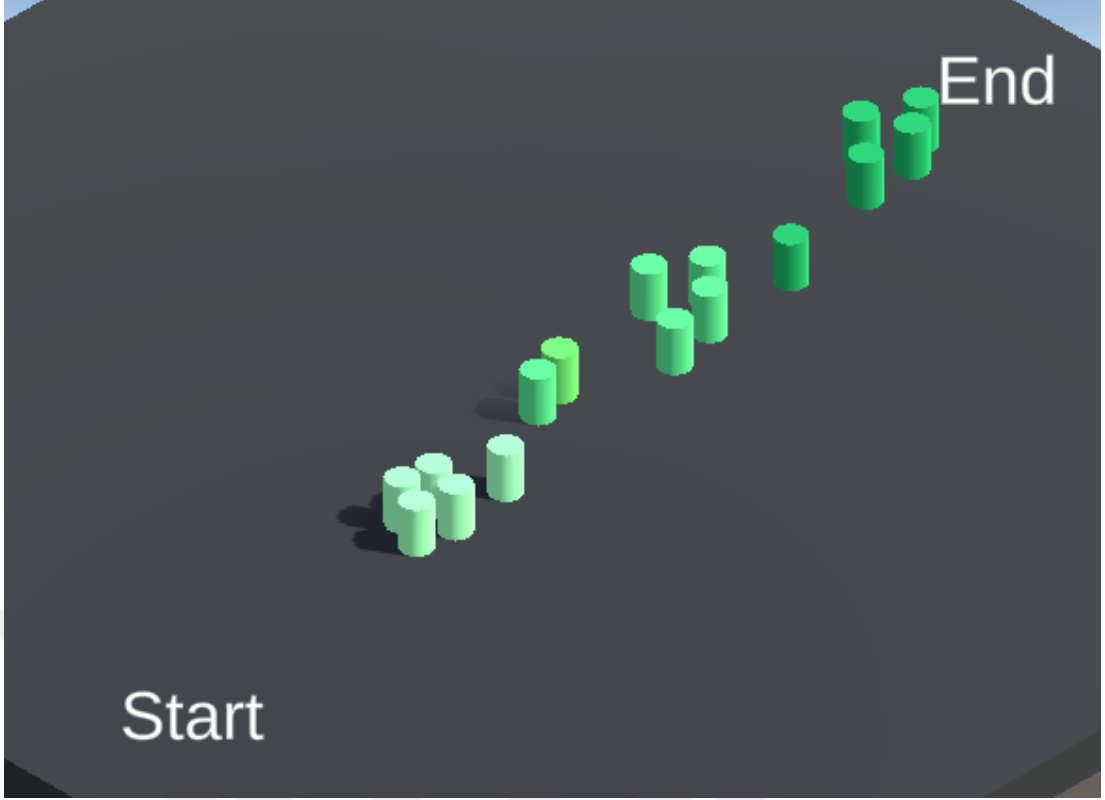
Şekil 4.2. Navigasyon ağı ajanının komut dosyası.

4.1. Yapay Zekaların Oluşturulması

Yapay zekalar oluşturulurken Unity bileşenlerinden navigasyon ağı ajanı bileşeni kullanıldı ve ajanlar navigasyon ağı üzerine yerleştirildi. Oluşturulan yapay zekalar başlangıçta renk komut dosyasına sahip olmayıp olaylar tetiklendiğinde komut dosyaları üzerine eklenebilecek şekilde programlandı.

4.2. Duyguların Eklenmesi

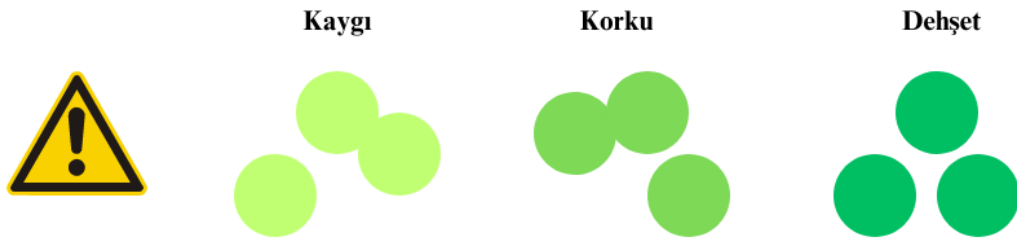
Tek ajanın olduğu bir ortamda ajan komutu verilen noktaya navigasyon özelliğini kullanarak hareket edebilir hale getirildi ve aynı ajanı çoğaltıp aynı özelliğe sahip ajanlar elde edildi. Birden fazla ajan oluşturulduğunda her ajan için hareket şekli birbirlerine benzerdi, benzerliğin azaltılması için her ajan için hız değeri değiştirildi ve bu durumda bazı ajanlar geride kalırken diğerleri varış noktasına hızlıca varabildi. Ancak buradaki temel problemler ajanlar arasındaki farklılıkların gözlemlenmesinin ve yönetilmesinin zorluğuydu. Bunun üstesinden gelmek için her bir ajan için Plutchik'in duygu çarkından faydalanarak bir renk tanımlandı. Tanımlanan renk korku duygusunu temsil eden yeşil renk olarak seçildi. Bu senaryoda ajanların bir hedeften korku duygusuyla kaçtığı varsayıldı ve ajanlar bu duyguyla bağdaştırıldı. Duygu çarkındaki duygu şiddetlerinin, rengin yoğunluğu ile doğru orantılı olduğu göz önüne alınıp korku duygusunun da rengin yoğunluğuyla birlikte artması sağlandı. Aynı şekilde ajanların hızı da rengin yoğunluğuyla doğru orantılı olacak şekilde belirlendi. Yeşil renk ne kadar yoğun olursa ajanların aynı derece yoğun korku duygusuna sahip olup daha hızlı kaçınması sağlandı. Oluşan sonuç Şekil 4.3'te gösterilmiştir.



