

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SEZGİSEL ARAMA ALGORİTMALARIYLA
DENİZ TAKTİK HARP OYUNU BENZETİMİ**

YÜKSEK LİSANS TEZİ

Bilgisayar Bilimleri Müh. Ahmet Turan DEĞERLİ

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Tez Danışmanı : Doç. Dr. Cemil ÖZ

Eylül 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SEZGİSEL ARAMA ALGORİTMALARIYLA DENİZ TAKTİK HARP OYUNU BENZETİMİ

YÜKSEK LİSANS TEZİ

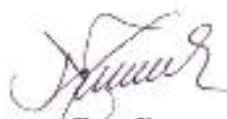
Bilgisayar Bilimleri Müh. Ahmet Turan DEĞERLİ

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 11 / 09 /2009 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



Doç. Dr.
Cemil ÖZ
Jüri Başkanı



Doç. Dr.
Nejat YUMUŞAK
Üye



Doç. Dr.
İbrahim ÇİL
Üye

TEŐEKKÜR

Bu alıőmada bana yol gsteren ve destek olan danıőman hocam Do. Dr. Cemil Z'e teőekkr ederim.

alıőmalarımda kullandığım vektrel haritaların izimindeki katkılarından dolayı Bil. Tek. Soner AVŐAR'a ve tezimi okuyup deđerli dőncelerini benimle paylaőan Bil. Yk. Mh. Remzi AKDAĐ ile Bil. Yk. Mh. İbrahim ŐŐANECİ'ye teőekkrlerimi sunarım.

Bu tez alıőması, kıymetli eőim Glőah DEĐERLİ'ye adanmıőtır.

Ahmet Turan DEĐERLİ

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ	vi
TABLolar LİSTESİ.....	vii
ŞEKİLLER LİSTESİ.....	viii
ÖZET	x
SUMMARY	xi

BÖLÜM 1.

GİRİŞ.....	1
------------	---

BÖLÜM 2.

ASKERİ BENZETİM SİSTEMLERİNİN İNCELENMESİ.....	4
2.1. Benzetim Yazılımları	4
2.2. Askeri Benzetim Yazılımları.....	4
2.3. Müşterek Harekât Alanı Benzetimi (Joint Theater Level Simulation - JTLS)	5
2.3.1. Müşterek Harekât Alanı Benzetimi'nin genel özellikleri.....	6
2.3.2. Müşterek Harekât Alanı Benzetimi yol bulma algoritmasının analizi.....	7

BÖLÜM 3.

KOMUTA KONTROL BİLGİ SİSTEMLERİ VE HARP OYUNLARINDA

GÖSTERİM SEMBOLOJİLERİ.....	10
3.1. Gösterim Sembolojileri.....	10
3.1.1. NTDS sembolojisi.....	10
3.1.2. MIL-STD 2525C sembolojisi.....	11

3.1.3. APP-6A sembolojisi.....	14
3.2. Gösterim Sembolojilerinin Değerlendirmesi	16

BÖLÜM 4.

GRAF(ÇİZGE) TEORİSİ VE YOL BULMA ALGORİTMALARI	18
4.1. Graf (Çizge) Teorisi.....	18
4.2. Graflarda Arama ve Yol Bulma Algoritmaları.....	21
4.2.1. Habersiz arama algoritmaları.....	22
4.2.1.1. Dijkstra algoritması.....	22
4.2.1.2. Dijkstra algoritmasının çalışması.....	23
4.2.1.3. Dijkstra algoritmasının karmaşıklığı.....	23
4.2.2. Sezgisel arama algoritmaları	24
4.2.2.1. A-Yıldız (A*) algoritması	25
4.2.2.2. A-Yıldız algoritmasının çalışması	25
4.2.2.3. Manhattan uzaklığı yöntemi	27
4.2.2.4. Euclidean uzaklığı yöntemi	28
4.2.2.5. Diagonal uzaklığı yöntemi	28
4.2.2.6. A-Yıldız algoritmasının karmaşıklığı	29

BÖLÜM 5.

SEZGİSEL ARAMA ALGORİTMALARIYLA DENİZ TAKTİK HARP OYUNU BENZETİMİ YAZILIMININ GELİŞTİRİLMESİ.....	31
5.1. Geliştirilen Yazılımın Özellikleri	31
5.2. Kullanılan Teknoloji ve Araçlar	32
5.3. Yazılımın Kullanıcı Arayüzü	33
5.3.1. Harita ekranı	33
5.3.1.1. Haritaların yüklenmesi ve coğrafi alanın yazılım tarafından tanınması	34
5.3.1.2. Haritalar üzerinde kullanılan gösterim sembolojileri.....	37
5.3.2. Yönetim ekranı	39
5.4. Yazılımın Teknik Altyapısı.....	41
5.5. Yazılımın Test Modu İşlemleri	42

BÖLÜM 6.	
TARTIŞMA VE ÖNERİLER	44
KAYNAKLAR.....	51
ÖZGEÇMİŞ	53

SİMGELER VE KISALTMALAR LİSTESİ

C4I	: Command, Control, Communications, Computer, and Intelligence
JDK	: Java Development Kit
JRE	: Java Runtime Environment
JTLS	: Joint Theater Level Simulation
KKBS	: Komuta Kontrol Bilgi Sistemleri
NATO	: North Atlantic Treaty Organization
NTDS	: Navy Tactical Data System
STANAG	: Standardization Agreement

TABLolar LİSTESİ

Tablo 3.1. NTDS sembolojisinden örnek semboller	11
Tablo 3.2. MIL-STD-2525C sembolojisinden örnek semboller	12
Tablo 3.3. MIL-STD-2525C standardında dolgulu simgelerin renk değerleri	13
Tablo 3.4. MIL-STD-2525C standardında dolgusuz simgelerin renk değerleri.	14
Tablo 3.5. APP 6A sembolojisinden örnek semboller.....	15
Tablo 3.6. APP 6A standardında belirtilen simgelerin renk değerleri.....	15
Tablo 5.1. Benzetim yazılımı geliştirirken kullanılan teknoloji ve araçlar	32
Tablo 5.2. Harita üzerindeki renkler ve coğrafi karşılıkları.....	34
Tablo 5.3. Izgara hücresinin uzunluğuna göre harita üzerindeki düğüm sayısı.....	36
Tablo 5.4. Geliştirilen benzetim yazılımında kullanılan simge renk değerleri.....	38
Tablo 5.5. Geliştirilen benzetim yazılımında kullanılan simgeler.	38
Tablo 5.6. Benzetim yazılımı seçenekler bölümündeki kontroller ve işlevleri.....	40
Tablo 6.1. Benzetim yazılımına uygulanan senaryo-I ve sonuçları	44
Tablo 6.2. Benzetim yazılımına uygulanan senaryo-II ve sonuçları.....	46
Tablo 6.3. Benzetim yazılımına uygulanan senaryo-III ve sonuçları.....	47

ŞEKİLLER LİSTESİ

Şekil 2.1. JTLS yazılımından bir ekran görüntüsü.....	6
Şekil 2.2. JTLS içerisinde yer alan nesnelerin hareket tarzları.....	7
Şekil 2.3. Dijkstra algoritması kullanılarak hedef noktaya giden rotanın belirlenmesi.....	9
Şekil 3.1. Aynı taktik resmin NTDS ve MIL-STD-2525B sembolojileriyle gösterimi.....	16
Şekil 4.1. Düzenli graf örnekleri.....	18
Şekil 4.2. Tamamlanmış graf örnekleri.....	19
Şekil 4.3. Üç düğümden oluşan bir basit yönsüz graf.....	19
Şekil 4.4. Üç düğümden oluşan bir yönlü graf.....	20
Şekil 4.5. Üç düğümden oluşan bir çoklu graf.....	20
Şekil 4.6. Dört düğümden geçen bir yol.....	21
Şekil 4.7. Ağaç şeklinde bir graf.....	21
Şekil 4.8. Harita üzerinde yol bulma işlemi.....	22
Şekil 5.1. Sezgisel Arama Algoritmalarıyla Deniz Taktik Harp Oyunu Benzetimi... 31	31
Şekil 5.2. Benzetim yazılımının harita ekranı.....	34
Şekil 5.3. Izgara boyutunun çok küçük olduğu durumlarda rota hassasiyeti.....	35
Şekil 5.4. Izgara boyutunun büyük olduğu durumlarda rota hassasiyeti.....	36
Şekil 5.5. Balistik füze taşıyan nükleer denizaltının sadece derin suları kullanarak seyri.....	37
Şekil 5.6. Benzetim yazılımını yönetim ekranı.....	39
Şekil 5.7. Farklı hızlardaki iki temasın yazılım içerisindeki hareketi.....	41
Şekil 5.8. Test sonuçlarından bir ekran görüntüsü.....	42
Şekil 6.1. Senaryo-I'e göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı.....	45
Şekil 6.2. Senaryo-I'e göre algoritmaların en kısa rotayı hesaplama süreleri.....	45

Şekil 6.3. Senaryo-II'ye göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı	46
Şekil 6.4. Senaryo-II'e göre algoritmaların en kısa rotayı hesaplama süreleri	46
Şekil 6.5. Senaryo-III'e göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı	48
Şekil 6.6. Senaryo-III'e göre algoritmaların en kısa rotayı hesaplama süreleri	48

ÖZET

Anahtar Kelimeler: Deniz Taktik Harp Oyunu Benzetimi, Graflar (Çizgeler), Yol Bulma Algoritmaları, Sezgisel Algoritmalar, A-Yıldız (A*) Algoritması, Dijkstra Algoritması, NATO Military Symbols for Land Based Systems, USA Common Warfighting Symbology, Joint Theater Level Simulation (JTLS).

Harp oyunlarına olan ilgi, gelişen teknoloji ile birlikte, gerek oyun amaçlı, gerekse eğitim amaçlı olsun artmaktadır. Gerçek hayatta ancak çok büyük bedellerle elde edilebilecek tecrübeler, sanal ortamlarda çok daha az maliyetle ve sifıra yakın zayıat ile elde edilebilmektedir. Olmuş olayların tekrar değerlendirilmesi ve gelecekte olabilecek olayların sonuçlarının analiz edilmesi işlemleri, bu yazılımlar ile yapılabilmektedir.

Günümüz dünyasında harp oyunu benzetimleri, orduların eğitiminde önemli yer teşkil etmeye başlamışlardır. Bu teknoloji ile çeşitli kuvvetler, sanal ortamlarda müşterek harekâtlar gerçekleştirebilmektedir. Ancak bu benzetim yazılımlarının gerçeğe en uygun şekilde tasarlanmış olmaları büyük önem taşımaktadır. Bu yazılımlarda insan zekâsının en iyi şekilde yansıtılmış olması, benzetimden alınacak verimin en üst düzeye çıkmasını sağlayacaktır.

Bu çalışma çerçevesinde, askeri benzetim sistemleri ve özelde Müşterek Harekât Alanı Benzetimi (JTLS) incelenmiş ve bu benzetimlerin yol bulma algoritmaları analiz edilmiştir. Java teknolojisi kullanılarak bir deniz taktik harp oyunu benzetimi geliştirilmiştir. Benzetim içerisinde yer alan haritalar, ızgaralara (düğümlere) bölünerek çizgeler elde edilmiştir. Harp unsurlarının hareketleri, elde edilen bu çizgeler üzerinde, en kısa rota problemlerinin çözümünde kullanılan A-Yıldız ve Dijkstra algoritmaları ile sağlanmıştır. Kullanılan bu algoritmaların karşılaştırmalı analizleri yapılmış ve benzetim içerisindeki performansları incelenmiştir.

Benzetimde yer alan harp unsurlarının haritalar üzerindeki gösterimlerinde, NATO Military Symbols for Land Based Systems (APP-6A) ve USA Common Warfighting Symbology (MIL-STD-2525C) standartlarına uygun sembolojiler kullanılmıştır.

THE NAVAL TACTICAL WARFARE GAME SIMULATION WITH HEURISTIC SEARCH ALGORITHMS

SUMMARY

Keywords: Naval Tactical Warfare Game Simulation, Graphs, Pathfinding Algorithms, A-Star (A*) Algorithm, Dijkstra Algorithm, Military Symbols for Land Based Systems, Common Warfighting Symbology, Joint Theater Level Simulation (JTLS).

With developing technology, the interest in warfare game is increasing for both game and training purposes. Although some experiences can be gained with very high costs in real life they can be obtained with zero casualties and much less cost in virtual environments. The operations evaluating of the events that happened in the past and analyzing the results of the events that will happen in the future can be done with these simulations.

In today's world, the naval tactical warfare game simulations constitute an important place in education of the armies. With this technology, various forces can perform joint operations. However, it is significantly important to be designed of the simulation software in the most appropriate way. By this software mirroring the human intelligence in the best way is going to provide to go to the highest level of the yield taken from the simulation.

Within the framework of this study, military simulation systems and in particular the Joint Theater Level Simulation (JTLS) has been examined and pathfinding algorithms of these simulations have been analyzed. A naval tactical warfare game simulation has been developed by using Java technologies. Graphs were obtained by dividing of the maps in this simulation into the grids (nodes). The movements of the military aspects have been made by using A-Star and Dijkstra algorithms using for solving the shortest route problems obtained on these graphs. Comparative analyses of these algorithms used in the simulations have been done and the performance has been investigated.

To show the warfare elements on the map in this simulation, NATO Military Symbols for Land Based Systems (APP-6A) and USA Common Warfighting Symbology (MIL-STD-2525C) symbologies belonging standards have been used.

BÖLÜM 1. GİRİŞ

Modern teknoloji ve silahlarla donatılmış harp nevelerinin eğitim ihtiyacı ve gelişen teknoloji ile artan maliyetler, tatbikat hazırlıkları ve tatbikat sonrası analizler için harcanan zamanın uzun olması, harp oyunlarına olan ihtiyacı artırmıştır. Ülkeler, kendi ordularının eğitiminde ve uluslararası işbirliği çerçevesinde gerçekleştirdikleri eğitimlerde, harp oyunlarının kullanımına önem vermektedirler.

Günümüzdeki harplerin göğüs göğüse çarpışmadan ziyade sanal ortamlarda olması ve bilgi gücünün ön plana çıkması tatbikatların sanal ortamlarda yapılması ve analiz edilmesi gereksinimini ortaya koymuştur.

Harp oyunu benzetimleri (simülasyonları), sadece ülke savunmasında görevli kurumların eğitiminde değil aynı zamanda dünya üzerindeki oyun pazarının da vazgeçilmez bir parçası olmuştur. Cazip doğasının yanında sanal ortamda stratejik karar verme, yönetme ve zafer elde etme imkânı sağlaması, harp oyunlarına olan ilgiyi her geçen gün arttırmaktadır.

Diğer taraftan iletişim araçların gelişmesi ve savaş sistemlerinin iletişim yeteneğinin artmasıyla birlikte ağ yetenekli harp vazgeçilmez bir hal almıştır. Harp sahasından çeşitli sistem ve algılayıcılarla elde edilen bilgiler birleştirilerek avantaja dönüştürülebilmektedir. Ağ destekli oyunlar ve benzetimler farklı coğrafyalardaki kullanıcıları aynı sanal ortamda buluşturmaktadır.

Oyunların gerçeğe yakın oluşu ve gerçek dünyadan izlenimler bırakması, oyuncuların oyundan daha fazla zevk almasını sağlamakta, bu oyunlar eğitim amaçlı olarak kullanıldığında, eğitimin başarı düzeyini arttırmaktadır.

