



**T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ**

**Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü**

Lisans Bitirme Projesi I

**MS. PACMAN OYUNU İÇİN AKILLI
ALGORİTMALAR KULLANILARAK YAZILIM
KONTROLÖRÜ GELİŞTİRİLMESİ**

**Hasan GÜLBABA
1621221012**

**Enes Behlül YENİDÜNYA
1521221048**

Bitirme Projesi Danışmanı: Prof. Dr. Ayşe Şima UYAR

İstanbul, Haziran 2020



**T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ**

**Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü**

Lisans Bitirme Projesi I

**MS. PACMAN OYUNU İÇİN AKILLI
ALGORİTMALAR KULLANILARAK YAZILIM
KONTROLÖRÜ GELİŞTİRİLMESİ**

Hasan GÜLBABA
1621221012

Enes Behlül YENİDÜNYA
1521221048

Bitirme Projesi Danışmanı: Prof. Dr. Ayşe Şima UYAR

<u>Jüri Üyeleri:</u>	<u>İmza:</u>
Dr. Öğr. Üyesi	
Dr. Öğr. Üyesi	
Dr. Öğr. Üyesi	

İstanbul, Haziran 2020

İÇİNDEKİLER

İÇİNDEKİLER	iii
Şekil Listesi	v
Tablo Listesi	vi
1. GİRİŞ	2
1.1 Konunun Tanımı.....	2
1.2 Ms. Pac-man Oyunu.....	2
1.2.1 Ms. Pac-Man Yarışması	4
1.2.2 Kısmi Algılanabilirlik (Partial Observability).....	4
1.2.3 Ms. Pacman'ın Harita Görüş Çeşitleri	4
2. PROJE PLANI	6
2.1 Projenin Amacı.....	6
2.2 İş Paketleri.....	6
2.3 Gantt Diagramı.....	7
3. KURAMSAL BİLGİLER VE LİTERATÜR TARAMASI	8
3.1 Monte Carlo Tree Search (MCTS).....	8
3.2 Ms. PacMan Gelişim Süreci ve Kullanılmış Algoritmalar	9
3.3 Hayaletlerin Gelişim Süreci ve Kullanılmış Algoritmalar	11
4. ÇÖZÜMLEME VE TASARIM.....	12
4.1 Ms. Pac-Man Yarışması İçin Sağlanan Sınıflar.....	12
4.1.1 Executor Sınıfı	12
4.1.2 Game Sınıfı	12
4.1.3 MyPacman Sınıfı.....	12
4.1.4 Hayalet Sınıfları	12
4.2 Hayaletler İçin Oyun İçi Mesajlaşma.....	13
4.3 ÖNERİLEN YÖNTEM.....	14
4.3.1 Ms. Pac-man İçin Kural Tabanlı Yaklaşım.....	14
4.3.2 Hayaletler İçin Kural Tabanlı Yaklaşım	14
4.4 Ms. Pac-Man İçin Tasarlanan Kontrolörde Kullanılmış Stratejilerin Akış Çizelgeleri	14
4.4.1 Ms. Pac-Man Kontrolörünü İçin Tasarlanmış Genel Akış Çizelgesi ...	14
4.4.2 Hayaletlerden Kaçmak İçin Tasarlanan Akış Çizelgesi	15
4.5 Karşılaşılan Zorluklar.....	17
5. GERÇEKLEME VE TEST	18
5.1 Ms. Pac-man İçin Tasarlanan Kontrolör İçin Oluşturulan Stratejiler.....	18
5.1.1 Hayaletlerden Kaçış Stratejisi	18
5.1.2 Ms-Pac-Man'ın Yenilebilir Durumdaki Hayaletleri Kovalaması	18

5.1.3	Görünürde Pil Yoksa Daha Önce Gidilmemiş Bir Konuma Gitme Stratejisi	18
5.2	Hayaletler İçin Tasarlanan Kontrolör İçin Oluşturulan Stratejiler	18
5.2.1	Haritayı 4’e Bölme Stratejisi.....	18
5.2.2	Yörünge Stratejisi.....	19
5.2.3	Yapılması Planlanan Stratejiler	20
6.	KAYNAKÇA.....	21

Şekil Listesi

Şekil 1 Ms. Pac-Man Oyunu [1].....	3
Şekil 2 PO görüş limiti türleri	4
Şekil 3 Gantt Diagramı	7
Şekil 4 Monte Carlo Ağaç Araması Ağaç Yapısı	8
Şekil 5 Monte Carlo Arama Ağacı Algoritması	9
Şekil 6 Gelişmiş Sinir Ağları.....	10
Şekil 7 Arama Ağacı Algoritması.....	10
Şekil 8 Karınca kolonisi	10
Şekil 9 Ms. Pac-Man Kontrolörünü İçin Tasarlanmış Genel Akış Çizelgesi	15
Şekil 10 Hayaletlerden kaçış için tasarlanmış akış diyagramı	16
Şekil 11 Haritayı 4'e Bölme Stratejisinin Harita Üzerindeki Hayali Görüntüsü	19
Şekil 12 Yörünge Stratejisinin Harita Üzerindeki Hayali Görüntüsü	20

Tablo Listesi

Tablo 1 İş Paketleri	6
----------------------------	---

**MS. PACMAN OYUNU İÇİN AKILLI
ALGORİTMALAR KULLANILARAK YAZILIM
KONTROLÖRÜ GELİŞTİRİLMESİ**

1. GİRİŞ

1.1 Konunun Tanımı

Tarihin başlangıcından beri canlılar o anki amaçlarını gerçekleştirebilmek için çeşitli algoritmalar kullanmışlardır. Teknolojinin gelişmesiyle de birçok araç gereç kullanarak bu amaçlarını daha kolay gerçekleştirebilir hale gelmişlerdir. Artan karmaşık problemlerimizi çözdürdüğümüz bilgisayarlar hayatımızın önemli bir parçası haline gelmiştir. Biyolojik ve doğal zekanın modellenmesiyle büyük başarılar elde edilmiştir ve gelişmeler akıllı sistemler olarak adlandırılmıştır.

