Daha İnandırıcı Oyun Karakterleri İçin Bayes ve Q-Learning Tabanlı Yaklaşım

A Bayesian Q-Learning Based Approach to Improve Believability of FPS Game Agents

OSMAN YILMAZ

Dr. Öğr. Üyesi UFUK ÇELİKCAN
Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2018

OSMAN YILMAZ 'ın hazırladığı "Daha İnandırıcı Oyun Karakteri İçin Bayes ve Q-Learning Tabanlı Yaklaşım" adlı bu çalışma aşağıdaki jüri tarafından BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI YÜKSEK LİSANS TEZİ olarak Kabul edilmiştir.

Prof. Dr. Haşmet GÜRÇAY
Başkan .....

Dr. Öğr. Üyesi Ufuk ÇELİKCAN
Danışman .....

Prof. Dr. Tolga K. ÇAPIN
Üye .....

Dr. Öğr. Üyesi Zümra KAVAFOĞLU
Üye .....

Dr. Öğr. Üyesi Serdar ARITAN
Üye .....

Bu tez Hacettepe Üniversitesi Fen Bilimler Enstitüsü tarafından YÜKSEK LİSANS TEZİ olarak onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimler Enstitü Müdürü

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin / raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan **" Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi / H. Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- o Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir. (1)
- o Enstitü / Fakülte yönetim kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren .... Ay ertelenmiştir. (2)
- o Tezimle ilgili gizlilik kararı verilmiştir. (3)

05 /09 /2018

(İmza)

Öğrencinin Adı SOYADI

Osman YILMAZ

"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"

(1) Madde 6. 1. Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir

(2) Madde 6. 2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. Şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü ve fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

(3) Madde 7. 1. Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir*. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.
Madde 7. 2. Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

* Tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu tarafından karar verilir.

Aileme ve arkadaşlarıma

ETİK

Hacettepe Üniversitesi Fen Bilimler Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğim,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumundan ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

05/09/2018

Osman YILMAZ

ÖZET

Daha İnandırıcı Oyun Karakteri İçin Bayes ve Q-Learning Tabanlı Yaklaşım

Osman YILMAZ

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Dr. Öğr. Üyesi Ufuk Çelikcan

Eylül 2018, 69 sayfa

Oyun programlamada ulaşılmak istenen hedeflerden biri de gerçek hayatta yer alan kavramları ve karakterleri oyunlara uyarlamaktır. Bu yaklaşım, daha ilgi çekici hareketler sergileyen oyun karakterleri sunmak için benimsenmektedir. En yüksek ödül mantığını ele alan yöntemler oyun karakterinin aynı örüntüleri sergilemesine ve tekrara düşmesine sebep olur. Aynı zamanda bu durum oyunun oynanabilirliğini azaltır. Bu tür tekrarlayıcı kalıpları önlemek için, Naïve Bayes ile Q-öğrenme yaklaşımına dayalı bir davranış algoritması geliştirilmiştir. Geliştirilen algoritmanın geçerliliği kullanıcı testleri ile karşılaştırmalı olarak ortaya konulmuştur. Bu testler sonucunda, algoritmanın öğrenmede kullandığı oyun verisi miktarı arttıkça davranış öğrenme algoritmasının daha iyi bir performans gösterdiği ve oyun karakterinin daha ilgi çekici hale geldiği görülmüştür.

Anahtar kelimeler: Naive Bayes, Q-Learning, Yapay Zekâ Karakteri, İlgi Çekici Davranış, Pekiştirmeli Öğrenme, Davranış Seçimi.

ABSTRACT

A Bayesian Q-Learning Based Approach to Improve Believability of FPS Game Agents

Osman YILMAZ

Master of Science, Department of Computer Engineering

Supervisor: Dr. Öğr. Üyesi Ufuk ÇELiKCAN

September 2018, 69 pages

One of the goals of modern game programming is adapting the life-like characteristics and concepts into games. This approach is adopted to offer game agents that exhibit more engaging behavior. Methods that prioritize reward maximization cause the game agent to go into same patterns and lead to repetitive gaming experience, as well as reduced playability. In order to prevent such repetitive patterns, we explore a behavior algorithm based on Q-learning with a Naïve Bayes approach. The algorithm is validated in a formal user study in contrast to a benchmark. The results of the study demonstrate that the algorithm outperforms the benchmark and the game agent becomes more engaging as the amount of gameplay veri, from which the algorithm learns, increases.

Key words: Naive Bayes, Q-Learning, Game Agent, Engaging Behaviour, Reinforcement Learning, State Selection.

TEŞEKKÜR

Tez çalışma süreci boyunca sürekli yanımda olan ve desteklerini hiç eksik etmeye aileme en içten teşekkürlerimi sunarım. Araştırma, geliştirme ve yazım sürecinde yapılan çalışmalarda gerekli yönlendirmeyi ve desteği sağlayan tez danışman hocama teşekkürü bir borç bilirim. Ayrıca bu yoğun süreçte gerekli anlayış ve yardımı gösteren iş arkadaşlarıma ve özellikle yakın çevremde yer alan özverisiyle, zekâsıyla, güveniyle ellerinden gelen hiçbir şeyi eksik etmeyen dostlarıma sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÇİZELGELER

Sayfa

ŞEKİLLER

Sayfa

SIMGELER VE KISALTMALAR

Kısıtlamalar

YZK	Yapay Zekâ Karakteri
MKS	Markov Karar Süreci
NV	Naïve Bayes
PÖ	Pekiştirmeli Öğrenme
QValue	Q Tablo Değeri
D	Durum (State)
E	Eylem (Action)
DE	Durum – Eylem Çifti
FPS	First Person Shooter
S_t	Bulunduğu Durum
S_{t+1}	Sonraki Durum
SD	Sağlık Değeri
RU	Rakip Uzaklığı
HD	Hız Değeri
AG	Atak Gücü
KZ	Kalan Zaman
QNBY	Q Öğrenme ve Naive Bayes Yaklaşımı
ABDY	Açgözlü Benzeri Davranış Yaklaşımı
KDİ	Karakterin Davranış İnandırıcılığı
OZ	Oyunun Zorluğu
KÖZ	Karakteri Öldürme Zorluğu
OOS	Oyunun Oynanabilirlik Seviyesi

1. GİRİŞ

1.1.Genel Bakış

Gelişen makine öğrenme teknolojisi diğer birçok sektöre olduğu gibi oyun sektörüne de katkıda bulunmuştur. Birçok makine öğrenme algoritması oyunlarda da kullanılmaya başlanmıştır. Bu algoritmaların oyunlarda uygulandığı alanlardan biri de yapay zekâ karakterleridir. Oyun geliştiricisi üreteceği yapay zekâ karakterinin hem yapması gereken işlevleri en ince ayrıntısı ile tanımlamakta hem de bu oyunu oynayacak kişilerin ilgisini ve heyecanını canlı tutmaya çalışmaktadır. Oyunlarda kullanıcının ilgisini çekecek seviyede yapay zekâ karakterleri tanımlamak emek isteyen bir iştir. Üretilen karakterlerin zorluk derecesi veya yaptığı hareketlerin mantık seviyesi oyuncunun bir anda oyundan soğumasına sebep olabilmektedir. Sürekli tekrara düşen ve aynı hareketleri tekrar eden yapay zekâ karakterleri, oyuncunun oyuna karşı ilgisini azaltmaktadır. Bu yüzden daha zeki ve oyuncuyu şaşırtan sonuçlar sergilemesi hedeflenmektedir. Bunun için de insan rol model olarak kullanılmaya çalışılmıştır. Üretilecek yapay zekâ karakterleri için insanı taklit edebilir mi, insansı hareketler sergileyebilir mi veya insan gibi kararlar alabilir mi gibi sorular üzerinde araştırmalar yapılmıştır. Bilinen bazı yöntemler kullanılarak yetenekleri ve hareketleri sınırlı karakterler üretilmektedir. Karakterleri yönetmek için sonlu durum makinaları ve kural tabanlı yöntemler kullanılmaktadır. Kural tabanlı yöntemlerde tüm kurallar koda dökülmektedir. Bu yöntemlerin geliştirici acısından birçok zorluğu yer almaktadır. Geliştirici üreteceği karakterin yeteneklerini ve davranışlarını içeren gereksinimleri tüm detayı ile çıkartması gerekmektedir. Hem bu gereksinimlerin çıkartılması hem de bunların koda dökülmesi zaman almaktadır. Oyuncu açısından da bu yöntemlerle üretilen karakter belli bir süre sonra tahmin edilebilir seviyeye ulaşmakta ve oyuncunun kolayca karakterin ne yapacağını öngörebilmesine sebep olmaktadır.

Daha iyi sonuçlar elde etmek için öğrenme teknikleri kullanılmaya başlanmıştır. Bu şekilde oyun içinde daha zekice hareket eden karakterler oluşturulması hedeflenmiştir. Yapay zekâ karakterinin gelişimi ve karakterin oyun içerisinde daha başarılı olması için birçok makine öğrenme tekniği oyun programlamada kullanılmaktadır. Bunlardan bazıları denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenmedir. Denetimli öğrenmede önceden işaretlenmiş veri kümesi bulunmaktadır ve bu veri kümesi üzerinden o anda bulunan girdiye karşılık gelen

çıktı bilgisine ulaşılmaktadır. Denetimsiz öğrenmede veri kümesi önceden işaretlenmemiştir. Kümeleme işlemi yapılarak bu veri üzerinden bilgiler çıkarılmaya çalışılmaktadır. Pekiştirmeli öğrenmede ise herhangi bir veri kümesi bulunmamaktadır. Karakterin yaptığı davranışlara ceza veya ödül verilerek karakterin öğrenme işlemi gerçekleştirilmektedir [1]. Alınan ödül veya cezalara göre karaktere yapmış olduğu seçim işleminin doğru veya yanlış olduğu bilgisi verilmektedir. Bu şekilde bir sonraki hamlelerini daha etkili bir şekilde yapması hedeflenmektedir. Öğrenme işlemi gerçekleştirilirken dikkate alınması gereken bazı kısıtlar bulunmaktadır. Bunlardan ilki oluşturulan öğrenme modelinin karmaşıklık seviyesidir. Modelin karmaşıklığı arttıkça yapılan hesaplamalar artacaktır ve yapılan işlemler daha karmaşık hale gelecektir. Bununla birlikte kullanılan veri kümesinin büyüklüğü ve çeşitliliği, öğrenme hızına etki edeceği için ayrı bir öneme sahiptir. Pekiştirmeli öğrenme gibi makine öğrenme tekniklerinde ayrıntılı bir model kullanılması halinde öğrenme işlemi zorlaşacaktır ve aynı zamanda işlemlerde kullanılacak olan parametrelerin ayarlanması zaman alacaktır [2].

1.2.Tezin Amacı ve Kapsamı

Bu tez çalışmasında oyunlarda yer alan yapay zekâ karakterinin çevresel faktörlere göre mantıklı karar vermesi ve oyun içerisinde doğal, inandırıcı ve ilgi çekici bir davranış sergilemesi öğretilecektir. Oyun esnasında Q-öğrenme ve Naive Bayes yaklaşımları kullanılarak karakterin sahip olduğu teknik özellikler ve çevresel faktörler yardımıyla davranış seçim işlemi gerçekleştirilecektir. Bu seçimin daha etkili ve sonuç odaklı olması için yapay zekâ karakterine nasıl davranması gerektiği ve hangi durumlara gitmesi gerektiği öğretilecektir. Bu öğrenme işlemi esnasında karakterin geçmişte sahip olduğu verilere karşı sergilediği tutum kullanılacaktır. Ele alınan veri kümesi, karakterin oyun esnasında sahip olduğu teknik özellikleri, çevresel bilgileri ve bu verilere sahipken hangi davranış durumunu seçtiği bilgisini içermektedir. Bu veri kümesinde yer alan bilgi aslında karakterin geçmişte oyuncu ile karşılaştığında nasıl tepki göstermiş olduğunun bilgisini içermektedir [3].

1.3.Tez Planı

1. Bölüm tez ile ilgili giriş kısımlarından ve problemin tanımından oluşmaktadır. 2. Bölüm tez ile ilgili taranan kaynakların özetini içermektedir. 3. Bölüm tezde

kullanılan yaklaşımların tanım ve detayını içermektedir. 4. Bölüm tezde bahsedilen yöntemin nasıl uygulanacağını ve hangi algoritmaların kullanılacağını göstermektedir. 5. Bölüm farklı insanlar kullanılarak yapılan deney çalışmalarını ve elde edilen verileri içermektedir. 6. Bölüm ise tez ile ilgili varılan sonucu anlatmaktadır.

1.4.Bilim ve Teknolojiye Faydası

Yapay zekâ karakterinin daha insansı ve mantıklı hareket sergilemesi için yapılmış olan bu tezde birden fazla bilimsel yaklaşım birleştirilerek kullanılmış ve olumlu sonuçlar gözlemlenmiştir. Bu yaklaşımlar birleştirilmesiyle sadece tek tür oyunlarda veya tek tip oyun karakterine değil farklı türlerde oyunlarda da uygulanması amacıyla oluşturulmuştur. Belirli durum uzayı olan ve bu durum uzayı içinde hareketini sağlayan her türlü oyun karakterine uygulanabilir yapıda tasarlanmıştır. Sadece oyunlarda değil aynı zamanda pekiştirmeli öğrenmenin uygulanabildiği diğer alanlarda da küçük eklemelerle kullanılabilir. Örneğin, bir robota bir alışkanlık kazandırmak veya yeni bir şeyi öğretmek için bu yaklaşımdan faydalanılabilir.

1.5.Problemin Tanımı

YZK'nin oyun içerisinde hedefine ulaşması ve normal bir şekilde hareketini devam ettirmesi, durumlar arasında eylemleri kullanarak bir durumdan başka bir duruma geçmesiyle gerçekleşmektedir. Başlangıçta rastgele atanmış bir durumdan hareketine başlayan YZK, daha sonra diğer durumlardan birine geçiş yapması gerekmektedir. Ancak durum sayısının artmasıyla bu seçim işlemi giderek zor bir hal alır ve bu seçim işlemini bir mantığa veya politikaya bağlı gerçekleştirmesi gerekmektedir. Bulunduğu her durumda diğer n durumdan biri arasında seçim yapmakta ve bu işlemi hedefine ulaşana kadar n defa tekrar etmektedir. Bu problem n-kollu kumar makinası(n-armed bandit) problemiyle benzerlik göstermektedir. Başka bir deyişle sıralı karar verme problemi olarak da ifade edilebilmektedir.

1.6. N-Kollu Slot Makinası Problemi

Bu problem ismini oyun salonlarında yer alan slot makinesinden almaktadır. Sıradan slot makinelerinin sadece bir kolu vardır. Bu problemde slot makinelerinin birden fazla kolu olduğu kabul edilmektedir. Her biri, birbirinden farklı getiri sağlayan kollardan oluşmaktadır [17]. Hangi kolun size ne kadar kazandıracağını

veya ortalama ne getiriyi verdiğini bilmemekle beraber sadece çeşitli kollar ile oynayarak hangi kolun en iyi olduğu hakkında bilgi alınabilmektedir. Burada bilgi almak için kullandığınız her bir kol sonucu gözlemleriniz sizin için ödül anlamına gelmektedir. Çünkü bu gözlemleriniz bir sonraki seçimlerinizde vereceğiniz kararları etkilemektedir. Bu ödüller ve kazanımlar, kollar arasında bir bağ kurarak ve bir denge oluşturarak daha iyi sonuca ulaşmanızı sağlamaktadır. Sonuca ulaşmak için n adım bulunmaktadır ve her bir adımda n koldan biri seçilerek ilerlenmektedir [18].

1.7.Öğrenme

Öğrenme işleminin oyunlarda nerelerden ve nasıl ilham alındığı konusunu tam olarak kavrayabilmek için biraz temele inmek ve diğer canlılarda özellikle insanda nasıl olduğu konusuna biraz değinmek gerekir. İnsanda öğrenme güdüsü, anne karnından hatta kimi kaynaklara göre daha öncesine yani genetik mirasa kadar dayanan bir yapıya sahiptir. Farklı insanların farklı spor dallarını kolaylıkla öğrenmesi o kişinin genetik koduyla ilişkili olduğu varsayılmaktadır. Atalarının başarılı olduğu spor etkinliklerinde çocuğunda başarılı olması buna bağlanmaktadır. Aynı şekilde duyu organlarının anne karnında gelişmesiyle birlikte aile bireylerinin davranışı çocuğun anne karnındaki öğrenme sürecini etkilemektedir. Çocuk doğduktan itibaren ise tamamen taklit etme güdüsü ile harekete geçer ve çevresindeki canlılardan bir şeyler kapmaya başlar. Bir yandan büyümeye devam eden bebek diğer yandan bütün duyu organları açık bir şekilde çevresini taramaya başlar. Annesinin yemek yapmasını verilen mamayı parmaklarıyla küçük parçalara ayırarak taklit eden bebek, diğer yandan babasının işe gidişini yerde bulduğu çantayı taşımaya çalışarak taklit eder. Taklit yeteneği sadece bununla sınırlı değildir. Evde yaşayan evcil hayvanları bile taklit ederek onların yaptığı hareketleri yapmaya çalışır. Bazı aileler bu taklit öğrenmeden özellikle faydalanarak çocuklarının fiziksel ve psikolojik gelişimine önemli katkıda bulunurlar [4]. Bunu sadece çocuklarının kendilerini taklit etmesiyle değil çeşitli oyunlar oynatarak, videolar izleterek hatta televizyondaki bazı öğretici programları izleterek sağlarlar. Çevresiyle daima bir alış veriş içerisinde olan bebek için o an verilen her bilgi değerlidir ve hızlı bir şekilde kaydeder. Bebeğin daha sonra harekete geçireceği bütün davranışları bu kaydettiği bilgiler doğrultusunda gerçekleşir. Davranışların geçmiş öğrenmeleriyle ilişkili olduğu varsayılır bu

yüzden. Dolayısıyla öğrenmeyi etkileyen faktörlerin çoğu insan davranışını dolaylı yoldan etkiler. Öğrenmenin gerçekleştiği çevresel etkenler öğrenmeye direkt etki ederken ileride kazanılacak olan davranış ve alışkanlıklarında etkilemiş olur [5]. Çevresel etkenler öğrenenin sahip olduğu koşullar, öğretenin sahip olduğu koşullar ve bulunulan çevre koşullarını içerir. Örneğin, öğrenen kişinin yaşı, fiziksel özellikleri nasıl ki öğrenmeye etki ediyorsa aynı şekilde öğreten kişinin fiziksel özellikleri ve ortamın sahip olduğu mevsimsel özellikler de aynı şekilde öğrenme üzerinde çeşitli etkiye sahiptir [6].

Öğrenme eylemi insan gelişimi için yeterince öneme sahip olmakla birlikte günümüzde başka canlıların hatta yapay oluşumların eğitiminde de sıkça kullanılmaktadır. İnsanların yaşamını kolaylaştırması için çeşitli canlılar eğitilerek çeşitli görevlerde kullanılır. Bunların başında köpekler gelmektedir. Köpekler özellikler insan güvenliği açısından çeşitli öğrenme teknikleri kullanılarak farklı birçok eğitime tabi tutulurlar. Bu öğrenme tekniklerinin başında ödül-ceza sistemi gelmektedir. Yaptığı hareketin doğru bir hareket olduğunu ona öğretmek için ödül olarak yemek verilir veya ceza olarak hiçbir şey vermeyerek yaptığı davranışın yanlış olduğu ve bir daha yapmaması gerektiği öğretilir. Köpeklerin eğitiminde kullanılan bu ödüllendirme ve cezalandırma sistemi aynı şekilde sentetik yani yapay oluşumlar için de kullanılır [7]. Bu yapay oluşumlardan kasıt herhangi bir sektörde kullanılan yapay zekâ içeren robotlar veya çeşitli oyun ve videolarda bulunan yapay zekâ içeren karakterlerdir. Bu sentetik oluşumların geliştirilmesindeki amaç yine insanların hayatını kolaylaştırmak olduğu için temelde örnek alınan öğreticinin insan olması hedeflenir. Yapay oluşumların gerçekleştirebileceği şeylerin hayal gücü insan zekâsıyla ölçülür ve rol model olarak da genellikle insan ele alınır [8].

1.8. Makine Öğrenmesi ve Çeşitleri

Makine Öğrenmesi, elde edilen verilerden yola çıkarak yeni şeyler öğretilmesi için kullanılan yöntemlerin tamamına verilen ad olarak tanımlanabilir. Gözetimli, gözetimsiz, yarı gözetimli ve pekiştirmeli öğrenme gibi çeşitleri bulunmaktadır. Gözetimli öğrenme daha önce elde edilen veri içerisinde yer alan bazı parametreler arasında bağlantı kurarak yeni bir model oluşturma mantığına dayanır. Gözetimli denmesinin sebebi ise makinenin öğrenmesi gereken sonuç bilgileri de veri ile birlikte verilir ve makine bu veriler doğrultusunda öğrenme

işlemini gerçekleştirir. Gözetimli öğrenme sınıflandırma ve regresyon olarak iki farklı kategori de ele alınabilir. Sınıflandırmada, verilen girdi ve sonuç setinde yer alan bilgiler göz önünde bulundurularak gruplandırma işlemi yapılır. Örneğin, motorlu araç kullanıcı kitlesinin belirli özelliklerini göz önünde bulundurarak kullandıkları aracın markasını belirlenebilir. Regresyon ise elde bulunan verilerden yola çıkarak en yakın tahmini sonuca ulaşmaktır. Örneğin, uzaklık, fiyat ve gidilecek yer bilgisi elimizde olan bir uçuş veri setinde farklı bir uzaklıktaki yerin uçuş fiyatının yaklaşık değerini regresyon yöntemiyle elde edebilir. Gözetimsiz öğrenme de aslında tıpkı gözetimli öğrenme gibi daha önce elde edilen veri içerisinde yer alan özellikler arasında ilişki kurarak yeni bir model oluşturma mantığına dayanır. Ancak burada sonuç verisi, girdi verisi ile birlikte verilmez. Herhangi bir sonuç verisi ve işaretlenmiş bir veri olmadığı için burada öğrenme, var olan giriş verileri arasında daha küçük kümeler oluşturma işlemiyle gerçekleşir. Yani herhangi bir işaretleme işlemi yapılmadan kümeleme teknikleri yardımıyla model oluşturularak öğrenme işlemi gerçekleştirilir. Yarı-Gözetimli öğrenme hem işaretlenmiş hem de işaretlenmemiş veri setlerini kullanarak öğrenme işlemini gerçekleştirir. Veri setinin çeşitliliğinden doğru sonuca ulaşma ihtimali artar. İşaretlenmemiş verilere ulaşmak işaretlenmiş verilere ulaşamaya kıyasla daha kolaydır. İnternetteki birçok siteyi tarayarak birçok veri elde edilebilir. Yarı-Gözetimli öğrenmede işaretlenmemiş büyük verilere ek olarak işaretlenmiş küçük veri setiyle birleştirilerek öğrenme işlemi daha etkin bir şekilde yapılabilmektedir [9].