Harp oyunu benzetimlerinde çeşitli güçlere komuta eden kişiler (oyuncular), verdikleri emirlerin oyun karakterleri tarafından gerçek insanlar gibi yerine getirilmesini beklemektedirler. Oyun karakterlerinin, çeşitli durumlar karşısında kendi kendilerine mantıklı kararlar verebilmeleri (yapay zeka) ve durum muhakemesi yapabilmeleri oyunun inandırıcılığını artırmaktadır.

Günümüzde, ülkemiz dâhil birçok ordu, harp oyunu benzetimlerini eğitim ve tatbikatlarda etkin olarak kullanmaktadır [1, 2]. Benzetim yazılımlarının askeri ve sivil alanlarda etkinliği artmakta ve bu yazılımların geliştirilmesi için ayrılan bütçeler her geçen gün büyümektedir. Örneğin Microsoft firması Xbox oyun konsolu ve içerisindeki oyunların gelişimi için 2 milyar doların üzerinde harcama yapmıştır. Amerika Birleşik Devletleri'nin yıllık bilim ve teknoloji bütçesi 1.6 milyar dolardır. Aynı zamanda ABD ordusu, pilotlarının eğitimi için Microsoft firmasının Flight Simulator yazılımının özelleştirilmiş bir sürümünü kullanmaktadır. Diğer taraftan İngiliz Kraliyet Donanması, kendi ihtiyaçlarına dönük olarak geliştirilmek üzere, ticari bir oyun olan Fleet Command'ın lisansını talep etmiştir [3]. Fleet Command benzetim yazılımı aynı zamanda, ABD Deniz Kuvvetleri Akademisi (United States Naval Academy) ve Deniz Harp Okulu (Naval War College)'nda eğitim amaçlı olarak kullanılmaktadır [3,4].

Bütün bu gelişmeler, benzetim yazılımlarının askeri alanlarda daha da yaygınlaşacağını birer göstergesidir.

Bu tez çalışması çerçevesinde, bir deniz taktik harp oyunu benzetimi geliştirilmiş ve bu benzetim yazılımında çeşitli yol bulma (pathfinding) algoritmaları kullanılmıştır. Yol bulma işlemi, oyun karakterleri veya araçların (gemiler, denizaltılar vb.) verilen emirlere göre, insan gibi düşünerek hareket etmesi, düşmandan gizlenerek yol alması ve muhtemel kritik bölgelerden uzak durarak ilerlenmesini kapsamaktadır. Haritalar üzerindeki bilgiler ışığında kendi harekât güzergâhını belirleyen yapay zekâ karakterleri, hedef noktaya uygun rotayı kendileri tayin etmektedirler.

Geliştirilen benzetimde yer alan harp unsurlarının haritalar üzerindeki gösterimlerinde, NATO Military Symbols for Land Based Systems (APP-6A) ve

USA Common Warfighting Symbology (MIL-STD-2525C) standartlarına uygun sembolojiler kullanılmıştır.

Benzetim içerisinde yer alan haritaların ızgaralara bölünmesi suretiyle düğümlerden oluşan bir harekât sahası (çizge) elde edilmiştir. Bu saha üzerinde yer alan harp unsurlarının hareketleri, A-Yıldız ve Dijkstra algoritmaları kullanılarak sağlanmıştır. Söz konusu algoritmaların karşılaştırmalı analizleri yapılmış ve benzetim içerisindeki performansları incelenmiştir.

A-Yıldız algoritmasının, kısa mesafeli, az düğümlü ve az engelli ortamlarda çok başarılı olduğu, ancak düğüm ve engel sayısının arttığı durumlarda algoritma performansının düştüğü, bununla birlikte rota hesaplama süresinin uzadığı görülmüştür.

Dijkstra algoritmasının tüm düğümleri taraması sebebiyle, yakın mesafeli hedeflerde ve az engelli ortamlarda A-Yıldız algoritmasına göre daha düşük performans gösterdiği ancak düğüm ve engel sayısının arttığı durumlarda performans düşüşünün çok az olduğu tespit edilmiştir.

Rota tayin etme işlemlerinde, hedef noktanın uzaklığı ile düğüm ve engel sayılarının göz önüne alınması suretiyle, A-Yıldız ve Dijkstra algoritmalarını tercihli kullanan melez yapıların oluşturulmasının benzetim veya oyun performansını arttıracığı değerlendirilmektedir.

BÖLÜM 2. ASKERİ BENZETİM SİSTEMLERİNİN İNCELENMESİ

2.1. Benzetim Yazılımları

Benzetim yazılımlarının amacı, katılımcının gerçek hayattan koparılı olarak, belli süre dahi olsa farklı bir rolde olmasını sağlamaktır. Katılımcı, sahip olduğu yeni role göre benzetim ile etkileşim göstermeye zorlanmaktadır. Böylece yapılan etkileşim sonucunda elde edilen tecrübe, katılımcının gerçek sistemlere olan ünsiyetini artırmaktadır. Aynı zamanda katılımcı benzetim içerisinde verdiği kararların sonuçlarını tecrübe edebilmektedir. Sanal sistemler üzerinde yapılan çalışmalar, aynı zamanda insan veya sistem kaynaklı hataları asgariye indirmektedir. Diğer taraftan benzetim sistemleri, dil ve kültürel eğitimlerde de yaygın olarak kullanılmaktadır.

2.2. Askeri Benzetim Yazılımları

Gelişen teknoloji ve karmaşıklaşan savaş sistemleri, askeri personelin eğitim ihtiyacını her geçen gün artırmaktadır. Ancak savaş sistemlerinin çok büyük bedellerle elde edilebilmeleri nedeniyle bu sistemler üzerinde eğitim yapmak zordur. Eğitim için kullanılan her silah veya teçhizatın bir bedeli vardır. Bu bedel bazen gerçek sistemler kadar maliyetli olabilmektedir.

Dünyada ve ülkemizde kolluk kuvvetleri, personel eğitimlerinde benzetim ve benzetici (simülator) sistemleri kullanmaktadır. ABD, Irak ve Afganistan'da görev yapan birliklerinin eğitim ve sarsıntı sonrası tedavilerinde benzetim ve sanal gerçeklik ortamlarını kullanmaktadır [5].

Benzetim sistemleri, tekrar kullanılabilirlik açısından önemli bir avantaj sağlamaktadırlar. Gerçek bir silah sistemi üzerinde yapılabilecek eğitim, zaman ve personel durumu ile sınırlı olmakla birlikte, benzetim sistemi vasıtasıyla yapılan eğitimlerde yan unsurlar sistem içerisinde bulunduğundan daha kolay gerçekleştirilmektedir.

Örneğin, bir geminin atış kontrol sisteminin harbe hazır olabilmesi için radarlarının devrede olması, emniyet tedbirlerinin alınmış olması, görevli tüm personelin savaş yerlerinde olmaları vs. gerekmektedir. Oysa benzetim sistemini kullanan gerçek şahıs haricindeki tüm çevresel değişkenler (diğer personel ve sistemler) benzetim içerisinde sanal olarak tanımlanabilir. Böylece hedef kişinin eğitimi çevresel faktörler olmadan da gerçekleştirilebilir.

Diğer taraftan benzetim sistemlerinin her zaman gerçek sistemin yerini tutacağı söylenemez. Zaten amaç da bu değildir. Bu nedenle yapılacak olan eğitimlerin bir bölümünün bu sistemler tarafından gerçekleştirilmesi ve belli bir seviyeye gelen kişilerin daha sonra gerçek sistem üzerinde eğitilmesi de önemli bir avantaj sağlamaktadır.

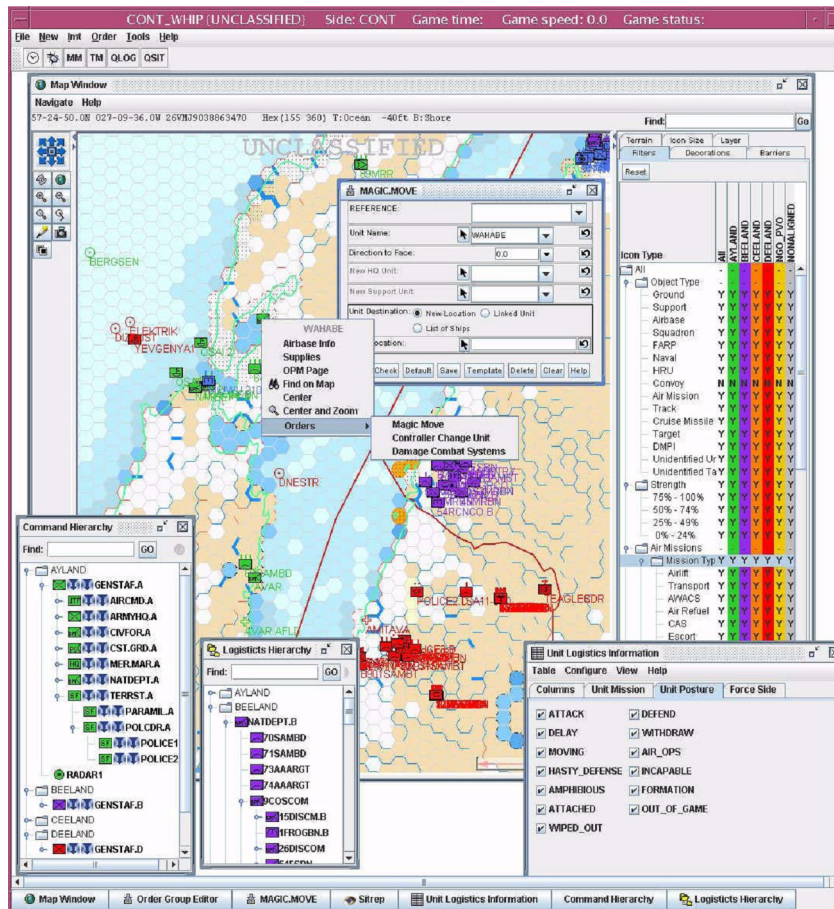
Bu bölümde, deniz taktik harp oyunu konusunda geliştirilmiş olan Müşterek Harekât Alanı Benzetimi (Joint Theater Level Simulation - JTLS) oyunu ve oyunda kullanılan yol bulma algoritmaları incelenmiş, oyuna özgü genel değerlendirmeler yapılmıştır.

2.3. Müşterek Harekât Alanı Benzetimi (Joint Theater Level Simulation-JTLS)

Müşterek Harekât Alanı Benzetimi (JTLS), interaktif ve çok taraflı oynanabilen, koalisyon ve müşterek güçlerin operasyonlarını kara, hava ve deniz ortamlarında resmedebilen bir oyundur. Bahse konu yazılıma ilişkin detaylar, müteakip başlıklarda yer almaktadır.

2.3.1. Müşterek Harekât Alanı Benzetimi'nin genel özellikleri

JTLS sistemi, senaryo hazırlanması, model işlemlerinin yerine getirilmesi ve harekât sonuçlarının analiz edilmesi amacıyla altı ana yazılım ve farklı görevlerdeki birçok bileşeni içerisinde barındırmaktadır. JTLS, tek veya birden çok bilgisayar üzerinde işletilebilmekte, ağ arayüzü desteği olması sebebiyle ağa bağlı herhangi bir terminalden sisteme erişim sağlanabilmektedir. Ayrıca kara, hava ve deniz güçlerinin detaylı bir şekilde modellenmesini sağlamakla birlikte Lanchester stratejilerine göre muharebe davranışları gerçekleştirebilmektedir. Benzetim, içerisindeki güçler, askeri, sivil veya karışık olarak ayarlanabilmektedir. Yazılım aynı zamanda Kuzey Atlantik Savunma Anlaşması Örgütü (North Atlantic Treaty Organization - NATO) üyesi ülkeler tarafından askeri eğitim amaçlı olarak kullanılmaktadır [2]. Joint Theater Level Simulation yazılımının ekran görüntüsü Şekil 2.1' de yer almaktadır.



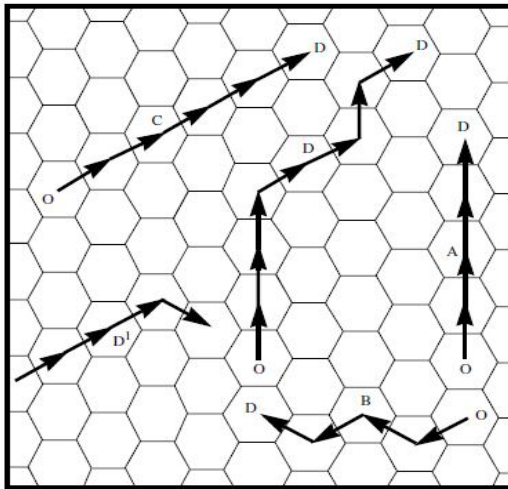
Şekil 2.1. JTLS yazılımından bir ekran görüntüsü

JTLS sistemi içerisindeki yazılımlar ile oyun senaryoları hazırlanabilmekte, bunlar doğrulanabilmekte, emirler verilebilmekte ve tüm bu işlemler yazılı mesajlar ve grafik görüntüleriyle kullanıcılara gösterilebilmektedir. Sistem, harekât planı analizleri ile vatan savunması, kaçakçılık, terörizm ve doğal afetlere ilişkin senaryoları da gerçekleyebilmektedir [6].

2.3.2. Müşterek Harekât Alanı Benzetimi yol bulma algoritmasının analizi

JTLS, oyun içerisinde harekât işlemlerini Dijkstra algoritmasına göre gerçekleştirmektedir. Kara ve deniz ortamlarında rota belirleme işlemleri Dijkstra algoritması, hava ortamlarında ise değiştirilmiş Dijkstra algoritması kullanılmaktadır [7].

Bahse konu ortamlarda yapılacak hareketler, bulunulan noktadan hedef noktaya asgari zamanda/en kısa yolu kullanılacak şekilde gerçekleştirilmektedir. Hareket emri verilen unsur, kara ortamlarında mümkün olan en hızlı şekilde hareket edebilir. Ancak arazinin yapısı hareket hızının düşmesine sebep olabilmektedir. Benzetim içerisinde yer alan araç ve nesnelerin harita üzerindeki hareket tarzları Şekil 2.2.' de yer almaktadır.



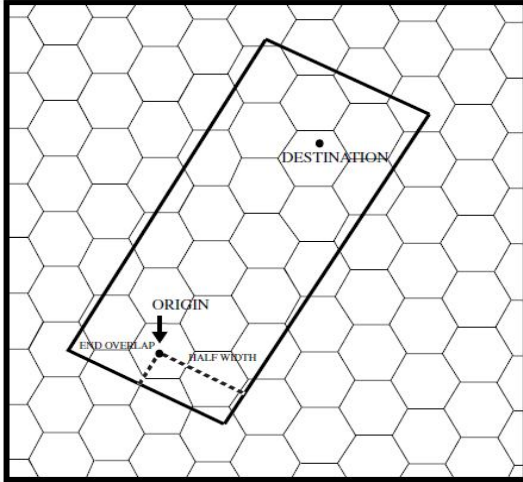
Şekil 2.2. JTLS içerisinde yer alan nesnelerin hareket tarzları

Deniz ortamlarında hareket, karadan biraz farklıdır. Benzetim içerisinde deniz araçlarının yapabileceği azami hızın yanında seyir yapılan suyun derinliği de gemi rotasına etki etmektedir. Benzetim veritabanında harita üzerindeki her bir noktanın (altıgen bölgelerin) su derinliği bilgisi yer almaktadır. Aynı zamanda deniz araçlarının her birinin su çekimi değerlerine bağlı olarak seyir yapabilecekleri asgari derinlik bilgisi de (minimum depth) benzetim içerisinde bulunmaktadır. Örneğin, hareketi istenen geminin seyir için gerekli asgari derinlik bilgisi, harita üzerindeki su derinliği bilgisinden büyükse o bölgeden hareket gerçekleştirilemeyecektir.