Şekil 4.3. Renkleri kullanarak yapay zekaların duygu şiddetlerinin ayarlanması.

Yeşil renk için içten dışa duygu seviyeleri dehşet, korku, kaygıdır. Duygular şekilde de görüldüğü gibi rengin yoğunluğu ne kadar fazlaysa ajanın hızı da aynı orantıda fazla olacak şekilde ayarlandı.

Her ne kadar mantık çerçevesine yeşil rengi korkuyu ifade etmek olarak yerleştirilmiş olsa bile simülasyon ortamında ajanların bu duygu durumunu durduk yere tetiklemesi simülasyonun gerçekçiliğini azaltmış oldu. Gerçek hayatta bir varlığın korku duygusunu hissetmesi için fizyolojik veya psikolojik olarak buna sebep olacak bir durumun bulunması gerekir. Bunun için simülasyona korku durumunu tetikleyecek bir nesne eklendi ve ajanlar tekrar gözlemlendi. Şekil 4.4'te sonuçlar gösterilmiştir.

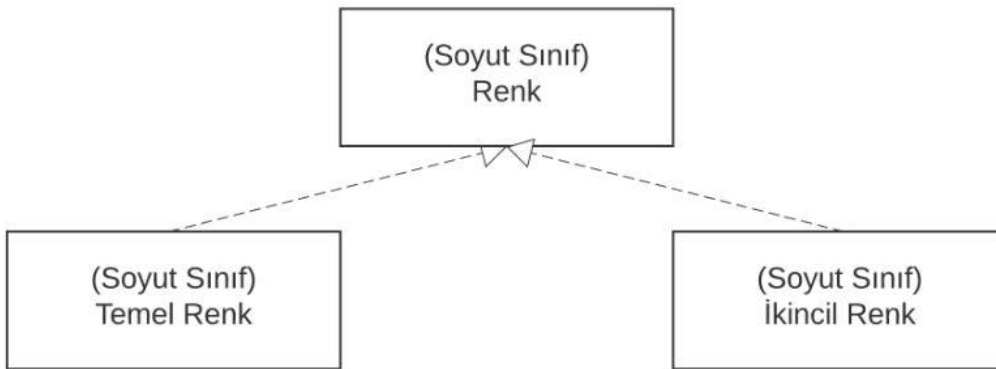


Şekil 4.4. Renkleri kullanarak yapay zekaların duygu şiddetlerinin ayarlanması.

4.3. Nesne Yönelimli Sistem

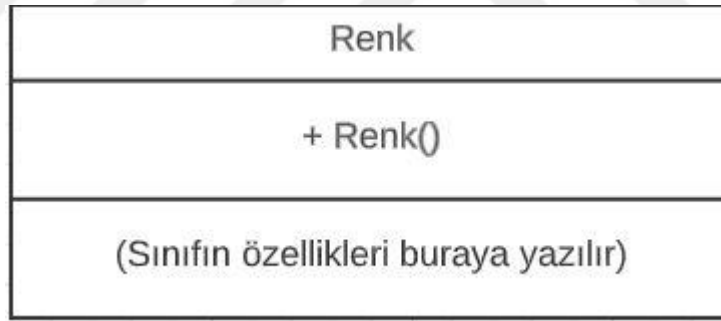
Nesne yönelimli programlama, bir programlama paradigmadır ve programların organizasyonunu ve tasarımını düzenlemek için bir yaklaşım sunar. NYP, gerçek dünyadaki nesnelerin özelliklerini ve davranışlarını programlara yansıtmak amacıyla kullanılır. NYP'nin temelinde nesne kavramı vardır. Bir nesne, verileri ve bu veriler üzerinde işlemleri bir araya getiren bir varlıktır. Nesneler, sınıflar tarafından tanımlanan şablonlardır. Bir sınıf, bir nesnenin özelliklerini ve davranışlarını içeren bir yapıdır. NYP'nin soyutlama, kapsülleme, kalıtım ve polimorfizm olmak üzere dört tane ana prensibi vardır.