1.9. Pekiştirmeli Öğrenme

Makine öğrenmesi yaklaşımlarından biri olan pekiştirmeli öğrenme, uygulanan yapının bulunduğu çevre ile etkileşime geçerek ödül sistemi yardımıyla nasıl hareket etmesi gerektiğini öğrenmesi mantığına dayanmaktadır [10]. Pekiştirmeli öğrenme uygulama alanı bakımından çeşitlilik gösterir. Pekiştirmeli öğrenme, robot kontrolü ve basit tahta oyunlarına uygulanmasıyla birlikte durum sayısı fazla olan oyunlarda da diğer öğrenme yöntemleriyle ile birlikte kullanılmaya başlanmıştır. Pekiştirmeli öğrenmenin güncel uygulama alanlarından bazıları aşağıdaki gibidir:

1.Tahta oyunları

Genellikle mevcut pozisyonuna dayanarak bir sonraki hamleyi öğrenmek için pekiştirmeli öğrenme kullanılmaktadır.

2. Bilgisayar oyunları

Atari oyunları dâhil birçok oyun türünde uygulanmaktadır.

3. Robot kontrolü

Robotlara yönelmeyi, eğilmeyi, el hareketleri yapmayı, kafa sallamayı, oyun oynamayı veya daha karmaşık işlevleri pekiştirmeli öğrenme yardımıyla öğretilmektedir.

4. Reklamcılık

Uygulamalarda bir kullanıcıya doğru zamanda gösterilecek bir reklamı seçmek için pekiştirmeli öğrenme kullanılmaktadır [11].

Pekiştirmeli öğrenme, oyun sektöründe yapay zekâ karakterinin yönlendirilmesinde de kullanılmaktadır. Sadece tek bir karakterin yönlendirilmesi değil aynı zamanda sürü olarak ifade edilen çok karakterli sistemlerde de kullanılmaktadır. Bütün bunların yanında, kontrol sistemleri mühendisliğinde kontrol teorisi ile birlikte pekiştirmeli öğrenmeden faydalanılır. Sürekli çalışan dinamik sistemlerin kontrolünde, bir kontrol eylemini, geciktirme veya aşma olmadan uygun değer olarak kontrol etmek ve kontrol dengesini sağlamak için kullanılmaktadır [12]. Robotların gelişimine ve yeni şeyler öğrenmesine kolaylaştırmak için pekiştirmeli öğrenmeden yararlanılmaktadır. Çevre ile etkileşim yoluyla en uygun davranışı keşfetmesi sağlanmakta ve bu şekilde robota etkili bir öğrenme işlevi kazandırılmaktadır [13]. Pekiştirmeli öğrenmede amaç tıpkı robotlarda olduğu gibi YZK'nin çevre ile etkileşime geçmesidir. Bu yüzden çevre büyük bir öneme sahiptir. Bu bağlamda çevrenin matematiksel bir modeli oluşturulması gerekmektedir. Bu modeli anlamlı hale getirmek ve kolaylaştırmak için markov karar süreci(MKS) kullanılmaktadır. Bazı makine öğrenmesi algoritmalarında dinamik programlama teknikleri bu tarz problemleri çözümlemek için kullanılırken, pekiştirmeli öğrenmede MKS kullanılmaktadır. Klasik dinamik programlama yöntemleri ile pekiştirmeli öğrenme algoritmaları arasındaki temel fark, pekiştirmeli öğrenmede matematiksel modelinin zor olduğu ve klasik yöntemlerin mümkün olmadığı büyük problemler ele alınmaktadır [14].

1.10. Markov Karar Süreçleri

Markov karar süreçleri çıktıların kısmen rastgele olduğu ve kısmen bir karar verenin kontrolünde bulunduğu durumlarda karar verme modellemesi için matematiksel bir çerçeve oluşturmaktadır. Her bir t anında, süreç birden fazla durumda olabilmektedir ve karar verici bulunduğu duruma bağlı herhangi bir eylemi seçebilmektedir. Markov karar süreçleri markov zincirlerine benzemektedir. Aralarındaki fark eylemlerin ve ödüllerin eklenmesidir. Eğer her bir durum için sadece bir eylem varsa ve tüm ödüller aynıysa markov zinciri olarak ele alınabilir. Markov karar süreçleri tanımlama grupları ile ifade edilir. Bu problemin veya çevrenin durumuna göre değişebilir. Basit olarak bir markov karar süreci aşağıdaki gibi 4-tuple ile ifade edilebilir.

$$(\mathrm{S}, \mathrm{A}, \mathrm{P_a}(\mathrm{s}, \mathrm{s}'), \mathrm{R_a}(\mathrm{s}, \mathrm{s}'))$$

S: "State" kelimesinin ilk harfini ifade eder ve durum anlamına gelir.

A: "Action" kelimesinin ilk harfini ifade eder ve eylem anlamına gelir.

$\mathrm{P_a}(\mathrm{s}, \mathrm{s}') = \mathrm{P}(\mathrm{s_{t+1}} = \mathrm{s}' | \mathrm{s_t} = \mathrm{s}, \mathrm{a_t} = \mathrm{a})$: Bir t anındaki s durumu ile t anındaki a eylemi bilinirken t+1 anındaki s_{t+1} gelme olasılığı.

$\mathrm{R_a}(\mathrm{s}, \mathrm{s}')$: s durumundan s' durumuna a eylemini kullanarak geçerken almış olunan anlık ödül [15].

MKS'lerdeki temel zorluk, durumlar arası eylemleri kullanarak yapılan geçiş işleminde karar veren bir politika belirlemektir. Bir karar süreci bu şekilde bir politika ile birleştirildiğinde, çeşitli problemler için uygulanabilmektedir. Örneğin, seçilecek olan bir politika ile maksimum ödül değeri ele alınarak karar işlemi gerçekleştirilmektedir.

2. LİTERATÜR ÖZETİ

Geçmişten günümüze yapay zekâ karakterinin gelişimi için çeşitli araştırmalar yapılmıştır. YZK'nin durumlar arasındaki geçişi, gideceği yolu bulması, ne zaman saldırması ve ne zaman hangi yöne gitmesi gerektiği gibi birçok yönlendirme işlemi kural tabanlı olarak yapılabilmektedir. YZK'nin yapması gereken hareketler oyun başlangıcında kurallar belirlenerek oluşturulmaktadır. Gelişen oyun teknolojisi ve büyüyen oyun pazarında daha çok insana hitap edebilmek için daha gerçekçi oyunlar yapma eğilimi ortaya çıkmıştır. Oyunlarda yer alan karakterlere gerçekçilik kazandırmak kural tabanlı sistemlerle hiç de o kadar kolay olmadığı görülmüştür. Çünkü gerçekçi davranış olayı çok karmaşık kodlar gerektiriyordu ve oyunlardaki hız faktörü gibi etkenlerinde göz önünde bulundurulması gerekiyordu. Sadece bunlarla kalmayıp parametrelerin belirlenmesi ve hesaplama zamanlamasını da dikkate alması gerekiyordu. Bütün bu zorluklar insanları yeni teknikleri ve yaklaşımları araştırmaya ve kullanmaya yöneltmiştir. Kural tabanlı yönlendirmeler ile birlikte durum makinaları kullanılmaya başlanmıştır. Karakterin geçiş yapması gereken durumları sonlu durum makinelerinde tutarak YZK'nin hareketi sağlanmıştır. Örneğin, kılıç taşıyan bir savaşçının yapması gereken hareketler basit bir sonlu durum makinesinde tutularak yönlendirme işlemi ile gerçekleştirilmiştir. Sağlık durumuna göre bekleme, atağa geçme ve kendini koruma gibi durumlar arasında geçişler yaparak rakibiyle savaşmaktadır. Hatta rakibini kılıç darbesiyle öldürdükten sonra sevinç gösterisi olarak dans etme durumuna geçmekte ve bu duruma atanmış olan dans animasyonunu sergilemektedir [19]. Sonlu Durum makinaları sayesinde YZK'nin kontrolü biraz daha kolaylaşmıştır. Bununla birlikte durum sayılarının fazla olduğu oyun türlerinde farklı teknikler denenmeye başlanmıştır. Ancak, yetenek sayısı arttıkça, durum makineleri çok iyi ölçeklenemediği ve bakımının zorlaştığı anlaşılmıştır. Bunun yanında karar ağaçları kullanılmaya başlanmıştır. Karar ağaçları iç içe geçmiş birden fazla kuralın formüle dönüştürülmesiyle oluşmaktadır. Popüler bir yöntem olan karar ağaçlarının uygulanması oldukça kolay ve hızlıdır. Ancak kuralların doğru ve düzgün bir şekilde oluşturulması için dikkatli bir başlangıç tasarımı gerektirmektedir. Davranış Ağaçları ise, bu eksikliğin giderilmesi ve yönetim sisteminde daha sonradan ilave edilen özelliklerinde kolay bir şekilde uyum sağlamasına izin vermektedir. Görev yönetiminin ölçeklenebilir olması için, diğer

bir deyişle, artan karmaşıklıkları izlenebilir kılmak için tasarlanan sistemin esnek olması ve uygulama sırasında istenilen özelliklerin kolayca değiştirilebilir olması gerekmektedir. Özellikle sistemin modüler olması beklenir ve sisteme ait bir bölümdeki değişiklikler sistemin farklı bir bölümünü etkilememelidir. Davranış ağaçları, ağaç yapılarındaki geçişleri sistemin diğer kısımlarını fazla etkilemeden sağlar. Davranış ağaçlarına düğümler eklemek veya çıkarmak, tek bir bağlantının değiştirilmesiyle sağlanabilirken, sonlu durum makinelerinde geçişleri yeniden bağlamak yorucu bir iş olabilmektedir. Davranış ağaçları, hiyerarşik sonlu durum makinalarına benzer şekilde hiyerarşik bir yapı üzerine inşa edilir. Hiyerarşik sonlu durum makinalarında durumlar arasındaki geçişleri ayrık olaylar tarafından tetiklenirken, davranış ağaçlarında devam eden durumlar arasındaki geçişler etkin veya devre dışı bırakılmış görevler üzerine kurulur [20]. Davranış ağaçları video oyunlarındaki davranışları kontrol etmek için uygulanmaktadır. Her ağaç, alt davranışlardan oluşan bir davranıştır. Yapay zekâ oyunu açısından bunun en önemli avantajı, programcı olmayanların da her davranışın açık semantiğini anlayabilmeleri ve kullanabilmeleridir. Çünkü davranış ağaçlarının uygun bir grafik biçiminde gösterilmesi mümkündür. Mevcut davranışlardan yeni davranışlar oluşturmak istenildiğinde bu grafik üzerinde gerekli çalışmalar yapılarak bu işlem yapılabilmektedir. Oyunlarda karakter davranış tasarımında kullanılan diğer bir yaklaşımda genetik algoritmasıdır. Oyunları ve üretilen karakterleri geliştirmek için genetik programlama kullanılmaktadır [21]. Bununla birlikte YZK‘nin yönlendirmesinde A* arama algoritması ve sezgisel fonksiyonlar da etkin bir şekilde kullanılmaktadır. Sezgisel arama mantığıyla yaygın olarak uygulanan A* algoritması, düğümleri sıralamak için değerlendirme fonksiyonunu kullanan bir algoritmadır. Bu algoritma da temel olarak bir değerlendirme fonksiyonu aşağıdaki gibidir:

$$f(n) = g(n) + h(n)$$

g (n), başlangıç düğümünden gidilecek düğüme olan gerçek maliyettir. Bir başka deyişle en uygun yolu bulma maliyeti olarak da ifade edilmektedir.

h (n), n düğümünden hedef düğüme giden en uygun yolun maliyetinin tahmini olarak ifade edilebilir. Bu problem ile ilgili sezgisel bilgilere de bağlıdır.

Yol arama probleminde tüm n düğümleri için h(n) hesaplanır ve hedef düğüme ulaşmak için en iyi yol A* algoritması tarafından bulunur. Yol arama problemlerinde A* algoritmasının uygulanması esas olarak problemin anlaşılması ve elde edilen çözümün yanı sıra, problem çözme ile ilgili bazı bilgileri aramak olarak kabul edilir. Diğer birçok yol arama algoritmasından farklı olarak A* algoritması, harita üzerinde yer alan tüm düğümler üzerinden geçiş yapmaz [22]. A*, bir haritada yer alan iki düğüm arasında en kısa yolu bulmaya çalışır. Haritadaki düğümleri bir sezgisel arama stratejisi kullanarak tarama işlemi yapar [23]. Ancak bu zaman ve bellek maliyetini yükseltmektedir. Daha sonra A* algoritması üzerinde çeşitli araştırmalar yapılarak zaman ve bellek maliyeti üzerinde çalışılmalar yapılmıştır [24]. Yol arama problemlerinde kullanılan diğer bir yöntem ise olasılık tabanlı yol bulma algoritmasıdır. A* algoritması veya arama tabloları gibi yol bulma tekniklerinde, kaynak ve hedef konumları aynıysa, genelde aynı yol oluşturulmaktadır. Öte yandan, olasılık tabanlı yol bulma yönteminde, kaynak ve hedef konumları aynı olsa bile farklı yollar üretebilir [16].

Bazı yaklaşımlarda ise oyundaki karakterin gerçekçiliğini belirlemek oyunun bir parçası haline getirilmiştir. Oyunun daha iyi hale gelmesi için, oyunculardan oyunda yer alan karakterin davranışlarını değerlendirmesi istenmektedir. Bazı oyunlarda ise Turing testinde olduğu gibi karakterleri gerçekçiliği sorgulanmaktadır. Turing testinde genellikle odada bulunan bir kişi başka bir odada bulunan iki konuğa sorular sorarak tanımlamaya çalışır. Konuklardan biri insandır, diğeri ise bir makinedir; sorgulayıcının görevi hangisinin insan olduğunu belirlemektir. Eğer sorgulayıcı bunu doğru söyleyemiyorsa, o zaman makine zeki kabul edilir [25]. Ters Turing olarak bilinen başka bir yaklaşımda ise, üretilen botların diğer botları insan ya da bot mu diye tanımlama becerilerinde kullanıldığı sistemdir. Son yıllarda bazı araştırmacılar ise çevrimiçi oyunlarda botları otomatik olarak tespit etmek için yöntemler geliştirmeye çalışmaktadır. Bu tür yöntemlere ihtiyaç duyulmasının nedeni, botların bazı oyunlarda haksız rekabet etmesini engellemektir [26].

Daha sonraları makine öğrenmesiyle oyun programlamasının tanışmasıyla beraber yeni yaklaşımlar ortaya çıktı. Bu durum oyunlara olan ilginin giderek artmasına ve oyuncu yelpazesinin de giderek genişlemesine sebep oldu. Oyuncuları arasında beceri, tercih ve deneyimin farklı olması oyunları kişiye özel

hale getirme ihtiyacı doğurdu. Bu sayede oyun deneyimlerinin büyümesine ve oyuncu görevlerinin artmasına yol açtı. Bu unsurlar oyunlarda modelleme ve deneyime dayalı uyumu giderek daha da önemli bir hale getirdi. Oyun stilini tanıyabilen ve modelleyen oyun motorları üretilmeye çalışıldı ve kullanıcının mevcut duygusal ve zihinsel durumunu tespit ederek ona göre kişiselleştirilmesi için gerekli çalışmalara başlandı. Oyuncu deneyim modelleme çalışması ile kullanıcı modeli yapımı için yapay zekâ teknikleri ile oyuncuların tecrübeleri elde edilmeye başlandı.

Oyuncu deneyim modelleri farklı başlıklar altında toplanmıştır. Oyuncu tarafında ifade edilen veriler nesnel oyuncu deneyim modeli olarak ele alınmıştır. Oyuncu tepkilerinden elde edilen verilere öznel oyuncu deneyim modeli adı verilmiştir. Oyuncu ile oyun arasındaki etkileşim yoluyla elde edilen bağlamsal ve davranışsal verilere ise oyun tabanlı deneyim modeli denmiştir [27]. Bir model deneyimini geliştirmenin en doğrudan yolu oyunculara kendi oyun deneyimlerini sormaktır ve bu verilere dayanan bir model inşa etmektir. Öznel modelde ilk ağızdan toplanan raporlar değerlendirilir. Uzmanlar veya dış gözlemciler tarafından dolaylı olarak bildirilen raporlar, potansiyel olarak güvenilir olan oyuncu deneyimine ek açıklamalar sağlar. Bununla birlikte, üçüncü kişi değerlendirmesi bu modelde ele alınmaz. Nesnel oyuncu deneyimi modelleme de oyun sırasında veya sonrasında oyuncuların yorumları anketler aracılığıyla alınır. Oyuncuların deneyimlerini karşılaştırmaları istenerek, Likert ölçeğinde veya sıralı bir şekilde verilen soru formlarını oyunun bir veya daha fazla oturumunda cevaplamaları istenir. Likert ölçeği, pozitif ve negatif farklı uçlarda yer alan yanıt seçenekleriyle davranış ölçmek için kullanılan bir yaklaşımdır. Bunun yanında, oyuncu deneyimi bir duygu akışına bağlanabilir, genellikle oyun sırasında meydana gelen olaylar tarafından tetiklenir. Oyunlar, oyuncunun fizyolojisindeki değişimleri etkileyebileceğinden oyuncu da duygusal tepkilerini ortaya çıkarabilir veya oyuncu yüz ifadesiyle tepki verebilir. Duruş ve konuşma da oyuncunun dikkatini ve odak seviyesini değiştirir. Bu tür bedensel değişikliklerin izlenmesi, oyuncunun duygusal tepkilerini tanıma ve sentezleme konusunda yardımcı olabilir. İşte bu model de nesnel model olarak karşımıza çıkmaktadır. Oyuncu ve oyun arasındaki etkileşimlerden türetilen herhangi bir eleman oyun tabanlı modelin temelini oluşturur. Bu, oyuncunun sistem öğelerine verilen yanıtlardan türetilen davranışından alınan parametreleri

içerir. Oyun tabanlı bir oyuncu deneyim modelinin verileri oyun etkileşiminin istatistiksel, mekânsal ve zamansal özellikleridir. Bu özellikler genellikle dikkat, meydan okuma ve katılım gibi zihinsel durum düzeyleriyle eşleştirilir. Bir görev için harcanan zaman ve performans gibi özellikler veya bir atış oyununda seçilen silahlar gibi oyuna özgü özellikler bu modelin verileri olabilmektedir [27]. Oyun esnasında veri toplanarak oluşturulan modeller yukarıdaki gibi modellere veya bu modellerin türevlerine benzerlik gösterir.

1960'larda ve 1970'lerde oyun oynama stratejileri, ünlü Bernstein'ın satrancı ve Samuel'in kontrol oyuncuları gibi insanlar tarafından oluşturulmuştur. Oyun alanındaki yapay zekânın muhteşem başarılarından biri olarak da Garry Kasparov'un Deep Blue'lu bir düelloda yenilgisiydi. Bununla birlikte, Deep Blue, insan bilgisine dayanan kaba kuvvet yaklaşımını benimsemekte, bu yüzden insan zekâsının anlaşılmasına pek katkıda bulunmamaktadır. Oyunlar ile yapay zekâ iç içe olmasına rağmen, insani bir rekabet performansı elde etmek için insan uzmanlığı ve hesaplama gücünden yardım almanın gerekli olduğu düşünülmektedir [28].

Başka bir yaklaşımda ilk amaç olarak yüksek etkileşimli oyunlar ile insan benzeri davranışa ve farklı beceri seviyelerine sahip sentetik karakterler yaratmaktır. Bununla birlikte, bu tür görevlerdeki sentetik karakterlerin gerçekçiliğini değerlendirmek için bir yöntem geliştirmek ve test etmektir. Gelişiminin bir parçası olarak, karakter dört boyutta ele alınmıştır. Bunlar zaman, saldırganlık, beceri ve taktik bilgisidir. Karakterin farklı parametre değerleri ile yeni sürümleri oluşturulmuş ve daha sonra onları uzman oyunculara karşı oynatarak değerlendirme yapılmıştır [29].

Oyunlarda uyarlanabilir öğrenme ile ilgili oluşturulan örnek sistemlerden biri ALIGN sistemidir. ALIGN sistem mimarisi dört kavramsal sürece ayrılmıştır; çıkarsama, içerik birikimi, müdahale kısıtlaması ve adaptasyon gerçekleştirmedir. Bunlar, oyun verilerinin eğitimsel uyarlamaya uygun olarak çıkartılması, çıkartılan verilerinin toplanması, oyun deneyiminin geçici olarak bir görünümünü geliştirmek için önceki uyarlama ve oyun bağlamına dayalı uyarlamalarının iyileştirilmesi için uygun şekilde oluşturulmuş aday elemanlara dayalı adaptasyon seçimidir [30].

Yapay zekâ karakterleri, oyunların etkileşim ve oynanabilirlik konusunda gelişim sağlamasına yardımcı olmaktadır. Bu gelişimi artırmak için öğrenme yaklaşımlarından yararlanılmaktadır. Pekiştirmeli öğrenme, oyuncu olmayan bu karakterlerin davranış modellerini oluşturmada umut verici bir yaklaşım olmasına rağmen, genel bir keşif aşaması gerekmektedir. Öte yandan taklitçi öğrenme, YZK'nin davranış modelini, rakibin eylemlerini gözlemleyerek önceden inşa etmek için etkili bir yaklaşımdır. Ancak taklit yoluyla öğrenme, YZK'nin performansını taklit ettiği rakiplerinin performansı ile sınırlar [31]. Pekiştirmeli öğrenme algoritması, YZK'nin bir ortamla etkileşim yoluyla bir görev öğrenmesini sağlar [32]. Çevreden aldığı ödüle dayanarak belirli bir durumda hangi eylemin en iyi yapılacağını öğrenir. Durum-eylem(DE) değerlerinin haritalandırılması politika olarak adlandırılır. Bu değerleri saklamak için popüler yaklaşımlardan biri tablo yaklaşımıdır [33]. Arama tablosu yaklaşımı, her bir durum eylemi çiftinin değerlerini saklamak için bir arama tablosu kullanır ve bu değerler, YZK çevre ile etkileşim kurduğunda ve öğrendiğinde değiştirilir. Diğer bir popüler yaklaşım, durum eylem arasında haritalama yapmaktır[34]. Ayrıca, kullanılan yaygın bir diğer algoritma da yapay sinir ağlarıdır. Bu yaklaşımla ilgili temel sorunlardan biri, problem alanı için uygun bir topoloji bulmaktır [35].