Benzetimde yer alan haritada, denizden karaya ve karadan denize hareket edemeyecek araçlar için, kıyı bölgeleri ve sahilleri içeren noktalar “geçilemez” olarak işaretlenmiştir. Böylece yol bulma algoritmasının bu bölgelerde arama yapması engellenmiştir.

JTLS yazılımı içerisinde, hedeflenen noktaya en kısa zamanda ulaşmak için aşağıdaki adımlar uygulanır.

- a. Hedef noktaya gitme emri verildiğinde, bulunulan nokta ile hedef noktayı içeren dikdörtgen şeklinde bir alan belirlenir (Şekil 2.3). Arama işlemi bu alan içerisinde yapılacağından, hedefe giden rotayı bulmak için yapılan işlem sayısı ve harcanan zaman sınırlandırılmış olur.
- b. Harita üzerindeki bu dikdörtgen alan içerisinde Dijkstra algoritması işletilir. Merkez noktası söz konusu alan içerisinde yer alan her bir bölge (altıgen) için bu işlem sürdürülür.
- c. Söz konusu alan içerisinde bulunan “geçilemez” alanlar hesaba katılarak en uygun rota belirlenir.



Şekil 2.3. Dijkstra algoritması kullanılarak hedef noktaya giden rotanın belirlenmesi

Tespit edilen rota üzerinde seyir yapan geminin hızı, JTLS sistemi içerisinde iki farklı etken göz önünde bulundurularak katılarak hesap edilmektedir.

- a. Geminin hasar (damage) durumu
- b. Deniz suyundan kaynaklanan derinlik çarpanı (depth factor)

Benzetim yazılımı içerisinde geminin hasar durumunun tespiti, sadece maddi hasarların meydana gelmesine bağlı olarak değil, aynı zamanda personel zayıtı ve savaş sistemlerinde meydana gelen arızalarla ilişkili verimlilik düşüşü de hesaba katılarak gerçekleştirilmektedir. Deniz suyundan kaynaklanan derinlik çarpanı, harita üzerindeki her bir altıgen noktanın yazılım veritabanında tutulan değeri alınarak tespit edilmektedir [7].

Örneğin, azami sürati 40 denizmili/saat, hasar durumu 0.8 ve derinlik çarpanı 0.45 olarak bilinen bir harp gemisine 30 denizmili/saat hız ile hareket emri verildiğinde;

Hasar Durumuna Bağlı Azami Hız: $40 \times 0.80 = 32$ denizmili/saat

Su Derinliğine Bağlı Azami Hız: $32 \times 0.45 = 14$ denizmili/saat

olacaktır. 30 denizmili/saat ile gitme emri verilen söz konusu geminin benzetim içerisinde 14 denizmili/saat hız ile seyir yaptığı görülecektir.

BÖLÜM 3. KOMUTA KONTROL BİLGİ SİSTEMLERİ VE HARP OYUNLARINDA GÖSTERİM SEMBOLOJİLERİ

3.1. Gösterim Sembolojileri

Stratejik seviyede elde edilen bilgiler ile harekât sahasına ait taktik resmin harita ve Komuta Kontrol Bilgi Sistemleri (KKBS) üzerinde gösterimine yönelik olarak, çeşitli devletler ve organizasyonlar tarafından gösterim sembolojisi standartları oluşturulmuştur. Söz konusu semboloji standartları, gerçeğe uygunluk açısından, harp oyunu benzetimleri içerisinde aynı şekilde kullanılmaktadır.

Bu bölümde, Amerika Birleşik Devletleri tarafından geliştirilen Naval Tactical Data System (NTDS) ve Common Warfighting Symbology (MIL-STD-2525C) standartları ile NATO tarafından geliştirilen Military Symbols for Land Based Systems (STANAG) 2019 (APP 6A) standardı incelenmiştir.














3.1.1. NTDS sembolojisi

Amerika Birleşik Devletleri tarafından, 1950'li yıllarda, saldırı ve savunma işlemlerine ilişkin bilgilerin toplanıp değerlendirildiği ve bilgi akışını otomatikleştirildiği hızlı karar verebilen sistemler geliştirilmiştir. Bu sistemler, 1960'lı yılların başında harp gemilerinde de kullanılmaya başlanmıştır. Önceleri yazı tahtası, kâğıt harita ve asetatlar üzerinde oluşturulan taktik resimler, gelişen teknoloji ile bilgisayar ortamında meydana getirilmiştir. Ortaya çıkarılan taktik resim, mevcut durumun değerlendirilmesi ve en uygun kararın verilebilmesi amacıyla kullanılmaktadır.

Harp gemilerinde oluşturulan Savaş Harekât Merkezleri (SHM) içerisinde, çevredeki temas (gemi, denizaltı, uçak, helikopter vs.) bilgilerinin değerlendirilip kullanılması amacıyla Deniz Taktik Veri Sistemi (Navy Tactical Data System – NTDS) meydana getirilmiştir. Böylece tehditlere karşı daha hızlı hareket edebilme ve hata yapma olasılığını azaltma imkânı elde edilmiştir.

Deniz Taktik Veri Sistemi (NTDS) ve algılayıcılar (sensors) vasıtasıyla toplanan veriler, kablosuz veri bağlantıları ile diğer gemilere de aktarılarak bilgi paylaşımı sağlanmıştır. Elde edilen söz konusu verilerden taktik resmin çıkarılması amacıyla, NTDS sembolojisi meydana getirilmiştir. NTDS sembolojisi, NATO ve ABD tarafından geliştirilen diğer standartların temelini oluşturmuştur [8]. Tablo 3.1 de NTDS sembolojisinden örnek semboller yer almaktadır.

Tablo 3.1. NTDS sembolojisinden örnek semboller

NTDS Sembolojisi		Temasın Kimliği (ID)			
		Tanınmıyor	Dost	Düşman	Tarafsız
Temasın Sınıfı	Hava				
	Suüstü				
	Denizaltı				
	Kendi Temasımız				

3.1.2. MIL-STD 2525C sembolojisi


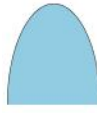




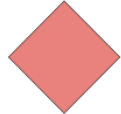









Common Warfighting Symbology (MIL-STD-2525C) standardı, ABD Savunma Bakanlığı (Department of Defence – DoD) tarafından, müşterek harekât, operasyon ve

eđitim dokümanlarında kullanılması ve Komuta, Kontrol, Muhabere, Bilgisayar ve İstihbarat Sistemleri (Command, Control, Communications, Computer, and Intelligence (C4I) systems) ile oluşturulan taktik resmin gösterimi amacıyla geliştirilmiştir.

Amerika Birleşik Devletleri'nde *C4I For The Warrior (C4IFTW)* konseptinin Haziran 1992 tarihinde imzalanmasıyla birlikte harekât alanının doğru bir şekilde gerçek zamanlı (real-time) olarak komuta kontrol bilgi sistemleri üzerinde gösterimi amaçlanmıştır. Komuta, Kontrol, Muhabere, Bilgisayar ve İstihbarat (C4I) sistemleri ile silah yönetim konsolları arasındaki bilgi deęişiminin ortak sembollerle yapılması sağlanmıştır [9].

Common Warfighting Symbology standardı, NATO tarafından geliştirilen Military Symbols for Land Based Systems (STANAG) 2019 (APP 6) standardı temel alınarak geliştirilmiştir [9]. Daha sonraki yıllarda ABD ve NATO karşılıklı olarak birbirlerinden faydalanarak standartlarını güncelleştirmişlerdir. Bu açıdan bahse konu iki standardın sembolleri benzerlik göstermektedir. Tablo 3.2'de MIL-STD-2525C standardından örnek semboller yer almaktadır.









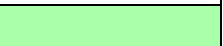


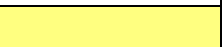



Tablo 3.2. MIL-STD-2525C semboljisinden örnek semboller

MIL-STD-2525C Sembolojisi		Temasın Kimliği (ID)			
		Tanınmıyor	Dost	Düşman	Tarafsız
Temasın Sınıfı	Hava				
	Suüstü				
	Denizaltı				
	Kendi Temasımız				

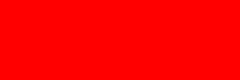




Ayrıca MIL-STD-2525C standardında NTDS sembolojisinin zayıf yönlerinin giderilmesi amaçlanmıştır [8].

Söz konusu standart, tespit edilen temasların/unsurların sistemler içerisinde gösterimine ilişkin çeşitli renk değerleri belirlenmiştir. Bu renk değerleri, komuta kontrol sistemlerini veya eğitim benzetimlerini kullanan kişilerin, unsurları hızlı bir şekilde kavramasını sağlamakta etkili olmaktadır. Diğer taraftan bu semboloji standardı, hem siyah-beyaz hem de renkli ekranlarda gösterilmek üzere tasarlanmıştır [9]. Tablo 3.3’de MIL-STD-2525C standardında belirtilen renk değerleri yer almaktadır.

Tablo 3.3. MIL-STD-2525C standardında dolgu simgelerinin renk değerleri

Temasın Kimliği (ID)	Renkler	Renk Değerleri		
		Koyu	Orta	Açık
Düşman (Hostile, Suspect, Joker, Faker)	Kırmızı			
		RGB (200, 0, 0)	RGB (255, 48, 49)	RGB (255, 128, 128)
		HSL (0, 255, 100)	HSL (0, 255, 152)	HSL (0, 255, 192)
Dost (Friend, Assumed Friend)	Mavi			
		RGB (0, 107, 140)	RGB (0, 168, 220)	RGB (128, 224, 255)
		HSL (138, 255, 70)	HSL (138, 255, 110)	HSL (138, 255, 192)
Tarafsız (Neutral)	Yeşil			
		RGB (0, 160, 0)	RGB (0, 226, 0)	RGB (170, 255, 170)
		HSL (85, 255, 80)	HSL (85, 255, 113)	HSL (85, 255, 213)
Tanımsız (Unknown, Pending)	Sarı			
		RGB (225, 220, 0)	RGB (255, 255, 0)	RGB (255, 255, 128)
		HSL (42, 255, 110)	HSL (42, 255, 128)	HSL (42, 255, 192)
Sivil (Civilian) (İsteğe Bağlı Olarak Kullanılır)	Mor			
		RGB (80, 0, 80)	RGB (128, 0, 128)	RGB (255, 161, 255)
		HSL (213, 255, 40)	HSL (213, 255, 64)	HSL (213, 255, 208)

Tablo 3.4. MIL-STD-2525C standardında dolgunsuz simgelerin renk deęerleri.


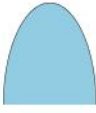




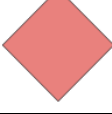









Temasın Kimlięi (ID)	Renkler	Renk Deęerleri	
		RGB Deęeri	Renk
Düşman (Hostile, Suspect, Joker, Faker)	Kırmızı	RGB (200, 0, 0)	
Dost (Friend, Assumed Friend)	Mavi	RGB (0, 255, 255)	
Tarafsız (Neutral)	Yeşil	RGB (0, 255, 0)	
Tanımsız (Unknown, Pending)	Sarı	RGB (255,255,0)	
Sivil (Civilian) (İsteęe Baęlı Olarak Kullanılır)	Mor	RGB (255, 0, 255)	

3.1.3. APP-6A sembolojisi

Military Symbols for Land Based Systems (STANAG) 2019 (APP 6A) standardı, NATO tarafından 1999 yılında geliştirilmeye başlanmış ve 2008 yılında APP-6B ismiyle son halini almıştır. Common Warfighting Symbology (USA) standardı ile büyük benzerlikler göstermektedir. Tablo 3.5'te APP 6A standardından örnek semboller yer almaktadır.

Bahse konu standardın NATO üyesi ülke sınırları içerisinde, C4I sistemlerinde, hava durumu gösterimlerinde, haritalarda, uçakların pilot köşklerinde ve dięer ilgili elektronik görüntüleme sistemlerinde kullanımı amaçlanmıştır [10]. Aynı zamanda elle oluşturulan harita ve taktik resimlerde de kullanılan standart içerisinde bir dizi iki boyutlu (2B) simge yer almaktadır. Tablo 3.6'da APP-6A standardında belirtilen ve simgeler için kullanılan renk deęerleri yer almaktadır.

Tablo 3.5. APP 6A sembolojisinden örnek semboller

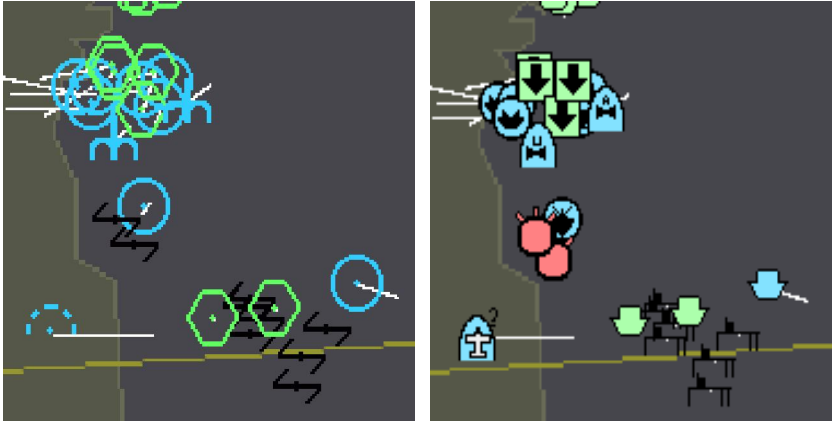
APP 6A Sembolojisi		Temasın Kimliği (ID)			
		Tanınmıyor	Dost	Düşman	Tarafsız
Temasın Sınıfı	Hava				
	Suüstü				
	Denizaltı				
	Kendi Temasımız				

Tablo 3.6. APP 6A standardında belirtilen simgelerin renk değerleri

Temasın Kimliği (ID)	Renkler	Renk Değerleri	
		Simgeler	Dolgu Simgeler
Dost (Friend, Assumed Friend)	Mavi	RGB (0, 255, 255)	RGB (128, 224, 255)
Tanımsız (Unknown, Pending)	Sarı	RGB (255, 255, 0)	RGB (255, 255, 128)
Tarafsız (Neutral)	Yeşil	RGB (0, 255, 0)	RGB (170, 255, 170)
Düşman (Hostile, Suspect, Joker, Faker)	Kırmızı	RGB (255, 0, 0)	RGB (255, 128, 128)
Hava (Weather)	Mor	RGB (128, 0, 128)	RGB (226, 159, 255)
Hava (Weather)	Kahverengi	RGB (128, 98, 16)	RGB (210, 176, 106)
Çizgiler, Alanlar (Boundaries, lines, areas, text, icons and frames)	Mor	RGB (0, 0, 0)	RGB (0, 0, 0)

3.2. Gösterim Sembolojilerinin Değerlendirmesi

NTDS semboljisi dünya üzerinde birçok ülkede ve bu ülkeler tarafından geliştirilmiş sistemler içerisinde yaygın olarak kullanılmaktadır. Ancak gösterim ve anlaşılabilirlik açısından sıkıntılar yaşanmaktadır. Şekil 3.1’de NTDS ve Common Warfighting Symbology (MIL-STD-2525B) ile aynı taktik resmin gösterimi yer almaktadır. Common Warfighting Symbology standardına göre oluşturulan taktik resmin daha anlaşılır olduğu görülmektedir [11].