Soyutlama, karmaşık sistemlerin daha basit ve daha anlaşılır parçalara ayrılmasını sağlar. Nesne yönelimli programlamada, gerçek dünyadaki nesneler programlama dilinde soyutlanarak, önemli özellikleri odaklanılmasını sağlar. Tez kapsamında soyutlama bütün renk sınıflarında olması gereken özelliklerin ve davranışların renklerin türediği sınıf üzerinde soyutlanmış bir şekilde tutulmasını sağlar. Soyutlanmış ana renk sınıfından temel renkler ve ikincil renklerin özelliklerini belirlemek üzere iki sınıf daha oluşturulur bu şekilde temel renklerin sahip olduğu baskınlık soyutlanmış temel renk sınıfında tanımlanabilir. Bu şekilde temel renk sınıfından kalıtım alan renkler baskınlık özelliğine kendi içlerinde varsayılan olarak sahip olur. Şekil 4.5'te soyutlanmış sınıflarla tasarlanmış sınıf diyagramı gösterilmiştir.



Şekil 4.5. Soyutlamayla oluşturulmuş sınıflar.

Renk soyut sınıfına tüm renklerde olması gereken yapıcı yöntem olarak renk oluşturucu tanımlandı. Yapıcı yöntem bir sınıfın nesne örnekleri oluşturulduğunda otomatik olarak çağrılan özel bir yöntemdir. Yapıcı, nesne oluşturulduğunda başlangıç değerlerini atamak ve nesnenin başlatılmasını sağlamak için kullanılır. Yapıcı yöntemler, sınıfın ismiyle aynı isme sahiptirler. Bu sayede bir sınıfta birden fazla yapıcı yöntem olabilir, ancak isimleri aynı olmalıdır. Yapıcı yöntemler, nesne örneği oluşturulduğunda otomatik olarak çağrılır ve herhangi bir çağrı komutuna ihtiyaç duyulmaz. Yapıcı yöntemler genellikle sınıfın özelliklerini tanımlamak için kullanılır. Nesne oluşturulduğunda yapıcı yöntem çalıştığından, sınıfın durumunu önceden tanımlanmış başlangıç değerleriyle ayarlamak için kullanılabilir. Ayrıca, yapıcı yöntemler parametre alabilir ve farklı başlangıç durumlarını destekleyebilir. Tez kapsamında yapıcı yöntemler, renk sınıflarından bir yapay zekâ oluşturulduğunda yapay zekaların duygularının veya renklerinin baskınlık derecesini ayarlamak için kullanıldı. Bu şekilde ekstra bir yöntem kullanmadan yapay zekalar daha oluşturulurken istenilen renk ve duygu özelliklerini sahip olması sağlandı. Şekil 4.6'da renk sınıfı içeriğindeki yapıcı metod gösterilmiştir.



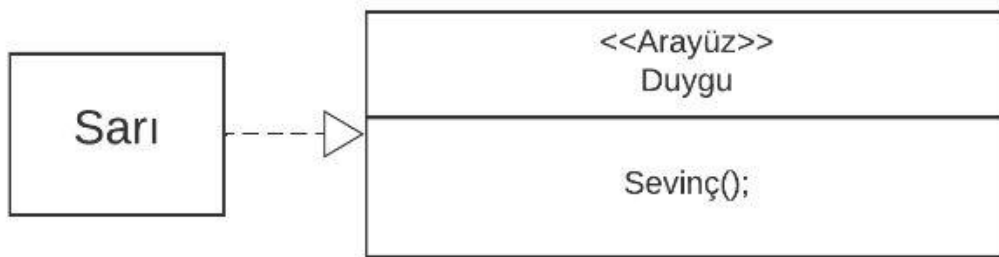
Şekil 4.6. Sınıf ve yapıcı yöntemi.

Bu yöntem sayesinde bu sınıfın özelliklerini taşıyan bir nesnenin otomatik olarak o sınıfın rengine dönüştürülmesi sağlandı.