Pekiştirmeli öğrenme kullanarak karakterin gelişimini sağlayan FALCON, alınan ödüllerin gelecekteki kararları etkileyebileceği ve öğrenme süresinin uzun olmadığı gerçek zamanlı bir sistemdir. Karakter, doğrudan oyundan elde edilen ödülleri alarak öğrenir ve herhangi bir insan denetimi ve müdahalesini içermez. Elde edilen ödüller, rakipler ile etkileşimde bulunan sonuçlara dayanmaktadır. Bu şekilde, karakterin eğitilmesi için mükemmel bir modele gerek yoktur. Sonuç olarak, karakter yeterli eğitimle sağlanmışsa, farklı rakiplerle karşı karşıya kaldığında da davranışlarını ayarlayabildiği görülmüştür [36]. Pekiştirmeli öğrenme, yorumlama ve birleştirmeye odaklanarak öğrenmeyi hızlandırmak için uzman olmayan insanların geri bildirimlerinden faydalanır [37]. Örneğin, farklı renklerden oluşan sıralı odaları bulunan bir sistemde karakterin yaptığı oda geçişlerinin doğruluğuna göre ödül ve ceza verilerek daha hızlı öğrenmesi gerçekleştirilmektedir [38].

Pekiştirmeli öğrenme algoritmalarının başında Q-öğrenme algoritması gelmektedir [14]. Q-öğrenme algoritması ödül verme mantığına dayanır. Ödül verme mantığı oyun türüne göre değişiklik göstermektedir. Bazı oyunlarda kazanma veya

yenilgiye ödül verilirken bazılarında yapılan hamlenin sonucuna ödül verilmektedir [39]. Bir eylem kazanç ya da kayıpla sonuçlanır ve bir kazanç için +1 takviye ve bir kayıp için -1 takviye verilir. Sonuç olarak YZK takviye değerini yükseltmeye çalışır. Kazanmaya karşılık gelen durumları hedef durumlar olduğu için, kayıpla sonuçlanan durumları tercih etmemeyi öğrenecektir [1]. Diğer bir pekiştirmeli öğrenme algoritması ise SARSA'dır. "State-Action-Reward-State-Action" kelimelerinin kısaltılması ile bu isim verilmiştir [35]. SARSA bir politika içi algoritmadır ve öğrenme politikasını izler, Q öğrenme ise bir politika dışı algoritmadır ve bazı yakınsama gerekliliklerini yerine getirerek ilerler. SARSA algoritmasında bir sonraki eylem ve durum seçildikten sonra Q değeri güncellenirken, Q öğrenme algoritmasında ilk önce Q değeri güncellenir ve bir sonraki eylem bir sonraki adımda seçilir. Pekiştirmeli öğrenme algoritmalarının bir diğeri avantajlı öğrenme algoritmasıdır. Q öğrenme algoritmasına büyük oranda benzerlik gösteren bu algoritmada Q değeri yerine A değeri kullanılır ve bu A değeri hesaplamasında k ölçek faktörü ile bir önceki eylem seçimi arasındaki zaman değişimi kullanılır [40].

Öğrenmede model, çevre dinamiklerinin oluşumunu temsil etmektedir. Model, geçiş olasılığını, mevcut s durumu, a eylemi ve çevre dinamikleri yardımıyla bir sonraki durumu öğrenmek için kullanır. Geçiş olasılığı başarılı bir şekilde öğrenilirse, YZK mevcut durum ve eylem verildiğinde belirli bir duruma geçme olasılığını bilecektir. Bununla birlikte, modele dayalı algoritmalar, durum alanı ve eylem alanı büyüdükçe pratikliği azalmaktadır. Fakat modelsiz algoritmalar bilgisini güncellemek için deneme yanılma yöntemine güvenir. Sonuç olarak, tüm durum ve eylem birleşimlerini saklamak için alana ihtiyaç duyulmaz. Bazı büyük oyunlarda çevrenin karmaşık olmasından ve durum veya eylem uzayının büyüklüğünden dolayı çok anlamlı öğrenme gerçekleşebilmesi için mevcut bilgilerin kapsamlı bir şekilde kullanılarak mikro yönetim birimlerinin oluşturulması gerekir [41]. Bunun yanında video ve atari oyunlarında pekiştirmeli öğrenme ve derin öğrenme birleştirilerek yeni denemeler yapılmaktadır [42, 43].

Modern bilgisayar oyunları, etkileşim için çok sayıda olasılık sunan ve bilgisayar grafikleri kullanılarak görüntülenen karmaşık ve dinamik sanal dünyalar oluşturur. Bilgisayar kontrollü karakterlerin eylemleri genellikle sabit bir repertuardan geçtikleri için tekrara düşerler ve böylece oyunda sıkılmaya neden olur.

Dolayısıyla, YZK öngörülemez hareket ederse, oyunda daha iyi bir etkileşim ortaya çıkar [44].

Bazı araştırmalarda ise karakterin davranış kısımlarının beceri ve inandırıcılık düzeyi üzerinde etkili olduğu araştırılmıştır. Karar verme süresi, taktik sayısı, saldırganlık durumu ve nişan alma becerileri gibi davranış parçaları üzerinde araştırmalar gerçekleştirilmiş ve daha sonra insana karşı karakterin varyasyonlarını oynatılmıştır. Sonuç olarak da toplanan değerlerde karar verme süresi ve nişan alma becerilerinin daha insansı olduğu sonucuna varılmıştır [29]. Ayrıca tekrarlayan animasyonlar kesilip bunu bir yönteme veya öğrenme tekniğine bağlayarak karakterin daha çok insan gibi görünmesi sağlanmaya çalışılmıştır [45]. Oyunlarda karakterin yanıtı ve davranışı, olabildiğince gerçekçi olacak şekilde tasarlanmaktadır. Örneğin, bir bot görünürlük bölgesinde ve ötesinde bir oyuncuyu algılayabilir yapılabilir. Fakat oyunun daha ilgi çekici ve oynanabilir olmasını sağlamak için, karakter sonlu bölgelerin ötesindeki rakibini tespit edememelidir [46]. Oyun geliştiricileri, gerçek zamanlı strateji oyunlarında oyuncu verilerini yaygın olarak kullanmaktadırlar. Bunları gözlemlerden ve oyun verilerinden elde ederek Bayes modeli kullanıp bazı tahminlerde bulunurlar [47]. Uyarlanabilir bir yapay zekâyı besleyebilecek yüksek kaliteli ve sağlam tahminler için bundan yararlanırlar [48]. Olasılık tabanlı çıkarımda ön bilginin ayrıştırılması ve değişkenlerin koşullu bağımsızlıkları hesaplamanın yapılabilir olmasında önemli rol oynar [49]. Hedefe yönelik sıralı karar verme görevleri için eğitmenlerin geri bildirimde bulunmalarından yararlanılarak deneyimlerden öğrenebilen ve bu bilgiyi öğrenmeyi hızlandırmak için kullanabilen sistemler tasarlanmaktadır. Amaç, insana daha uyumlu modellerin gelişimini sağlamaktır [50].

Bir diğer yaklaşım ise pekiştirmeli öğrenme ile olasılık tabanlı yaklaşımları birleştirerek karakterin ivmeli bir şekilde gelişmesini sağlamaktır [51]. Bayes ve pekiştirmeli öğrenme algoritmaları farklı bakış açılarıyla birleştirilerek oyunlarda denenmektedir [52]. Bayes ile direkt eylemlerin olasılıklarını hesaplayarak, çevre modelleri veya ödüller üzerinde dağılımı kullanarak en uygun eylemi bulma mantığına dayanan bir yaklaşım sunar [53,54]. Bununla beraber yine davranış verilerinin tutulmasından faydalanılır [55]. Davranış verileri kaydedilerek karakterin gelişimi için kullanılır [56]. Son olarak bu yaklaşım sadece oyun sektöründe değil aynı zamanda enerji sektörü gibi farklı alanlarda da denenmektedir [57].

3. KULLANILAN YAKLAŞIMLAR

3.1. Naïve Bayes Yaklaşımı

Naive Bayes yaklaşımı, bir veri kümesi içerisinde yer alan verileri sınıflandırmak için Bayes teoremini kullanarak sınıflandırma yaklaşımıdır. Bayes yaklaşımı, bir X olayının, diğer bağımsız bir Y olayının gerçekleştiği durumda ortaya çıkma ihtimalinin hesaplanmasına dayanır. Buna koşullu olasılık da denir.

$$P(X|Y) = \frac{P(Y|X).P(X)}{P(Y)} \qquad (1)$$

Bir olayın meydana gelmesi birden fazla olayın varlığına bağlı olabilir. Örneğin, elimizde birden fazla X olayı, Y_1 olayı ve Y_2 olayı olsun. Bu olayların gerçekleşmesi birbirinden bağımsız olduğunu varsayalım. Bu durumda Y_1 ve Y_2 olayı gerçekleşmişken X olayının gerçekleşme ihtimali, ayrı ayrı Y_1 olayı gerçekleşmişken X olayının meydana gelme ihtimali, Y_2 olayı gerçekleşmişken X olayının meydana gelme ihtimali ve tek başına X olayının gerçekleşme ihtimalinin çarpımıyla elde edilir. Herhangi bir olayın gerçekleşme olasılığının sıfıra eşit olması durumunda, çarpma nedeniyle sonucu sıfırlayacaktır. Bu nedenle, "smoothing" yaklaşımlarından yararlanılır. En çok kullanılan smoothing yaklaşımlarının başında "add -1 smoothing" ve "add - k smoothing" gibi algoritmalar gelmektedir [62].

$$P(X \mid Y_i, \dots, Y_n) = P(X).P(Y_1, X) \dots P(Y_n, X) \qquad (2)$$

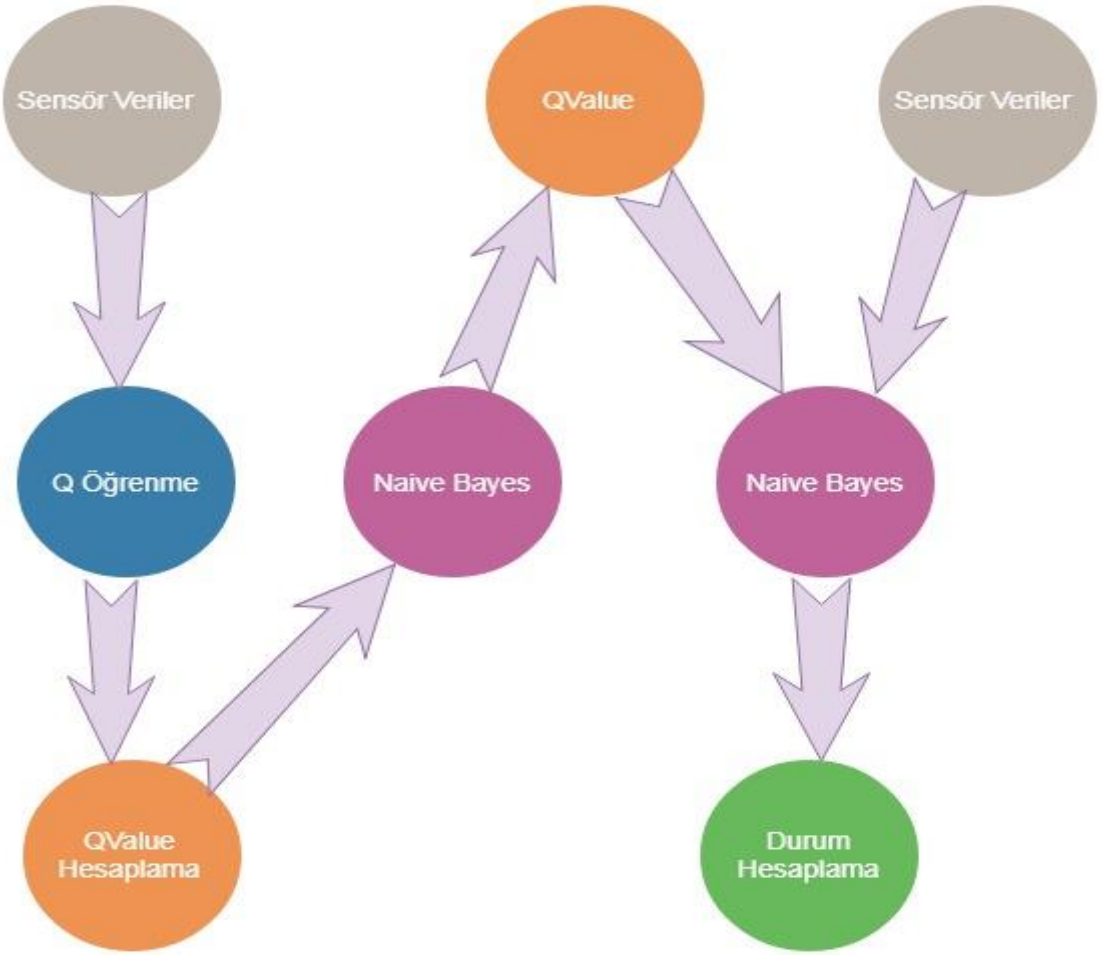
$$P(X \mid Y_i, \dots, Y_n) = P(X) \prod_{i=1}^{n} P(Y_i, X) \qquad (3)$$

3.2. Q Öğrenme

Q öğrenme, pekiştirmeli öğrenme yöntemleri arasında çok bilinen yöntemlerden biridir. Takviye öğrenme yaklaşımında, YZK belirlenen bir hedefe ulaştığında pozitif bir ödül verilir. Bu değer 1, 5, 10 gibi pozitif bir değer olabilir. Bunun dışında, YZK'nin gitmesini istemediğimiz tehlikeli yerler de işaretlenebilir ve YZK için negatif bir ceza puanı verilebilir. YZK, en yüksek Q değerine sahip bir sonraki eylemi seçer. Büyük bir Q değerine sahip bir eylemin, hedefe ulaşmak için iyi bir yol olduğu düşünülmektedir. Bununla birlikte, en yüksek Q değerini seçmek, daha iyi bir yol bulmak için fırsatları sürekli olarak düşürür. Bu nedenle, YZK bazen bir sonraki eylemi rastgele seçer. Bu rastgele seçim, yeni durumları keşfetmek ve henüz bulunmayan daha iyi bir yolu bulmak için tercih edilir. YZK'nin amacı yalnızca hedefine ulaşmak ise, olumlu ödüller vermek yeterli olabilir. Bununla birlikte, karakterin engellerden sakınmasını ve hedefe ulaşması isteniyorsa, engelleri içeren durum alanına negatif ödül verilebilir. Olumsuz bir ödül, kötü bir sonucu temsil eder ve YZK için kötü durumları tercih etmeme işlemi bu şekilde öğretilir. YZK, ödüllerin toplamını arttırmak için harekete geçtiğinden, olumsuz ödül bulunan durum YZK'nin ilgisini çekmeyecektir [58]. Burada 4 tanımlama grubu içeren MKS kullanılmaktadır. (S, A, pr, pt) S durum dizisi, A eylem dizisi, pt S'den A'ya geçiş olasılığını tutan bir geçiş modelidir ve pr, S'den A'ya geçiş sırasında ödül alma ihtimalini tutan bir modeldir.

Pekiştirmeli öğrenme deki amaç YZK'nin geçiş model(pt) ve ödül model(pr) bilmeden YZK'nin ödül değerini en yüksek değere çıkarmaktır. Q öğrenmede ise YZK geçmiş tecrübelerinden yararlanarak Q(s,a) değerini tahmin etmeye çalışır. Daha sonra bu Q değerine göre bir eylem seçer. Her adımda gerçekleştirilecek bir işlemi seçmek için kullanılan strateji, algoritmanın performansı için çok önemlidir. Diğer bazı pekiştirmeli öğrenme algoritmalarında olduğu gibi, keşif(araştırma) ile kullanım arasında denge bulunması gerekir. Bu dengeyi sağlamak için yaygın olarak kullanılan iki yöntem, yarı-uniform rasgele keşif ve Boltzmann araştırmasıdır. Yarı uniform rastgele araştırmada, En iyi eylem olasılığı için bir p seçilir ve olasılık 1-p ile bir eylem rasgele seçilir. Bazı durumlarda, p başlangıçta oldukça düşüktür ve bu durum araştırmaya teşvik eder. Boltzman da, seçilen bir eylemin olasılığı, Q değerinin güncel tahmini ile birlikte artırılır. Burada iyi eylemlerin zayıf eylemlerden daha fazla seçilme eğiliminde olduğu görülür [54].

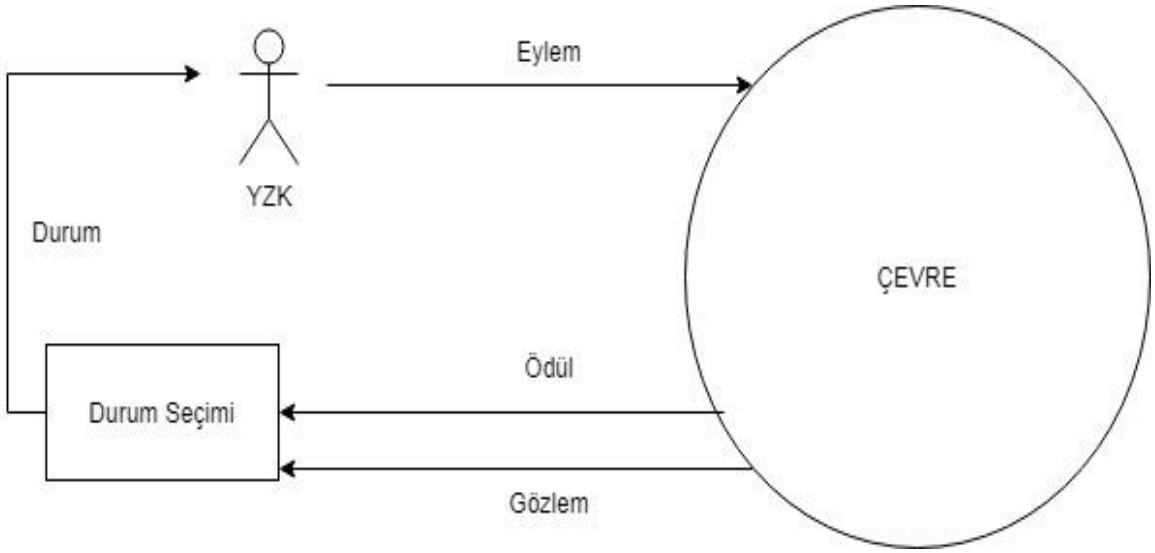
3.3. Model Yapısı



Şekil 3.1 Model Yapısı

4. UYGULANAN YÖNTEM

Başlangıçta bilgisiz olarak doğaya bırakılan YZK başlangıçta varsayılan olarak seçilen durum ne ise ilk olarak o şekilde davranır. Daha sonra çevreden gelen bir uyarı ile karşılaştığında anlık sahip olduğu bilgiler çerçevesinde gitmesi gereken duruma karar verir. Sensör veri olarak da adlandırdığımız bu bilgiler YZK'nin sahip olduğu karakteristik özellikler(sağlığı, atak gücü vb.) ve çevre ile etkileşiminden elde ettiği diğer bilgilerin(rakibe olan uzaklığı vb.) bütünüdür. Bununla birlikte davranış öğrenme işleminde daha önceki seçimlerinde Q öğrenmeden elde edilen QValue değeri de kullanılmaktadır.



Şekil 4.1 YZK-Çevre Etkileşimi

QValue değeri, Q öğrenme algoritması kullanılarak hesaplanan bir değerdir. İlk olarak başlangıç verisi ile oynamaya başlayan karakter daha sonra insanlara karşı oynatılarak çeşitli veri setleri elde edilir. Her bir oyuncuya karşı YZK'nin sergilediği davranışlar (geçiş yaptığı durumlar), bu durumlara geçerken sahip olduğu sensör veriler ve QValue değeri oyun sonunda kaydedilmektedir. Q öğrenme yaklaşımı kullanılarak YZK'nin hedef durumlara ulaşana kadar izlediği yoldaki bütün geçtiği durumlara geriye doğru Q Value değeri hesaplanarak eklenir. Ve oyun sonunda YZK'nin bu durumlara geçerken sahip olduğu sensör verilere karşılık gelen QValue değeri bilgisi tutulmuş olur. Aynı şekilde sensör veriler ve QValue değerine karşılık gelen durum bilgileri de kaydedilmiş olur. Elde edilen bu veri setleri üzerinde Naive Bayes yaklaşımı kullanılarak birebir verilere sahip veri tabloları

oluşturulur. Daha sonra veri tabloları oyun esnasında davranış seçimi aşamasında hızlı bir şekilde sonuca ulaşmak için kullanılır. Bu şekilde YZK'nin insanlara karşı oynarken hangi değerlere sahipken hangi davranış durumunu seçeceği bilgisi Naive Bayes ve Q öğrenme yaklaşımlarıyla elde edilmiş olur. QValue bilgisi YZK'nin geçmişte insanlara karşı oynarken hangi davranış yolunu daha çok seçtiği anlamına gelir. Bu yol, durum geçişleri bilgisini içerir.(Bekleme-Yürüme-Koşma-Atak şeklinde). Ancak sadece bu QValue değerine göre durum tercihini gerçekleştirmek, YZK'nin ezbere hareket etmesi anlamına gelir ve belli bir sure sonra tekrara düşmesine sebep olacaktır. O yüzden bu tercih yerine veri seti olarak mevcut olan Q Value dağılımı ve durum dağılımı üzerinde Naive Bayes yaklaşımı kullanılarak sonuca varmaya çalışılmıştır. Bu şekilde hem QValue değerine göre hem de geçmişteki insanlara karşı sergilediği tercihlerin sıklığına göre karar vermeyi öğrenmiş olacaktır.

4.1. Veri Toplama Algoritması

Oyun esnasında YZK, oyuncu ile etkileşime geçtiği anda o an sahip olduğu sensör bilgileri ve durum bilgileri tutulmaktadır. QValue değeri ise oyun sonunda hesaplanıp daha önceden tutulan bu durum bilgilerine eklenmekte ve bu şekilde bütün bilgilerin kaydedilmektedir (Algoritma 1). Oyun sonunda yazma işlemlerini kullanarak kaydettiğimiz bu verilerden daha sonra oyun esnasında kullanabileceğimiz sonuçlar çıkartılmaktadır.

Algoritma 1: Veri Toplama Algoritması

```
1:Prosedür: DataCollection ( )
2: do{
3:    if(detectCollision) then
4:     createGameAgentStateInfo()
5:    end if
6:              if(TimeCheck( ) = 0 || PlayerHealthCheck ( ) = 0) then
7:              Game_Result = 1;
8:              else
9:              Game_Result = 0;
10:             end if
11:     }while( FinishGame( ) );
12: if(Game_Result = 1) then
13: WriteData(GameAgentStateInfo);
14: end if
15: Prosedür Sonu
```

4.2. QNBY Algoritması

Veri toplama aşamasında toplanan veri kümesinde başlangıç olarak hangi sensör verilere sahipken hangi QValue değerinin gelmesi gerektiği bilgisi Q öğrenme ve Naive Bayes Yaklaşımı(QNBY) ile elde edilmektedir (Algoritma 2). Bunun için sensör veriler ve QValue bilgisini içeren veri kümesindeki dağılım üzerinde oyun başlangıcında Naive Bayes yaklaşımı uygulanmaktadır.