Şekil 3.1. Aynı taktik resmin NTDS ve MIL-STD-2525B semboljileriyle gösterimi.

MIL-STD-2525C ve APP-6A standartlarının NTDS semboljisine göre çeşitli üstünlükleri aşağıda sıralanmıştır.

- a. Semboller içerisinde harflerin de kullanılması simgenin daha kolay anlaşılmasını sağlamakta ve kullanıcının tepki süresini azaltmaktadır. Örneğin, yapılan testlerde, suüstü unsurlar için kullanılan semboller için, MIL-STD-2525B standardının NTDS’e göre %60 daha kısa sürede algılanabildiği tespit edilmiştir [11].
- b. İçleri dolu simgelerin kullanımı, üst üste binme durumlarında (unsurlar birbirlerine çok yakın olduklarında) okunabilirliği arttırmaktadır.

c. Unsurlara ait simgeler, kullanıcılara, NTDS sembolojisine göre daha fazla bilgi vermektedir. Örneğin, MIL-STD-2525C sembolojisindeki simgeler unsurların platform (Firkateyn, Mayın Avlama Gemisi, Amfibi Çıkarma Gemisi vb.) bilgilerini de içermektedirler.

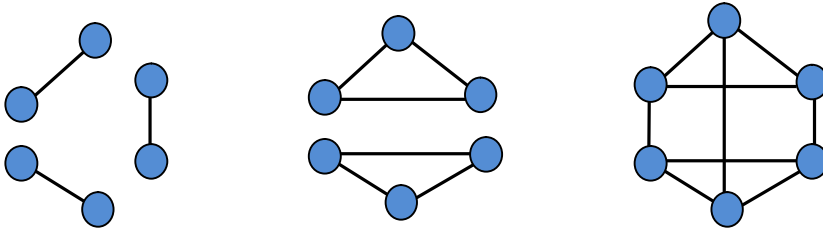
BÖLÜM 4. GRAF(ÇİZGE) TEORİSİ VE YOL BULMA ALGORİTMALARI

4.1. Graf (Çizge) Teorisi

Graf, düğüm olarak adlandırılan noktalar ve bu noktaları birleştiren hatlardan oluşan ve geometrik bir bilgi vermeyip, sadece düğümler arasındaki ilişkiyi gösteren çizgiler topluluğudur. Graf, durumları modelleyen matematiksel bir nesnedir.

Graflar, geometrik değil ilişki bilgileri içerirler. Bir graf üzerindeki her bir noktaya düğüm ve bu düğümler arasındaki her bir çizgiye kenar adı verilmektedir. Kenarların uzunluk ve şeklinin, doğrusal veya eğrisel oluşunun ve düğümlerin konumunun bir önemi yoktur. Her bir kenar, iki düğüm arasındaki bir ilişkiyi simgelemektedir [12].

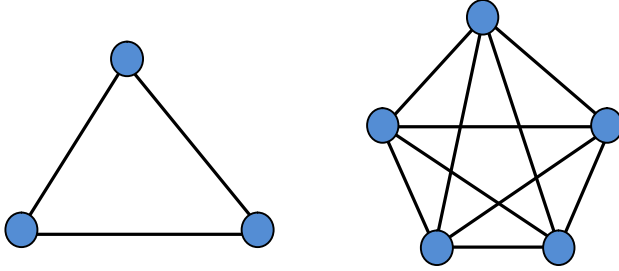
Bir graf içerisinde, d_1 ve d_2 düğümlerini bağlayan bir k kenarı mevcutsa bu iki düğüm komşudur (adjacent). Diğer bir ifadeyle, k_1 ve k_2 kenarları en az bir d düğümüne sahip iseler bu kenarlar komşudur. Bir graf içerisindeki düğümlere bağlanan kenar sayısı o düğümün derecesini ifade etmektedir. Tüm düğümlerinin dereceleri eşit olan graflara düzenli graf (regular graph) denilmektedir. Şekil 4.1'te düzenli graf örnekleri yer almaktadır.



Şekil 4.1. Düzenli graf örnekleri

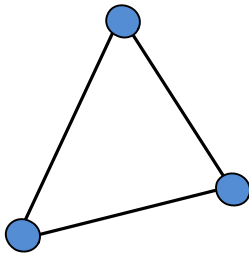
Bir graf içerisindeki tüm düğümler diğer düğümlere doğrudan bağlantıya sahip iseler bu tür graflara tamamlanmış graf (completed graph) denilir. Tamamlanmış graf

içerisinde n adet düğüme karşılık $n(n-1)/2$ adet kenar bulunur. Şekil 4.2 da tamamlanmış graf örnekleri yer almaktadır.



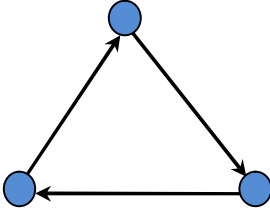
Şekil 4.2. Tamamlanmış graf örnekleri

Graf yapısındaki kenarlar değer alabilir ve bu değerler grafin yapısına katılabilir. Bir grafin üzerindeki kenarların değerleri eşit değilse ve her biri farklı bir değer alabiliyorsa bu tip graflara maliyetli veya ağırlıklı graf (weighted graph) denir. Bütün kenarların değeri aynı ise, ağırlıkların bir anlamı olmayacağından bu graf, maliyetli graf olarak anılamaz ve her kenarın değerinin 1 olduğu basit graf gibi değerlendirilir. Basit graflar (simple graphs), yönsüz, paralel kenarı olmayan ve döngü içermeyen graflardır (Şekil 4.3).



Şekil 4.3. Üç düğümden oluşan bir basit yönsüz graf

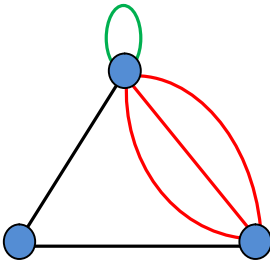
Eğer bir graftaki kenarlar yön bilgisine sahipse bu tür graflara yönlü graf (directed graf) denir. Bu yön bilgisi bağlantının nereden başlayıp nereden bittiğini belirtir. Yön bilgisi olan graflarda eğer iki yönde de bağlantı bulunuyorsa, gösterim için ters yönde iki ayrı kenar kullanılır. Graf yapısında bütün kenarlar aynı çeşittir. Yani ya hepsi yönlüdür ya da değildir (Şekil 4.4). Yol ağını temsil eden bir grafta trafiğin tek ya da çift yönlü oluşu, yönlü graflar için bir örnektir.



Şekil 4.4. Üç düğümden oluşan bir yönlü graf

Yönlü graflarda, bir düğümden diğer düğüme gitmek mümkün ise bu işlemin tersinin de olabileceği varsayılır. Ancak bu durum her zaman mümkün değildir. Bazı graflarda düğümler arası geçişler, sadece tek yöne izin verebildiği gibi, çift yönlü olan kenarın gidiş ve dönüş maliyetleri de farklı olabilir. Örneğin, çift yönlü ve maliyetleri farklı grafların kullanıldığı bir oyunda, nehir içerisinde ilerleyen bir teknenin, akıntı yönünde ve akıntıya ters yönde aldığı aynı uzunluktaki yol için harcadığı enerji farklı olacaktır.

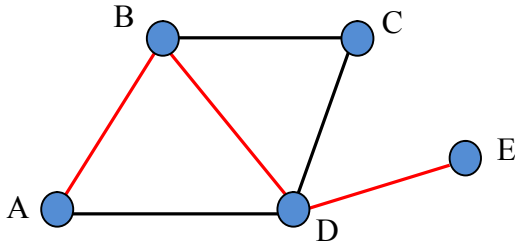
Başladığı düğüme geri dönen ve aynı düğümden iki kez geçmeyen yolculuğa döngü (loop) denilmektedir. Bir graftaki kenar sayısı düğüm sayısına eşit veya fazlaysa, o graf en az bir döngü içeriyor demektir. Yönsüz, paralel kenarı olan ve döngü içermeyen graflara çoklu graf (multi graph) denir ve basit grafların yeterli olmadığı durumlarda kullanılır (Şekil 4.5).



Şekil 4.5. Üç düğümden oluşan bir çoklu graf

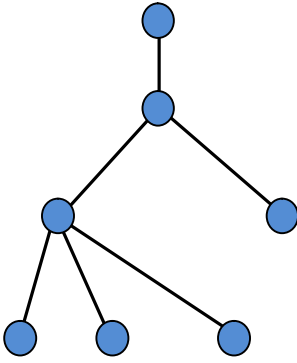
Yalancı graflar (pseudo graphs), yönsüz, paralel kenarı olan ve döngü içeren graflardır ve çoklu grafların yeterli olmadığı durumlarda kullanılırlar. Yönsüz grafların en temel halidir.

Bir graf üzerinde bir veya daha fazla düğümden ve kenardan geçen rota yol (path) olarak isimlendirilmektedir. Yollar, geçtikleri düğümlerin yazılması ile ifade edilirler. Şekil 4.6'de yer alan yol, $\{A,B,D,E\}$ şeklinde gösterilir.



Şekil 4.6. Dört düğümden geçen bir yol

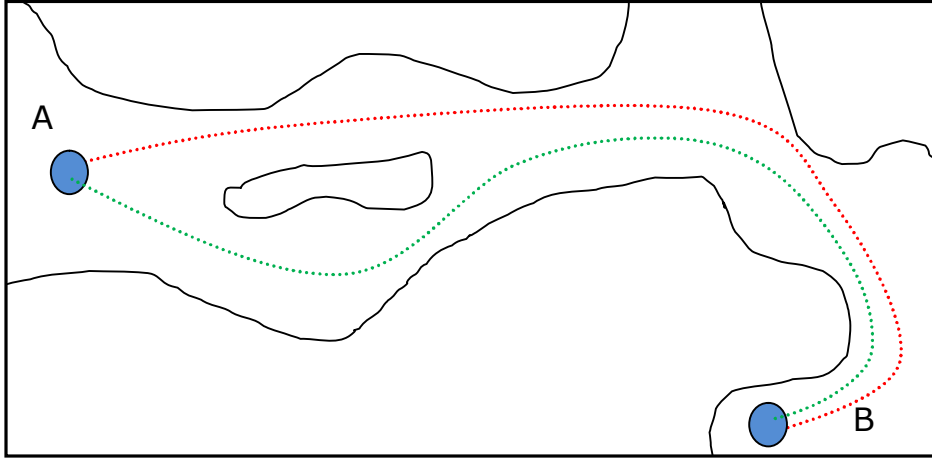
İçinde döngü barındırmayan grafa ağaç adı verilmektedir. Bir ağaçtaki kenar sayısı düğüm sayısının bir eksiği kadardır (Şekil 4.7).



Şekil 4.7. Ağaç şeklinde bir graf

4.2. Graflarda Arama ve Yol Bulma Algoritmaları

Graflar, günümüzde elektronik devreler, ulaşım ağları, bilgisayar ağları ve veritabanları gibi birçok alanda kullanılmaktadır. Gerek oyunlarda, gerekse bahse konu alanlardaki problemlerin çözümlerinde, çizgeler üzerindeki arama ve yol bulma işlemleri ön plana çıkmaktadır. Örneğin, Şekil 4.8’de yer alan harita üzerinde A noktasından B noktasına gidilecek uygun yolun/rotanın tayini çizgeler kullanılarak gerçekleştirilebilmektedir.



Şekil 4.8. Harita üzerinde yol bulma işlemi

Graflar üzerindeki arama algoritmaları, Kör/Habersiz Arama Algoritmaları ve Sezgisel Arama Algoritmaları olarak ikiye ayrılabilir.

4.2.1. Habersiz arama algoritmaları

Habersiz Arama Algoritmalarından bazıları aşağıda sıralanmıştır.

- a. Derinlik Öncelikli Arama –Depth First Search (DFS)
- b. Genişlik Öncelikli Arama –Breadth First Search (BFS)
- c. Düşük Maliyetli Arama (Dijkstra) –Uniform Cost Search (UCS)
- d. Sınırlı Derinlikte Arama -Depth-Limited Search (DLS)
- e. Tekrarlı Derinlikte Arama –Iterative Deeping Search (IDS)
- f. İki Yönlü Arama –Bidirectional Search (BS)

4.2.1.1. Dijkstra algoritması

Dijkstra En Kısa Yol Algoritması (Dijkstra Shortest Path Algorithm), Edsger Wybe Dijkstra tarafından 1956 yılında geliştirilmiştir [13]. Ağırlıklı ve yönlü graflarda kullanılan Dijkstra algoritması, seçilen bir başlangıç noktasına (başlangıç düğümüne) göre en kısa yolu bulmaktadır. Ağırlıklı graflarda kullanılmasına karşın, graf

içerisindeki kenarların ağırlıkları eksi değer almamalıdır. Dijkstra algoritması içerisinde, gidilecek bir sonraki düğüm, Greedy yaklaşımı kullanılarak tespit edilir. Greedy yaklaşımı, herhangi bir graf üzerinde hedefe giden en uygun yolun bulunması için kullanılan bir karar verme yöntemidir.

Bir grafın herhangi bir noktasından başlayan en az dallanan ağacı bulmak için, o noktadan itibaren toplam düğüm sayısının bir eksiği kadar yineleme (iteration) yapılması gerekmektedir [12].

4.2.1.2. Dijkstra algoritmasının çalışması

Dijkstra algoritması, başlangıç düğümünden hedef düğümüne varmak için en kısa yol ağacını oluşturacak şekilde çalışır. Algoritmanın çalışması esnasında gidilen her bir düğümün başlangıç düğümüne olan uzaklığı hesap edilir. Söz konusu düğümlerden uzaklığı en küçük olan düğüm işaretlenir ve bir sonraki düğüme geçilir. Böylece en kısa yol ağacı oluşturularak hedef düğüme kadar ilerlenir. Algoritmanın çalışması sona erdiğinde elde edilen ağaç, kaynak düğüm ile hedef düğüm arasındaki en kısa yolu içerecektir.

4.2.1.3. Dijkstra algoritmasının karmaşıklığı

Dijkstra Algoritmasının çalışması esnasında, kaynak (başlangıç) düğümünün etiketlenmesi işlemi 1, başlangıç düğümü haricindeki düğümlerin etiketlenmesi işlemi $(n-1)$, düğümlerin etiketlerinin güncellenmesi ve ebeveynlerinin atanması işlemi ise $(n-1)$ kere yapılmaktadır.

Her işlemde en çok $(n-2)$ karşılaştırma yapılmakta, etiketleri gözden geçirme işleminde $(n-1)$ düğüm ele alınmakta ve her birinde toplama, karşılaştırma ve 2 adet atama işlemi ile birlikte 4 işlem yapılmaktadır. Bu aşamaya kadar toplam $(n-1)(5n-5)$ tane işlem olmaktadır.

Son olarak, düğümlerin ebeveynlerinin takip edilerek en kısa yolun elde edilmesi işleminde en çok $(n-1)$ düğümün ebeveyni izlenmekte ve n tane işlem yapılmaktadır. Toplamda, n düğümlü bir çizge için, en kötü durumda $(5n^2-8n+5)$ tane işlem olmaktadır.

Bu çerçevede, Dijkstra algoritmasının karmaşıklığının $O(n^2)$ olduğu görülmektedir [14].