Nesnelerin renklere uygun davranış göstermesi için iki farklı çözüm ortaya sürüldü. Bu çözümlerden ilki temel duyguları varsayılan olarak tanımlayıp, polimorfizm ile renk sınıflarından türetilen sınıflarda kalıtılan duygular üzerinde değişiklik yapabilmenin mümkün kılınması. İkinci çözüm yolu ise kalıtım alınacak renk sınıfını soyut sınıf kabul edip, bu sınıf üzerinden türetilen sınıflar üzerinde tanımlama işlemi yapmak. Birinci çözüm yolu hazır bir taslak oluşturmak için kullanıldığında daha elverişli bir çözüm olmaktadır. Geliştiriciyi duyguları tanımlama zahmetinden kurtarır ve kalıtılan sanal yöntemleri varsayılan olarak kullanmasını sağlar.

Geliştirici eğer isterse tanımlanan sanal yöntemleri ezerek kendi istediği şekilde kullanılabilir. Nesne yönelimli programlama dillerinde “sanal” anahtar kelimesi, bir sınıfta tanımlanan bir metodu, miras alan alt sınıflar tarafından ezilebilir hale getiren bir özelliktir. Bu şekilde, alt sınıflar, üst sınıfın metodunu kendi ihtiyaçlarına göre yeniden tanımlayabilir. Bu şekilde nesne yönelimli programlamanın dört ana prensibinden birisi olan çok biçimlilik (polimorfizm) sağlanmış olur. Polimorfizm aynı isimdeki yöntemlerin veya özelliklerin farklı şekillerde davranabilmesini sağlayan bir prensiptir. Bu şekilde, farklı sınıfların aynı metodu farklı şekillerde uygulayabilmesini ve çalışma zamanında doğru yöntemi çağırabilmesini sağlar. Bir üst sınıfta sanal olarak işaretlenen bir yöntem, alt sınıflar tarafından ezme anahtar kelimesiyle yeniden tanımlanabilir.

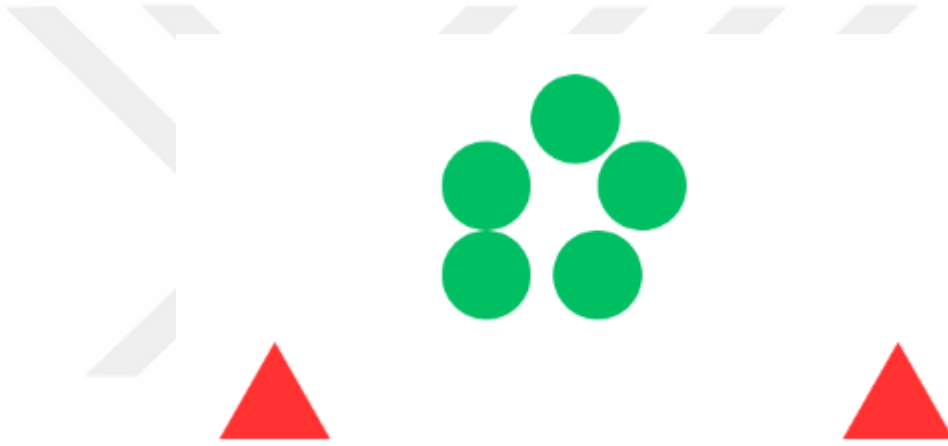
İkinci çözüm yolu için geliştirici türetilen her renk için duyguları kendi tanımlamalıdır. Bu durumda renk sınıfları istenildiği şekilde ve davranışta oluşturulabilir. Bu yöntem geliştiriciye büyük esneklik sağlayan bir yöntem olsa da hazır kalıp kullanılmadığı takdirde her renk için tanımlama yapmak oldukça uzun bir zaman almaktadır. Bu için çözüm olarak duyguların bir arayüz içinde tanımlanması gerekir. Tanımlanan arayüzle duygular renk sınıflarına eklenerek bu sınıfların arayüzün anlaşma koşulundaki duyguyu gerçekleştirme sağlanır. Tez çalışmasında kullanılan sarı rengin sevinç duygusunu temsil ettiği düşünülerek sarı sınıfının duygu arayüzünü sağlaması beklenir ve duygu arayüzünün koşulu olan sevinç yöntemini kendi içinde tanımlamasını zorunlu kılar. Bu tasarıma göre elde edilen diyagram Şekil 4.7’de gösterilmiştir.



Şekil 4.7. Arayüz eklenmiş sınıf.

Renkler nesnenin ne olduğunu belirtirken arayüzler de o nesnenin sahip olduğu özelliği belli edecek şekilde bir sistem oluşturulduktan sonra bu sistem üzerinden yeni ajanlar üretip simülasyon üzerinden sonuçlar toplanmaya başlandı.

