



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

**Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü**

**Lisans Bitirme Projesi I**

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

**Hasan GÜLBABA  
1621221012**

**Enes Behlül YENİDÜNYA  
1521221048**

**Bitirme Projesi Danışmanı: Prof. Dr. Ayşe Şima UYAR**

**İstanbul, Ocak 2020**



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

**Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü**

**Lisans Bitirme Projesi I**

**MS. PACMAN OYUNU İÇİN AKILI  
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, Ocak 2020

# 1. İÇİNDEKİLER

<b>1</b>	<b>İÇİNDEKİLER .....</b>	<b>ii</b>
<b>2</b>	<b>GİRİŞ .....</b>	<b>2</b>
2.1	Konunun Tanımı .....	2
<b>3</b>	<b>GENEL BİLGİLER VE LİTERATÜR TARAMASI.....</b>	<b>3</b>
3.1	Ms. Pacman .....	3
3.2	Ms. PacMan Gelişim Süreci ve Kullanılmış Algoritmalar .....	4
3.3	Ms. PacMan Ghost Gelişim Süreci ve Kullanılmış Algoritmalar .....	5
3.4	Ms.PacMan Yarışması .....	6
3.5	Ms. Pac-Man Yarışması İçin Sağlanan Sınıflar .....	6
<b>4</b>	<b>ÖNERİLEN YÖNTEM.....</b>	<b>8</b>
4.1	Oyun İçi Mesajlaşma .....	8
4.2	Monte Carlo Search Tree(MCST) .....	9
<b>5</b>	<b>PROJE PLANI .....</b>	<b>11</b>
5.1	Projenin Amacı .....	11
5.2	İş Paketleri ve Gantt Diagramı .....	11
5.3	Birinci Dönem Yapılanlar ve Sonuçlar .....	13
5.4	Karşılaşılan Zorluklar .....	14
<b>6</b>	<b>KAYNAKÇA.....</b>	<b>15</b>

## Şekil Listesi

Şekil 1	Ms. Pac-Man Oyunu [1] .....	3
Şekil 2	Gelişmiş Sinir Ağları .....	4
Şekil 3	Arama Ağacı Algoritması .....	5
Şekil 4	Karınca kolonisi .....	5
Şekil 5	1 PO görüş limiti türleri .....	8
Şekil 6	Monte Carlo Arama Ağacı Algoritması .....	9
Şekil 7	Monte Carlo Arama Ağacı Algoritması .....	10

## **Tablo Listesi**

Tablo 2.2 İş Paketleri .....	10-11
------------------------------	-------

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

## **2. GİRİŞ**

### **2.1 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.

### 3. GENEL BİLGİLER VE LİTERATÜR TARAMASI

#### 3.1 Ms. Pacman

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. Ekranın sol tarafından sağ tarafına doğru bir geçit vardır. Köşelerde birer tane olmak üzere toplam 4 adet güç pili bulunmaktadır. Oyunun amacı pilleri ve rengi maviye dönmüş hayaletleri yiyerek toplanılabildiği kadar puan toplamaktır.

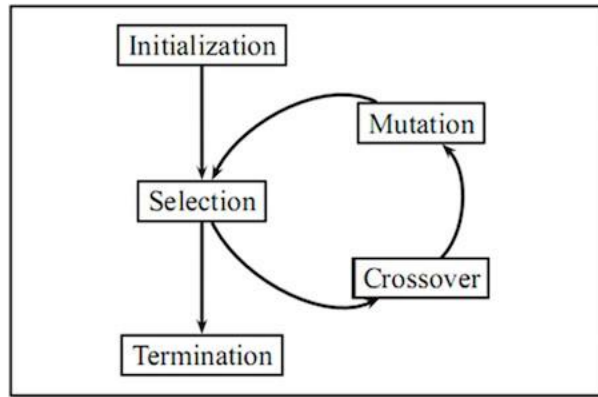
Oyuncu (Ms. Pac-Man) pilleri yiyerek ve hayaletlerden kaçınarak puan kazanır ve herhangi bir hayaletle teması halinde can kaybeder. Bir güç pili yemesi halinde, önceden tanımlanmış süre bitene kadar hayaletler yenilebilir duruma geçer, renkleri maviye dönüşür ve yenilmeleri halinde fazladan puan kazanır. Tüm piller yendiği takdirde yeni bir seviyeye geçilmiş olur, seviyeler arttıkça hız artar ve hayaletleri



yiyebilme süresi 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.