4.2.2. Sezgisel arama algoritmaları

Sezgisel algoritma ve yöntemler, çeşitli sebeplerle kesin bir çözümü olmayan veya çok karmaşık hesaplamalar neticesinde sonuca ulaşılabilen problemlerde tercih edilmektedir.

Graf arama algoritmalarında, sezgisel yöntemler genellikle, bulunulan düğümden bir sonraki düğüme en uygun çözümü tahmin etme işlemlerinde tercih edilmektedir. Bu işlemler esnasında, hedef düğüm gözetilerek, en kısa yolu oluşturan düğümlerin tespit edilmesine çalışılır.

Sezgisel yöntemler her zaman en kısa yolun belirlenmesinde doğru sonucu bulunmasını garanti etmeyebilirler ancak işlem miktarını ve dolayısıyla kaynak kullanımını önemli ölçüde azaltmaları sebebiyle kabul edilebilir çözümler üretmektedirler.

Sezgisel Arama Algoritmalarının bazıları aşağıda sıralanmıştır.

- a. Dal-Sınır Arama - Branch and Bound Search (B&BS)
- b. En İyi Öncelikli Arama - Best-First Search (BFS)
- c. Greedy Search (GBFS)
- d. A* Arama - A* Search (A*)
- e. IDA* Arama - Iterative Deeping A* Search (IDA*)
- f. Tepe Tırmanma - Hill Climbing Search (HCS)

- g. Isıl İşlem Arama - Simulated Annealing (SA)
- h. Tabu Arama - Tabu Search (TS)
- i. Genetik Arama - Genetik Search (GA)

4.2.2.1. A-Yıldız (A*) algoritması

A* algoritması, içerisinde barındırdığı buluşsal fonksiyon sebebiyle yapı olarak sezgisel (heuristic) algoritma olarak isimlendirilmektedir. A* algoritmasında kullanılan sezgisel fonksiyonu $f(n)$ ile ifade edecek olursak;

$$f(n) = g(n) + h(n) \quad (4.1)$$

$g(n)$ = Başlangıç düğümünden mevcut düğüme kadar gelme maliyeti.

$h(n)$ = Mevcut düğümünden hedef düğüme varmak için tahmin edilen mesafe.

4.2.2.2. A-Yıldız algoritmasının çalışması

Dijkstra algoritmasında kullanılan başlangıç düğümünden itibaren bir noktadan dışarıya doğru bütün yönlerde ilerleme metodunun aksine, A* algoritmasında hedef düğüme doğru, doğrudan bir maliyet hesaplanır. Algoritma ilerleme yönünü bu maliyetin artıp azalmasına göre ayarlamaktadır. Böylece karmaşık ortamlarda dahi, algoritma $g(n)$ fonksiyonunu kullanarak hedeften uzaklaştığını anlayabilir. $g(n)$ parametresi aynı zamanda maliyet ile hedefe yaklaşma arasında bir denge unsurudur. Diğer bir deyişle $g(n)$ fonksiyonu, algoritmanın hızı ile doğruluğu arasında belirleyici bir rol oynamaktadır.

Algoritmanın çalışması esnasında öncelikle kaynak düğümün komşu düğümleri ziyaret edilir. Ardından $f(n)$ değeri en düşük olan düğüm öncelikli olmak üzere hedef düğüm bulunana kadar ilerlenmeye devam edilir. Her bir adımda, bir önceki düğüm, gidilen düğümün ebeveyni olarak işaretlenir. Böylece hedef düğüme ulaşıldığında düğümlerin ebeveynleri takip edilerek başlangıç ve hedef düğümleri arasındaki yol elde edilir.

Gidilmesi muhtemel düğümler, algoritma içerisinde “açık liste” isimli bir listede tutulmaktadır. Açık listede bulunan ve ziyaret edilen düğümler tekrar kontrol edilmemesi için “kapalı liste”ye alınmaktadır. Kapalı listede yer alan bir düğüm, daha kısa bir yol bulunması durumunda tekrar açık listeye alınmaktadır.

Açık liste içerisindeki düğümler, başlangıç düğümünden hedef düğüme kadar olan toplam maliyetlerine göre sıralıdır. Böylece öncelik sıralamasına sahip bir liste elde edilir. Bahse konu toplam maliyet, mevcut düğüm ile başlangıç düğümü arasındaki gerçek maliyet ile mevcut düğüm ile hedef düğüm arasındaki tahmini maliyetin toplamıdır.

Hedef düğüme ulaşıldığında veya açık listede düğüm kalmayınca algoritma sonlandırılır.

A* algoritması, sezgisel fonksiyonun hesapladığı tahmini maliyetin gerçek maliyetten fazla olmadığı durumlarda en kısa yolu verir. Ancak tahmini maliyet gerçek maliyetten fazla olursa bulunan yol, gerektiği kadar kısa değildir [15].

A* algoritması içerisinde sezgisel maliyet sıfır olarak alınır, diğer bir ifadeyle $h(n)$ 'in değeri sıfıra eşitlenirse, algoritma Dijkstra algoritması gibi davranış sergileyecektir. Diğer taraftan $g(n)$ fonksiyonunun değeri sıfıra eşitlenirse algoritma, Best-First Search algoritması gibi davranış gösterecektir. İlk durumda taranacak düğüm sayısı ve işlem zamanı artacak, ikinci durumda ise hedefe yaklaşma-uzaklaşma kontrolü yapılamayacaktır.

A* algoritmasının daha hızlı çalışabilmesi için geçilmesi istenmeyen düğümlere çok yüksek değerler verilebilir. Ancak en doğru çözüm, bu düğümlerin kapatılması veya çizgeden çıkarılması olacaktır. Ayrıca yoğun bellek kullanımı, A* algoritmasının en önemli dezavantajlarından birisi olduğundan, kullanılan bellek miktarının artırılması algoritmanın daha hızlı çalışmasını sağlayacaktır.

Algoritma içerisinde kullanılan sezgisel yöntemler sebebiyle, kaynak düğümünden hedef düğüme ve hedef düğümünden kaynak düğüme giden en kısa yollar farklı çıkabilir. Bu

nedenle yeterince hızlı sistemlerde, kaynak düğümden hedef düğüme ve hedef düğümden kaynak düğüme giden en kısa yollar hesaplanır ve elde edilen iki sonuçtan en uygunu seçilerek kullanılabilir [16].

Ayrıca A-Yıldız algoritmasının çeşitli iyileştirmeler içeren Iterative Deepening A* (IDA*), Memory-Bounded A* (MA*), Simplified Memory Bounded A* (SMA*) ve Recursive Best-First Search (RBFS) şeklinde türevleri mevcuttur [17].

A* algoritmasında, $h(n)$ sezgisel maliyetin (hedefe olan uzaklığın) hesaplamasında kullanılan yöntemlerden bazıları aşağıda sıralanmıştır [18].

- a. Manhattan Uzaklığı Yöntemi
- b. Euclidean Uzaklığı Yöntemi
- c. Diagonal Uzaklığı Yöntem

4.2.2.3. Manhattan uzaklığı yöntemi

Manhattan Uzaklığı Yöntemi, sezgisel maliyet hesaplamalarında kullanılan standart yöntemdir. Mevcut düğüm ile hedef düğüm arasındaki yatay ve dikey mesafelerin toplanmasını esas almaktadır.

$$h(n) = D * (|mdX - hdX| + |mdY - hdY|) \quad (4.2)$$

$h(n)$: Sezgisel Maliyet Fonksiyonu

D : Komşu Düğüme Gitme Asgari Gerçek Maliyeti

mdX : Mevcut Düğümün X Koordinatı

mdY : Mevcut Düğümün Y Koordinatı

hdX : Hedef Düğümün X Koordinatı

hdY : Hedef Düğümün Y Koordinatı

4.2.2.4. Euclidean uzaklığı yöntemi

Euclidean Uzaklığı Yöntemi, en çok kullanılan sezgisel maliyet hesaplama yöntemlerinden birisidir. Mevcut düğümden hedef düğüme olan yatay ve dikey uzaklıkların karelerinin toplamının karekökü alınarak hesaplanır. Bu uzaklıkların kareleri alınması sebebiyle aykırı değerlerin maliyet fonksiyonuna etkisi Manhattan uzaklığı yöntemine göre daha fazladır.

$$h(n) = D * \sqrt{(mdX - hdX)^2 + (mdY - hdY)^2} \quad (4.3)$$

$h(n)$: Sezgisel Maliyet Fonksiyonu

D : Komşu Düğüme Gitme Asgari Gerçek Maliyeti

mdX : Mevcut Düğümün X Koordinatı

mdY : Mevcut Düğümün Y Koordinatı

hdX : Hedef Düğümün X Koordinatı

hdY : Hedef Düğümün Y Koordinatı

4.2.2.5. Diagonal uzaklığı yöntemi

Diagonal Uzaklığı Yöntemi'nde sezgisel maliyeti hesaplama işlemi diğer metotlardan farklı bir şekilde gerçekleşmektedir. Toplam Sezgisel maliyet, düz ve çapraz geçişler için ayrı ayrı hesap edildikten sonra elde edilir. Düz ve çapraz geçiş sezgisel fonksiyon sonuçları, komşu düğüme gitme asgari gerçek maliyeti ile çarpılarak toplam sezgisel maliyet elde edilmiş olur [18].

Komşu düğüme geçiş asgari gerçek maliyeti düz (D_1) ve çapraz (D_2) geçişlerde farklı olabilmektedir. Ayrıca bu yöntemdeki işlem yoğunluğu sebebiyle, işlem zamanı maliyeti yüksek olmaktadır.

$$h_{\text{Çapraz}}(n) = \min(|mdX - hdX|, |mdY - hdY|) \quad (4.4)$$

$$h_{\text{Düz}}(n) = (|mdX - hdX| + |mdY - hdY|) \quad (4.5)$$

$$h(n) = (D_2 \times h_{\text{Çapraz}}(n)) + D_1 \times (h_{\text{Düz}}(n) - 2 \times h_{\text{Çapraz}}(n)) \quad (4.6)$$

$h(n)$: Toplam Sezgisel Maliyet Fonksiyonu
$h_{\text{Düz}}(n)$: Düz Geçişlerde Sezgisel Maliyet Fonksiyonu
$h_{\text{Çapraz}}(n)$: Çapraz Geçişlerde Sezgisel Maliyet Fonksiyonu
D_1	: Düz Geçişlerde Komşu Düğüme Gitme Asgari Gerçek Maliyeti
D_2	: Çapraz Geçişlerde Komşu Düğüme Gitme Asgari Gerçek Maliyeti
mdX	: Mevcut Düğümün X Koordinatı
mdY	: Mevcut Düğümün Y Koordinatı
hdX	: Hedef Düğümün X Koordinatı
hdY	: Hedef Düğümün Y Koordinatı

4.2.2.6. A-Yıldız algoritmasının karmaşıklığı

A-yıldız algoritmasının zaman karmaşıklığı, sezgisel fonksiyona bağlı olarak değişmektedir. En kötü durumda, incelenen toplam düğüm sayısı ve en kısa yol çözümünün uzunluğuna bağlı olarak üsseldir. Ancak arama uzayı ağaç yapısında ve hedef nokta bir tane ise zaman karmaşıklığı çok terimlidir (polynomial). Bu durumda sezgisel fonksiyon (h) aşağıdaki gibidir [17].

$$|h(x) - h^*(x)| = O(\log h^*(x)) \quad (4.7)$$

$h(x)$: Tahmin edilen sezgisel maliyet (x düğümünden hedef düğüme tahmini maliyet)
$h^*(x)$: Sezgisel parametrenin en uygun olduğu durum (x düğümünden hedef düğüme gerçek maliyet)

Diğer bir deyişle, h sezgisel fonksiyonun hata oranı, kaynak düğümünden hedef düğüme olan gerçek uzaklığın logaritmasından daha hızlı büyümeyebilir.

A-Yıldız algoritmasının alan karmaşıklığı, zaman karmaşıklığından daha sorunsaldır. En kötü durumda, bütün düğümlerin hafızada tutulması sebebiyle $O(b^m)$ olarak ifade edilmektedir [19].

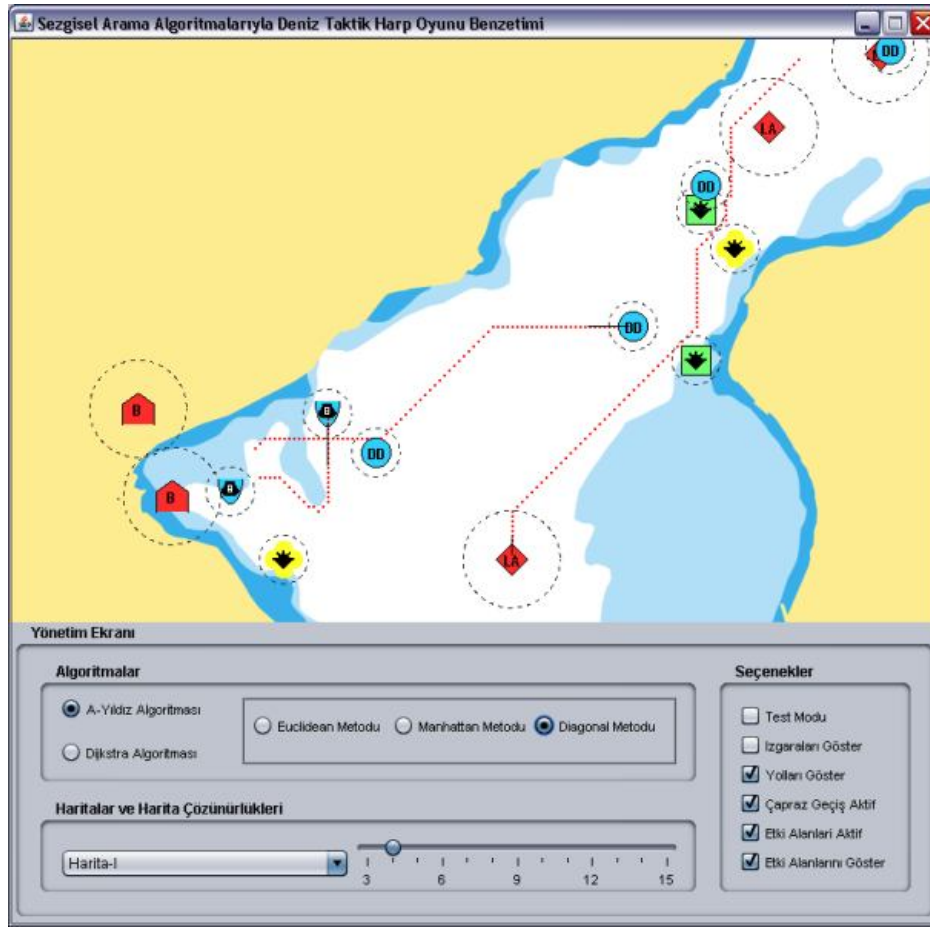
b : Algoritmanın Dalların Faktörü

m : Arama Ağacındaki Azami Derinlik

BÖLÜM 5. SEZGİSEL ARAMA ALGORİTMALARIYLA DENİZ TAKTİK HARP OYUNU BENZETİMİ YAZILIMININ GELİŞTİRİLMESİ

5.1. Geliştirilen Yazılımın Özellikleri

Deniz Taktik Harp Oyunu Benzetimi, harita üzerindeki dost ve düşman unsurların, kabiliyetleri ölçüsünde, taktik harekât ve manevralar yaparak karşı unsurlara üstün gelme mücadelesini resmeden bir yazılımdır (Şekil 5.1).