Duygular ve renkler bağdaştırıldıktan sonra her renk için farklı simülasyon türleri oluşturuldu. Simülasyonlar gerçek hayattaki bazı canlıların davranışlarından esinlenerek canlandırıldı. İlk simülasyon örneği için vahşi doğadan bir örnek canlandırıldı. Örnekte avcılarının ve avın olduğu senaryoda simülasyonda gerçekleşen tepki ve duygular simüle edildi. Simülasyonda avcı ajanın kırmızı sınıftan kalıtım alması sağlanıp hiddet arayüzünün özellikleri tanımlandı. Av olan ajan için ise yeşil sınıftan kalıtım alması sağlanıp korku arayüzünün özellikleri tanımlandı. Simülasyonun taslak görüntüsü Şekil 4.8’de gösterilmiştir.



Şekil 4.8. Av ve avcı simülasyonu.

Simülasyonda avcılarının hızı belirlenirken avcıya en yakın av belirlenip ava doğru birim vektör, avcının duygu şiddeti ile temel avcı hızı çarpılarak elde edilen sonuç avcının hızı olarak belirlendi. Avların hızı belirlenirken kendisine belirli bir mesafede bulunan bütün avcılarının zıt yönünde olan birim vektörler toplanıp bu mesafede bulunan avcılarının sayısına bölündü ve elde edilen vektör, avın duygu şiddeti ve temel av hızıyla çarpılarak elde edilen sonuç avın hızı olarak belirlendi.

4.4. Duygu ve Renk Değişimi

Simülasyonda şu ana kadar sabit renk ve duygular kullanıldı ancak gerçek hayatta varlıkların duyguları buldukları durumlara ve diğer etkenlere bağlı olarak değişiklik gösterebilir. Bunun simüle edilmesi için bir önceki örnekten faydalanarak, avcılarının herhangi bir duyguya sahip olmadıkları durumda bir ava yaklaştıklarında duygu durumlarındaki renk değişimi kırmızı olarak gözlemlenir, aynı şekilde avlar kendilerine bir avcı yaklaştıklarında duygu durumlarındaki renk değişimi yeşil olarak gözlemlenir.

Yazılımsal olarak bunun gerçekleştirilmesi için başlangıçta herhangi bir duyguya sahip olmayan avcı ajanının bir ava yaklaşım tetikleyici olayını tetikledikten sonra bileşen ekle yöntemini kullanarak kırmızı komut dosyasının eklenmesi gerekir, av için de aynı şekilde tetikleyici olayının çalışması sağlanıp yeşil komut dosyasının av eklenmesi gerekir. Komut dosyası eklendiğinde sınıfın yapıcı yöntemi çalışıp nesne için duygu ve rengi ekler, tetikleyicinin çıkış olayı tetiklendiğinde ise bileşenin sil yöntemi kullanılarak renk komut dosyasının silinmesi gerekir. Renk komut dosyası silindiğinde sınıfın yıkıcı yönteminin çalışması sağlanıp nesne için renk ve duygunun silinmesi sağlanır. Bu işlem bir döngü halinde simülasyon devam ettiği sürece çalıştırılarak yapay zekaların birbirleriyle olan etkileşimi ve duygu değişimleri gözlemlenir. Renk değişiminin akış şeması Şekil 4.9'da gösterilmiştir.



Şekil 4.9. Renk değişimi akış şeması.

4.5. Duyguların Modellenmesi

Yapay zekanın insan duygularını tam olarak deneyimlemesi veya hissetmesi mümkün değildir. Duygusal deneyimler, canlıların karmaşık beyin yapıları, fizyolojik etkenler ve kişisel deneyimlerden kaynaklanır

Duygular modellenirken iki farklı kategoride incelendi. Bunlardan ilki olan temel duygular da kendi içinde sevinç, korku, üzüntü ve öfke olmak üzere dört farklı duygu olarak incelenmiştir. Bu duygular davranışlar üzerinde yoğun bir etkiye sahip olup karar verme ölçüsünü büyük oranda belirlemektedirler. Diğer duygular ise ikincil duygular olarak incelenmiştir. Bu duygular davranışlar üzerinde etkilidirler ancak temel duygular ikincil duyguların davranışlarını ve renklerini bastırabilir.

Duygular modellenirken şu unsurlar dikkate alınmıştır:

1. Duyguyu tetikleyen unsur nedir?
2. Hangi davranışlarla bu duygu modellenebilir?
3. Duygunun şiddet derecesi nedir?

4.5.1. Temel duygular

Dört temel duygunun ilki olan sevinç duygusunu tetikleyen unsurlar, farklı bir yapay zekâ, ortamda bulunan bir nesne ve habitat olarak kabul edildi. Uygulama için bir insanın kendisine sevinç veren bir insan, hayvan veya bitkinin yakınında bulunduğu gösterdiği duygu değişimi simülasyonda kullanıldı. Davranış modelleri içinse zıplama animasyonu eklendi, kendisine sevinç veren nesneye doğru yönelme hareketi ve sevinçli olduğunu belli eden sesler çıkarması sağlandı. Minimum duygu şiddetinde(sükûnet) ajanın sadece durumdan memnun olduğunu belli eden animasyonlar ve sesler kullanıldı. Maksimum duygu şiddetinde(coşkunluk) çevresindeki yapay zekalar için sevinç durumunun duygu şiddetini arttıracak bir kaynağa dönüşmesi sağlandı. Simülasyon modelindeki sevinç duygusu Şekil 4.10'da gösterilmiştir.