Akıllı algoritmaların farklı kullanım alanları vardır, bu alanlardan birisi de bilgisayar oyunlarıdır. Son yıllarda üretilen oyunların grafik, animasyon ve ses özelliklerindeki ilerlemelere rağmen birçok bilgisayar oyununun temel işleyiş mantığı birbirine benzemektedir. Ancak akıllı algoritmalar sayesinde bu tekdüzelikten kurtulabilir ve oyun içinde kullanıcıya ya da bilgisayara karşı kendi kendine kararlar alabilecek yazılım kontrolörü tasarlanabilir.

Akıllı algoritma tekniklerinin uygulandığı en ünlü oyunlardan bir tanesi de Ms. Pac-man'dir. Ms. Pac-man bu kontrolörler sayesinde kullanıcı müdahalesi olmadan pilleri yiyerek ve hayaletlerden kaçarak hayatta kalabilir ve yüksek puanlara ulaşabilir.

1.2 Ms. Pac-man Oyunu

Ms. Pac-man, General Computer Corporation tarafından geliştirilmiş olup 1982 yılında Midway tarafından yayınlanmış bir oyundur. 1980 yılında Namco tarafından ilk defa geliştirilen Pacman'ın değiştirilmiş halidir.



Şekil 1 Ms. Pac-Man Oyunu [1]

Oyuncu Ms. Pac-Man'i sonu olmayan bir labirent içinde yönlendirir. Labirent pillerle doludur ve 4 adet farklı renklerde hayalet içerir: Blinky, Pinky, Inky ve Sue. Toplamda 4 farklı labirent vardır. Her 4 seviye geçişinde ilk labirente dönülür. Ekranın sol tarafından sağ tarafına doğru iki adet geçit vardır. Her haritada toplam 4 adet güç pili bulunmaktadır. Oyunun amacı pilleri ve rengi maviye dönmüş hayaletleri yiyerek toplanılabildiği kadar puan toplamaktır.

Oyuncu (Ms. Pac-Man) pilleri yiyerek ve hayaletlerden kaçınarak puan kazanır ve herhangi bir hayaletle teması halinde can kaybeder. Bir güç pili yemesi halinde, önceden tanımlanmış süre bitene kadar hayaletler yenilebilir duruma geçer, renkleri maviye dönüşür ve yenilmeleri halinde fazladan puan kazanır. Tüm piller ve güç pilleri yendiği takdirde yeni bir seviyeye geçilmiş olur, seviyeler arttıkça hayaletleri yiyebilme süresi azalır ve hayaletleri yedikten sonra hayaletlerin labirent içine giriş yapmadan önce içinde bekledikleri kafeste geçirecekleri süre azalır. Her 10 bin puanda bir yeni bir can kazanır, oyun başlangıcındaysa 3 adet cana sahiptir. Bir hap yemek Ms. Pac-Man'e 10 puan kazanır ve hayalet yemek 200 puan kazanır, ancak hayalet yemeye devam edilmesi durumunda bu puan sürekli ikiye katlanarak kazanılır. 4 hayaletin de aynı anda yenilmesi durumunda toplamda 3000 puan kazanılmış olur.

1.2.1 Ms. Pac-Man Yarışması

Bu projemizde <http://www.pacmanvghosts.co.uk/> sitesinden sağlanan platformu kullanarak, Ms. Pac-Man ve hayaletler için kontrolörler geliştireceğiz.

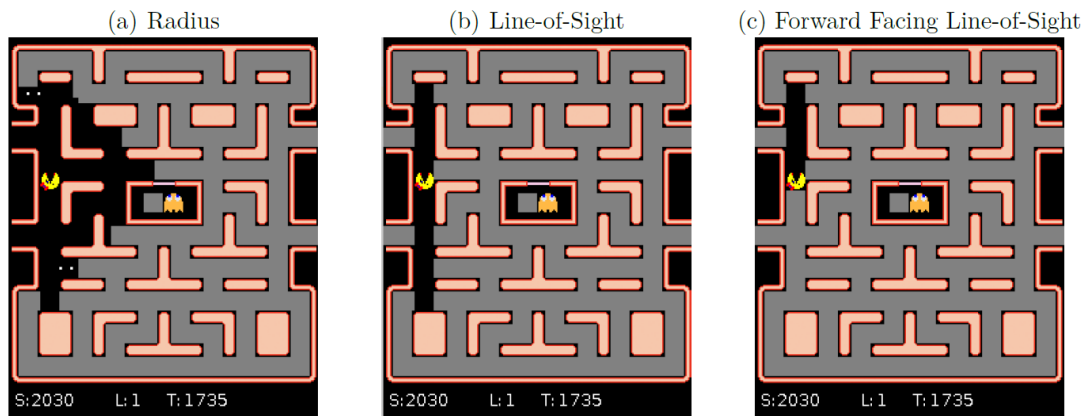
Yarışmada yalnızca Java dili kullanılmalı, performans artırımı amacıyla izlek(thread) kullanılmamalıdır. Ayrıca Ms. Pac-Man'ın ve 4 adet hayaletin bir sonraki hamlesi için 40ms süreleri vardır. Bu sürenin tamamını Ms. Pac-Man tek başına kullanabilecekken, hayaletler kendi aralarında paylaşırlar.

1.2.2 Kısmi Algılanabilirlik (Partial Observability)

Kullandığımız platformun bir özelliği de partially observable(PO) olmasıdır. PO, kontrolörün oyunu yalnızca Ms. Pac-Man'ın görüş alanı ile kısıtlı olarak yönetmesi durumudur. Bu durumda kontrolöre giden bilgiler yalnızca o anda oyuncunun görebildiklerinden ibarettir. Örneğin hayaletlerden herhangi birinin konum bilgisi, oyuncu eğer o hayaleti görmekteyse kontrolöre iletilir, ya da oyuncunun görüş alanı dışındaki bir pilin yenilip yenilmediği bilgisine erişilemez.