Şekil 5.1. Sezgisel Arama Algoritmalarıyla Deniz Taktik Harp Oyunu Benzetimi

Bu çalışma çerçevesinde, harita üzerindeki söz konusu unsurların buldukları noktadan emredilen hedef noktaya akıllı olarak gitmelerini sağlayacak algoritmalar kullanılmıştır. Harekât emri alan unsur, harita üzerinde bulunduğu coğrafi konumu ve diğer unsurların konumlarını da hesaba katarak en uygun yolu bulabilmektedir.

En uygun yolu hesap edip o yol üzerinde harekete geçen araç, diğer araçların yer değiştirmesi, hizmet dışı kalması vb. durumlarda yolunu dinamik olarak değiştirebilmekte ve son andaki koşulları dikkate alarak kendisine yeni rotalar tayin edebilmektedir.

5.2. Kullanılan Teknoloji ve Araçlar

Benzetim yazılımı, temelinde Java teknolojisi olmak üzere, çeşitli araçlar kullanılarak geliştirilmiştir. Bahse konu teknoloji ve araçlar Tablo 5.1’de yer almaktadır.

Tablo 5.1. Benzetim yazılımı geliştirirken kullanılan teknoloji ve araçlar

Özellik	Teknoloji/Araç
Programlama Dili	Java 1.6
Geliştirme Ortamı	Netbeans IDE 6.7.1
Grafik Kütüphanesi	Java 2D
Platform	JDK 6 Update 15
Kullanılan Algoritmalar	A-Yıldız ve Dijkstra
Sezgisel Yöntemler	Euclidean, Manhattan ve Diagonal

Yazılımın Java teknolojisi kullanılarak geliştirilmesinin başlıca sebepleri şunlardır;

- a. Java dilinin nesneye dayalı programlama yapısı nedeniyle, yazılım hem hızlı bir şekilde geliştirilmiş, hem de genişletilmeye uygun esnek bir yapıya kavuşmuştur.

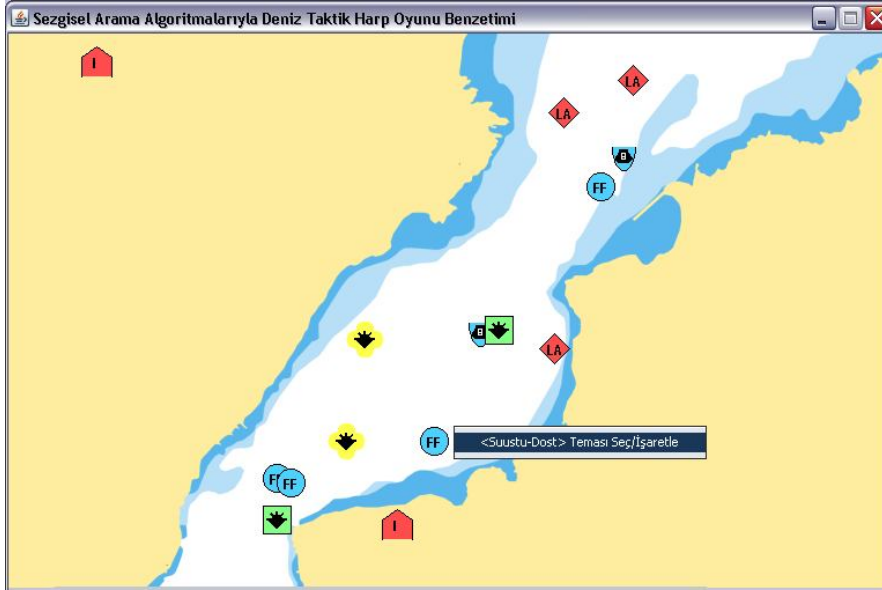
- b. Yazılım performansına etki eden önemli özelliklerden birisi olan çok-kanallı yapı, Java programlama dili tarafından desteklenmektedir.
- c. Bu teknoloji ile benzetim yazılımı platform bağımsız olarak geliştirilmiştir. Böylece Java Runtime Environment (JRE) kütüphanesi yüklenebilen herhangi bir işletim sistemi üzerinde çalıştırılabilmektedir.
- d. Benzetim yazılımı, Java Web Start teknolojisine uygun şekilde kodlandığından Internet ağı üzerinden de çalıştırılabilmektedir.

5.3. Yazılımın Kullanıcı Arayüzü

Benzetim yazılımı Harita Ekranı ve Yönetim Ekranı olmak üzere iki ana bölümden oluşmaktadır.

5.3.1. Harita ekranı

Harita Ekranında, yüklenen haritalar ve bu haritalar üzerindeki temaslar izlenmektedir. Tüm grafik işlemlerinin sonucu bu ekran üzerinden izlenmektedir. Ayrıca harita ekranı üzerinde fare kontrolleri bulunmakta, haritada bulunan temasların seçimleri fare ile yapılabilmektedir (Şekil 5.2).







Şekil 5.2. Benzetim yazılımının harita ekranı

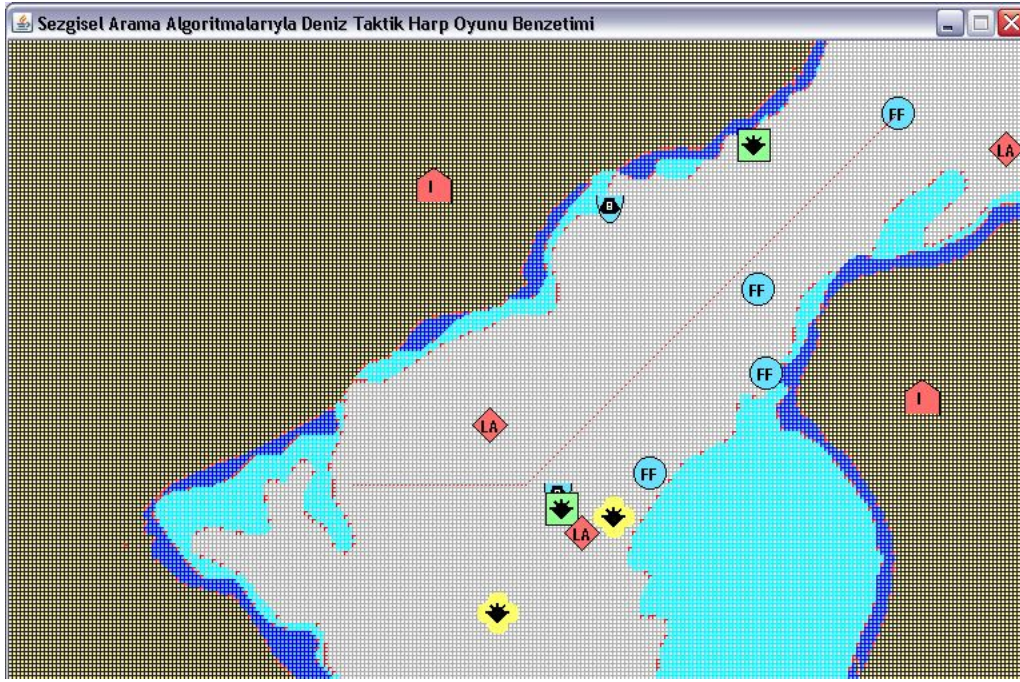
5.3.1.1. Haritaların yüklenmesi ve coğrafi alanın yazılım tarafından tanınması

Benzetim yazılımı içerisinde çeşitli haritalar bulunmaktadır. Bu haritalar, kara ve deniz ile birinci ve ikinci derece sığ sular şeklinde bölgelere ayrılmış vektörel haritalardır. Söz konusu bölgeler Tablo 5.2’de yer almaktadır.

Tablo 5.2. Harita üzerindeki renkler ve coğrafi karşılıkları

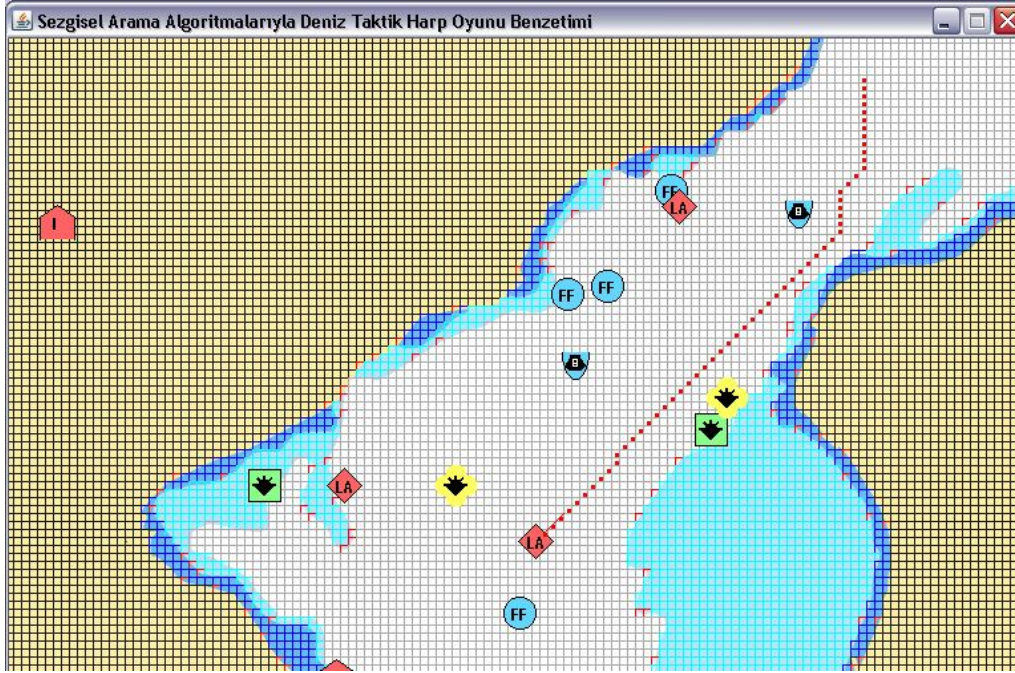
Harita Üzerindeki Renk	Karşılığı
	Karasal Alanlar
	Derin Sular
	I. Derece Sığ Sular
	II. Derece Sığ Sular

Haritaların yüklenmesi yazılım içerisindeki seçme kutusu vasıtasıyla yapılmaktadır. Seçilen haritanın ekrana yüklenmesini müteakip, harita arka planda ızgaralara bölünmektedir. Izgaraların büyüklüğü, benzetim yazılımını kullanan kişi tarafından yazılım yönetim arayüzünden ayarlanabilmekte ve değeri 3–15 benek (pixel) arasında değişmektedir. Şekil 5.3'te ızgara boy ve eninin 3 benek olduğu durumda seyir yapan temas görülmektedir.



Şekil 5.3. Izgara boyutunun çok küçük olduğu durumlarda rota hassasiyeti

Izgara boyutunun çok küçük olduğu durumlarda belirlenen rotaların hassasiyetinin artmasına karşın düğüm sayısı ve işlem yükü de artmaktadır. Diğer taraftan ızgara boyutu büyütüldüğünde ise rota hassasiyetinin azalmasına karşın işlem yükünde gözle görülür bir azalma sağlanmaktadır (Şekil 5.3 ve Şekil 5.4).



Şekil 5.4. Izgara boyutunun büyük olduğu durumlarda rota hassasiyeti

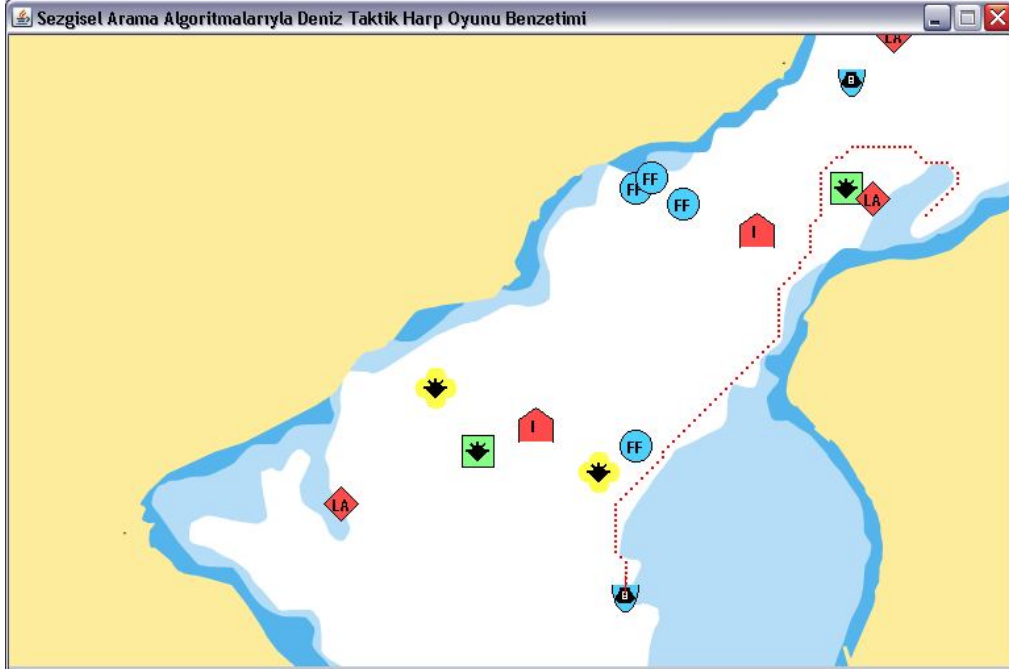
Izgara boyutunun değişimine göre, 768x480 benek büyüklüğündeki bir harita üzerinde taranan ızgara(düğüm) sayısı Tablo 5.3’de yer almaktadır.

Tablo 5.3. Izgara hücresinin uzunluğuna göre harita üzerindeki düğüm sayısı.

Izgara Hücresinin Uzunluğu (benek)	Harita Üzerindeki Düğüm Sayısı
3	41377
4	23353
5	14938
6	10449
8	5917
10	3773
12	2665
15	1716

Her bir ızgara hücresinin merkez noktasının harita üzerinde denk geldiği coğrafi konum, yazılım tarafından işaretlenmekte ve unsurların hareketi bu coğrafi bilgiler doğrultusunda gerçekleştirilmektedir.

Örneğin suüstü gemileri, derin sular, birinci ve ikinci derece sığ sularda hareket edebilirken; denizaltılar, sadece derin sularda hareket edebilmektedirler. Şekil 5.5'te hareket emri verilen bir denizaltının akıllı olarak rotasını bulabildiği ancak sadece derin suları kullanarak seyir yaptığı görülmektedir.



Şekil 5.5. Balistik füze taşıyan nükleer denizaltının sadece derin suları kullanarak seyri.

5.3.1.2. Haritalar üzerinde kullanılan gösterim sembolojileri




Gösterim sembolojileri üçüncü bölümde ayrıntılı olarak incelenmiştir. Bu tez çalışması çerçevesinde, değişen ihtiyaçlara yönelik olarak geliştirilen ve önceki sembolji standartlarında yer alan eksiklikleri tamamlayan Common Warfighting Symbology MIL-STD-2525C (USA) ve Military Symbols for Land Based Systems (STANAG) 2019 APP-6A (NATO)'ye ait sembolojiler kullanılmıştır. Geliştirilen benzetim yazılımında kullanılan simgelerin kimliklerine göre renk değerleri Tablo 5.4'de yer almaktadır.