Şekil 4.10. Sevinç duygulu yapay zekâ.

Öfke duygusunu tetikleyen unsurlar, farklı bir yapay zekanın alanına girmesi, yaralanmaya sebep olacak bir durumun oluşması ve rahatsız edici seslerin olması olarak kabul edildi. Uygulama için bir hayvanın yaşam alanına başka bir hayvan girdiğinde gösterdiği duygu değişimi simülasyonda kullanıldı. Davranış modelleri için saldırı animasyonu eklendi, diğer yapay zekaya doğru kovalama hareketi sağlandı, bağırma ve kükreme gibi sesler eklendi.

Rahatsızlık durumunda kızgınlığı belli edecek sesler, hiddet durumunda ise hızını ve kuvvetini arttırarak hedefe doğru kovalama hareketi yaptırıldı.

Korku duygusunu tetikleyen unsurlar, yapay zekanın algılayamadığı veya netleştiremediği durumlar, tehlike unsuru olan bir nesneye ya da yapay zekaya yakın olunması ve anlık bir durumla karşı karşıya kalınması olarak kabul edildi. Uygulama için bir avın avcıdan kaçarken gösterdiği duygu değişimi simülasyonda kullanıldı. Davranış modelleri için titreme ve korku kaynağından kaçma hareketi sağlandı. Kaygı durumundayken bakış açısını tehlike kaynağını doğru çevirmesi, dehşet durumunda fiziksel kapasitesini arttırarak tehlike kaynağından uzaklaşma davranışını göstermesi sağlandı.

Üzüntü duygusunu tetikleyen unsurlar, yapay zekanın bağ kurduğu bir nesne veya yapay zekanın zarar görmesi, güven duyduğu bir yapay zekanın ona saldırması ve yalnızlık olarak kabul edildi. Uygulama için aile ilişkisi bulunan iki yapay zekadan birisinin ölmesi durumunda ve yapay zekanın yalnız kaldığı durumlarda gösterdiği duygu değişimi simülasyonda kullanıldı. Davranış modelleri için ağlama animasyonu ve ağlama sesleri kullanıldı. Dalgınlık durumunda fiziksel hareketinde yavaşlama, yerine getirmesi gereken normal davranışlarda eksiklik gösterme, keder durumunda şiddetli sesler çıkarma ve hastalanma gibi davranışlar göstermesi sağlandı. Üzüntü duygulu yapay zekâ Şekil 4.11’de gösterilmiştir.



Şekil 4.11. Üzüntü duygulu yapay zekâ.

4.5.2. İkincil duygular

İkincil duygulardan ilki olan güven duygusunu tetikleyen unsurlar, aile başına sahip olma, beslenme ihtiyacını karşılayan başka bir yapay zekâ, hastalığını tedavi eden bir yapay zekâ olarak kabul edildi. Uygulama için aile kendisini tedavi edip onu iyileştiren ve besleyen yapay zekanın bir evcil hayvanının ona karşı hissettiği duygu değişimi simülasyonda kullanıldı. Davranış modelleri için güven duyduğu varlığa yakın olma ve onu tehdit olarak gördüğü varlıklardan koruma davranışları sağlandı. Kabullenme durumunda kabullenilen varlığı tehdit olarak görmeme hayranlık durumunda ise hayranlık duyulan varlık ona zarar verse bile o varlığa güven duygusunu sürdürme davranışlarını göstermesi sağlandı.

Beklenti duygusunu tetikleyen unsur yapay zekanın ihtiyaç durumuna göre belirlendiğinden net bir şey söylemek zordur. Açlık durumunda olan bir yapay zekâ beslenme beklentisine, yalnız kalan bir yapay zekâ ise sosyalleşme beklentisine girebilir. Uygulama için yalnız kalan bir yapay zekanın, sosyalleşme ihtiyacının duygu değişimine olan etkisi simülasyonda kullanıldı.

Tiksinme duygusunu tetikleyen unsur yapay zekanın tetikleyici alanına giren herhangi bir nesne olabilir. Uygulama için yapay zekanın alanına tiksinme duygusunu tetikleyecek bir nesne bırakıldı ve hissettiği durumlardaki duygu değişimi simülasyonda kullanıldı.