1.2.3 Ms. Pacman'ın Harita Görüş Çeşitleri

Ms. Pac-Man motoru, her biri görüş limiti uygulanan üç PO türünü desteklemektedir. Bunlar şekil 5'te görüldüğü üzere Radius, Line-of-Sight(LOS) ve Forward Facing Line-of-Sight(FF-LOS)'dır.[5]



Şekil 2 PO görüş limiti türleri

a) Yarıçap temelli PO: Görüş, Manhattan mesafesi veya Öklid mesafesi kullanılarak Ms. Pac-Man çevresindeki bir yarıçapla sınırlandırılabilir.[6]

b) Görüş Çizgisi(LOS): Alternatif olarak, vizyon görüş çizgisiyle sınırlandırılabilir, yani Ms. Pac-Man koridorlardaki düz çizgilerde sınırlı bir mesafe görebilir ve duvarlar görüşünü engeller. Köşelerden hiçbir şey görülemez.[6]

c) FF-LOS: Bu, MS. Pac-Man'in sadece seyahat ettiği yönde gözlemleyebileceği LOS üzerinde ek bir kısıtlamadır.

2. PROJE PLANI

2.1 Projenin Amacı

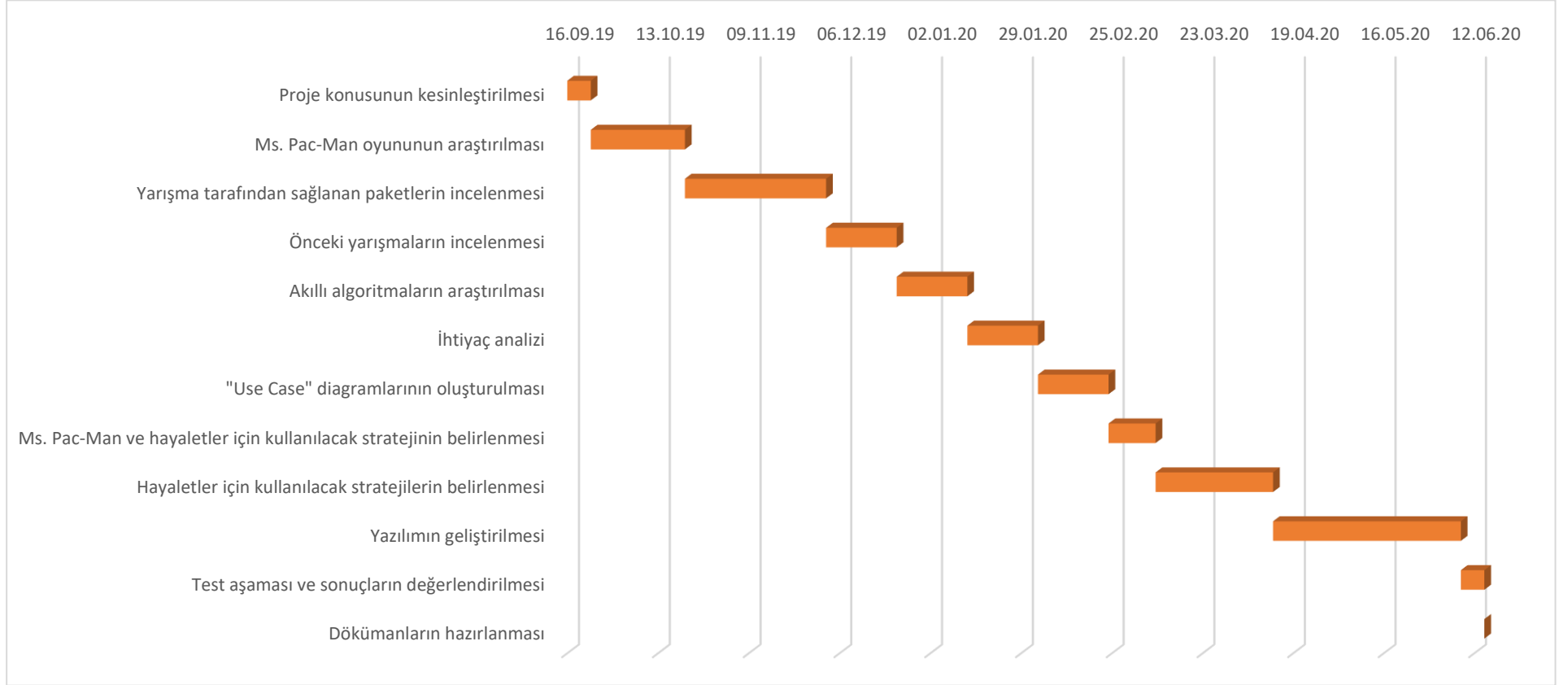
Yapılan yarışmalarda amaç, Ms. Pac-man için en yüksek puanı toplaması, hayaletler içinse Ms. Pac-Man'ın en az puanla oyunu bitirmesidir.

2.2 İş Paketleri

Görev	Başlangıç Tarihi	Süre (Gün)
Proje konusunun kesinleştirilmesi	16.09.19	7
Ms. Pac-Man oyununun araştırılması	23.09.19	21
Yarışma tarafından sağlanan paketlerin incelenmesi	14.10.19	42
Önceki yarışmaların incelenmesi	25.11.19	21
Akıllı algoritmaların araştırılması	16.12.19	21
İhtiyaç analizi	06.01.20	21
"Use Case" diagramlarının oluşturulması	27.01.20	21
Ms. Pac-Man ve hayaletler için kullanılacak stratejinin belirlenmesi	17.02.20	14
Hayaletler için kullanılacak stratejilerin belirlenmesi	02.03.20	35
Yazılımın geliştirilmesi	06.04.20	56
Test aşaması ve sonuçların değerlendirilmesi	01.06.20	14
Dökümanların hazırlanması	15.06.20	14