Tablo 5.4. Geliştirilen benzetim yazılımında kullanılan simge renk değerleri

Temasın Kimliği (ID)	Renkler	Renk Değerleri
Dost	Mavi	RGB (128, 224, 255)
Tanımsız	Sarı	RGB (255, 255, 128)
Tarafsız	Yeşil	RGB (170, 255, 170)
Düşman	Kırmızı	RGB (255, 128, 128)

Ayrıca geliştirilen benzetim yazılımında kullanılan simgelerden bazıları kimlik bilgisi, sınıfı ve platformuna göre Tablo 5.5’de yer almaktadır.

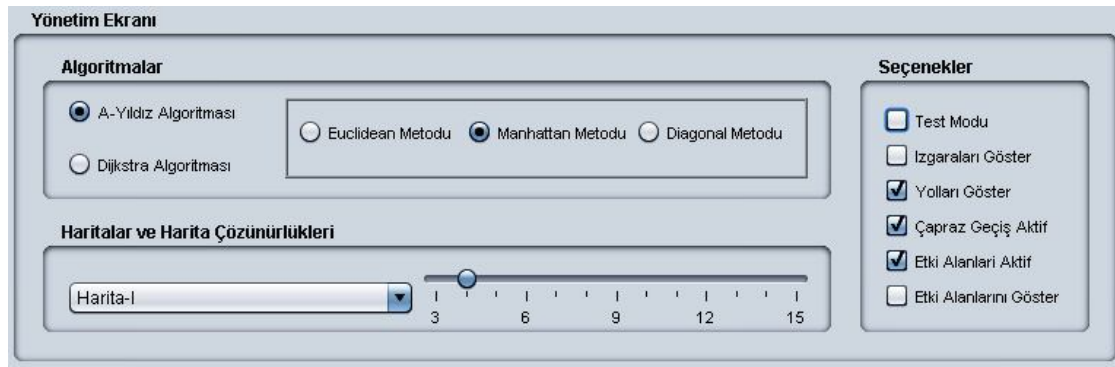
Tablo 5.5. Geliştirilen benzetim yazılımında kullanılan simgeler.

Sembol	Temasın Sınıfı	Temasın Kimliği	Platform
	Suüstü	Dost	Firkateyn/Korvet
	Suüstü	Düşman	Amfibi Çıkarma Gemisi
	Suüstü	Tarafsız	Mayın Avlama Gemisi
	Suüstü	Tanımsız	Mayın Avlama Gemisi
	Suüstü	Düşman	Hücumbot
	Denizaltı	Dost	Balistik Füze Taşıyan Nükleer Denizaltı
	Hava	Düşman	Bombardıman Uçağı
	Hava	Düşman	Avcı Uçağı

5.3.2. Yönetim ekranı

Yönetim ekranı, benzetim yazılımının özellik ve kabiliyetlerinin daha rahat gözlemlenmesi için tasarlanmış esnek bir arayüzdür (Şekil 5.6). Üç ana bölümden oluşmaktadır. Bunlar;

- Algoritmalar
- Haritalar ve Harita Çözünürlükleri
- Seçenekler



Şekil 5.6. Benzetim yazılımını yönetim ekranı.

Algoritmalar bölümünde, A-Yıldız ve Dijkstra algoritmalarının seçilebildiği düğmeler mevcuttur. Yazılım, dinamik olarak tasarlandığından, harita üzerindeki unsur harekete başlamış olsa bile kullanılan algoritma bu bölümden rahatlıkla değiştirilebilir. Söz konusu unsur yolunun kalan kısmını en son seçilen algoritmaya göre tamamlar.

A-Yıldız algoritması kullanıldığında, sezgisel yöntemin belirlenmesi için Euclidean, Manhattan ve Diagonal olmak üzere üç adet yöntem seçilebilmektedir. Bahse konu sezgisel yöntemler, unsur hareket halindeyken dahi değiştirilebilecek şekilde kodlanmıştır. Ayrıca sezgisel yöntemleri değiştiren düğmeler, Dijkstra algoritması seçildiğinde pasif duruma geçmektedir.

Haritalar ve Harita Çözünürlükleri bölümünde, yeni haritaların ekrana yüklenme işlemi yapılmaktadır. Böylece yazılımın özellikleri, farklı haritalar üzerinde test

edilebilmektedir. Ayrıca haritalar üzerindeki ızgaraların büyüklüklerinin ayarlanması da bu bölümden gerçekleştirilmektedir.

Seçenekler bölümünde yazılımın daha ayrıntılı kullanımını sağlayan kontroller bulunmaktadır. Bu kontrollerin işlevleri Tablo 5.6'da yer almaktadır.

Tablo 5.6. Benzetim yazılımı seçenekler bölümündeki kontroller ve işlevleri

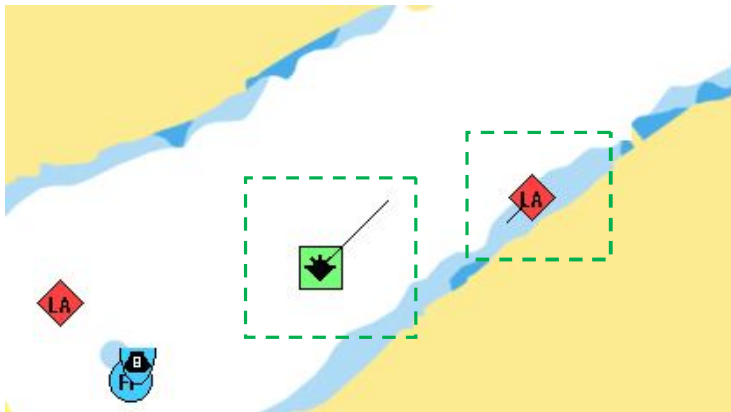
Seçenekler	Aktif Konumda	Pasif Konumda
Test Modu	Benzetim yazılımını Test Modu'na alır.	Benzetim yazılımını Test Modu'ndan çıkarır.
Izgaraları Göster	Izgaraları harita üzerinde gösterir.	Harita üzerindeki ızgaraları gizler.
Yolları Göster	Aracın üzerinde seyrettiği rotayı/yolu gösterir.	Aracın üzerinde seyrettiği rotayı/yolu gizler.
Çapraz Geçiş Aktif	Rota belirleme işlemlerinde araçlara ızgaralar arasında çapraz geçiş yapabilme izni verir.	Rota belirleme işlemlerinde araçlara ızgaralar arasında çapraz geçiş yapabilme iznini kaldırır.
Etki Alanları Aktif	Rota belirleme işlemlerinde temasların vuruş ve etki alanlarının dikkate alınmasını sağlar.	Rota belirleme işlemlerinde temasların vuruş ve etki alanları dikkate alınmaz.
Etki Alanları Göster	Temasların vuruş/etki alanlarının harita üzerinde halka ile işaretlenmesini sağlar.	Temasların harita üzerindeki vuruş/etki alanlarını kaldırır.

5.4. Yazılımın Teknik Altyapısı

Benzetim yazılımı içerisindeki ızgaralar arka planda “düğüm” olarak temsil edilmektedir. Böylece harita üzerindeki unsurların takibi bulunduğu düğümün tespit edilmesiyle kolayca gerçekleştirilebilmektedir. Söz konusu ızgaralar kare şeklinde tasarlanmıştır. Bu şekilde bölünen ızgaraların matris tablolarında tutulması ve yazılım içerisinde iki boyutlu diziler şeklinde tanımlanabilmesi olanağı elde edilmiştir.

Seçilen haritanın yüklenmesi işleminin hemen sonrasında temasların uygun coğrafi konumlara atanması işlemi yapılmaktadır. Örneğin, hiçbir denizaltı kara parçasını temsil eden bir düğümde yer alamaz. Aynı zamanda herhangi bir suüstü aracı, kara bölgesindeki bir düğümde gitme emrini yerine getiremez.

Yazılım, çok kanallı (multi-threaded) olarak çalışmaktadır. Bu sebeple, seçilen her bir unsur için ayrı bir kanal (thread) açılmaktadır. Seçilen temasın hareketi boyunca bu kanal izlenmekte, hareketinin bitmesini müteakip o unsura tahsis edilmiş olan kanal sonlandırılmaktadır. Bu yapı, yazılımın işletme performansını artırmakla birlikte, hızları farklı olan unsurların ayrı ayrı hareket etmesini ve bunların rahatlıkla takip edilebilmesini sağlamaktadır. Örneğin Şekil 5.7’de iki unsur farklı hızlarda hareket etmekte ve birim zamanda farklı mesafeler kat etmektedir. Bu hareketler ise çok kanallı yapı ile sağlanmaktadır.



Şekil 5.7. Farklı hızlardaki iki temasın yazılım içerisindeki hareketi.

5.5. Yazılımın Test Modu İşlemleri

Benzetim yazılımının özellik ve kullanılan algoritmalarının daha iyi anlaşılması için test modu seçeneği kullanılmaktadır. Yazılım test moduna alındığında, seçilen temaların hareketleri gözlenmez. Ancak seçilen temasın bulunduğu düğüm ile hedef düğüm arasında algoritmalar işletilerek sonuçları ekranda gösterilir (Şekil 5.8).



Şekil 5.8. Test sonuçlarından bir ekran görüntüsü.

İşletim sistemi, sistem kaynaklarını, anlık olarak farklı iş parçacıklarına ayırabileceğinden, test işlemleri 10 defa tekrarlanarak sonuçların ortalamaları alınmıştır. Böylece test işlemleri sırasında meydana gelebilecek ani değişimlerin test sonuçlarına etkisi azaltılmıştır.

Test işlemleri süresince;

- a. A-Yıldız ve Dijkstra algoritmaları kullanılarak en uygun rota hesaplaması yapılmaktadır. A-Yıldız algoritması için Euclidean, Manhattan ve Diagonal metotlarının sonuçları ayrı ayrı hesaplanmaktadır.
- b. Algoritmaların işlem yapma süreleri milisaniye cinsinden ölçülmektedir.
- c. Algoritmaların rota hesaplaması işlemi boyunca kaç adet düğüm üzerinde işlem yaptığı ölçülmektedir.

BÖLÜM 6. TARTIŞMA VE ÖNERİLER

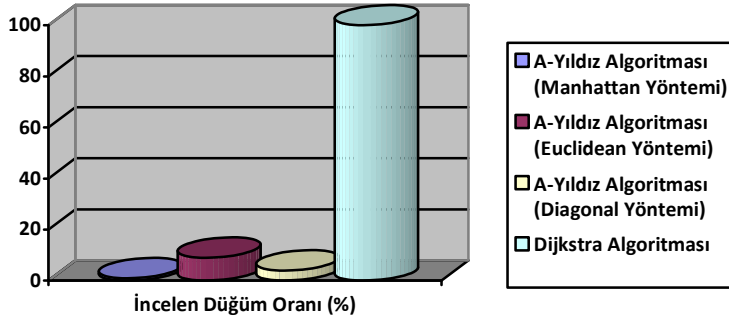
Bu tez çalışması çerçevesinde, akıllı algoritmalarla kendi rotalarını tayin edebilen harp unsurlarının bulunduğu bir Deniz Taktik Harp Oyunu Benzetimi geliştirilmiştir. Benzetim yazılımı içerisinde, rota bulma işlemleri için A-Yıldız ve Dijkstra algoritmaları kullanılmış ve bu algoritmaların performans ölçümleri yapılmıştır. Performans ölçümlerinin analiz edilmesi amacıyla, benzetim yazılımı içerisinde üç farklı senaryo işletilmiş ve elde edilen sonuçlar değerlendirilmiştir. Tablo 6.1'de Senaryo-I'e göre işletilen algoritma ve metotların performans değerleri yer almaktadır.

Tablo 6.1. Benzetim yazılımına uygulanan senaryo-I ve sonuçları

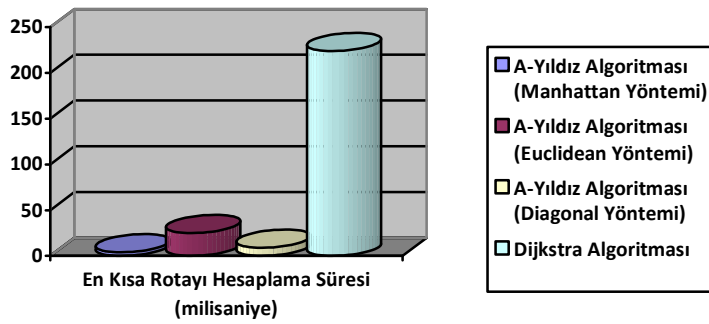
	Algoritma ve Yöntemler		Çizgedeki Toplam Düğüm Sayısı (adet)	Algoritma Tarafından İncelenen Düğüm Sayısı (adet)	En kısa Rotanın Ortalama Hesap Süresi (milisaniye)	Hesaplanan En Kısa Rotanın Uzunluğu (düğüm)
Senaryo I	A-Yıldız Algoritması	Manhattan Yöntemi	23353	240	4	31
		Euclidean Yöntemi	23353	2280	25	31
		Diagonal Yöntemi	23353	1152	9	31
	Dijkstra Algoritması		23353	23353	224	31

Benzetim yazılımı içerisinde analizi yapılan Senaryo-I'de, seçilen temas ile gitmesi istenilen hedef nokta arasında herhangi bir engel yoktur ve hedef noktanın başlangıç noktasına olan mesafesi kısadır. Bu durumda, A-Yıldız algoritması ile Dijkstra algoritmasının hesap ettiği en kısa yol uzunluklarının aynı (31 düğüm) olduğu görülmüştür. A-Yıldız algoritmasının en uygun rotayı hesaplama ortalama süresi, Dijkstra algoritmasına göre daha az çıkmıştır. Bu algoritmada kullanılan sezgisel fonksiyon sebebiyle, daha az düğüm incelendiğinden bu sonucun çıktığı

değerlendirilmektedir. Senaryo-I'e göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı Şekil 6.1'de yer almaktadır.



Şekil 6.1. Senaryo-I'e göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı



Şekil 6.2. Senaryo-I'e göre algoritmaların en kısa rotayı hesaplama süreleri

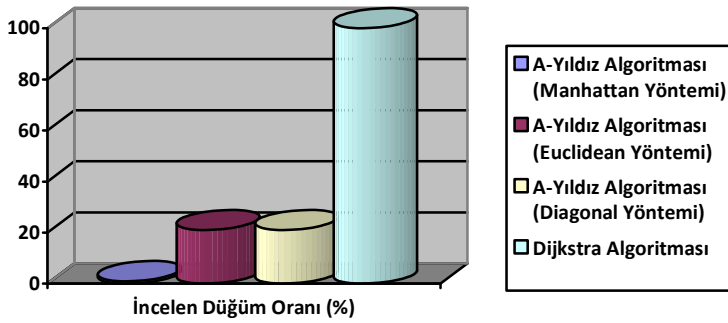
Dijkstra algoritması, rota hesaplama işleminde, doğası gereği bütün düğümleri taramakta ve kaynak düğümden her yöne uzaklıkları hesap etmektedir. Bu durum, rota hesaplama zamanının artmasına sebep olmuştur (Şekil 6.2).

Analizi yapılan diğer bir örnek olay Senaryo-II'de, seçilen unsura, orta uzaklıkta bir hedefe gitme emri verilmiştir. Bu senaryoda seçilen unsurun çevresinde çeşitli engeller bulunmaktadır. Bu durumda A-Yıldız ve Dijkstra algoritmalarının en kısa rota uzunluklarının (52 düğüm) aynı olduğu görülmüştür.