4.6. Duyguların Birleşimi

İki farklı duygu birleşerek yeni bir duygu oluşturabilir. Duygular kompleks deneyimlerdir ve çeşitli faktörlerin etkileşimiyle ortaya çıkarlar. Yapay zekalar farklı duygusal durumlar içerisindeyken, bu duygular arasında geçişler ve kombinasyonlar olabilir.

Bir yapay zekâ aynı anda sevinç ve güven hissedebilir. Sevinç, pozitif bir duygudur ve genellikle mutluluk, neşe ve memnuniyetle ilişkilidir. Güven ise, birine veya bir şeye olan inanç, emniyet ve rahatlık hissiyle bağlantılıdır. Sevinç ve güven duyguları, aşkın inşa edildiği ve sürdürüldüğü temel duygulardır.

Bunun gibi daha birçok örnek verilebilir ama tez kapsamında duyguların birleşimi sekiz farklı birleşim olabilecek şekilde kabul edilmiştir. Bu sekiz duygu insani duygular olup Tablo 4.1’de gösterilmiştir.

Tablo 4.1. Duygu Birleşimi Tablosu.

Birinci Duygu	İkinci Duygu	Sonuç
Beklenti	Sevinç	Optimizm
Beklenti	Öfke	Agresiflik
Güven	Korku	Teslimiyet
Sevinç	Güven	Aşk
Korku	Şaşkınlık	Haşmet
Üzüntü	Şaşkınlık	Kınama
Üzüntü	Tiksinme	Pişmanlık
Tiksinme	Öfke	Küçümseme

Tablo incelendiğinde birleşim için kullanılan duyguların temel ve ikincil duygular olduğu görülür. Bu durumdan hareketle tablonun yapısı nesne yönelimli olarak incelendiğinde yapay zekaların duygu birleşimi modelleri yazılım ortamında tasarlanabilir.

Yapay zekâ bir duygu durumundayken yeni bir duygu durumu tetiklenirse önce bu iki duygu durumunun birleşimi olup olmadığı kontrol edilir, eğer duygu birleşimi yoksa baskın olan duygu varlığını korur. Duyguların birleşimi olma durumunda ise öncelikle var olan duygu durumu iptal edilir. Ardından yeni renk sınıfı yapay zekaya tanımlanır, iki renk sınıfı olan yapay zekanın duygu yöntemleri ezilerek yeni tanımlanan duygu yapay zekaya eklenir.



5. SONUÇ

Bu tez çalışmasında, Unity 3B oyun motorunda renkleri kullanarak kontrollü ve gözlemlenebilir yapay zekâ simülasyonu tasarlandı. Simülasyon tasarımı için kullanılan yapay zekâ ve renk kavramları hakkında araştırma yapıldı. Oyun motorunda kullanılan bileşenler hakkında bilgiler verildi ve bu bileşenler uygulama kısmında kullanıldı. Simülasyon tasarımı için nesne yönelimli programlama modeli seçildi ve nesne yönelimli programlamanın temel ilkelerinin kullanımından bahsedildi. Simülasyondaki yapay zekaların başlangıç durumunda renge ve duyguya sahip olarak daha sonrasında duyguların belirli koşullar altında kazanılarak simülasyonun çalışması sağlandı. Simülasyonun amacı doğrultusunda yapay zekaların duygu durumunu belli eden renkler yapay zekâ modellerinin üst kısmına yerleştirildi.

Simülasyon tasarımı ve amacı tamamlandıktan sonra, simülasyon ortamına farklı yapay zekalar yerleştirildi. Renk sahibi olmayan yapay zekâ belirli koşullar altında duyguların tetiklenmesiyle renk bileşenini eklemektedir. Renk bileşenini ekleyen yapay zekâ yeni duygular kazanıp bu duyguların tetikleyeceği davranışları otonom bir şekilde gerçekleştirmektedir. Yapay zekanın otonom bir hareket gerçekleştirmesi için yapay zekaya navigasyon ağı üzerinde ağ ajanı bileşeni eklendi. Tez kapsamında yapay zekalar duygularının şiddetine göre davranışsal özelliklerinde artış veya azalma olacak şekilde geliştirildi. Farklı renklere sahip yapay zekalar tetikleyici alanlarına girdiklerinde rengin gerektirdiği duyguyla bir tepki verecek, çıktığında ise renk şiddeti ve duygu şiddeti azalarak normale dönecektir. Ayrıca yapay zekâ üzerindeki renklerin birleşerek yeni bir duyguyu oluşturması sağlandı. Yapılan testler sonucunda yapay zekaların duygu şiddetlerinin fiziksel özelliklere olan etkisinin fazla olduğu görülmüştür.