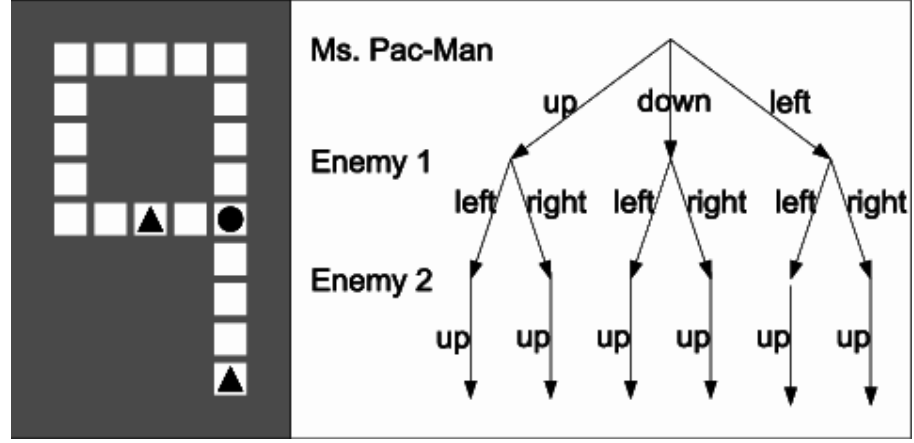
### 3.2 Ms. PacMan Gelişim Süreci ve Kullanılmış Algoritmalar

- 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),



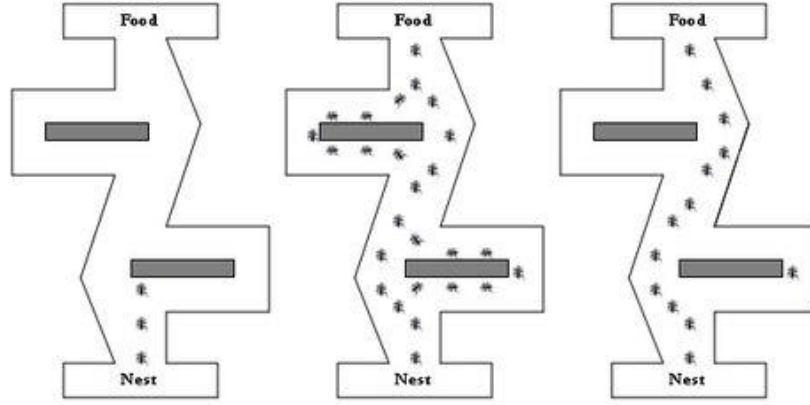
Şekil 2 Gelişmiş Sinir Ağları

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



Şekil 3 Arama Ağacı Algoritması

- Ant Colonies (Martin ve ark.2010),



Şekil 4 Karınca kolonisi

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

### 3.3 Ms. PacMan Ghost 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).

### **3.4 Ms.PacMan 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 arttırım amacıyla 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.

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, oyuncunun görüş alanı dışındaki bir pilin yenilip yenilmediği bilgisine erişilemez.

### **3.5 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 ihtiyaç duyulan nesneler ve metodlar bu sınıf vasıtasıyla çağrılır.

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

Game sınıfı içinde, oyunun oynandığı Maze(labirent) nesnesi, piller, pillerin yenildiği bilgisi, skor, oyunun bitip bitmediği gibi bilgiler tutulur.

MyPacman sınıfı içinde, Ms. Pac-Man'ın hareket stratejisinin belirlenir. Başlangıçta bu sınıf pacman'i rastgele hareket ettirecek basit kodlardan oluşur ve yarışmacıların bu sınıfta yaptığı değişikliklerle daha yüksek puan alması beklenir.

POGhost sınıfı temel olarak hayaletlerin hangi durumlarda nasıl davranacağını

belirlendiği 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.