Algoritma 2: QNBY Algoritması

1:Prosedür: QNBY ( )

2: $\mathrm{ComputedQValue} \leftarrow \mathrm{QValueHesaplama}(\,)$

3: ComputedQValue, SensorData ve Durumu Kaydet

4: SensorDataQValuePair, QValueStatePair Tablolarını Sıfırla

5: Tekrar Et:

6: $\mathrm{SensorDataQValuePair} \leftarrow \mathrm{NaiveBayes}(\,\mathrm{SensorData}, \mathrm{ComputedQValue}\,)$

7: $\mathrm{QValueStatePair} \leftarrow \mathrm{NaiveBayes}(\mathrm{ComputedQValue}, \mathrm{Durum})$

8: Kaydedilen Verilar Bitene Kadar Devam Et.

9: Prosedür Sonu

Aşağıda yer alan formüller (4-7) ile QValue ve sensör verileri içeren veri kümesi üzerinde sınıflandırma işlemi gerçekleştirilmektedir.

$$\mathrm{P}(\mathrm{QValue} \mid \mathrm{SensorData}) = \frac{\mathrm{P}(\mathrm{SensorData} \mid \mathrm{QValue})\ \mathrm{P}(\mathrm{SensorData})}{\mathrm{P}(\mathrm{QValue})} \qquad (4)$$

$$\mathrm{P}(\mathrm{QValue} \mid \mathrm{SD\ \ RU\ \ HD\ \ AG\ \ KZ}\,) = \mathrm{P}(\mathrm{QValue}\,).\,\mathrm{P}(\mathrm{QValue} \mid \mathrm{SD}) \qquad (5)$$

$$.\,\mathrm{P}(\mathrm{QValue} \mid \mathrm{RU})\,.\,\mathrm{P}(\mathrm{QValue} \mid \mathrm{HD})$$

$$.\,\mathrm{P}(\mathrm{QValue} \mid \mathrm{AG}).\,\mathrm{P}(\mathrm{QValue} \mid \mathrm{KZ})$$

$$\mathrm{argmax}\,\{\,\mathrm{P}(\mathrm{QValue})\,\} = \mathrm{argmax}\,\{\,\mathrm{P}(\mathrm{QValue} \mid \mathrm{SD\ \ RU\ \ HD\ \ AG\ \ KZ})\,\} \qquad (6)$$

$$\text{argmax}\{ P(\text{QValue})\} = \text{argmax}\{ P(\text{QValue}).P(\text{QValue}|\text{SD}).P(\text{QValue}|\text{RU}) \quad (7)$$
$$.P(\text{QValue}|\text{HD}).P(\text{QValue}|\text{AG}).P(\text{QValue}|\text{KZ})\}$$

QValue için sınıflandırma işlemi gerçekleştirildikten sonra birebir sensör veri ve QValue çiftini içeren veri tablosu elde edilmektedir. Aynı şekilde durum değeri için aşağıdaki formüller (8-10) kullanılarak sınıflandırma işlemi gerçekleştirilmektedir.

$$P(\text{Durum}|\text{QValue}) = \frac{P(\text{QValue}|\text{Durum}).P(\text{QValue})}{P(\text{State})} \quad (8)$$

$$\text{argmax}\{ P(\text{Durum})\} = \text{argmax}\{ P(\text{Durum}|\text{QValue})\} \quad (9)$$

$$\text{argmax}\{ P(\text{Durum})\} = \text{argmax}\{ P(\text{State}).P(\text{State}|\text{QValue})\} \quad (10)$$

Durum için sınıflandırma işlemi gerçekleştirildikten sonra birebir QValue ve durum çiftini içeren veri tablosu elde edilmektedir.

4.3. Davranış Karar Algoritması

Oyun esnasında performans kaybı yaşamamak için sensör verileri birleşik değer haline çevrilerek birebir sensör veri ve QValue çiftini içeren veri tablosu oluşturulmaktadır. Aynı şekilde elimizde bulunan QValue ve karşılık gelen durum bilgilerini içeren veri kümesi üzerinde de Naive Bayes yaklaşımı uygulanarak birebir QValue ve durum çiftini içeren veri tablosu elde edilmektedir. Oyun hızına etki etmemesi acısından oyun başlangıcında eğitme işlemi gerçekleştirilmektedir. Oyun esnasında ise davranış seçim işlemi yapılırken yani durumlar arasında geçiş işlemi yapılırken, ilk önce Qvalue değeri hesaplanması için karakterin o an sahip olduğu sensör veri birleşik değer haline çevrilerek karşılık gelen QValue değeri sensör veri ve QValue çiftini içeren veri tablosuna bakılarak elde edilir (Algoritma 3). Daha sonra bu QValue değerine göre daha önceden insanlara karşı oynanan oyunlarda kaydedilmiş veri kümesinden elde edilen Qvalue ve durum çiftini içeren veri tablosundan karşılık gelen durum YZK için seçilir.

Algoritma 3: Davranış Karar Algoritması

```
1: Prosedür: DavranisKarar( SensorData{SD, RU, HD, AG, KZ},
   SensorDataQValuePair, QValueStatePair )
2: YZK İçin Başlangıç Durumunu Ata
3: Tekrar Et:
4:          SensorData {SD, RU, HD, AG, KZ}  Al
5:          CombinedData ← Merge( SensorData {SD, RU, HD, AG, KZ} )
6:          OptimalQValue ← SensorDataQValuePair(CombinedData )
7:          KD ← QValueStatePair(OptimalQValue )
8:          YZK için KD' yi uygula.
9: Oyun Sonuna Kadar Devam Et.
10: Prosedür Sonu
```

4.4. ABDY Algoritması

Deney aşamasında sağlıklı karşılaştırma yapılabilmesi için kullanılan Açgözlü Benzeri Davranış Yaklaşımı'nda(ABDY) Q öğrenme algoritmasıyla QValue hesaplaması yapıldıktan sonra kaydedilen verilerden QValueStatePair tablosu oluşturulmaktadır (Algoritma 4). Karakterin davranış seçme işlemi sağlanırken aşırı saçma bir davranış seçmemesi için geçmişte toplanan verilerden oluşturulan bu tablodan en çok tercih edilen 3 farklı durum listesi elde edilmektedir. Oyun esnasında ise ABDY uygulandığında yapay zekâ karakterine bu 3 farklı en çok tercih edilen listeden rastgele bir tanesi atanmaktadır.

Algoritma 4: ABDY Algoritması

1: Prosedür: ABDY( )

2: $\mathrm{ComputedQValue} \leftarrow \mathrm{QValueHesaplama}(\,)$

3: $\mathrm{ComputedQValue}$, Durumu Kaydet

4: RandomSonucListesini Sıfırla

5: Tekrar Et:

6: $\mathrm{QValueStatePair} \leftarrow \mathrm{NaiveBayes}(\mathrm{ComputedQValue}, \mathrm{Durum})$

7: $\mathrm{randomPick} \leftarrow \mathrm{MostPreferedThreeState}(\mathrm{QValueStatePair})$

8: Kaydedilen Veriler Bitene Kadar Devam Et.

9: Prosedür Sonu

4.5. Q Öğrenme Algoritması

Hedef duruma ulaşıldığında Q öğrenme algoritması (Algoritma 5) yardımı ile QValue değerleri hesaplanıp geriye doğru bütün durumlar güncelleniyor.

$$\text{QValue} \rightarrow \pi(s) = \text{argmax}(Q[s, a])$$

$$R(\text{state}, \text{action}) = \text{Ödül Fonksiyonu (Reward Function)}$$

$$\text{QValue} = \text{argmax}\{Q(\text{state}, \text{action})\}$$

$$Q(\text{state}, \text{action}) = R(\text{state}, \text{action}) + \text{Gamma} * \text{Max}[Q(\text{next state}, \text{all actions})]$$

$$\text{QValue} = \text{argmax}\{R(\text{state}, \text{action}) + \text{Gamma} * \text{Max}[Q(\text{next state}, \text{all actions})]\}$$

Oyun sonunda başlangıç durumu ve bitiş durumu ile birlikte tüm durum listesi Q öğrenme algoritmasının kullanıldığı fonksiyona gönderiliyor ve aşağıdaki adımlar izleniyor.

Algoritma 5: Q-Learning Algoritması ile QValue Hesaplama [59]

1: Prosedür: QValueHesaplama ( )

2: $Q(s, a)$ Değerini Sıfırla

3: Tekrar Et(her bir bölüm için):

4: s Değerini Sıfırla

5: Tekrar Et(her bir adım için):

6: s Değerini Al

7: s İçin a Değerini Al

8: r ve s' Değerini Gözlemle

9: $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$

10: $s \leftarrow s'$;

11: $\text{QValue} \leftarrow Q(s, a)$

12: Adım Sonu

13: Bölüm Sonu

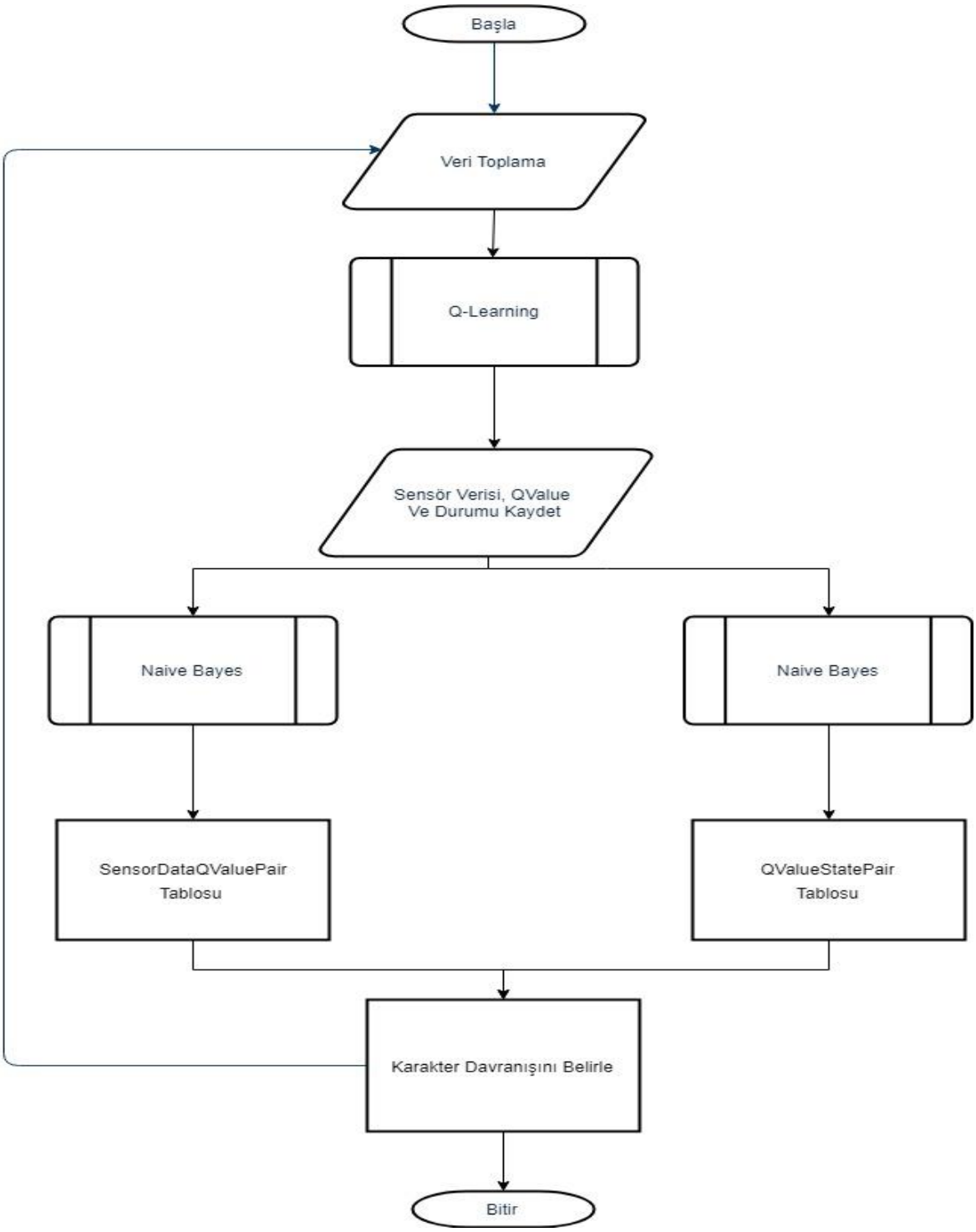
14: Hedef s 'i Bulana Kadar Devam Et.

15: Prosedür Sonu

QValue değerleri hesaplandıktan sonra bu değer ile birlikte karşılık gelen durum bilgisi ve o an sahip olduğu sensör bilgiler veri setine eklenir. Bu şekilde öğrenme işlemi için gerekli veriler elde edilmiş olur.

4.6. QNBY Akış Diyagramı

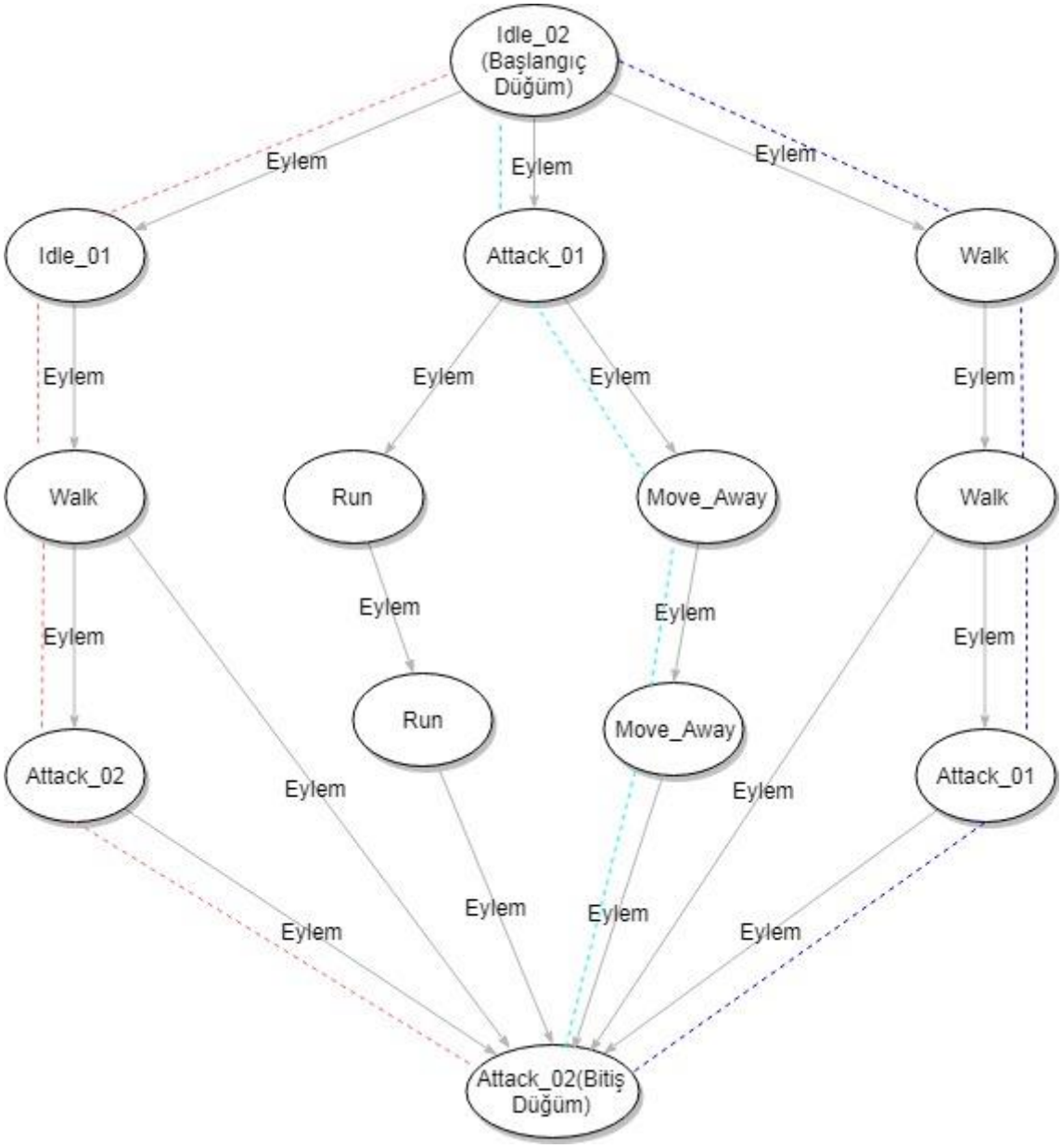
QNBY algoritmasına ait akış diyagramı aşağıda yer alan Şekil 4.2 ‘deki gibidir.



Şekil 4.2 QNBY Akış Diyagramı

4.7. Davranış Senaryosu

Şekil 4.3'de örnek olarak belirtildiği üzere, hedefe ulaşmak isteyen YZK başlangıçta "Bekleme_02" durumundan daha sonra tekrar "Bekleme _02" durumuna geçiyor oradan "Yürüme" durumuna geçtikten sonra "Atak_02" durumuna ve en son olarak "Atak_02" durumuna tekrar geçerek oyunu tamamlamış oluyor.



Şekil 4.3 Davranış Senaryosu

5.DENEYLER VE SONUÇLARI

5.1. Deneysel Yaklaşımlar

Bu tezde kullanıcı tabanlı deneyler gerçekleştirirken aşağıdaki bazı yaklaşımlar kullanılmıştır. Veri kaynağı olarak aşağıda yer alan veriler elde edilmektedir.

- Oyun oturum bilgileri,

- Anket veya skor bilgilerini,

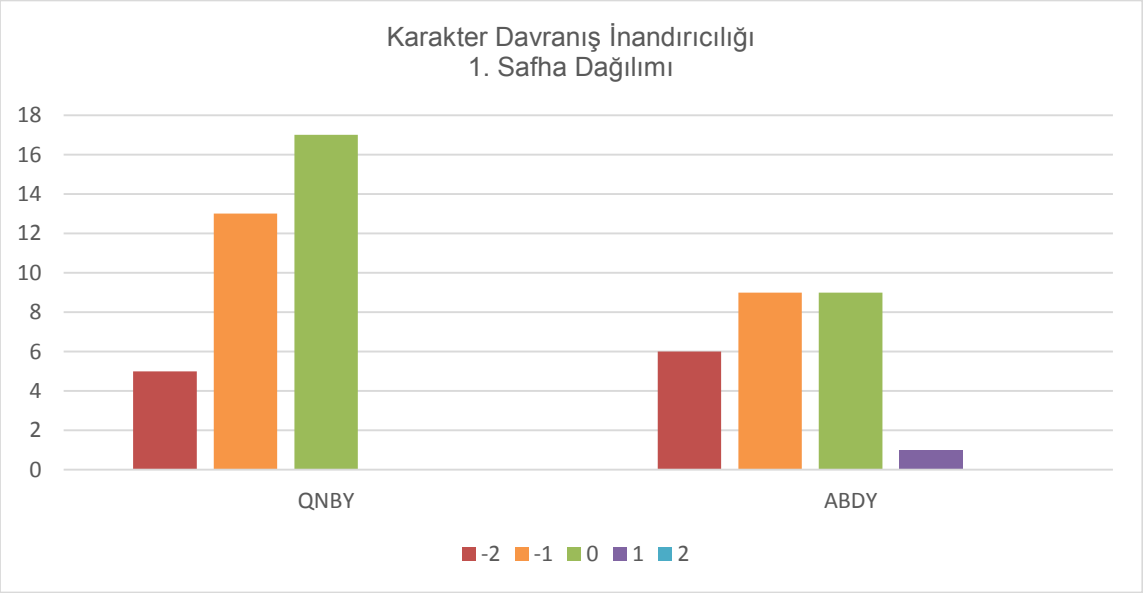
- Veri ve durum yol bilgilerini içeren dosyaları elde edilmektedir [60] .

Deney aşamasından sonra değerlendirme işlemi yapılırken 1 (çok kötü) ile 5 (çok iyi) arasında puanlandırma sistemiyle sorular puanlandırılmış ve her bir soruya verilen cevabın ortalaması alınarak deney gruplarının sonuçları karşılaştırılmıştır [61]. Burada yer alan puanlandırma yaklaşımını benzer bir puanlandırma sistemi bu tezde kullanılmaktadır. -2 (çok kötü) ve 2(çok iyi) arasında puanlar olan bir puanlandırma sistemi yer almaktadır. Bunun yanında oyun senaryosunda birebir yaklaşımı kullanılmıştır [36]. YZK'nin tek hedefi oyuncuyu öldürmek ve oyuncunun tek hedefi ise YZK'yi öldürmektir. Bazı yaklaşımlarda basit ödüllendirme sistemi kullanmak yerine daha karmaşık ödüllendirme sistemi kullanılmaktadır. Her bir vuruş için 0.5 veya her bir öldürme için 1 vermek yerine, ödül değerlerini yapılan tam hasarla orantılı olarak sağlamak gerektiğinden bahsedilmektedir. Bu tez çalışmasında ise sabit bir ödül değeri kullanılarak sonuç elde edilmektedir. Başka bir yaklaşımda geliştiriciler tarafından oluşturulan karakterin, bir UT(Unreal Tournament) botundan (non-learned Pogamut bot) daha fazla insansı olup olmadığını öğrenmek için test edilmiştir. Oluşturulan bot üç farklı mod arasında geçiş yapmak için sonlu bir durum makinesi kullanıyor. Her mod için ayrı ayrı politikaları öğreniyor. Arama modu, rakiplerle savaşmazken çevreden kaynakları (cephane ve sağlık paketleri gibi) toplamak için tasarlanmıştır. Bir düşman görünür olduğunda bot, atak moduna geçiyor. Hedef modu, rakibe ateş ederken saldırı modu sırasında tetikleniyor. Karşılaştırılan bot is Pogamut ta yapılan öğrenme işlevi olmayan bir bot. Kullanıcılara hazırlanan sorular ile puanlandırma yapılarak karşılaştırma yapılıyor [34]. Bu yaklaşıma benzer olarak bu çalışmada ABDY ile hareket eden karakter ile QNBY ile öğrenme işlemi gerçekleştiren YZK karşılaştırılmaktadır. Burada ABDY yaklaşımı ile YZK için durumlar arasında geçiş yaparken anlık seçebilmesi için en çok tercih edilen durumlar olarak belirlenen

durum değerlerinden biri rastgele atanmaktadır. Yukarıdaki yaklaşımda yer alan atak modu gibi oyuncuya yaklaştığında hep Atak_01 ve Atak_02 gibi fayda getiren durumları seçecek açgözlü benzeri yaklaşım kullanan bir YZK ile Q-öğrenme ve Naive Bayes kullanan YZK karşılaştırılmaktadır.