Simülasyonda ortaya çıkan problemi çözmek amacıyla renk şiddetlerinin hızlıca arttırılması yerine uzun sürece yayılıp duygular animasyon ve ses çıkarma gibi davranışlarla desteklenmiştir. Renklerin bağlı olduğu duygular analiz edilerek bu duygular için uygun davranışlar animasyonlar ve seslerle desteklenerek yapay zekalara eklenmiştir. İki tane farklı simülasyon modeli sunulup ilk modelde renklerin sınıf

yapısı oluşturulmuş soyutlamalar sağlanmıştır. Yöntemler sanal olarak oluşturulup geliştiricinin ihtiyacı doğrultusunda yöntemlerin ezilmesine mümkün kılınmıştır. Temel ve ikinci duygular için arayüzler oluşturulup bu arayüzler renk sınıfıyla bağdaştırıldı rengin duygu şeklinin modellenmesi doğrultusunda arayüzlerin içinde bulunacak olan yöntemler belirlendi. İkinci simülasyon modelinde çalışmaya hazır olan yapay zeka simülasyonu renk sınıflarına uygun arayüzler eklenerek ve bu sınıflardaki yöntemler tasarlanarak oluşturuldu. Renklerin birleşiminde oluşacak durumlar ve yeni duygular tanımlandı ve yapay zekalara dahil edilerek simülasyon tamamlandı.

Sonuç olarak pek çok yapay zekanın bulunduğu simülasyonlardaki kontrol zorluğunun önüne geçilip erişimi ve kontrolü kolay, rahatça gözlemlenebilen, değiştirilmesi ve eklenti yapması kolay olan yapay zekalar elde edildi. Elde edilen yapay zekâ sistemi temel olarak kullanılıp duygu ve davranışların modellenmesi sağlandı ve birçok yapay zekanın farklı duygu ve davranışlara sahip olduğu simülasyonlarda kullanılabilinecek bir sistem elde edilmiş oldu.

KAYNAKLAR

- [1] Wikipedia (2023, 20, Mayıs). Zeka <https://en.wikipedia.org/wiki/Intelligence> adresinden 5 Haziran 2023 tarihinde alınmıştır.
- [2] Picard, R. W., Vyzas, E., & Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE transactions on pattern analysis and machine intelligence*, 23(10), 1175-1191.
- [3] Breazeal, C. (2003). Emotion and sociable humanoid robots. *International journal of human-computer studies*, 59(1-2), 119-155.
- [4] Winston, P. H. (2016). Marvin L. Minsky (1927–2016). *Nature*, 530(7590), 282-282.
- [5] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2), 100-107.
- [6] Reynolds, Craig (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. Association for Computing Machinery. pp. 25–34.
- [7] Tarhan, D. E. (2020). Goethe'de Renk Fenomeni. *FLSF Felsefe ve Sosyal Bilimler Dergisi*, (29), 219-246.
- [8] Birren, F. (1961). *Color psychology and color therapy: a factual study of the influence of color on human life*. New Hyde Park, N.Y., University Books.
- [9] Lüscher, Max. (1969). *The Lüscher color test*. New York :Random House.
- [10] Ekman, P. (2003). *Emotions revealed: Recognizing faces and feelings to improve communication and emotional life*. New York, NY: Times Books.
- [11] Barrett, L. F. (2017). *How Emotions Are Made: The Secret Life of the Brain*. Houghton Mifflin Harcourt.
- [12] Plutchik, R. (1980). *Theories of Emotion*. Academic Press.
- [13] Ekman, P. (2003). *Emotions revealed: Recognizing faces and feelings to improve communication and emotional life*. New York, NY: Times Books.
- [14] Izard, C. E. (2009). *The psychology of emotions*. New York, NY: Guilford Press.
- [15] Tomkins, S. S. (2008). *Affect imagery consciousness: The complete edition*. New York, NY: Springer Publishing Company.
- [16] Microsoft (2023, 5, Mayıs). Visual Studio. <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022> adresinden 10 Mayıs 2023 tarihinde alınmıştır.

[17] Unity Documentation (2023, 6, Haziran). Unity. <https://docs.unity3d.com/Manual/index.html> adresinden 9 Haziran 2023 tarihinde alınmıştır.



ÖZGEÇMİŞ

Ad-Soyad : Emre Can ARSLAN

ÖĞRENİM DURUMU:

- **Lisans** : 2019, Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİM:

- 2022 yılı içinde Boom Games Yazılım ve Ticaret Anonim Şirketinde yazılım geliştirici olarak çalıştı.
- 2022-2023 yılları arasında Solvia Yazılım ve Danışmanlık Anonim Şirketinde yazılım geliştiricisi olarak çalıştı.