Tablo 1 İş Paketleri

2.3 Gantt Diagramı



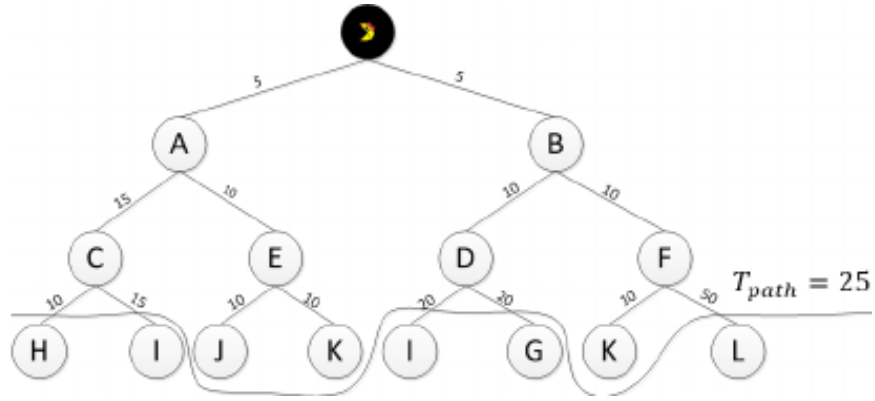
Şekil 3 Gantt Diagramı

3. KURAMSAL BİLGİLER VE LİTERATÜR TARAMASI

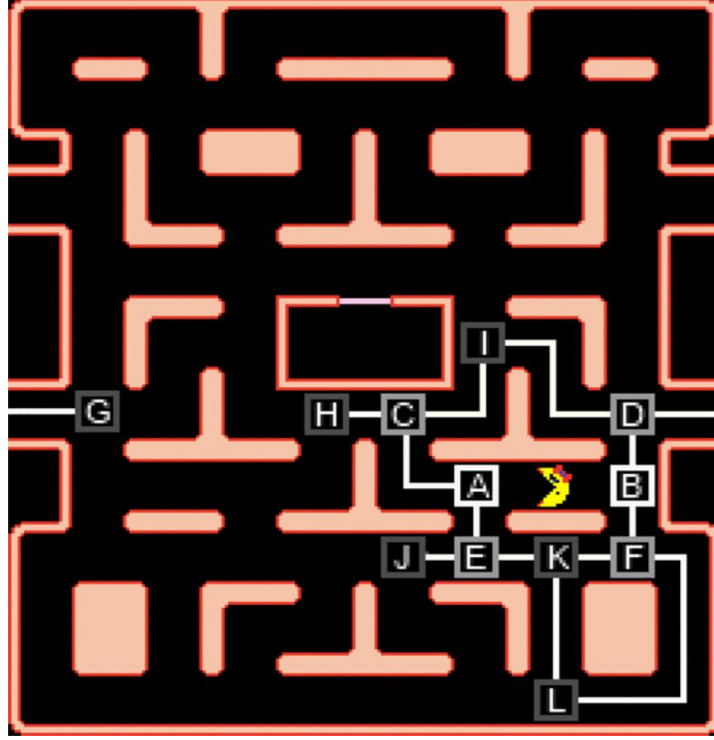
3.1 Monte Carlo Tree Search (MCTS)

Son yıllarda üretilen oyunların grafik, animasyon ve ses özelliklerindeki ilerlemelere rağmen birçok bilgisayar oyununun temel işleyiş mantığının birbirine benzemesi araştırmacıları MCTS gibi farklı oyunlara uygulanabilen algoritmalara yönlendirmektedir.

Monte Carlo Ağaç Araması, bir oyunun anlık durumuna bakarak, bir sonraki duruma götürecek en iyi hamleyi bulmak için belirli sayıda rastgele tahminlerde bulunur. [2] MCST'nin çalışma şekli, ilk önce bir botun halihazırda sahip olabileceği tüm olası hareketleri görselleştirmektir. Ardından, bu olası hamlelerin her biri için, bir oyuncunun yanıt verebileceği tüm hamleleri analiz eder, daha sonra da tepki olarak yapabileceği tüm olası tepki hamlelerini vb. dikkate alır. [3]



Şekil 4 Monte Carlo Ağaç Araması Ağaç Yapısı

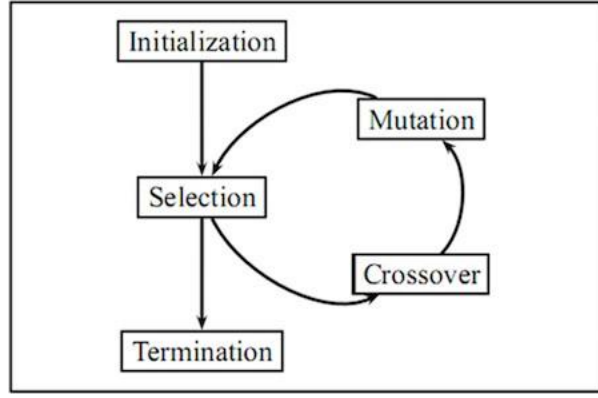


Şekil 5 Monte Carlo Arama Ağacı Algoritması

MCST kullanan bir bilgisayarın, bir insan bileşenine karşı harekete geçmeden önce geçtiği süreci vurgulamaktadır. Öncelikle sahip olduğu tüm seçeneklere bakıyor, yukarıdaki örnekte bu seçenekler ya savunmak, teknoloji geliştirmek ya da saldırı yapmak. Daha sonra, bundan sonraki her potansiyel hareket için başarı olasılığını öngören bir ağaç oluşturur. Yukarıda, başarı olasılığının en yüksek olduğu seçeneğin “saldırı” olduğunu görüyoruz (çünkü koyu kırmızı, daha yüksek bir ödül olasılığına eşittir), bu nedenle bilgisayar saldırmayı seçer. Oyuncu bir sonraki hamlesini yaptığında, bilgisayar tekrar ağaç yapım işlemini tekrarlar.