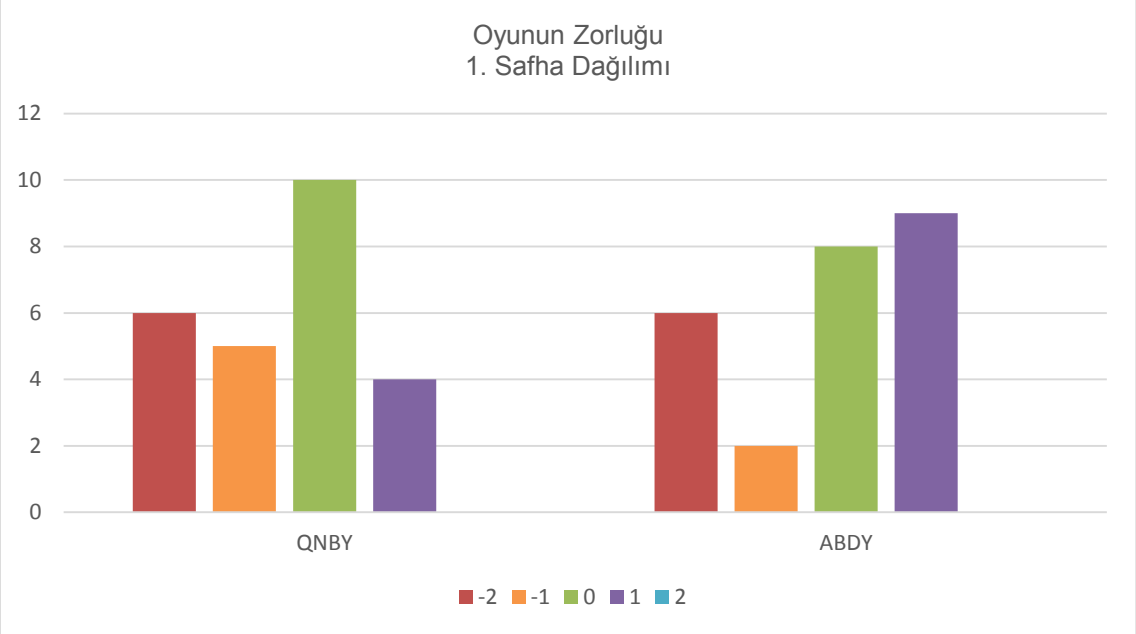
5.2.Deney Sonuçları

Aşağıdaki grafikler safhalara göre dağılımı ifade etmektedir. Her safhada, hazırlanan oyun 20 ile 35 yaş aralığında 5 farklı kişiye oynatılarak veriler elde edilmiştir. Bu 5 kişiye ilk önce QNBY yöntemi kullanarak 5 oyun oynatılmış ve her oyun sonunda 4 farklı sorudan oluşan anketi cevaplandırması istenmiştir. Bu ankette yer alan sorular karakterin davranış inandırıcılığı, oyunun zorluğu, karakteri öldürme zorluğu ve oyunun oynanabilirlik seviyesidir. Puanlandırma sisteminde yer alan -2 ,-1, 0, 1, 2 sırasıyla çok kötü, kötü, orta, iyi ve çok iyi anlamlarına gelmektedir. Burada yer alan -2,-1 cevapları olumsuz, 0 cevabı orta ve 1,2 cevabı da olumlu değerlendirme olarak ele alınmaktadır. QNBY için 5 oyun oynayıp puanlandırdıktan sonra aynı 5 kişiye ABDY yöntemini içeren 5 oyun daha oynatılıp ankette yer alan soruları puanlandırması istenmiştir. Toplanan bu puanlamalar doğrultusunda aşağıdaki sonuçlar elde edilmiştir.



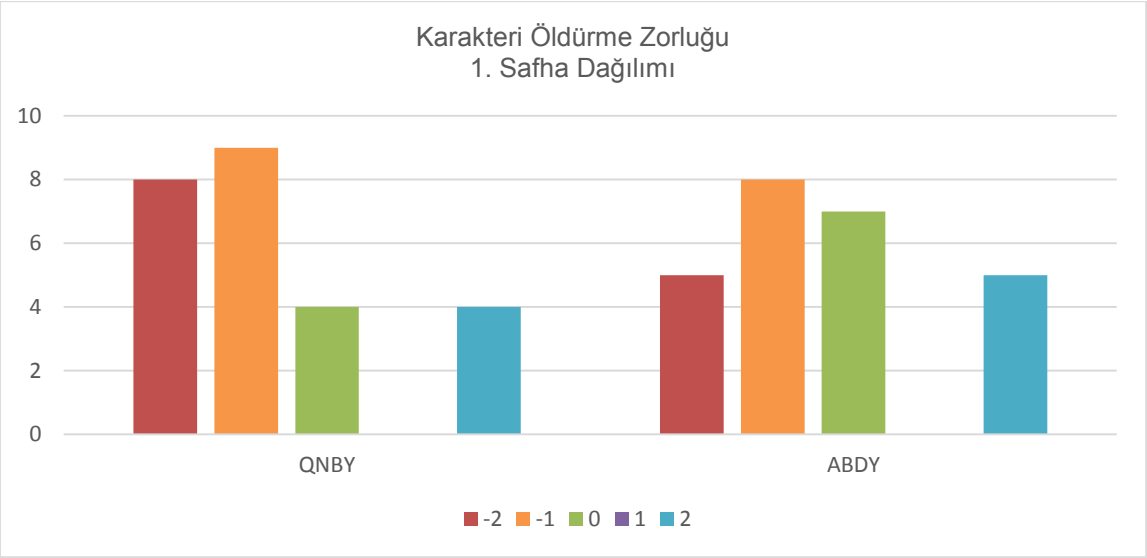
Şekil 5.1. 1.Safha Karakter Davranış İnandırıcılığı

1.Safha'da karakterin davranış inandırıcılığına %72 oranında olumsuz cevap verilirken %28 oranında orta cevabı verilmiştir. ABDY de ise %60 oranında olumsuz cevap verilirken %4 oranında olumlu ve % 36 oranında orta cevabı verilmiştir.



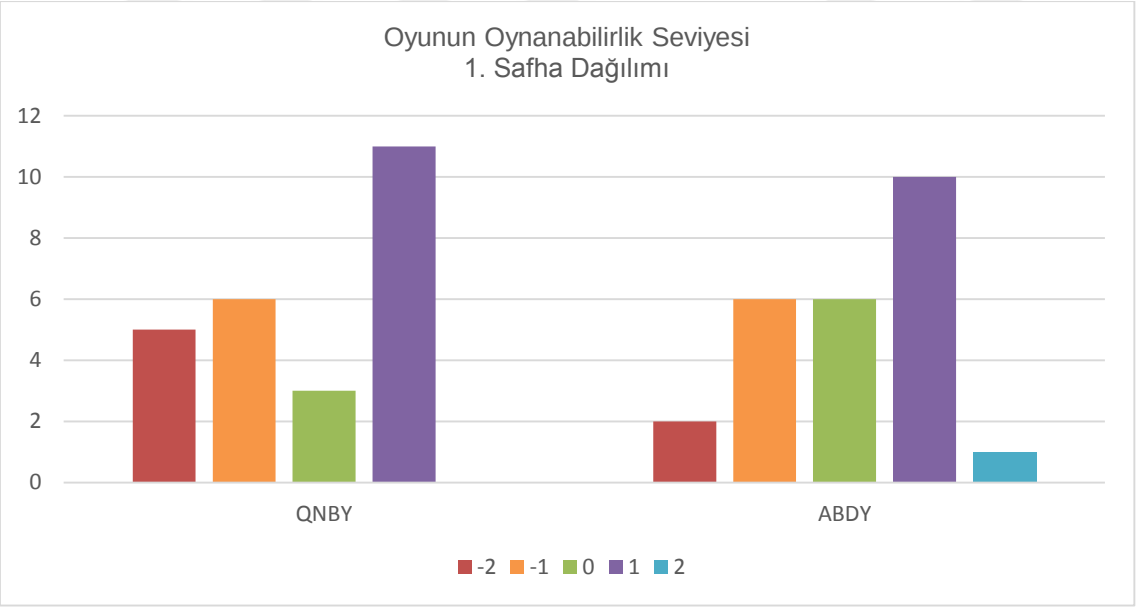
Şekil 5.2. 1.Safha Oyunun Zorluğu

1.Safha'da oyunun zorluğuna %44 oranında olumsuz cevap verilirken %16 oranında olumlu ve %40 oranında orta cevabı verilmiştir. ABDY de ise %32 oranında olumsuz cevap verilirken %32 oranında olumlu ve % 38 oranında orta cevabı verilmiştir.



Şekil 5.3. 1.Safha Karakteri Öldürme Zorluğu

Karakteri öldürme zorluğuna QNBY için %68 oranında olumsuz cevap verilirken %16 oranında olumlu ve %16 oranında orta cevabı verilmiştir. ABDY de ise %48 oranında olumsuz cevap verilirken %20 oranında olumlu ve % 32 oranında orta cevabı verilmiştir.

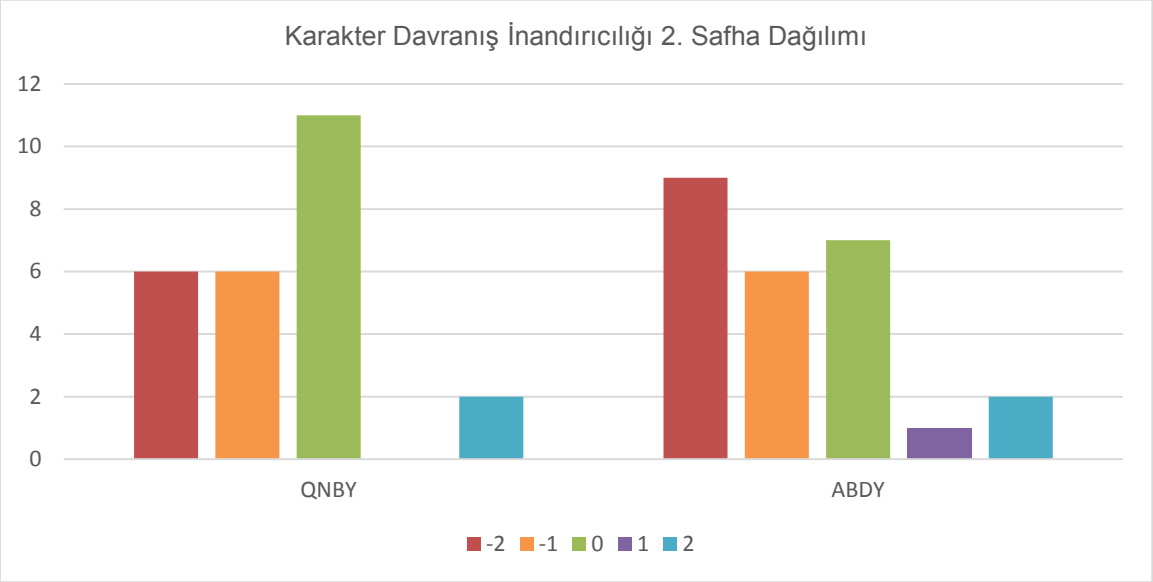


Şekil 5.4. 1.Safha Oyunun Oynanabilirlik Seviyesi

Oyunun oynanabilirlik seviyesine QNBY için %44 oranında olumsuz cevap verilirken %44 oranında olumlu ve %12 oranında orta cevabı verilmiştir. ABDY de ise %32 oranında olumsuz cevap verilirken %44 oranında olumlu ve %24 oranında

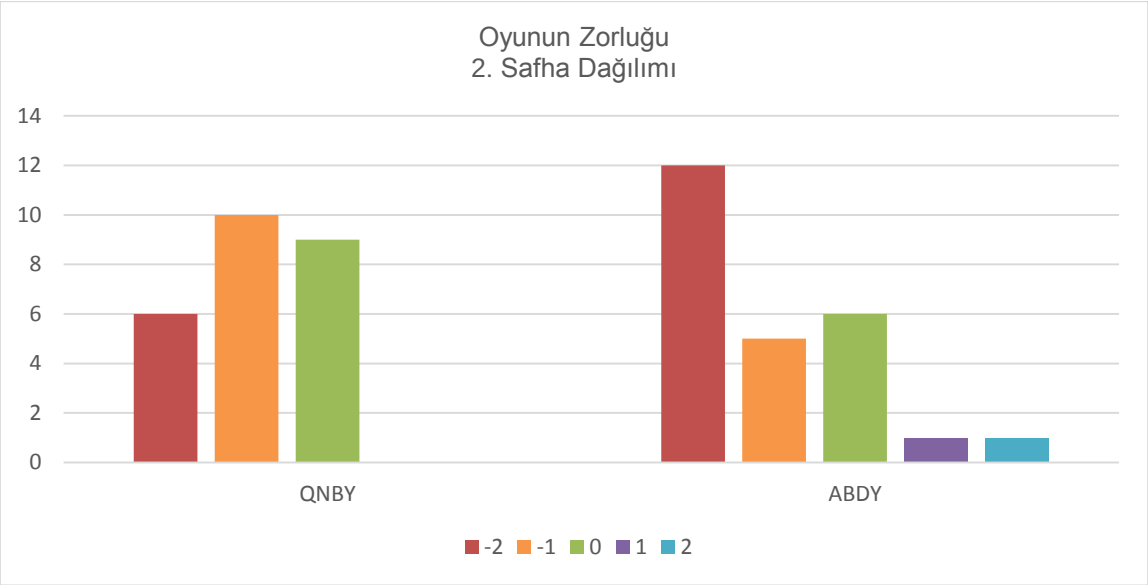
orta cevabı verilmiştir. Buradan da anlaşıldığı üzere 1.Safha'da ABDY, QNBY'ye oranla daha iyi sonuçlar almıştır.

2.Safha'da hazırlanan oyun 5 farklı kişiye oynatılarak elde edilmiştir. Burada 1.Safha'dan farklı olarak hazırladığımız öğrenme metodu daha önceden 1.Safha'da oynayan 5 kişiden toplanan veriyi kullanmaktadır. Sonuç olarak, QNBY için kötü olarak verilen cevap ABDY'e oranla fazla olsa da, çok kötü olarak verilen cevap ABDY'nin gerisinde kalmıştır. Buna karşılık orta, iyi ve çok iyi olarak verilen cevap sayısı ABDY'yi küçük farklarla da olsa geçtiği görülmektedir.



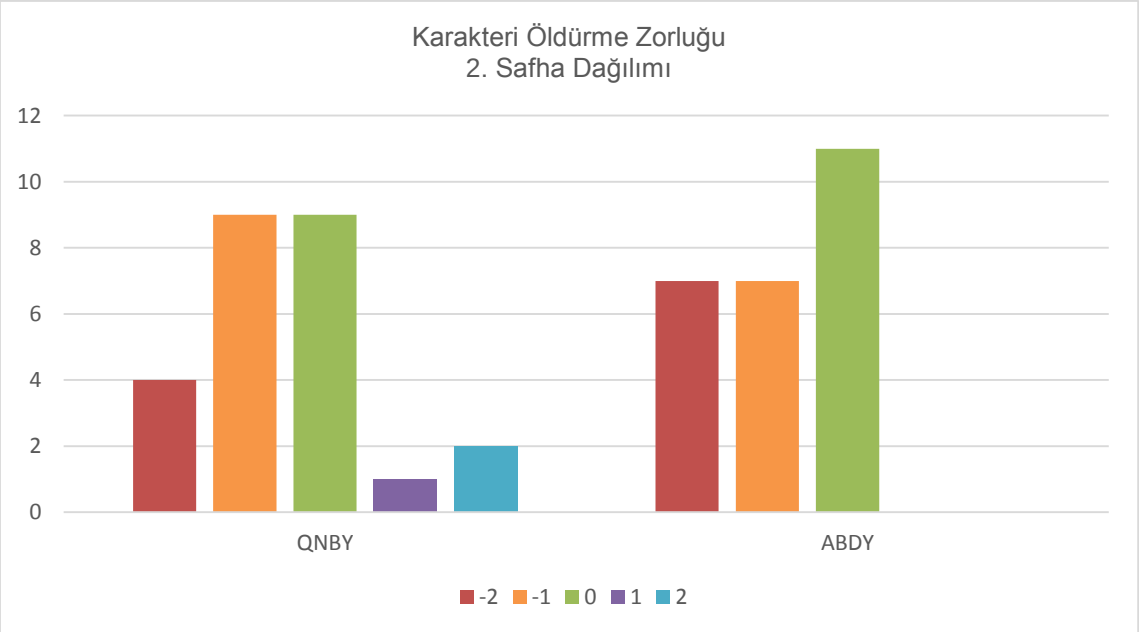
Şekil 5.5. 2.Safha Karakter Davranış İnandırıcılığı

2.Safha'da QNBY için karakterin davranış inandırıcılığına %48 oranında olumsuz cevap verilirken %8 oranında olumlu ve %44 oranında orta cevabı verilmiştir. ABDY de ise %60 oranında olumsuz cevap verilirken %12 oranında olumlu ve % 28 oranında orta cevabı verilmiştir.



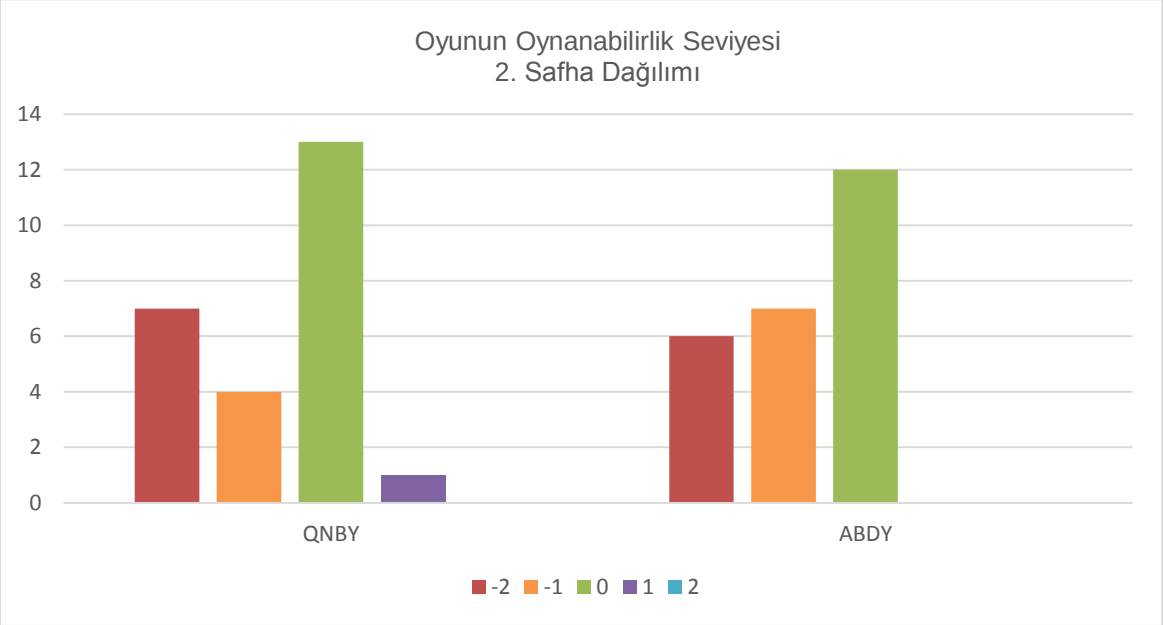
Şekil 5.6. 2.Safha Oyunun Zorluğu

2.Safha'da oyunun zorluğuna %64 oranında olumsuz cevap verilirken %36 oranında orta cevabı verilmiştir. ABDY de ise %68 oranında olumsuz cevap verilirken %8 oranında olumlu ve %24 oranında orta cevabı verilmiştir.



Şekil 5.7. 2.Safha Karakteri Öldürme Zorluğu

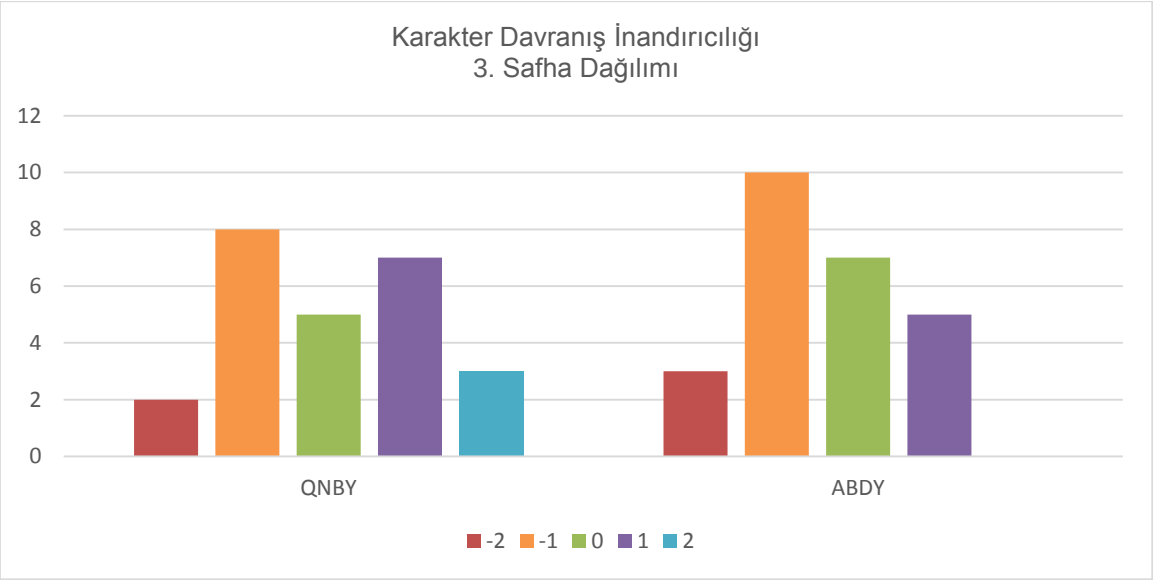
Karakteri öldürme zorluğuna QNBY için %52 oranında olumsuz cevap verilirken %12 oranında olumlu ve %36 oranında orta cevabı verilmiştir. ABDY de ise %56 oranında olumsuz cevap verilirken % 44 oranında orta cevabı verilmiştir.