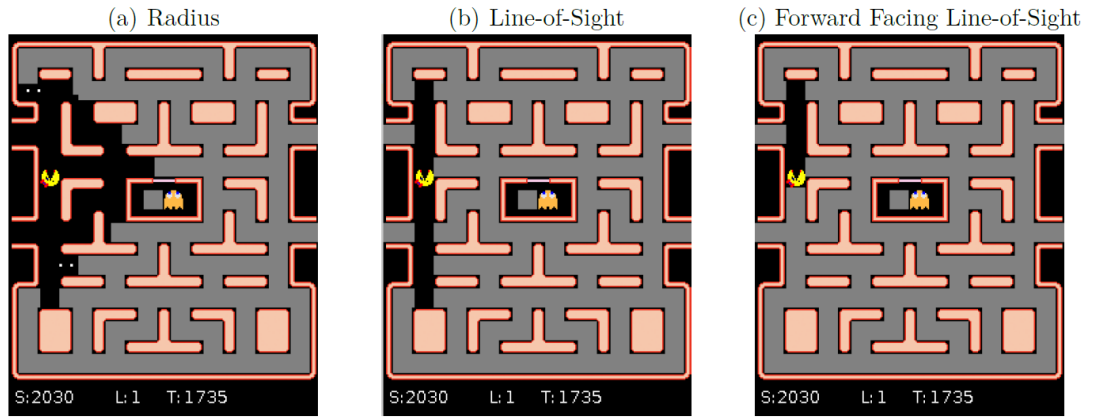
Hayaletin en önemli işlevi pac-man'ı kovalama ve onu yemektir. Bu işlemler yine bu sınıftaki metodlar ile gerçekleşir. "getApproximateNextMoveTowardsTarget()" metodu ile pac-man yenilebilir olduğu sürece (herhangi bir powerPill almamış ise) hayaletler pac-man'ı kovalama görevini gerçekleştirir, eğer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

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. 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]

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 5 1 PO görüş limiti türleri

**a) Yarıçap temelli PO:** Görüş, Manhattan mesafesi veya Öklid mesafesi kullanılarak Bayan 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.

### 4.1 Oyun İçi Mesajlaşma

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

önemli bilgiler hayaletler arası iletilir.

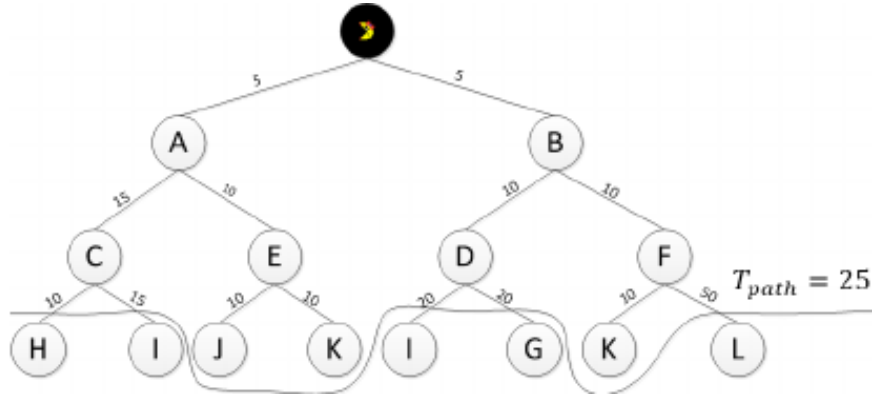
Mesaj tipi	Tanım
Pacman seen	Pacman in görüldüğünü diğer hayaletlere iletme mesajı
I am	Mesajı gönderenin nerede olduğunu bildiren mesaj
I am heading	Mesajı gönderenin nereye gittiğini bildiren mesaj

#### 4.2 Monte Carlo Search Tree(MCST)

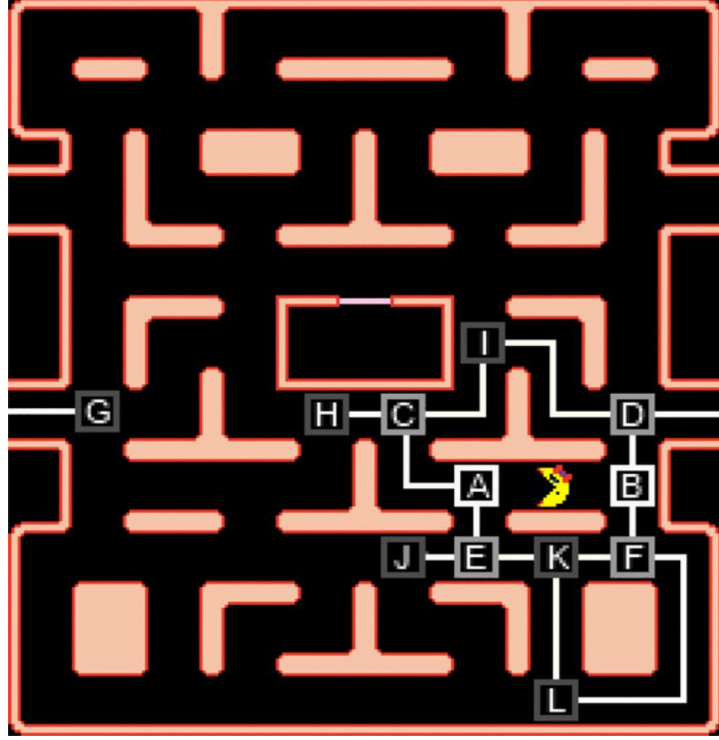
Son yıllarda üretilen oyunların grafik, animasyon ve ses özelliklerindeki ilerlemelere rağmen birçok bilgisayar oyununun temel işleyiş mantığının birbirine benzemesi araştırmacıları MCST gibi farklı oyunlara uygulanabilen algoritmalara yönlendirmektedir.