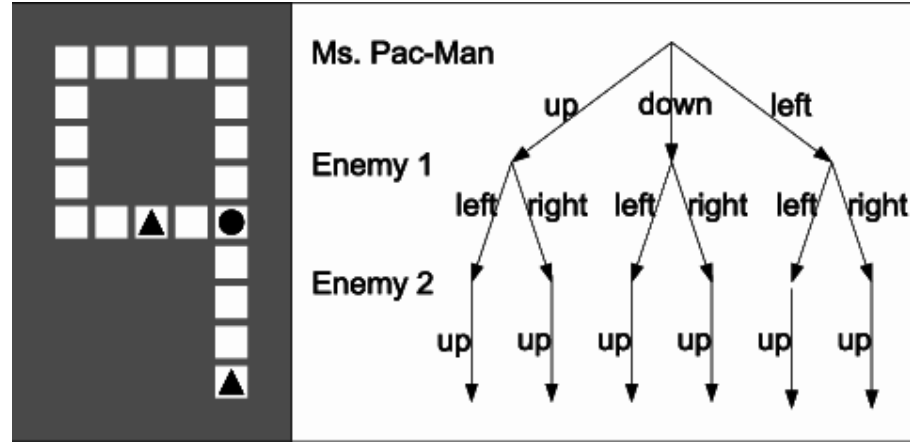
3.2 Ms. PacMan Gelişim Süreci ve Kullanılmış Algoritmalar

- Genetik Programlama (Alhejali ve Lucas 2010, 2011, Brandstetter ve Ahmadi 2012),
- Parametrelerin optimizasyonu için Genetik Algoritmalar (Thawonmas 2010),
- Gelişmiş Sinir Ağları (Burrow ve Lucas 2009, Keunhyun ve Sung-Bae 2010),



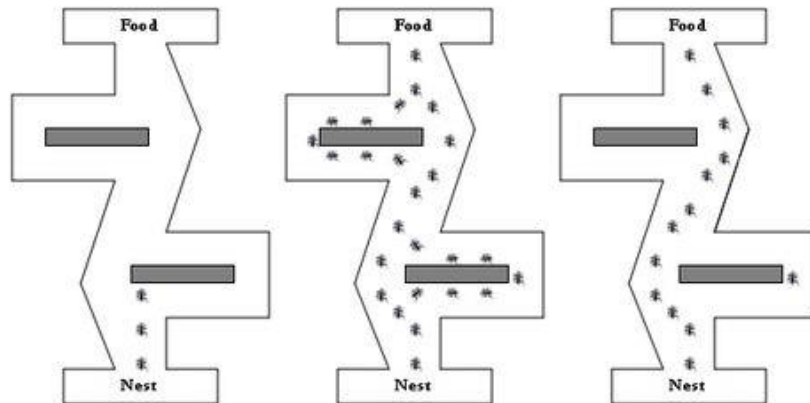
Şekil 6 Gelişmiş Sinir Ağları

- Robles ve Lucas, basit bir **Tree-Search algoritması** kullanarak oyun üzerinde çalışmışlardır.



Şekil 7 Arama Ağacı Algoritması

- **Ant Colonies (Martin ve ark.2010),**



Şekil 8 Karınca kolonisi

- Monte Carlo Ağacı Arama (Samothrakis ve ark.2011, Ikehata ve Ito 2011, Alhejali ve Lucas 2013),

3.3 Hayaletlerin Gelişim Süreci ve Kullanılmış Algoritmalar

- Monte Carlo Ağacı Arama (Nguyen and Thawonmas 2011, 2013),
- Etki Haritaları (Svensson ve Johansson 2012),
- Sinir Ağları (Jia-Yue ve ark.2011),
- Genetik Algoritma + Kuralları (Gagne ve Congdon 2012),
- Bayan Pac-Man ve Ghosts denetleyicilerinin Rekabetçi Birlikte Evrimi (Cardona ve ark. 2013).

4. ÇÖZÜMLEME VE TASARIM

Kontrolör oluşturmada önce yarışma tarafından sağlanan paketi analiz etmek, oyun için gerekli algoritmayı geliştirebilmek adına önem arz etmektedir.

4.1 Ms. Pac-Man Yarışması İçin Sağlanan Sınıflar

Ms. Pac-Man oyunu, Game, Executor, MyPacman (Ms. Pac-Man) ve MASController (Hayaletler) ana sınıflarının ortaklaşa çağrılmasıyla çalışır. Main sınıfı içinde Executor nesnesi oluşturulur, ve oluşturulan bu nesneye Ms. Pac-Man ve hayaletlerden oluşan bir kontrolör dizisi parametre olarak verilir.

4.1.1 Executor Sınıfı

Executor sınıfı aracılığıyla bir adet Game nesnesi, hayaletler ve pacman için kontrolörler oluşturulur. 40 ms aralıklarla sürekli olarak, oyun, pacman ve hayaletler güncellenir.

4.1.2 Game Sınıfı

Game sınıfı içinde, oyunun oynandığı Maze(labirent) nesnesi, hayaletler, piller, pillerin yenildiği bilgisi, skor, oyunun bitip bitmediği gibi bilgiler tutulur. Kontrolör yazabilmek gerekli olan tüm bilgiler bu sınıftadır. Sınıfa ait methodlar çağrılarak bu bilgilere erişilir.

4.1.3 MyPacman Sınıfı

MyPacman sınıfı, kontrolörü oluşturduğumuz sınıftır, Ms. Pac-Man'in hareket stratejisi belirlenir. Başlangıçta bu sınıf pacman'i rastgele hareket ettirecek basit kodlardan oluşur ve yarışmacıların bu sınıfta yaptığı değişikliklerle daha yüksek puan alması beklenir.

Yarışmacı, bu sınıf içindeki getMove metodunu düzenleyerek bir sonraki oyun adımı için Ms. Pac-Man'in hareket etmesi gereken yönü belirler. Bu yönler, yukarı, aşağı, sağ ve sol olacak şekilde 4 adettir. Oyun çalıştığı sürece bir döngü halinde sürekli olarak koşan getMove metodu, her bir oyun adımı için gereken yönü bu şekilde sağlar.

4.1.4 Hayalet Sınıfları

4 adet hayaletin her biri için kendi adları ile oluşturulan sınıflardır. Bu sınıflar hayaletler için kontrolör oluşturmada kullanılır. MyPacman sınıfında olduğu burada

da hayaletlerin hareket stratejileri belirlenir.

4.1.4.1 POGhost Sınıfı

POGhost sınıfı temel olarak hayaletlerin hangi durumlarda nasıl davranacağını belirlediği sınıftır. Hayaletlerin durumlara karşı verdiği tepkiler ve pac-man'ın durumuna karşı aldığı pozisyonlar için fonksiyonlar yer alır, "getApproximateNextMoveAwayFromTarget()" metodu ile pacman'den kaçma işlemini gerçekleştirir. Bu işlemi gerçekleştirirken pac-manın bulunduğu konuma bakar, pac-man powerPill'e yakınsa veya powerPill'i yediyse bu hayaletimiz için tehlike demektir ve kaçmaya başlaması komutu verilir.