Şekil 5.8. 2.Safha Oyunun Oynanabilirlik Seviyesi

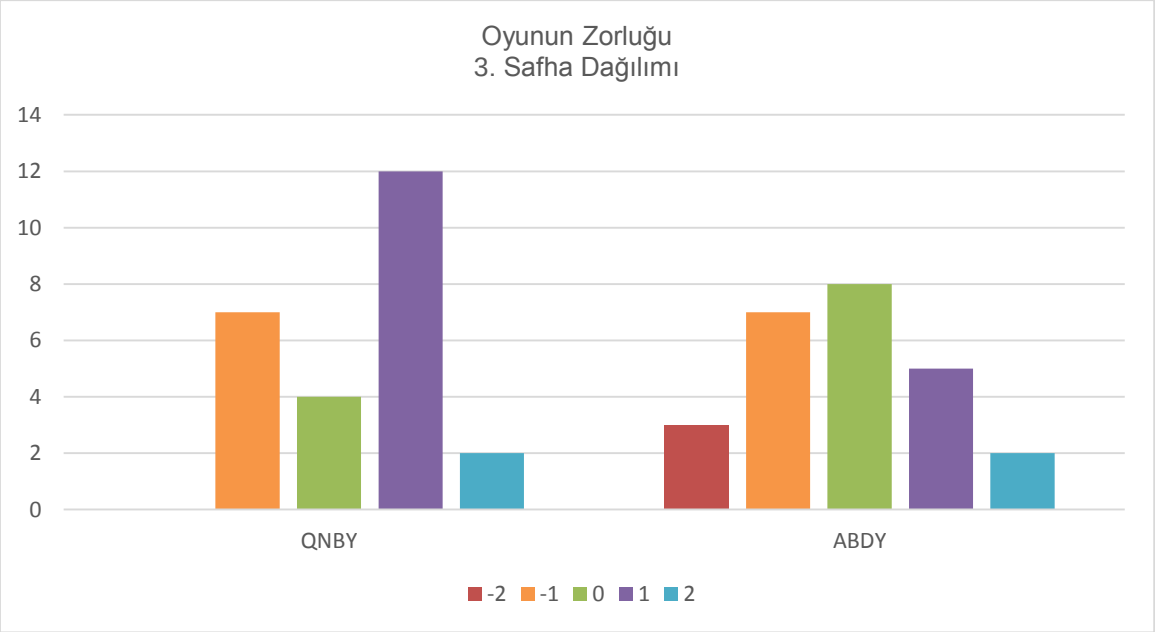
Oyunun oynanabilirlik seviyesine QNBY için %44 oranında olumsuz cevap verilirken %4 oranında olumlu ve %52 oranında orta cevabı verilmiştir. ABDY de ise %52 oranında olumsuz cevap verilirken %48 oranında orta cevabı verilmiştir Buradan da anlaşıldığı üzere 2.Safha'da QNBY'nin ABDY'den biraz daha iyi sonuçlar aldığı gözlemlenmiştir. Ayrıca 2.Safha'nın 1.Safha'ya göre daha iyi olduğu gözlemlenmiştir.

3.Safha'da hazırlanan oyun 5 farklı kişiye oynatılarak elde edilmiştir. Burada QNBY ve ABDY için daha önceden 1.Safha'da ve 2.Safha'da oynayan 5 farklı kişiden, toplamda 10 farklı kişiden toplanan veri kullanılmaktadır. Sonuç olarak, QNBY için çok kötü ve kötü olarak verilen cevap sayısı hem ABDY'ye göre azalmış hem de 1.Safha ve 2.Safhaya göre daha az tercih edilmiştir. Orta ve çok iyi olarak verilen cevap sayısı ABDY'yi geçtiği görülmektedir. Ancak asıl göze çarpan iyi olarak verilen cevap sayısındaki ciddi artıştır.



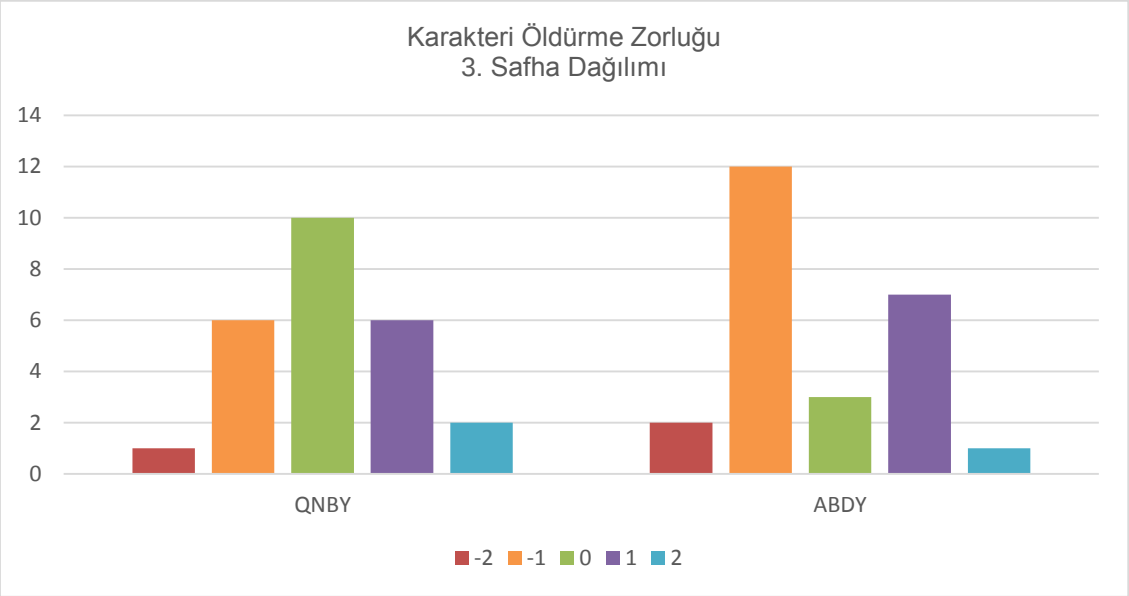
Şekil 5.9. 3.Safha Karakter Davranış İnandırıcılığı

3.Safha'da QNBY için karakterin davranış inandırıcılığına %40 oranında olumsuz cevap verilirken %40 oranında olumlu ve %20 oranında orta cevabı verilmiştir. ABDY de ise %52 oranında olumsuz cevap verilirken %20 oranında olumlu ve %28 oranında orta cevabı verilmiştir.



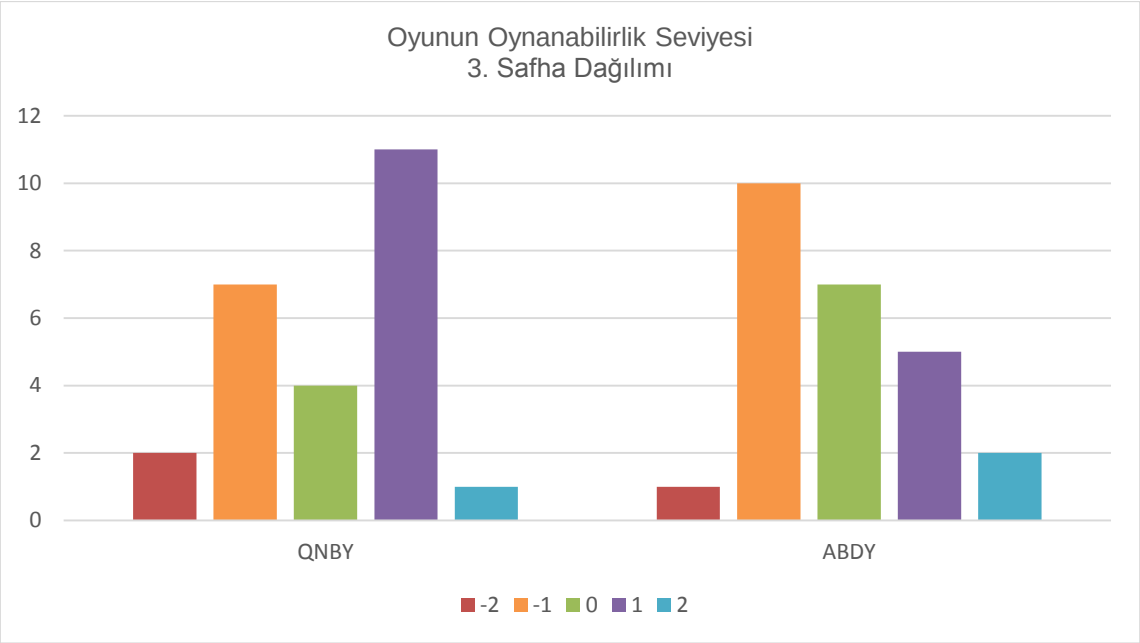
Şekil 5.10. 3.Safha Oyunun Zorluğu

3.Safha'da oyunun zorluğuna %28 oranında olumsuz cevap verilirken %56 oranında olumlu ve %16 oranında orta cevabı verilmiştir. ABDY de ise %40 oranında olumsuz cevap verilirken %28 oranında olumlu ve % 32 oranında orta cevabı verilmiştir.



Şekil 5.11. 3.Safha Karakteri Öldürme Zorluğu

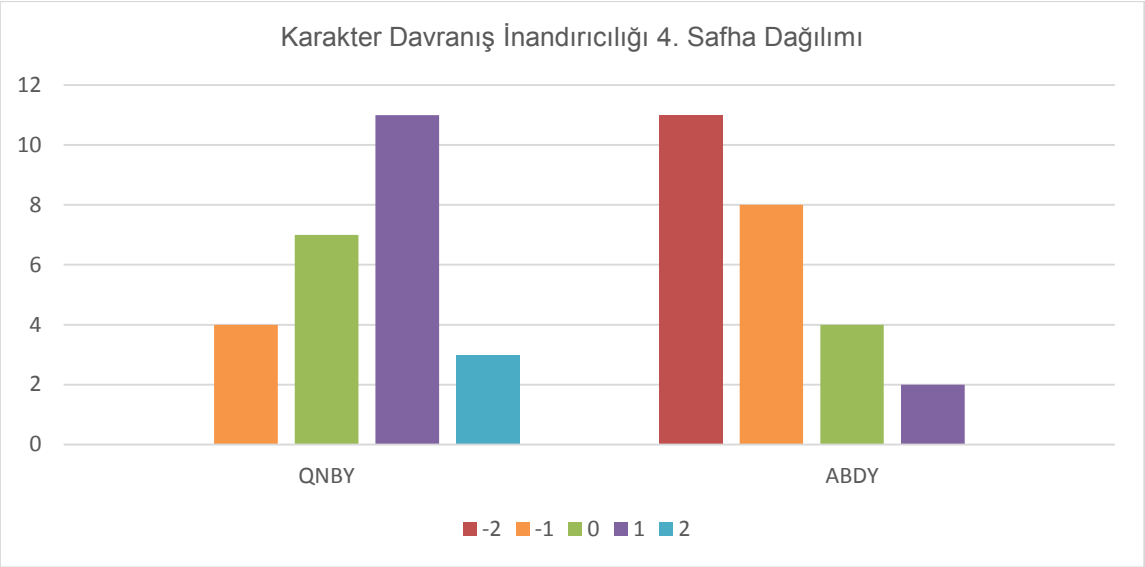
Karakteri öldürme zorluğuna QNBY için %28 oranında olumsuz cevap verilirken %32 oranında olumlu ve %40 oranında orta cevabı verilmiştir. ABDY de ise %56 oranında olumsuz cevap verilirken %32 oranında olumlu ve % 12 oranında orta cevabı verilmiştir.



Şekil 5.12. 3.Safha Oyunun Oynanabilirlik Seviyesi

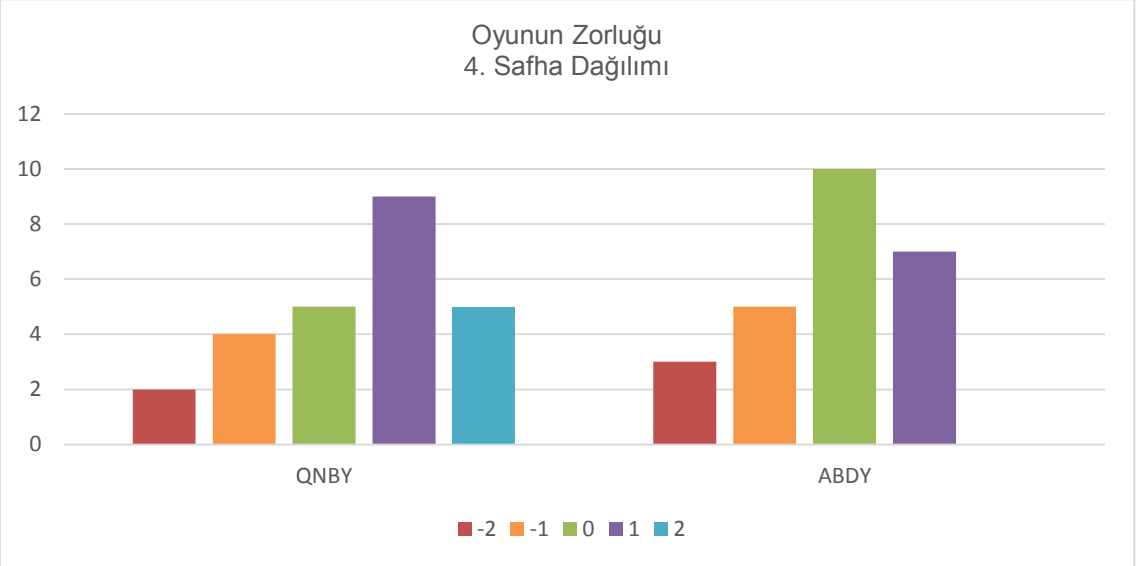
Oyunun oynanabilirlik seviyesine QNBY için %36 oranında olumsuz cevap verilirken %48 oranında olumlu ve %16 oranında orta cevabı verilmiştir. ABDY de ise %44 oranında olumsuz cevap verilirken %28 oranında olumlu ve %28 oranında orta cevabı verilmiştir. Buradan da anlaşıldığı üzere 3.Safha'da QNBY, ABDY'den daha iyi sonuçlar almıştır. Bununla birlikte 1. ve 2.Safha'ya göre de daha iyi olduğu gözlemlenmiştir.

4.Safha'da hazırlanan oyun 5 farklı kişiye oynatılarak elde edilmiştir. Burada QNBY ve ABDY için daha önceden 1.Safha, 2. Safha ve 3.Safha'da oynayan 5er kişiden toplamda 15 kişiden toplanan veri kullanılmaktadır. Sonuç olarak, QNBY çok kötü olarak verilen cevapta ciddi bir azalış göze çarpmaktadır. İyi ve çok iyi olarak verilen cevap sayısında da ciddi bir artış gözlemlenmiştir.



Şekil 5.13. 4.Safha Karakter Davranış İnandırıcılığı

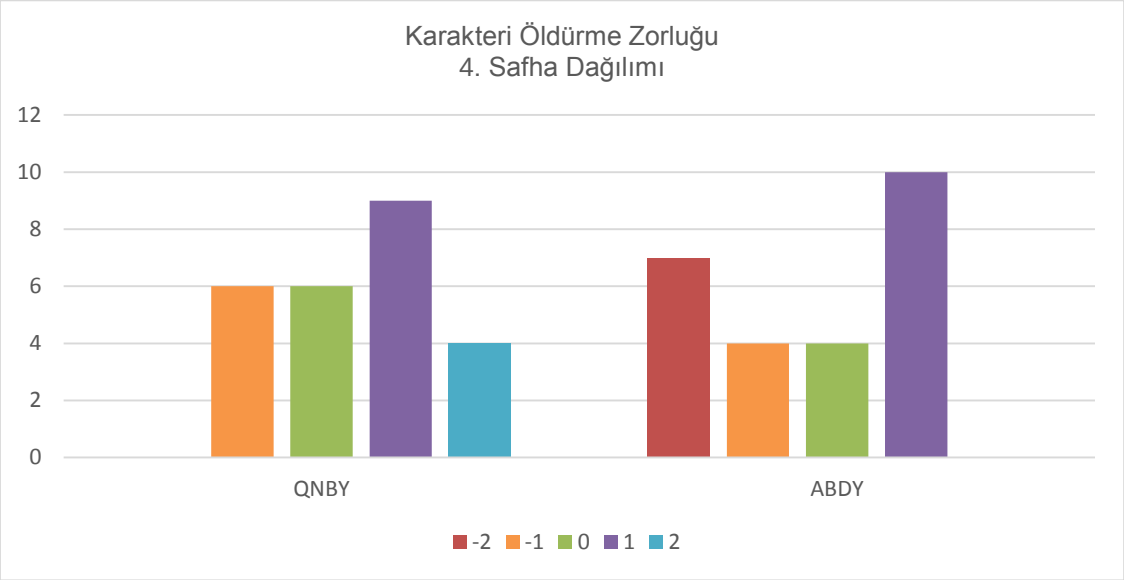
4.Safha'da karakterin davranış inandırıcılığına %16 oranında olumsuz cevap verilirken %56 oranında olumlu ve %28 oranında orta cevabı verilmiştir. ABDY de ise %76 oranında olumsuz cevap verilirken %8 oranında olumlu ve %16 oranında orta cevabı verilmiştir.



Şekil 5.14. 4.Safha Oyunun Zorluğu

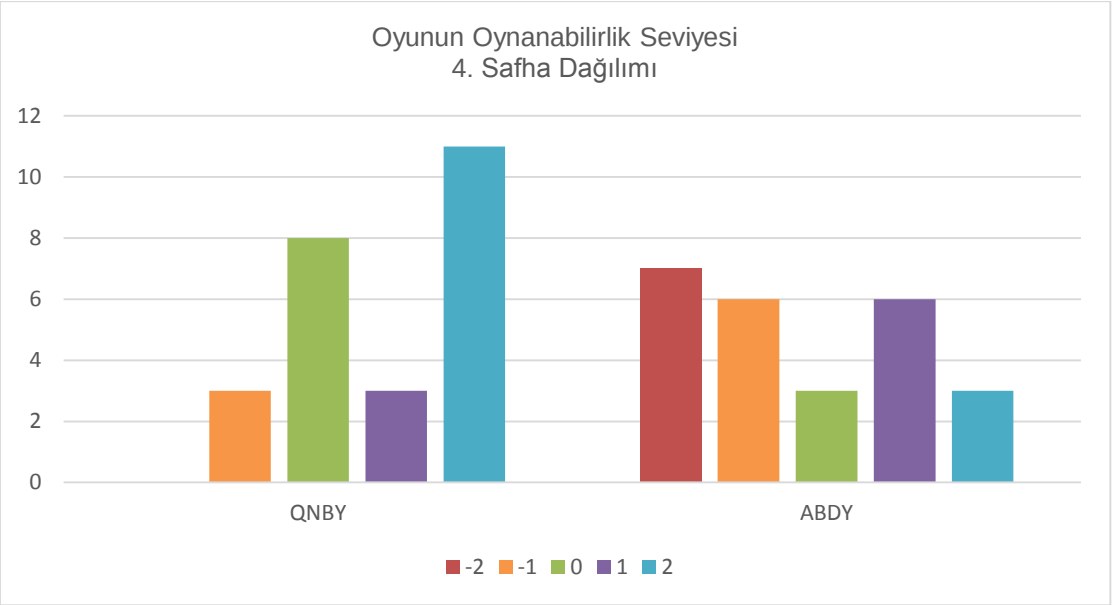
4.Safha'da oyunun zorluğuna %32 oranında olumsuz cevap verilirken %56 oranında olumlu ve %12 oranında orta cevabı verilmiştir. ABDY de ise %32

oranında olumsuz cevap verilirken %28 oranında olumlu ve %40 oranında orta cevabı verilmiştir.



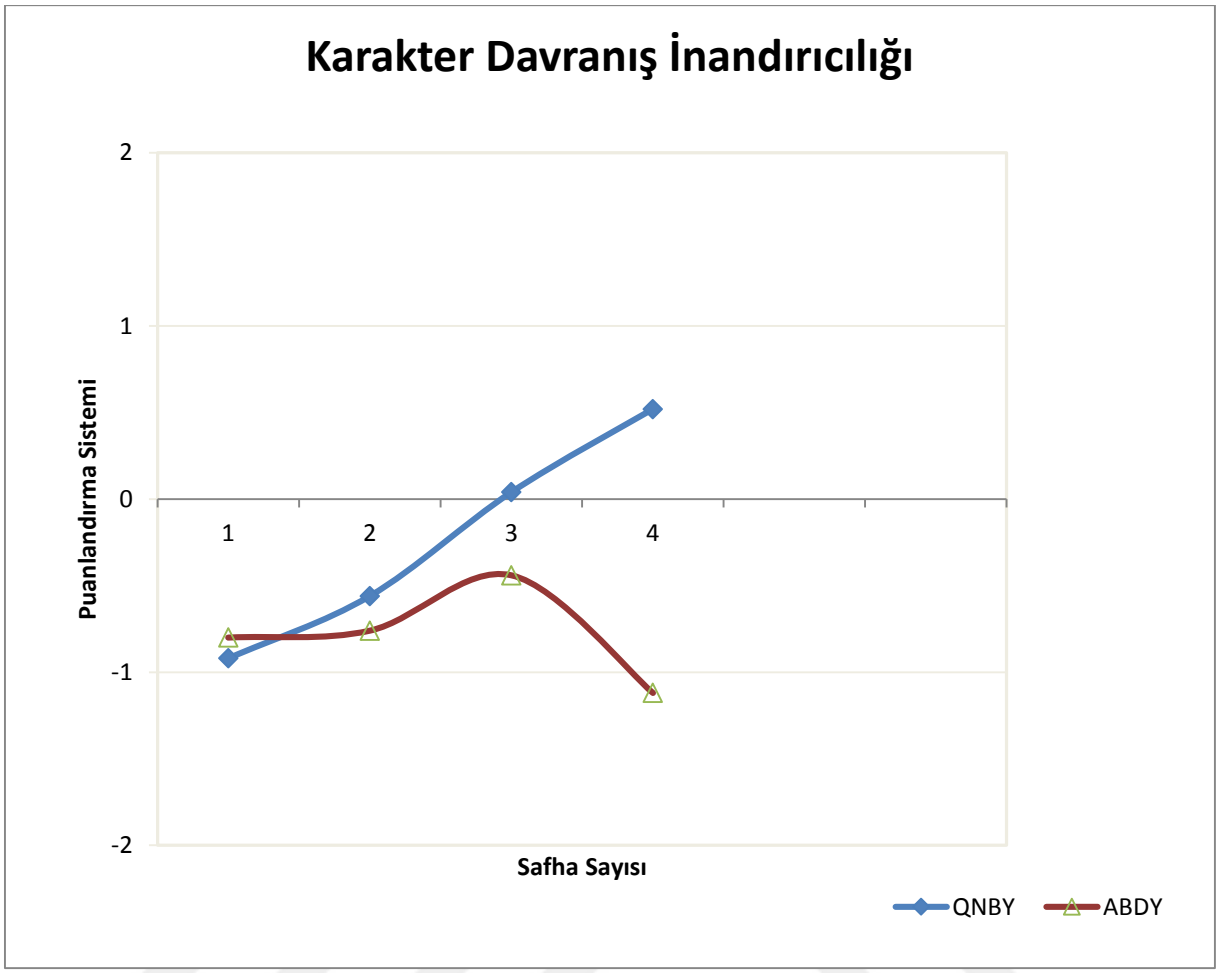
Şekil 5.15. 4.Safha Karakteri Öldürme Zorluğu

Karakteri öldürme zorluğuna QNBY için %24 oranında olumsuz cevap verilirken %52 oranında olumlu ve %24 oranında orta cevabı verilmiştir. ABDY de ise %44 oranında olumsuz cevap verilirken %40 oranında olumlu ve %16 oranında orta cevabı verilmiştir.