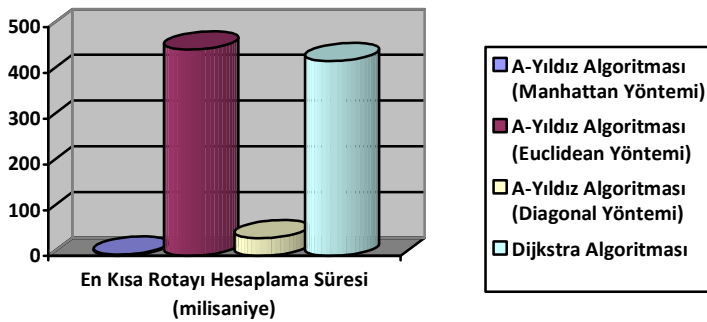
Tablo 6.2. Benzetim yazılımına uygulanan senaryo-II ve sonuçları

	Algoritma ve Yöntemler		Çizgedeki Toplam Düğüm Sayısı (adet)	Algoritma Tarafından İncelenen Düğüm Sayısı (adet)	En kısa Rotanın Ortalama Hesap Süresi (milisaniye)	Hesaplanan En Kısa Rotanın Uzunluğu (düğüm)
	A-Yıldız Algoritması					
Senaryo II	A-Yıldız Algoritması	Manhattan Yöntemi	23353	407	3	52
		Euclidean Yöntemi	23353	4999	451	52
		Diagonal Yöntemi	23353	4999	39	52
	Dijkstra Algoritması		23353	23353	426	52

Hedef noktanın başlangıç noktasına uzaklığının Senaryo-I'e göre yaklaşık 1.67 kat artmasına karşılık A-Yıldız algoritmasının (Euclidean yöntemi) taradığı düğüm sayısının 2.20 kat artması, artan düğüm sayısına karşılık algoritma performansının azaldığını göstermektedir (Tablo 6.2).



Şekil 6.3. Senaryo-II'ye göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı



Şekil 6.4. Senaryo-II'e göre algoritmaların en kısa rotayı hesaplama süreleri

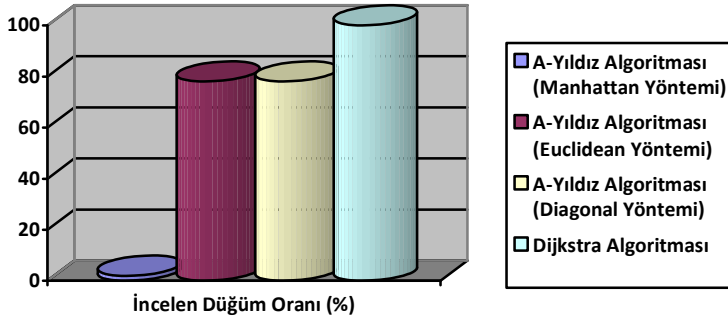
Senaryo-II'ye göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı Şekil 6.3'te yer almaktadır. Senaryo-II testlerinden elde edilen diğer bir sonuç ise Dijkstra algoritmasının rota hesaplama süresinin bazı durumlarda A-Yıldız algoritmasından (Euclidean) daha az olduğudur (Şekil 6.4).

Tablo 6.3. Benzetim yazılımına uygulanan senaryo-III ve sonuçları

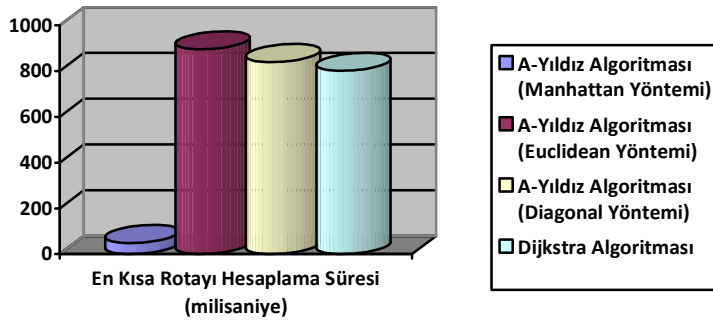
	Algoritma ve Yöntemler		Çizgedeki Toplam Düğüm Sayısı (adet)	Algoritma Tarafından İncelenen Düğüm Sayısı (adet)	En kısa Rotanın Ortalama Hesap Süresi (milisaniye)	Hesaplanan En Kısa Rotanın Uzunluğu (düğüm)
Senaryo III	A-Yıldız Algoritması	Manhattan Yöntemi	41377	1165	49	149
		Euclidean Yöntemi	41377	32324	895	149
		Diagonal Yöntemi	41377	32324	839	149
	Dijkstra Algoritması		41377	41377	801	149

Son incelenen senaryo olan Senaryo-III'te başlangıç noktası ile hedef nokta arasındaki mesafe uzun ve rota boyunca birçok hedef bulunmaktadır. Ayrıca bu senaryoda önceki iki senaryodan farklı olarak ızgara boyutu düşürülmüş, buna karşılık çizge üzerindeki düğüm sayısı arttırılmıştır. Bu durum belirlenen rotanın hassasiyetini artırmasına karşın, işlem yükünde de önemli bir artmaya da sebep olmuştur.

Bahse konu senaryoda A-Yıldız algoritması ve Dijkstra algoritmalarının hesapladıkları rota uzunluğunun aynı olduğu (149 düğüm) görülmüştür. Kaynak-hedef düğümler arası mesafe ve engeller sebebiyle A-Yıldız algoritmasının performansında gözle görülür bir düşüş meydana gelmiştir (Tablo 6.3). Bu performans düşüşünün asıl sebebinin engelle karşılaştığı algoritmanın, engelin etrafındaki düğümlerde zaman kaybetmesi olduğu değerlendirilmektedir. Senaryo-III'e göre algoritmaların inceledikleri düğümlerin tüm çizgeye oranı Şekil 6.5'te yer almaktadır.



Şekil 6.5. Senaryo-III'e g re algoritmaların inceledikleri d ğ mlerin t m  izgeye oranı



Şekil 6.6. Senaryo-III'e g re algoritmaların en kısa rotayı hesaplama s releri

Dijkstra algoritması, diğ r algoritmalara g re (Manhattan y ntemini kullanan A-Yıldız algoritması hari ) daha iyi bir performans sergilemiřtir (Şekil 6.6). Bu algoritmanın t m  izge yerine, kaynak ve hedef noktaları i ine alan sınırlandırılmıř bir alanda  alıřtırılmasının performansını daha fazla arttıracadıđ deėerlendirilmektedir.

Her  ç senaryonun incelenmesi sonucu elde edilen deėerlendirmeler ařađıdadır.

- A-Yıldız algoritmasında,  evredeki engel sayısı ve hedef nokta uzaklıđının performansa etkisinin  ok olduėu her  ç senaryoda da g r lmektedir.
-  evredeki engel sayısının algoritma hesap s resi  zerindeki etkisi, Dijkstra algoritmasında daha azdır.
- A-Yıldız algoritması sezgisel metotlarından Manhattan y ntemi, rota hesaplama zamanı ve taranan d ğ m sayısı a ısından Euclidean ve Diagonal y ntemlerine

göre daha başarılıdır. Euclidean ve Diagonal yöntemlerindeki matematiksel işlemlerin Manhattan yönteminden daha fazla oluşunun bu sonuca önemli bir etkisi bulunmaktadır.

d. Hızlı sonuç elde edilmesi istenen problemlerde, kabul edilebilir en kısa rota belirleyen A-Yıldız algoritmasının seçilmesi ve en kısa rotanın garanti edilmesi istenen problemlerde Dijkstra algoritması kullanılmasının uygun olacağı değerlendirilmiştir.

Benzetim yazılımlarının en önemli özelliklerinden birisi insan doğasına uygun davranışlar sergileyebilmesidir. Böylece benzetim, gerçek sistemi daha fazla anımsatacaktır.

Harp oyunu benzetimleri için en kritik problemlerden olan en kısa rotanın bulunması sorunu bu tez çalışmasında detaylı olarak ele alınmıştır. Harp oyunu benzetimleri içinde en çok kullanılan en kısa yolu bulma algoritmaları arasında A-Yıldız ve Dijkstra algoritmaları incelenmiştir. Farklı alanlarda başarılı olan bu algoritmaların harp sahası gerçeklerinin dikkate alındığı benzetimlerde nelerden etkilendiği ve başarı durumları araştırılmıştır. Çalışmalar çok geniş bir alana sahip olan harp oyunu benzetimlerinden Deniz Taktik Harp Oyunu Benzetimi üzerinde ilerletilmiştir. Mevcut Deniz Taktik Harp Oyunu benzetimleri incelenmiştir.

Bu araştırma kapsamında deniz harp sahası unsurlarını ve coğrafi ortamının gerçeklerini yansıtan bir Deniz Taktik Harp Oyunu Benzetimi gerçekleştirilmiştir.

Geliştirilmiş olan Deniz Taktik Harp Oyunu Benzetimi içerisinde rota tayin etme işlemlerinde akıllı algoritmalar kullanılmıştır. Coğrafi bilgiler ve dost-düşman harp unsurların hesaba katılması ile uygun rotaların tayin edilmesi, bu rotalar üzerinde unsurların hareket etmesi ve değişen şartlara göre rotaların dinamik olarak güncellenmesi benzetim yazılımı içerisinde başarılı ile gerçekleştirilen uygulamalardır.

Geliştirilen Deniz Taktik Harp Oyunu Benzetimi ile çeşitli çevresel ve harp faktörleri kullanılarak deneysel testler yapılmış ve en kısa rotayı bulma algoritmalarının performansları gözlenmiştir. Elde edilen sonuçlara göre genel olarak az engelli ve kısa

mesafeli hedeflerde A-Yıldız algoritması ile yol bulmanın daha kısa sürede ve yüksek başarı ile olduğu ortaya konmuştur. Harp unsurlarının ve coğrafi engellerin sayısının arttığı şartlarda, Dijkstra Algoritmasının performansının daha ön plana çıktığı görülmüştür. Bu nedenle geliştirilecek uygulamalarda, her iki algoritmanın da tercihli olarak kullanıldığı melez yapıların bulunmasının, uygulama performansını arttıracakları değerlendirilmektedir.

Şüphesiz ki Deniz Taktik Harp Oyunu Benzetimi bu çalışma çerçevesinde yapılanlardan ibaret değildir. Ancak bu tez çalışması ile gerçekleştirilen benzetim yazılım altyapısı temelinde, deniz harp oyunları benzetimleri ile ilgili başka problemler için de deneysel bir ortam oluşturulmuştur. Analizi yapılmak istenen ileri konular için, yazılım ve içerisindeki harp unsurlarına farklı özellik ve yeteneklerin eklenmesi suretiyle, yeni çalışmaların kolaylıkla yapılabilmesi mümkündür.

Dünya üzerindeki ülkelerin savunma ihtiyacının olduğu sürece, her geçen gün harp teknolojileri gelişecek ve bu teknolojilerle üretilen silah ve araçların eğitimi için harcanacak meblağlar artacaktır. Bu nedenle, savunma giderlerinin en verimli şekilde kullanımına yardımcı olan ve askeri personel için eğitim ortamı sağlayan harp oyunu benzetimleri, daha da önem kazanacaktır. Gelecekte bu benzetimler, yeni teknoloji ve ihtiyaçlara göre gelişecek ve üzerinde birçok bilimsel araştırmanın yapılacağı verimli sahalara olmaya devam edecektir.

KAYNAKLAR

- [1] Euroasian Star-2007 Tatbikatı, Atatürk Harp Oyunu ve Kültür Merkezi
http://www.hrf.tu.nato.int/03_activities/01_20071129_eurasianstar07.htm, 26 Nisan 2009.
- [2] ÇAYIRCI, E., NATO's Joint Warfare Centre Perspective on CAX Support Tools and Requirements, Joint Warfare Centre, Stavanger, Norway.
- [3] MACEDONIA M., Games, Simulation, and Military Education Dilemma, U.S. Army Simulation, Training, and Instrumentation Command, USA.
- [4] Fleet Command Real-Time Tactics Naval Warfare Simulation
http://en.wikipedia.org/wiki/Fleet_Command, 26 Nisan 2009.
- [5] YEH, S., NEWMAN, B.; LIEWER, M., PAIR, J., TRESKUNOV, A., REGER, G., ROTHBAUM, B., DIFEDE, J.; SPITALNICK, J.; MCLAY, R.; PARSONS, T.; RIZZO, A., A Virtual Iraq System for the Treatment of Combat-Related Posttraumatic Stress Disorder, Virtual Reality Conference, 2009. VR 2009. IEEE Volume , 14-18 March 2009
- [6] Joint Theater Level Simulation – JTLS, http://www.simscrip.com/solutions/military/Joint_Theater_Level_Simulation.html
<http://www.rolands.com/Home/Projects/JTLS/jtlsinfo.htm>, 24 Nisan 2009.
- [7] JTLS Analyst's Guide, January 2008. U.S. Joint Forces Command Joint Warfighting Center, Suffolk.
- [8] MCFADDEN, S., M., JEON, J., LI A., MINNITI, A., Evaluation of Symbol Sets for Naval Tactical Displays, Defense Research and Development, Canada.
- [9] Department of Defense Interface Standard, Common Warfighting Symbology (MIL-STD-2525B) , w/Change 2 , xi,43. pgs, 7 March 2007.
- [10] NATO Military Symbols for Land Based Systems APP6A, 2.pgs, October 1998.
- [11] Icons and Symbols - MIL-STD-2525 Symbology, Sailor-in-the-Simulation (SITS) Workshop, 27, 36. pgs, Basic Commerce and Industries, Dahlgren, VA. 2005.

- [12] KARAŞ, İ. R., 3B CBS'de Ağ Analizlerine Yönelik Coğrafi Veritabanının Otomatik Olarak Üretilmesi, Gebze Yüksek Teknoloji Enstitüsü, Kocaeli.
- [13] DIJKSTRA, E., From My Life, Department of Computer Science, The University of Texas at Austin, USA, 29 November 1993.
- [14] SELÇUK, F., YURTAY, N., YUMUŞAK, N., Ayrık İşlemsel Yapılar, Sakarya Kitabevi, 75. syf, Sakarya 2005.
- [15] BRACKEEN, D., Developing Games in Java, Basics of the A* Algorithm, New Riders Publishing, August 2003.
- [16] LUGER George F, Artificial Intelligence, Structures and Strategies for Complex Problem Solving, Addison Wesley, England 2002.
- [17] A* Search Algorithm, http://en.wikipedia.org/wiki/A*_search_algorithm, 01 Eylül 2009.
- [18] A*'s Use of the Heuristic, <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>, 02 Eylül 2009.
- [19] KANDEMİR, R., İleri Programlama Yöntemlerine Giriş, Ders Notları, Sezgisel Arama Stratejileri, A* Algoritması.

ÖZGEÇMİŞ

Ahmet Turan DEĞERLİ, 1980 yılında Kemaliye’de doğdu. 1998 yılında Haydarpaşa Teknik Lisesi Bilgisayarlı Nümerik Kontrol (CNC) bölümünden mezun oldu. Aynı yıl girdiği İstanbul Üniversitesi Bilgisayar Bilimleri Mühendisliği bölümünden 2002 yılında mezun oldu. 2009 yılında Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği bölümünde yüksek lisansını tamamladı. 2005 yılından beri Deniz Kuvvetleri Komutanlığı’nda mühendis sınıfı subay olarak görev yapmaktadır.