Hayaletin en önemli işlevi pac-man'ı kovalama ve onu yemektir. Bu işlemler yine bu sınıftaki metodlar ile gerçekleşir. "getApproximateNextMoveTowardsTarget()" metodu ile pac-man yenilebilir olduğu sürece (herhangi bir powerPill almamış ise) hayaletler pac-man'ı kovalama görevini gerçekleştirir, eğer pac-man bir powerPill'e belli bir birimden daha fazla yakınsa hayaletler pac-man'ı kovalamayı bırakır ve random bir yol seçerler.

4.1.4.2 POCCommGhost Sınıfı

POCommGhost sınıfının temel amacı ise hayaletlerin pac-man'ı kovalarken birbirlerine haber verip pac-man'ın konumu hakkında diğerlerini de haberdar etmektir. Hayaletlerin birbiri ile gerçekleştirdiği haberleşme işlemi yapay zekanın temellerinden olan sürü teorisi algoritmalarına (swarm-based algorithm) ve karınca teorisi algoritmalarına (ant colony algorithm) benzer olarak örnek gösterilebilir.

4.2 Hayaletler İçin Oyun İçi Mesajlaşma

Oyunun PO olmasından dolayı hayaletler birbirleri arasında iletişime ihtiyaç duymaktadır. Ms. Pac-Man'ın görüldüğü son konum, güç piline olan mesafesi gibi önemli bilgiler hayaletler arası iletilir.

Mesaj tipi	Tanım
Pacman seen	Pacman in görüldüğünü diğer hayaletlere iletme mesajı
I am	Mesajı gönderenin nerede olduğunu bildiren mesaj

I am heading	Mesajı gönderenin nereye gittiğini bildir- ren mesaj
--------------	---------------------------------------------------------

4.3 ÖNERİLEN YÖNTEM

Daha önce Ms Pacman için yapılan çözümlere bakıldığında, Kural Tabanlı Sistemler, Monte Carlo Search Tree, Evrimsel Algoritmalar, Yapay Sinir Ağları ve Takviyeli Öğrenme gibi yöntemlerin kullanıldığını görüyoruz. [1]

4.3.1 Ms. Pac-man İçin Kural Tabanlı Yaklaşım

Kural tabanlı yaklaşım dışında kullanılabilecek akıllı algoritmalar göz önünde bulundurulduğunda, yoğun performans gereksinimleri, karmaşık veri yapılarının kullanılması ve algoritmayı gerçekleştirmesinin kural tabanlı yaklaşıma göre daha zor olması kural tabanlı yöntemi seçmemizde önemli etkenlerdendir.

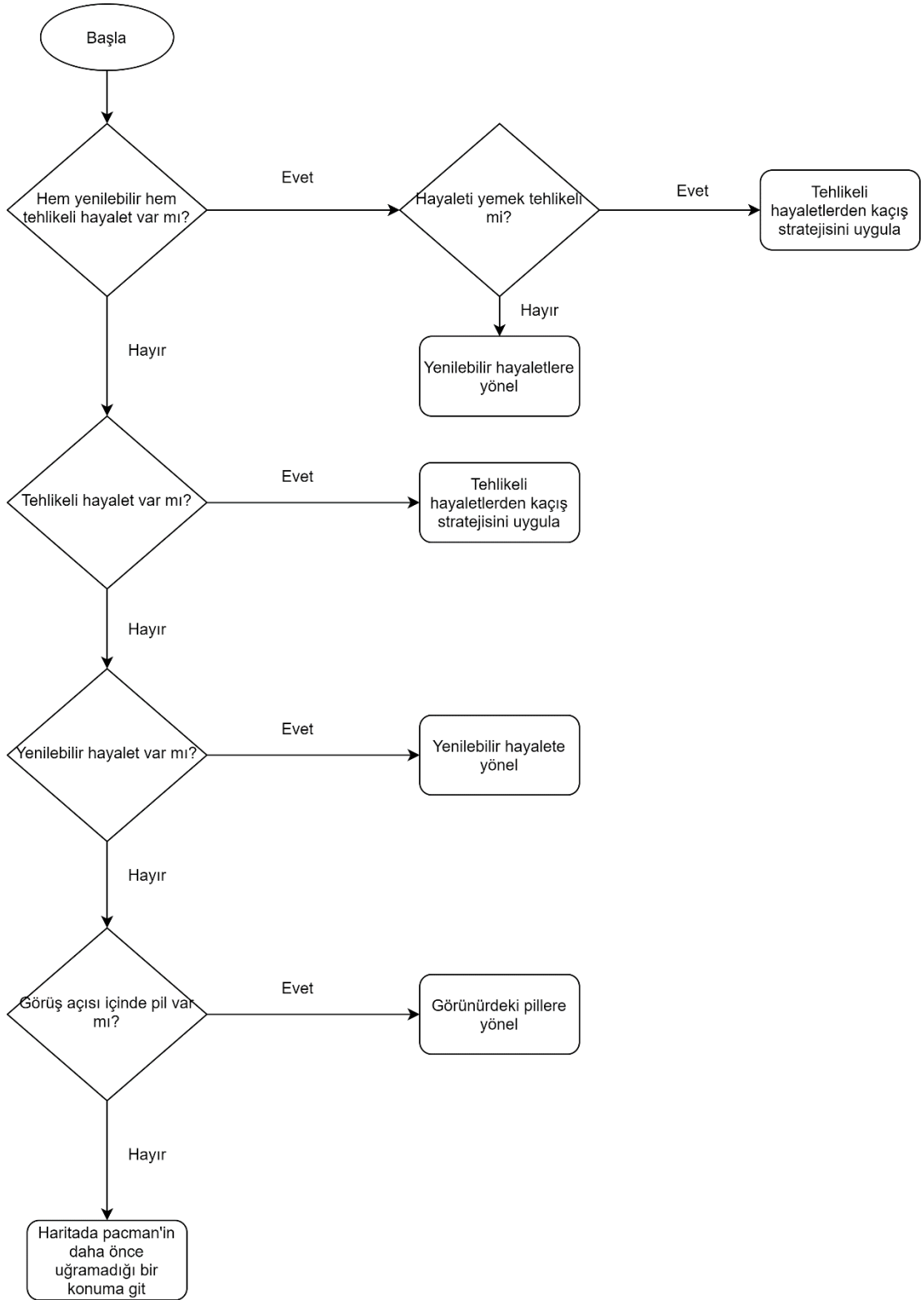
4.3.2 Hayaletler İçin Kural Tabanlı Yaklaşım