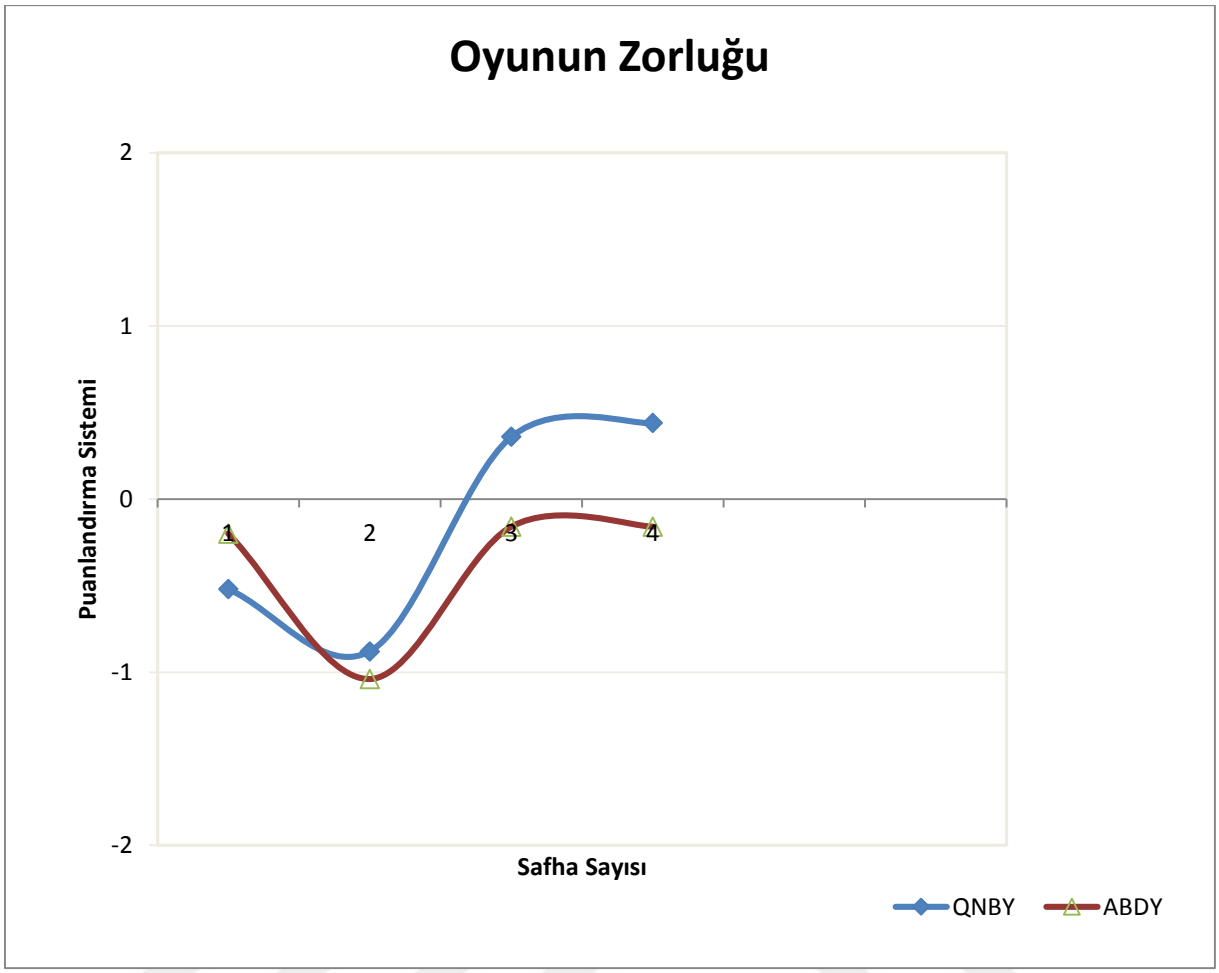
Şekil 5.16. 4.Safha Oyunun Oynanabilirlik Seviyesi

Oyunun oynanabilirlik seviyesine QNBY için %12 oranında olumsuz cevap verilirken %56 oranında olumlu ve %32 oranında orta cevabı verilmiştir. ABDY de ise %52 oranında olumsuz cevap verilirken %36 oranında olumlu ve % 12 oranında orta cevabı verilmiştir. 4.Safha'da QNBY, ABDY' ye ciddi anlamda fark attığı görülmüştür. Aynı şekilde diğer safhalara göre çok daha iyi olduğu gözlemlenmiştir.



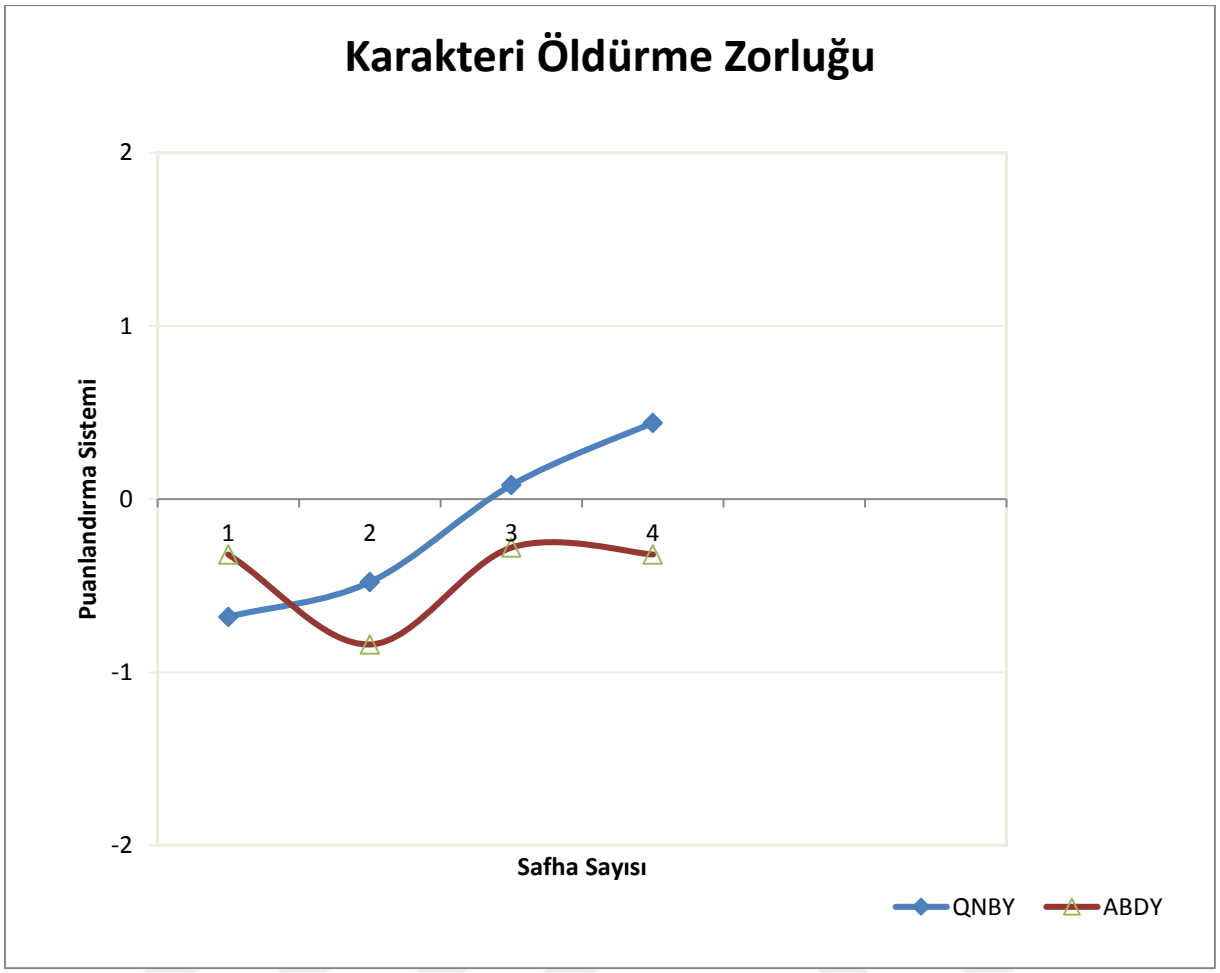
Şekil 5.17 Karakter Davranış İnandırıcılık Değişimi

Yukarıdaki grafik karakter davranış inandırıcılığının safhalara göre değişimini ifade etmektedir. Hem QNBY için hem de ABDY için her safhada farklı 5 farklı kişiye 5 oyun oynatılmış ve karakter davranış inandırıcılığına verilen puanların safhalara göre ortalaması alınmıştır. Bu bilgiler doğrultusunda yukarıdaki grafik elde edilmiştir. Sonuç olarak, hazırlanan öğrenme metodunda safhalara göre karakter davranış inandırıcılığının arttığı görülmüştür. 1.Safha için kötülerde seyrederken giderek artış göstermiş ve son safhada iyi seviyesine kadar ilerleme göstermiştir. Yine ABDY'ye 3.Safha'dan itibaren ciddi bir fark göze çarpmaktadır. ABDY ise daha çok kötü seviyesiyle orta seviyesi arasında değerler aldığı gözlemlenmiştir.



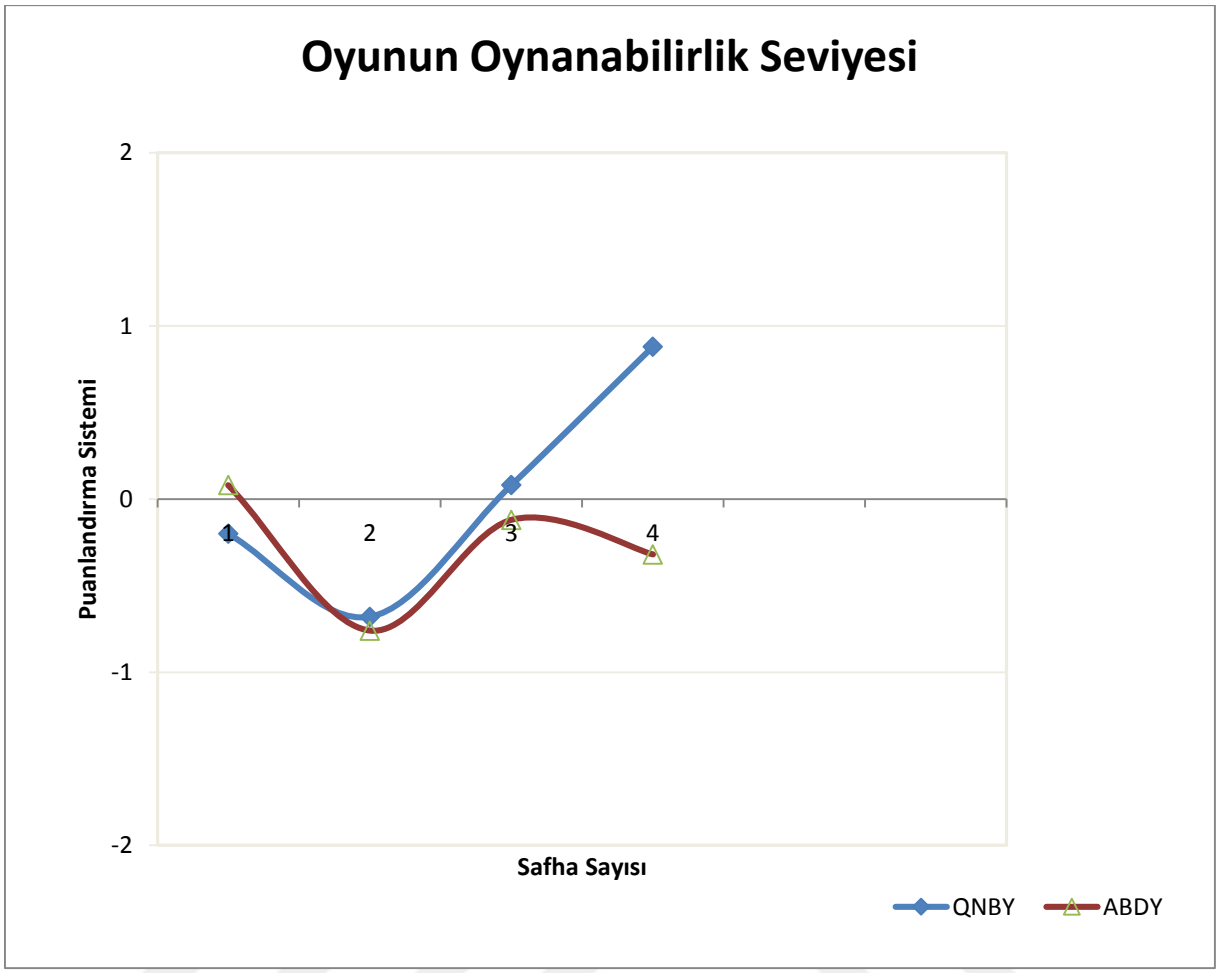
Şekil 5.18 Oyunun Zorluk Değişimi

Yukarıdaki grafik oyunun zorluğunun safhalara göre değişimini ifade etmektedir. Hem QNBY için hem de ABDY için oyunun zorluğuna verilen puanların safhalara göre ortalaması alınmıştır. Bu bilgiler doğrultusunda yukarıdaki grafik elde edilmiştir. Sonuç olarak, QNBY kullanıldığında safhalara göre oyunun zorluğunun arttığı görülmüştür. 1.Safha için kötülerde seyrederken giderek artış göstermiş ve son safhada orta ile iyi seviyesi arasında kalmıştır. ABDY'ye göre 3.Safha'dan itibaren pozitif yönde bir fark göze çarpmaktadır. ABDY ise kötü ile orta seviyeleri arasında devam ettiği gözlemlenmiştir.



Şekil 5.19 Karekteri Öldürme Zorluk Değişimi

Yukarıdaki grafik karakteri öldürme zorluğunun safhalara göre değişimini ifade etmektedir. Hem QNBY için hem de ABDY için karakteri öldürme zorluğuna verilen puanların safhalara göre ortalaması alınmıştır. Bu bilgiler doğrultusunda yukarıdaki grafik elde edilmiştir. Sonuç olarak, QNBY'de safhalara göre karakteri öldürme zorluğunun arttığı görülmüştür. 1.Safha'da kötü seviyelerine yakın seyrederken giderek artış göstermiş ve son safhada orta ile iyi seviyeleri arasında devam etmiştir. ABDY'ye göre 3.Safha'dan itibaren pozitif yönde bir artış göze çarpmaktadır. ABDY'nin ise kötü ile orta seviyeleri arasında puanlar aldığı gözlemlenmiştir.



Şekil 5.20 Oyunun Oynanabilirlik Seviyesi Değişimi

Yukarıdaki grafik oyunun oynanabilirlik seviyesinin safhalara göre değişimini ifade etmektedir. Hem QNBY için hem de ABDY için oyunun oynanabilirlik seviyesine verilen puanların safhalara göre ortalaması alınmıştır. Bu bilgiler doğrultusunda yukarıdaki grafik elde edilmiştir. Sonuç olarak, QNBY için safhalara göre oyunun oynanabilirlik seviyesinin arttığı görülmüştür. 1.Safha için kötü seviyelerine yakın seyrederken giderek artış göstermiş ve son safhada iyi seviyelerine kadar yaklaştığı gözlemlenmiştir. ABDY ile kıyaslandığında yine 3.Safha'dan itibaren pozitif yönde bir fark göze çarpmaktadır. ABDY'nin ise kötü ile orta seviyeleri arasında ilerlediği gözlemlenmiştir.

5.3. Deney Sonuç Değerlendirmesi

Table 5.1. Oyun Özelliklerinin Safhalara Göre Puanlandırma Dağılımı

SAFHA	YÖNTEM	PUAN	KDİ	OZ	KÖZ	OOS
1.	QNBY	(2,1)	% 0	% 16	% 16	% 44
		0	% 28	% 40	% 16	% 12
		(-2,-1)	% 72	% 44	% 68	% 44
	ABDY	(2,1)	% 4	% 32	% 20	% 44
		0	% 36	% 38	% 32	% 24
		(-2,-1)	% 60	% 32	% 48	% 32
2.	QNBY	(2,1)	% 8	% 0	% 12	% 4
		0	% 44	% 36	% 36	% 52
		(-2,-1)	% 48	% 64	% 52	% 44
	ABDY	(2,1)	% 12	% 8	% 0	% 0
		0	% 28	% 24	% 44	% 48
		(-2,-1)	% 60	% 68	% 56	% 52
3.	QNBY	(2,1)	% 40	% 56	% 32	% 48
		0	% 20	% 16	% 40	% 16
		(-2,-1)	% 40	% 28	% 28	% 36
	ABDY	(2,1)	% 20	% 28	% 32	% 28
		0	% 28	% 32	% 12	% 28
		(-2,-1)	% 52	% 40	% 56	% 44
4.	QNBY	(2,1)	% 56	% 56	% 52	% 56
		0	% 28	% 12	% 24	% 32
		(-2,-1)	%16	% 32	% 24	% 12
	ABDY	(2,1)	% 8	% 28	% 40	% 36
		0	% 16	% 40	% 16	% 12
		(-2,-1)	% 76	% 32	% 44	% 52

İnsanların verdikleri puanlar göz önünde bulundurularak aşağıdaki sonuçlara varılmıştır. QNBY 1.Safha'da %58 oranında olumsuz, %12,6 oranında olumlu ve % 29,4 oranında orta cevabı verilirken son safhada %24 oranında olumsuz, %54 oranında olumlu ve %22 oranında orta cevabı verilmiştir. ABDY ise 1.Safha'da %47,3 oranında olumsuz, %18 oranında olumlu ve % 34,7 oranında orta cevabı verilirken, son safhada %50 oranında olumsuz, %28,7 oranında olumlu ve % 21,3

oranında orta cevabı verilmiştir. Başlangıçta olumlu oy oranı ABDY' den %30 daha az iken son safhada 2 katına çıkmıştır. Buradan da QNBY'nin safhalara göre ilerleme gösterdiği kaydedilmiştir. Bunun yanında karakter davranış inandırıcılığı ilk safhada kötü sonuçlar alırken son safhada giderek iyi seviyesine yaklaşmıştır. Aynı şekilde karakteri öldürme zorluğu, oyun zorluğu ve oyunun oynanabilirlik seviyeleri de başlangıç safhasında kötü seviyelerdeyken son safhalara doğru iyi seviyelere yaklaştıkları gözlemlenmiştir. ABDY'de ise belirlenen bütün özellikler tüm safhalarda kötü ve orta seviyeleri arasında seyretmiştir.

5.4. Tasarlanan FPS Oyunu

Şekil 5.21 Veri Toplanması için tasarlanan oyun başlangıcı

Şekil 5.22 Hangar Alanı

Şekil 5.23 Yürüme Durumu

Oyun esnasında öğrenme sonucu YZK‘nin geçtiği durumlardan biri “Yürüme” durumudur. Bu durumda karakter yürüme hareketi yaparak hedefini takip eder.

Şekil 5.24 Bekleme_01 Durumu

Oyun esnasında öğrenme sonucu YZK‘nin geçtiği durumlardan biri “Bekleme _01” durumudur. Bu durumda karakter hareketsiz kalarak ve karşıya bakarak hedefin tepki göstermesini bekler.

Şekil 5.25 Bekleme_02 Durumu

Oyun esnasında öğrenme sonucu YZK'nin geçtiği durumlardan biri "Bekleme _02" durumudur. Bu durumda karakter hareketsiz kalır, kafasını sağa ve sola çevirerek etrafı gözetlemeye devam eder.

Şekil 5.26 Koşma Durumu

Oyun esnasında öğrenme sonucu YZK'nin geçtiği durumlardan biri "Koşma" durumudur. Bu durumda karakter koşar adımlarla hedefe doğru hızlı bir şekilde ilerlemeye devam eder.

Şekil 5.27 Kaçış Durumu

Oyun esnasında öğrenme sonucu YZK‘nin geçtiği durumlardan biri “Kaçış” durumudur. Bu durumda karakter koşar adımlarla hedeften uzaklaşır. Bu durum daha çok karakterin savunma mekanizmasını öğrenmesi için tasarlanıp eklenmiştir.