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 6 Monte Carlo Arama Ağacı Algoritması



Şekil 7 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.

## 5. PROJE PLANI

### 5.1 Projenin Amacı

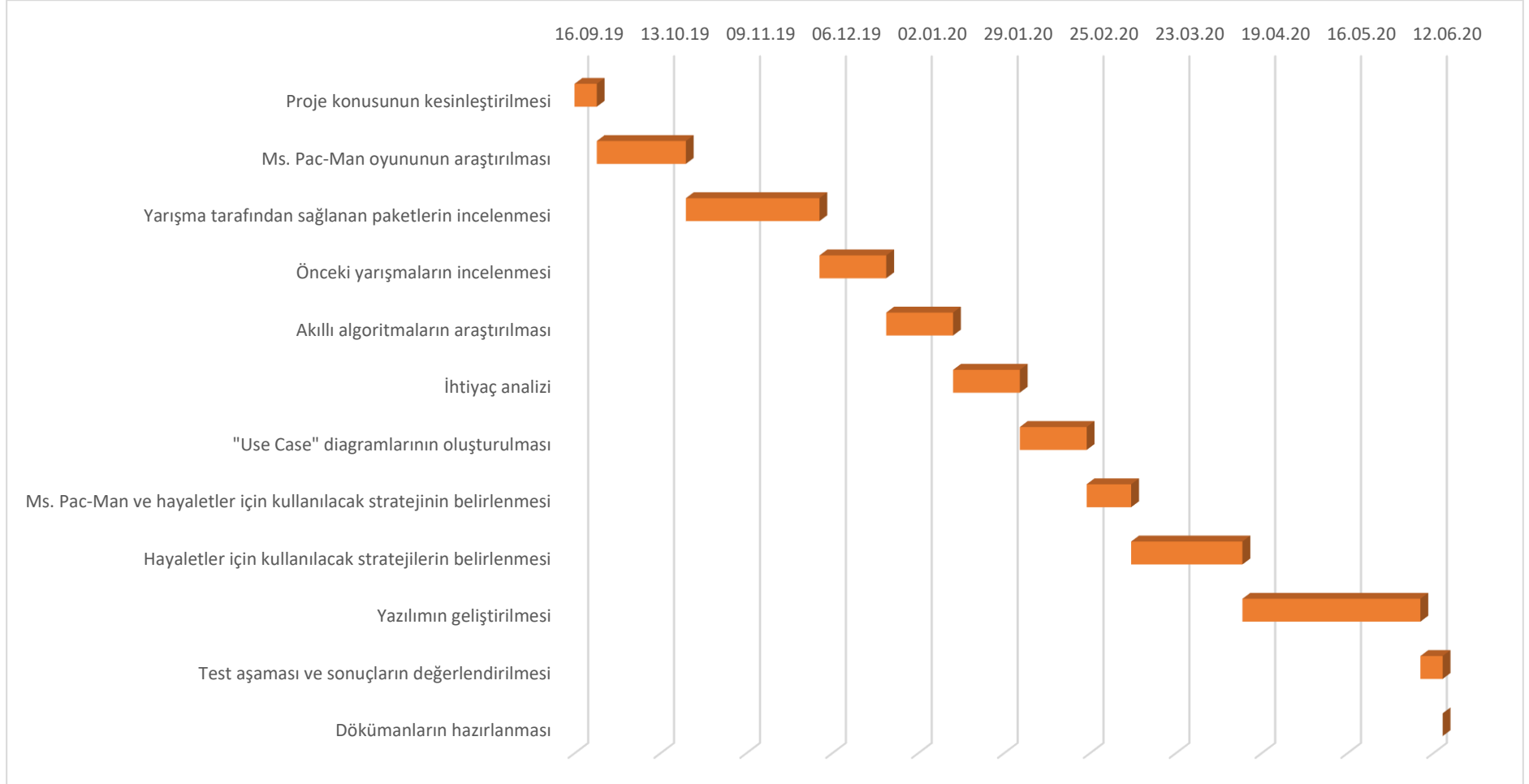
Yapılan yarışmalarda amaç, Ms. Pac-man için en yüksek puanı toplaması, hayaletler içinse Ms. Pac-Man'ın en az puanla oyunu bitirmesidir.