Ms. Pac-Man için geçerli olan durum hayaletler için de geçerlidir

4.4 Ms. Pac-Man İçin Tasarlanan Kontrolörde Kullanılmış Stratejilerin Akış Çizelgeleri

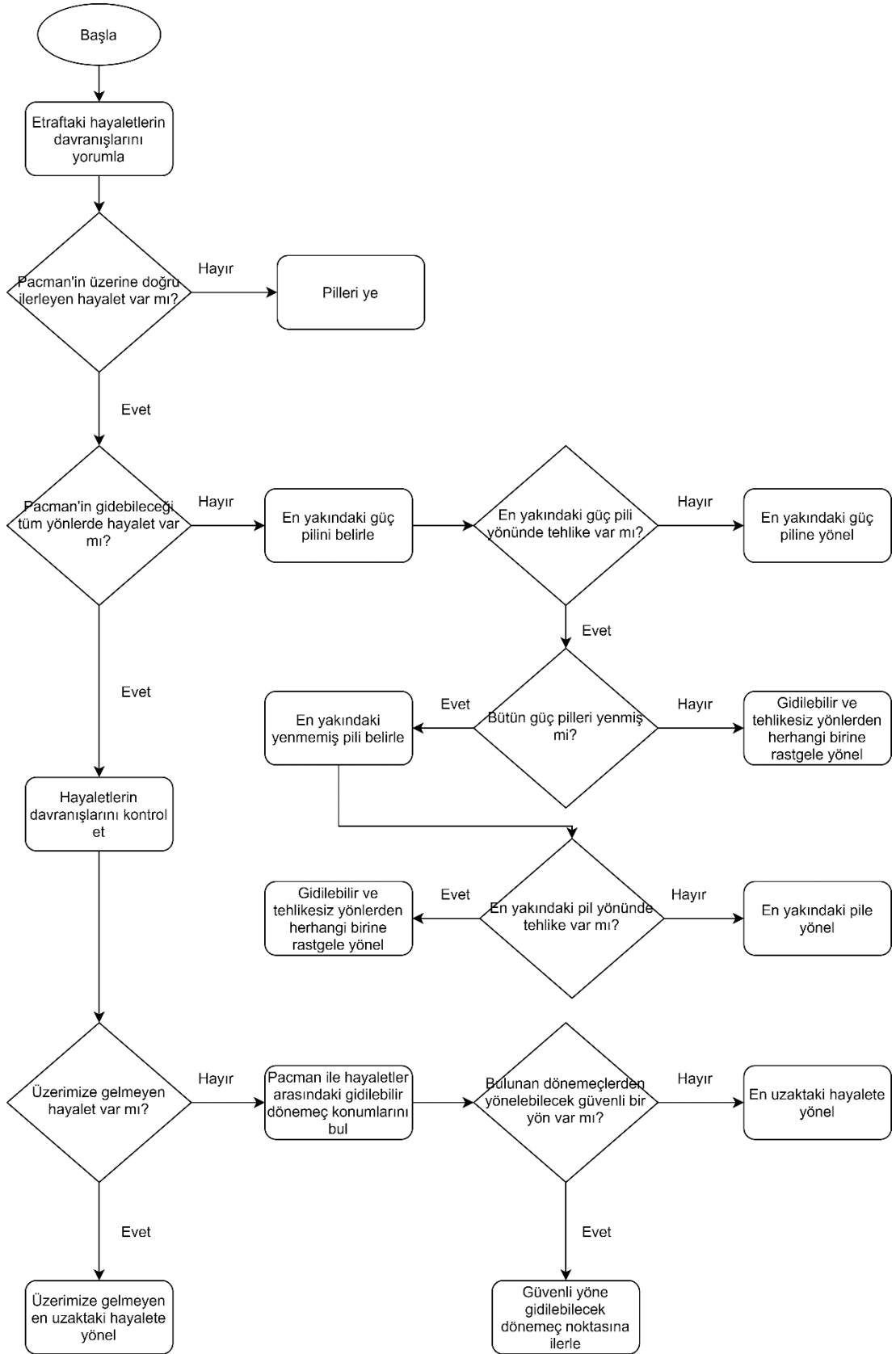
4.4.1 Ms. Pac-Man Kontrolörünü İçin Tasarlanmış Genel Akış Çizelgesi

Ms. Pacman için tasarlanan kontrolörün genel akışı şekil BURAYA BİR NUMARA VER UNUTMA te görüldüğü gibidir. Ms. Pac-Man, görüş açısı içindeki hayaletleri ve pilleri göze alarak döneceği yöne karar verecektir.



Şekil 9 Ms. Pac-Man Kontrolörünü İçin Tasarlanmış Genel Akış Çizelgesi

4.4.2 Hayaletlerden Kaçmak İçin Tasarlanan Akış Çizelgesi



Şekil 10 Hayaletlerden kaçış için tasarlanmış akış diyagramı

4.5 Karşılaşılan Zorluklar

Projemizi yaparken karşılaştığımız en büyük zorluk oyun içinde bize sunulan ortamın Partial Observability (kısmi gözlemlenebilir) olması. Bu ortamda zorlanmamızın sebebi ise oyun içindeki tüm noktaları (piller, güç pillerinin konumları gibi) kısmi gözlemlenebilirlik özelliği nedeniyle göremediğimizden dolayı Ms.pac-man için haritadaki tüm pillerin konumlarını öğrenmesi için ayrı bir yöntem geliştirmemiz gerekmektedir.

Bu yöntem için literatüre baktığımızda Pac-Man Arcade oyunu için kullanılmış ve iyi sonuç veren PO algoritmalarından radius kullanılması önerilmektedir.