Şekil 5.28 Atak_01 Durumu

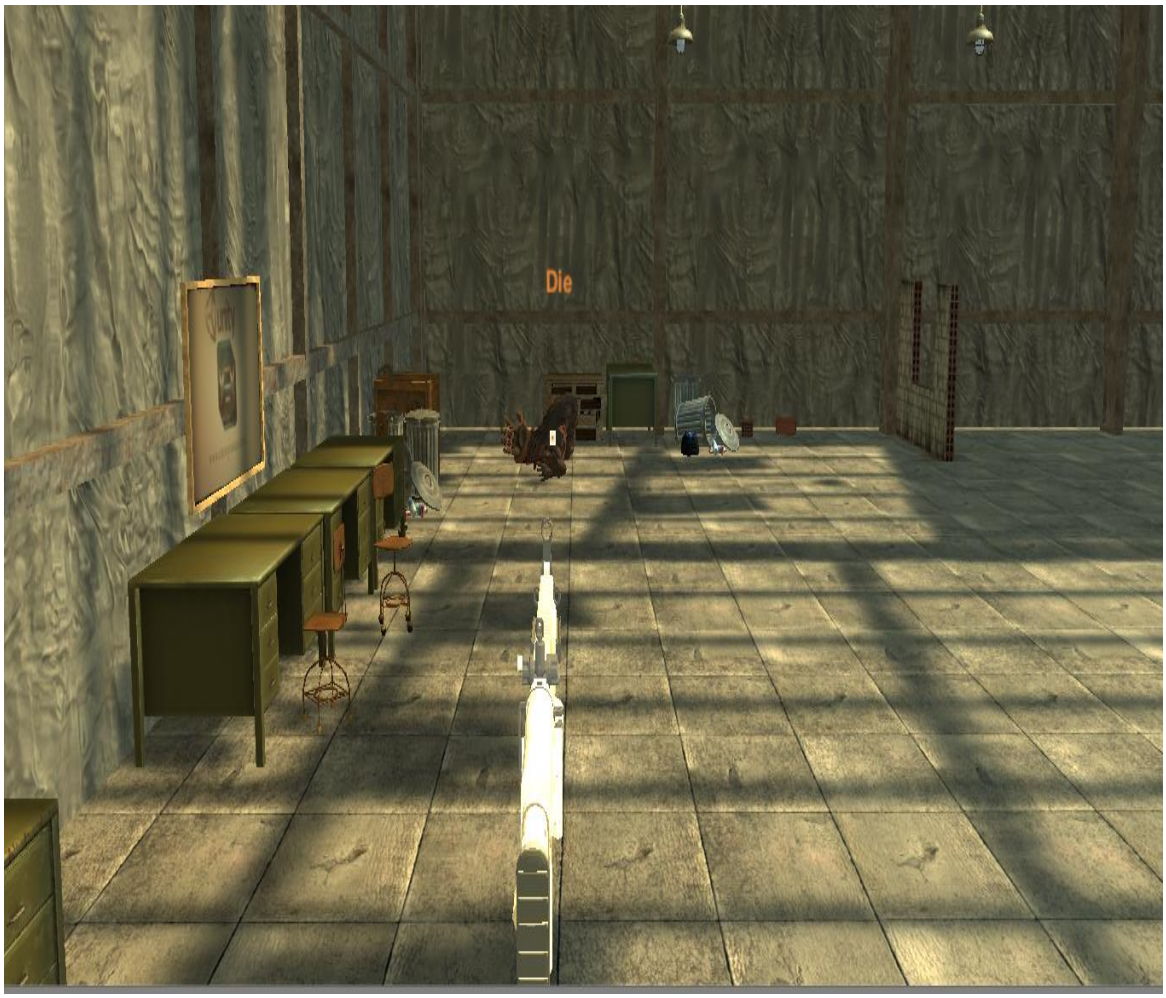
Oyun esnasında öğrenme sonucu YZK'nin geçtiği durumlardan biri "Atak_01" durumudur. Bu durumda karakter hedefe belirli bir mesafe yaklaştıktan sonra zıplayarak rakibine yumruk darbeleri atmaya çalışır ve rakibini zayıf düşürüp öldürmeyi hedefler.

Şekil 5.29 Atak_02 Durumu

Oyun esnasında öğrenme sonucu YZK'nin geçtiği durumlardan biri "Atak_02" durumudur. Bu durumda karakter hedefe belirli bir mesafe yaklaştıktan sonra Atak_01'e göre daha istikrarlı bir şekilde karşıya yumruk darbeleri atmaya devam eder ve rakibini zayıf düşürüp öldürmeyi hedefler.

Şekil 5.30 Zıplama Durumu

Oyun esnasında öğrenme sonucu YZK‘nin geçtiği durumlardan biri “Zıplama” durumudur. Bu durumda karakter zıplama hareketi yaparak belirlediği stratejiyi izler.



Şekil 5.31 Ölme Durumu

Oyun esnasında öğrenme sonucu YZK‘nin geçtiği durumlardan biri “Ölme” durumudur. Bu durumda karakter aldığı darbeler sonucu sağlığı giderek azalır ve sağlık değeri sıfır olduğunda bu duruma geçer. Bu durum oyun sonu olarak yani bitiş durumu olarak tanımlanmaktadır.

Tasarlanan FPS oyununda yer alan durum bilgileri yukardaki gibidir. Burada oyun başlangıcından sonuna kadar YZK‘nin geçmiş olduğu durum bilgileri tutulmuştur. Başlangıç durumundan başlayan YZK‘nin oyun bitimine kadar hangi durumlardan geçerek yani hangi yolu izlediği bilgisi elde edilmiştir. Bu bilgi, başlangıç durumundan oyunun bitmesi için gerekli koşulların sağlandığı bitiş durumuna geçene kadar sergilemiş olduğu hareketler listesini içerir. Başlangıç durumu tercihe göre bazen “Yürüme” bazen de “Bekleme_02” seçilmiştir. Bitiş durumu ise 3 farklı koşula göre belirlenmektedir. Bunlardan ilki YZK‘nin sağlığının sıfır olması yani “Ölüm” durumuna geçmesidir. İkincisi hedefin sağlık değerinin sıfır olması

oyunun sonlanmasını yani bitiş durumunu göstermektedir. Sonuncu olarak önceden belirlenmiş olan zaman değerinin bitmesi yine bitiş durumunu ifade etmektedir. Bu bitiş durumlarından hedefin sağlık değerinin sıfır olması durumunda ve zaman değerinin bitmesi durumuna ödül değeri olarak 100 verilir. Bunların dışında kalan bitiş durumuna ve aradaki bütün durumlara başlangıçta 0 ödül değeri verilmektedir.

5.5. Elde Edilen Örnek Veri kümesi İçeriği

Sınıflandırma aşamasında kullanılan özellikler aşağıdaki tabloda gösterildiği gibidir. Bunlar oyun ve test verisi toplama aşamasında kullanılan yapay zekâ karakterine ait özelliklerdir. Özellik sayısının az olması nedeniyle herhangi bir özellik seçim işlemi yapılmamıştır.

Tablo 5.2. Elde Edilen Veri kümesi Örneği

Sağlık	Rakip Uzaklığı	Hız Değeri	Atak Gücü	Kalan Zaman	QValue	Durum
60	60	70	70	35	90	Atak_01
90	80	60	80	23	90	Atak_02
60	70	70	90	16	70	Koşma
50	30	60	30	14	20	Bekleme_01
20	50	80	50	21	40	Kaçış
40	50	40	60	45	40	Bekleme_01
30	40	20	50	24	30	Kaçış
50	50	60	50	17	60	Yürüme

6.SONUÇ

Bu tez çalışmasında hedeflenen amaç daha inandırıcı ve eğlenceli oyun karakterlerinin üretimine yardımcı olmak ve bu sayede oyunu oynayan insanların oyuna olan ilgisini taze tutmaktır. Bu amaç doğrultusunda oyunda yer alan yapay zekâ karakterinin daha ilgi çekici olması için araştırmalar yapılarak karakterin davranışları geliştirilmeye çalışılmıştır. Araştırmalarda karşılaşılan bazı yaklaşımlarda, oyuncunun oyun içerisindeki tepkileri, oyuncu ile yapay zekâ karakterinin etkileşimleri toplanarak karakterin gelişiminde kullanılmış ve oyuncunun oyun genelindeki davranışı taklit edilerek karakterin gelişimi sağlanmıştır [27]. Bununla birlikte daha da temele inilerek yapay zekâ karakterinin sadece dövüşmesi veya sadece bomba kurabilmesi için öğrenme algoritmaları kullanılmıştır [2]. Ayrıca ateş etme stratejisini belirlemek, silah veya cephaneyi doğru kullanmasını öğrenmesi için çeşitli modellemeler yapılmıştır [39]. Bu tez çalışmasında ise karakterin geçmişte oyuncuya karşı sergilediği davranışlardan bilgi çıkartılarak karakterin davranış öğrenme işlemi geliştirilmeye çalışılmıştır. Geçmişteki başarılı davranışları kaydedilerek gelecekte daha az hataya düşmesi hedeflenmiştir. Bunu etkili bir şekilde uygulayabilmek için makine öğrenmesi tekniklerinden biri olan pekiştirmeli öğrenme yaklaşımından faydalanılmıştır. Buna ek olarak sınıflandırma işlemi için toplanan veriler üzerinde Naive Bayes yaklaşımı kullanılmıştır ve daha sonra hazırlanan oyun üzerinde uygulanmıştır. Hazırlanan senaryo doğrultusunda toplam 20 insana 100 oyun oynatılarak veriler toplanmış ve bu verilerden çeşitli çıkarımlar elde edilmiştir. Daha doğru ve sağlıklı sonuçlar almak için bu süreç safhalara bölünmüş ve açgözlü benzeri bir yaklaşım ile karşılaştırılmıştır. Oyunun ve üretilen karakterin nasıl sonuçlar verdiğini gözlemlemek için insanlardan oyun sonunda oyunla ilgili anketi doldurmaları istenmiştir. Burada oyun ve karakter hakkında insanların görüşleri alınmıştır. İnsanların görüşleri doğrultusunda, karakterin safhalar ilerledikçe daha mantıklı davranışlar sergilediği görülmüştür. Karakterin gelişimini ölçmek için kullanılan karakterin davranış inandırıcılığı, karakterin öldürme zorluğu ve oyunun oynanabilirlik derecesi, veri miktarı arttıkça ilk safhaya göre artarak %56 oranlarına ulaşmıştır. Aynı zamanda bu durumun oyuna olan etkisini ölçmek için kullanılan oyunun zorluğu da %52 oranlarına ulaşmıştır.

Sonuç olarak insanlara karşı oynatılarak toplanan veri miktarı arttıkça, QNBY algoritmasını kullanan karakterin ABDY algoritmasını kullanan karaktere göre daha doğru kararlar alarak daha ilgi çekici davranışlar sergilediği tespit edilmiştir. Aynı zamanda QNBY algoritmasını kullanan karakterin daha ilgi çekici davranışlar sergilemesi, oyunu da daha ilgi çekici hale getirmiştir.

KAYNAKLAR

[1] M. E. Harmon and S. S. Harmon, Reinforcement learning: a tutorial, *WL/ Agriculture and Agri-Food Canada, Wright-Patterson Air Force Base Ohio,* vol. 45433, pp. 237–285, **1996**.

[2] P. G. Patel, N. Carver, and S. Rahimi, Tuning computer gaming agents using Q-learning, *2011 Fed. Conf. Comput. Sci. Inf. Syst.*, pp. 581–588, **2011**.

[3] C. Thurau, T. Paczian, and C. *Bauckhage,* Is bayesian imitation learning the route to believable gamebots? , *1st Int. North-American Conf. Intell. Games Simulation, Game-On 'NA 2005*, no. November 2015, pp. 3–9, **2005**.

[4] A. N. Meltzoff, *Born to learn: What infants learn from watching us*, *Role Early Exp. Infant Dev.*, pp. 145–164, **1999** .

[5] Senemoğlu, N. Gelisim Öğrenme ve Öğretim, Gazi Kitapevi, Ankara, 10 – 25, **2005**.

[6] Bacanlı, H. *Gelisim ve Öğrenme*, Nobel Yayın ve Dağıtım, Ankara, 55, **2005**.

[7] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. P. Johnson, and B. Tomlinson, *Integrated Learning for Interactive Synthetic Characters*, *Proc. 29th Annu. Conf. Comput. Graph. Interact. Tech.*, p. 417, **2002**.

[8] J. Laird, *Design goals for autonomous synthetic characters*, Ann Arbor, vol. 1001, pp. 48109–2110, **2000**.

[9] M. A. Hasegawa-Johnson, J.-T. Huang, and X. Zhuang, *Semi-supervised learning for speech and audio processing*, *J. Acoust. Soc. Am.*, vol. 130, no. 4, pp. 2408–2408, **2011**.

[10] G. Brockman *et al.*, *OpenAI Gym*, pp. 1–4, **2016**.

[11] Hu, J. Reinforcement learning explained. O'Reilly Media, https://www.oreilly.com/ideas/reinforcement-learning-explained, (Nisan, **2018**).

[12] En.wikipedia.org.,Control theory, [https://en.wikipedia.org/wiki/Control_theory , (Mayıs, **2018**).

[13] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics :,” Int. J. Rob. Res., vol. 32, no. 11, pp. 1238–1274, **2015**.

[14] R. Giryes and M. Elad, “Reinforcement Learning: A Survey,” Eur. Signal Process. Conf., pp. 1475–1479, 2011.

[15] En.wikipedia.org. , Markov decision process, https://en.wikipedia.org/wiki/Markov_decision_process (Ocak, **2018**).

[16] T. C. H. John, E. C. Prakash, and N. S. Chaudhari, “Strategic Team AI Path Plans: Probabilistic Pathfinding,” *Int. J. Comput. Games Technol.*, vol. 2008, pp. 1–7, **2008**.

[17] S. Agrawal and N. Goyal, “Analysis of Thompson Sampling for the multi-armed bandit problem,” vol. 23, pp. 1–26, **2011**.

[18] V. Kuleshov and D. Precup, “Algorithms for multi-armed bandit problems,” *Unpublished*, vol. 1, pp. 1–32, **2010**.

[19] M. P. Eladhari and M. Mateas, “Semi-autonomous avatars in world of minds,” *Proc. 2008 Int. Conf. Adv. Comput. Entertain. Technol. - ACE ’08*, p. 201, **2008**.

[20] A. Klöckner, “Interfacing Behavior Trees with the World Using Description Logic,” *AIAA Guid. Navig. Control Conf.*, **2013**.

[21] S. E. Gaudl, J. C. Osborn, and J. J. Bryson, “Learning from play: Facilitating character design through genetic programming and human mimicry,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9273, pp. 292–297, **2015**.

[22] J. Yao, C. Lin, X. Xie, A. J. Wang, and C.-C. Hung, “Path Planning for Virtual Human Motion Using Improved A* Star Algorithm,” *2010 Seventh Int. Conf. Inf. Technol. New Gener.*, pp. 1154–1158, **2010**.

[23] A. G. Bayrak and F. Polat, “Formation preserving path finding in 3-D terrains,” *Appl. Intell.*, vol. 36, no. 2, pp. 348–368, **2012**.

[24] P. Su, Y. Li, Y. Li, and S. C. K. Shiu, “An auto-adaptive convex map generating path-finding algorithm: Genetic Convex A*,” *Int. J. Mach. Learn. Cybern.*, vol. 4, no. 5, pp. 551–563, **2013**.

[25] D. Livingstone, “Turing’s test and believable AI in games,” *Comput. Entertain.*, vol. 4, no. 1, p. 6, **2006**.

[26] P. Hingston, “A new design for a Turing Test for Bots,” *Proc. 2010 IEEE Conf. Comput. Intell. Games, CIG2010*, no. 2010, pp. 345–350, **2010**.

[27] G. N. Yannakakis, “Game AI Revisited Categories and Subject Descriptors,” *Most*, pp. 285–292, **2012**.

[28] W. Jaśkowski, K. Krawiec, and B. Wieloch, “Evolving strategy for a probabilistic game of imperfect information using genetic programming,” *Genet. Program. Evolvable Mach.*, vol. 9, no. 4, pp. 281–294, **2008**.

[29] J. E. Laird and J. C. Duchi, “Creating Human-Like Synthetic Characters with Multiple Skill Levels: A Case Study Using the Soar Quakebot,” *Pap. from 2001 AAAI Spring Symp. Artif. Intell. Interact. Entertain. I*, pp. 54–58, **2001**.

[30] N. Peirce, O. Conlan, and V. Wade, “Adaptive educational games: Providing non-invasive personalised learning experiences,” *Proc. - 2nd IEEE Int. Conf. Digit. Game Intell. Toy Enhanc. Learn. Digit. 2008*, pp. 28–35, **2008**.

[31] S. Feng and A. H. Tan, “Towards autonomous behavior learning of non-player characters in games,” *Expert Syst. Appl.*, vol. 56, pp. 89–99, **2016**.

[32] M. McPartland and M. Gallagher, “Creating a multi-purpose first person shooter bot with reinforcement learning,” *2008 IEEE Symp. Comput. Intell. Games, CIG 2008*, pp. 143–150, **2008**.

[33] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, “Evolutionary dynamics of multi-agent learning: A survey,” *J. Artif. Intell. Res.*, vol. 53, pp. 659–697, **2015**.

[34] B. Tastan and G. Sukthankar, “Learning Policies for First Person Shooter Games Using Inverse Reinforcement Learning,” *Proc. Seventh AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*, no. October, pp. 85–90, **2011**.

[35] M. Mcpartland and M. Gallagher, “Learning to be a Bot : Reinforcement Learning in Shooter Games,” *Proc. Fourth Artif. Intell. Interact. Digit. Entertain. Conf.*, pp. 78–83, **2008**.

[36] D. Wang, B. Subagdja, A. Tan, and G. Ng, “Creating human-like autonomous players in real-time first person shooter computer games,” *Proc. 21st Annu. Conf. Innov. Appl. Artif. Intell.*, pp. 173–178, **2009**.

[37] D. Wang and A. H. Tan, “Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game,” *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 2, pp. 123–138, **2015**.

[38] J. Macglashan, R. Loftin, M. L. Littman, D. L. Roberts, and M. E. Taylor, “A Need for Speed : Adapting Agent Action Speed to Improve Task Learning from Non-Expert Humans Categories and Subject Descriptors,” *Aamas 2016*, pp. 957–965, **2016**.

[39] F. G. Glavin and M. G. Madden, “Learning to shoot in first person shooter games by stabilizing actions and clustering rewards for reinforcement learning,” *2015 IEEE Conf. Comput. Intell. Games, CIG 2015 - Proc.*, no. September, pp. 344–351, **2015**.

[40] R. Bonse, W. Kockelkorn, and R. Smelik, “Learning agents in quake iii,” *Science (80-.).*, pp. 1–13, **2004**.

[41] S. Wender and I. Watson, “Applying reinforcement learning to small scale combat in the real-time strategy game StarCraft:Broodwar,” *2012 IEEE Conf. Comput. Intell. Games, CIG 2012*, pp. 402–408, **2012**.

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness,M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, **2015**.

[43] N. Justesen and S. Risi, “Learning macromanagement in starcraft from replays using deep learning,” *2017 IEEE Conf. Comput. Intell. Games, CIG 2017*, pp. 162–169, **2017**.

[44] C. Bauckhage, C. Thurau, and G. Sagerer, “Learning human-like opponent behavior for interactive computer games,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2781, pp. 148–155, **2003**.

[45] K. Lundqvist, “An Approach to Create Realistic Animations in Games,” **2016**.

[46] P. K. K. Loh and E. C. Prakash, “Performance simulations of moving target search algorithms,” *Int. J. Comput. Games Technol.*, vol. 2009, no. 1, **2009**.

[47] G. Synnaeve and P. Bessière, “A Bayesian Model for Plan Recognition in RTS Games applied to StarCraft,” no. November, **2011**.

[48] G. Synnaeve and P. Bessière, “Bayesian modeling of a human MMORPG player,” *AIP Conf. Proc.*, vol. 1305, pp. 67–74, **2010**.

[49] R. Le Hy, A. Arrigoni, P. Bessiére, and O. Lebeltel, “Teaching Bayesian behaviours to video game characters,” *Rob. Auton. Syst.*, vol. 47, no. 2–3, pp. 177–185, **2004**.

[50] B. Peng, R. Loftin, J. MacGlashan, M. L. Littman, M. E. Taylor, and D. L. Roberts, “Language and Policy Learning from Human-delivered Feedback,” *Proc. Mach. Learn. Soc. Robot. Work. (at {ICRA})*, **2015**.

[51] S. Ross STEPHANEROSS *et al.*, “A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes,” *J. Mach. Learn. Res.*, vol. 12, pp. 1729–1770, **2011**.

[52] F. Cao and S. Ray, “Bayesian hierarchical reinforcement learning,” *Adv. Neural Inf. Process. Syst.*, pp. 73–81, **2012**.

[53] D. Ramachandran and E. Amir, “Bayesian inverse reinforcement learning,” *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 2586–2591, **2007**.

[54] R. Dearden, N. Friedman, and S. Russell, “Bayesian Q-learning,” *Proc. fifteenth Natl. Conf. Artif. Intell. Appl. Artif. Intell.*, pp. 761–768, **1998**.

[55] T. Hong, J. Lee, K. E. Kim, P. A. Ortega, and D. Lee, “Bayesian reinforcement learning with behavioral feedback,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2016–January, pp. 1571–1577, **2016**.

[56] P. Pandey, D. Pandey, and S. Kumar, “Reinforcement Learning by Comparing Immediate Reward,” *IJCSIS) Int. J. Comput. Sci. Inf. Secur.*, vol. 8, no. 5, **2010**.

[57] Y. Wang, Q. Xie, A. Ammari, and M. Pedram, “Deriving a near-optimal power management policy using model-free reinforcement learning and Bayesian classification,” *Dac*, no. May 2014, pp. 41–46, **2011**.

[58] T. Fuchida, K. T. Aung, and A. Sakuragi, “A study of Q-learning considering negative rewards,” *Artif. Life Robot.*, vol. 15, no. 3, pp. 351–354, **2010**.

[59] Sutton, R. and Barto, *A. Reinforcement Learning*. Cambridge, Mass.: MIT Press ,**1998**

[60] F. Balducci, C. Grana, and R. Cucchiara, “Affective level design for a role-playing videogame evaluated by a brain–computer interface and machine learning methods,” *Vis. Comput.*, vol. 33, no. 4, pp. 413–427, **2017**.

[61] M. Kwak, D. Casper, and C. Talmage, “an Educational Game for Information Literacy and Student Engagement,” *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, pp. 3616–3625, **2018**.

[62] G. Synnaeve and P. Bessiere, “Special Tactics : a Bayesian Approach to Tactical ere To cite this version : Special Tactics : a Bayesian Approach to Tactical Decision-making,” 2012.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Osman Yılmaz

Doğum Yeri : Ağrı -TÜRKİYE

Medeni Hali : Bekar

E-posta : osmanyilmzz@gmail.com

Adres : Maraşal Çakmak Mah. Goktuğ Sok. Çağrı Apt. No: 21/4

Sincan / ANKARA

Eğitim

Lisans : Çankaya Üniversitesi Bilgisayar Mühendisliği

Yabancı Dil ve Düzeyi

İnglizce (İyi)

İş Deneyimi

2013'den beri TÜBİTAK' da çalışmaktadır.

Deneyim Alanı

Web Tabanlı Yazılım

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

Osman Yılmaz, Ufuk Çelikcan, " Q-Learning with Naïve Bayes Approach Towards More Engaging Game Agents" , International Conference on Artificial Intelligence and Data Processing IDAP 2018.

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

-

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 05/09/2018

Tez Başlığı / Konusu: Daha İnandırıcı Oyun Karakterleri için Bayes ve Q-Learning Tabanlı Yaklaşım

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 65 sayfalık kısmına ilişkin, 05/09/2018 tarihinde ~~şahsım~~/tez danışmanım tarafından *Turnitin* adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 1 'tür.

Uygulanan filtrelemeler:

1- Kaynakça hariç
2- Alıntılar hariç/~~dâhil~~
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

05.09.2018

Adı Soyadı:	Osman YILMAZ
Öğrenci No:	N13230208
Anabilim Dalı:	Bilgisayar Mühendisliği
Programı:	Tezli Yüksek Lisans
Statüsü:	☑ Y.Lisans ☐ Doktora ☐ Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Dr. Öğr. Üyesi Ufuk Çelikcan

(Unvan, Ad Soyad, İmza)