### 5.2 İş Paketleri ve Gantt Diagramı

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

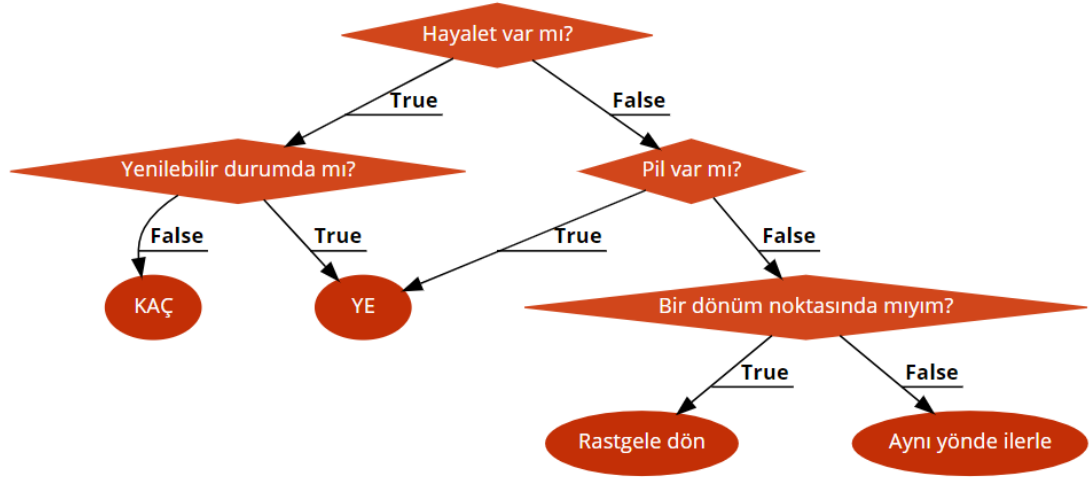
Tablo 2.2 İş Paketleri[1]





Şekil 2.2 Gantt Diagramı[2]

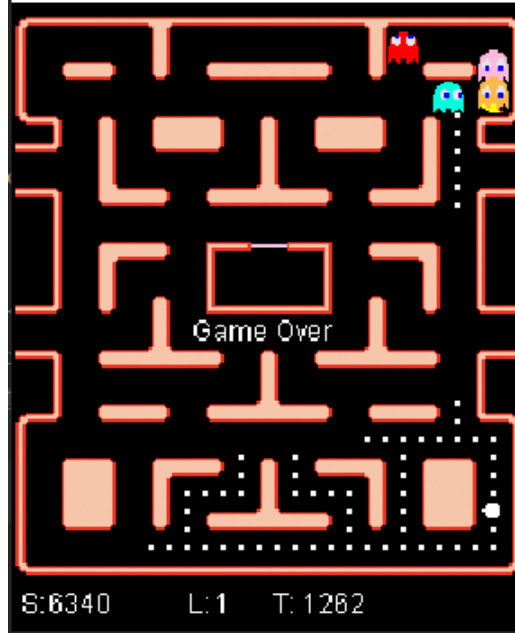
### 5.3 Birinci Dönem Yapılanlar ve Sonuçlar



Şekil 4.2.1 Ms. Pac-Man algoritması akış diagramı

```
1  Function getMove()
2      pacmanKonumu = getPacmanLocation()
3      if getPacmanLocation != NULL then
4          sonPacmanKonumu = pacmanKonumu
5          mesajGonder(PACMAN_GORULDU)
6
7      if !PACMAN_GORULDU then
8          for mesaj in mesajlar
9              if mesaj == PACMAN_GORULDU then
10                 sonPacmanKonumu = mesaj.pacmanKonumu
11
12      if sonPacmanKonumu != NULL then
13          return nexMoveTowardsTarget(sonPacmanKonumu)
```

Tüm hayaletler için sözde kod



Ulaştığımız en yüksek skor

#### 5.4 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.

## 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>