5. GERÇEKLEME VE TEST

5.1 Ms. Pac-man İçin Tasarlanan Kontrolör İçin Oluşturulan Stratejiler

5.1.1 Hayaletlerden Kaçış Stratejisi

İlk olarak hayaletlerin Ms-Pac-Man'e göre olan konumları bulunuyor. Örneğin Ms-Pac-Man bir hayalet gördüğünde, daha önceden tanımlamış olduğumuz tehlikeli yönler adlı dizi içersine hayaletin bulunduğu yönü Ms-Pac-Man orijininde kaydediyoruz. Dizi 4 elemandan oluşmaktadır ve her biri sağ, sol, yukarı ve aşağı yönlerini kontrol etmek içindir.

Görünürdeki tüm hayaletler kontrol edildikten sonra, tehlike olmayan yöne doğru Ms-Pac-Man yönlendirilir.

5.1.2 Ms-Pac-Man'in Yenilebilir Durumdaki Hayaletleri Kovalaması

Öncelikle görünürdeki hayaletlerin yenilebilir durumda oldukları kontrol edilir ve ardından en yakındakini yemek için gereken kovalama kodu işleme sokulur.

5.1.3 Görünürde Pil Yoksa Daha Önce Gidilmemiş Bir Konuma Gitme Stratejisi

Görüş açısı içinde pil yoksa daha önce ziyaret edilmiş tüm konumların saklandığı bir hashmap veri yapısı kullanılarak ziyaret edilmemiş konumlar bulunur ve Ms-Pac-Man bunlar arasından en yakın olana yönlendirilir. Bu sayede görünürde hayalet ve pil olmadığı durumlarda Ms-Pac-Man kararsız kalmaz ve yenmemiş pillere yönelmiş olur.

5.2 Hayaletler İçin Tasarlanan Kontrolör İçin Oluşturulan Stratejiler

5.2.1 Haritayı 4'e Bölme Stratejisi



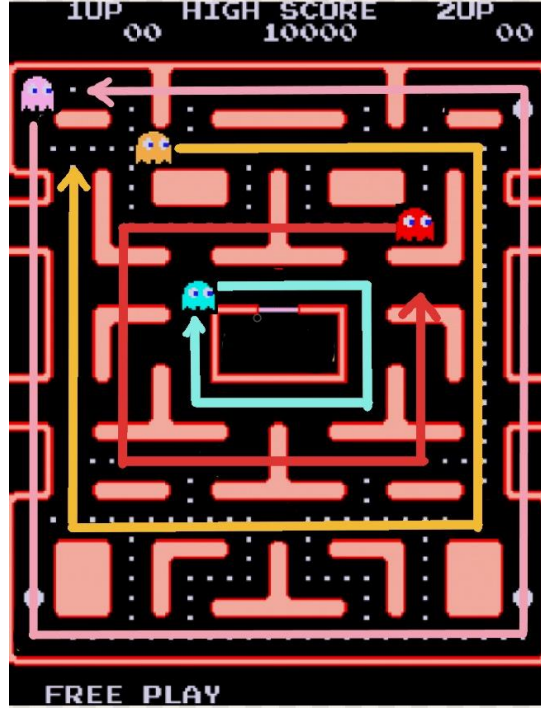
Şekil 11 Haritayı 4'e Bölme Stratejisinin Harita Üzerindeki Hayali Görüntüsü

Oyun haritasını hayali bir şekilde 4'e bölerek her hayalet oyunun farklı bir köşesine gönderilir ve her hayalet oyun haritasının her bir köşesinde dolanarak Ms.Pac-man'i aramaya başlar ve böylece Ms.Pac-man'in hayaletler tarafından görülme olasılığı artırılmış olur.

Oyunun her bir köşesine dağılmış olan hayaletlerden herhangi biri Ms. Pac-man'i görürse diğer hayaletlere haber vermektedir ve hayaletler oyunun dört bir köşesinden Ms. Pac-man'inin etrafını sarmaya ve sıkıştırmaya çalışmaktadır.

Hayaletler için düşünülen ek bir strateji ise her hayalete özel görevler yükleme işlemidir.

5.2.2 Yörünge Stratejisi



Şekil 12 Yörünge Stratejisinin Harita Üzerindeki Hayali Görüntüsü

Hayaletler için düşünülen ek bir strateji ise her hayalete özel görevler yükleme işlemidir. Hayaletler oyun başladıktan sonra kendi yuvalarından çıkarken her bir hayalet oyun sahasını daireler çizerek taramaya başlar ve her biri bir öncekinin zıttı yönünde hareket eder. Bu sayede oyun alanının büyük bir kısmının eş zamanlı olarak taranması sağlanır.

5.2.3 Yapılması Planlanan Stratejiler

- Hayaletlerden biri Ms. Pac-man i görüp yakalamak için birbirleriyle haberleştikten sonra eğer yakalama işlemi gerçekleşmezse hayaletlerin geri köşelere dönme işlemi.
- Hayaletler haberleştikten sonra Ms. Pac-man i yakalamak için harekete geçtikten sonra eğer Ms. Pac-man Power Pill alırsa yapılacak kaçma stratejisi.

6. KAYNAKÇA

- [1] Rohlfshagen Philipp, Liu Jialin, Perez-Liebana Diego, Lucas Simon M., “Pac-Man Conquers Academia: Two Decades of Research Using a Classic Arcade Game” (2017) s6.
- [2]<https://devhunteryz.wordpress.com/2018/03/30/monte-carlo-agac-aramasimonte-carlo-tree-search/>
- [3]<https://towardsdatascience.com/artificial-intelligence-in-video-games-3e2566d59c22>
- [4] S. M. Lucas, “Evolving a Neural Network Location Evaluator to Play Ms. Pac-Man,” in Computational Intelligence and Games. IEEE, 2005.
- [5] Price, W., & Schrum, J. (2019). Neuroevolution of Multimodal Ms. Pac-Man Controllers under Partially Observable Conditions. *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, 466–473. <https://doi.org/10.1109/CEC.2019.8790278>
- [6] Price, W., & Schrum, J. (2019). Neuroevolution of Multimodal Ms. Pac-Man Controllers under Partially Observable Conditions. *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, 466–473. <https://doi.org/10.1109/CEC.2019.8790278>