

GAZİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ



**BM455 YAPAY ZEKAYA GİRİŞ**

Uygulama Ödevi Raporu

191180005

Selin Cansu Akbaş

13.12.2023

## **Bölüm 1**

Gerçekleştirilen Platform: Pacman projesi, JetBrains tarafından geliştirilen PyCharm Integrated Development Environment (IDE) üzerinde gerçekleştirilmiştir. PyCharm, Python dilini destekleyen güçlü bir IDE'dir ve projenin geliştirilmesi ve yönetilmesi için sağlam bir temel sunar.

Programlama Dili: Pacman oyunu, Python programlama dilinde yazılmıştır. Python, okunabilirliği, basit yapısı ve geniş kütüphane desteği ile bilinen bir dil olup oyun geliştirmek için de uygun bir seçenektir.

## **Bölüm 2**

Kısa Tanım: Bu projede, klasik bir oyun olan Pacman'in Python dilinde bir versiyonu geliştirilmiştir. Oyuncu, labirent içinde dolaşan Pacman karakterini kontrol ederek, renkli noktaları yiyerek puan kazanmaya çalışır. Aynı zamanda, hayaletlerden kaçarak ve özel noktaları yiyerek geçici süreliğine hayaletleri avlayarak oyunu tamamlamaya çalışır. Proje, PyCharm IDE üzerinde geliştirilmiştir. Ayrıca, test süreçleri ile oyunun istikrarlı ve kullanıcı dostu bir deneyim sunması hedeflenmiştir.

## **Bölüm 3**

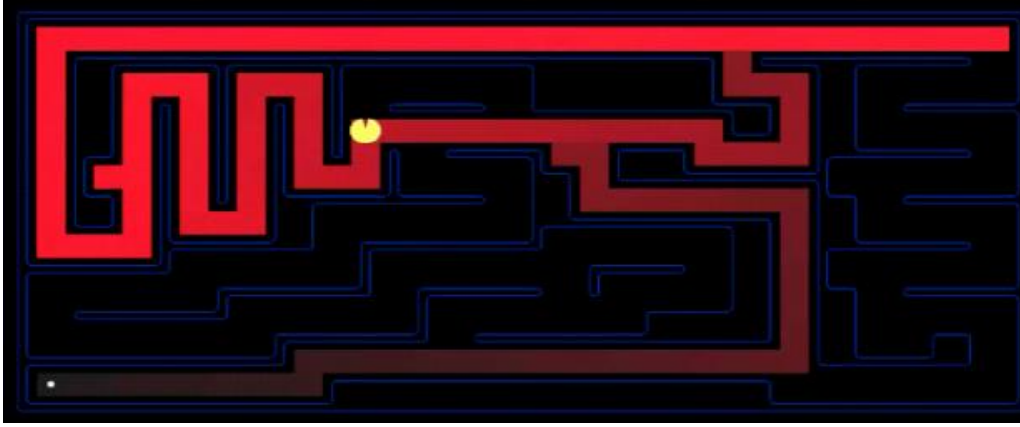
Yığın (Stack): Hayaletlerin ve Pacman'in hareketini takip etmek için kullanılan yığın veri yapısı. Hareket geçmişi bu veri yapısı içinde tutulur. Pacman'in yaptığı hareketler (örneğin, sağa, sola, yukarı veya aşağı gitme) bu yığına eklenir ve geri alınabilir.

Kuyruk (Queue): Hareket etmeyen özel noktaların süresini takip etmek için kullanılan kuyruk veri yapısı. Hareket eden nesnelerin sırasını korumak amacıyla kuyruk kullanılmıştır.

Sınıflar (Classes): Oyun içindeki farklı nesnelerin ve özelliklerinin temsil edilmesi için sınıflar kullanılmıştır. Örneğin, Pacman, hayaletler, noktalar ve labirent gibi unsurlar ayrı sınıflarda modellenmiştir.

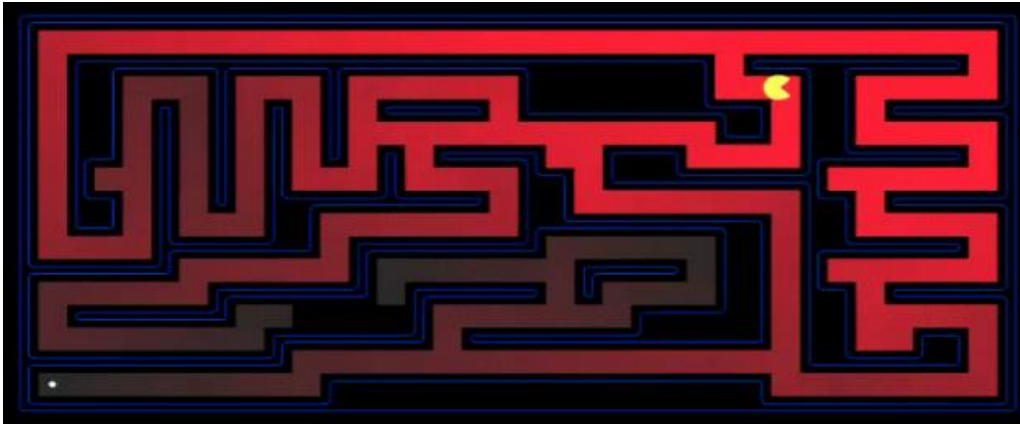
## **Bölüm 4**

### **Depth First Search**



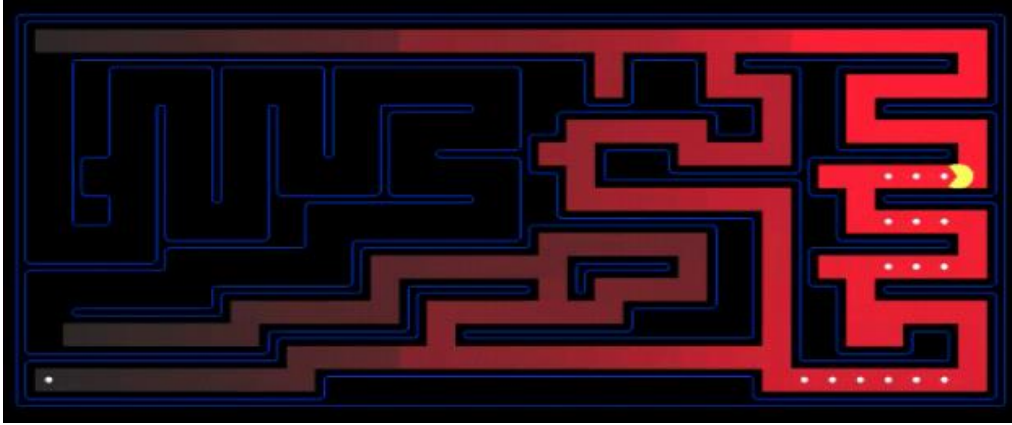
*Depth First Search (DFS):* DFS, labirent içinde derinlemesine bir arama yapar. Pacman, bir yöne gittiğinde bu yöne devam eder ve ilerlediği yolda sona ulaşana kadar gider. DFS, basit bir yapıya sahiptir ancak kısa vadeli bir stratejiye dayanır. En kısa yolu garantilemez, ancak derinleme avantajı sayesinde alternatif yolları keşfeder.

### **Breadth First Search**



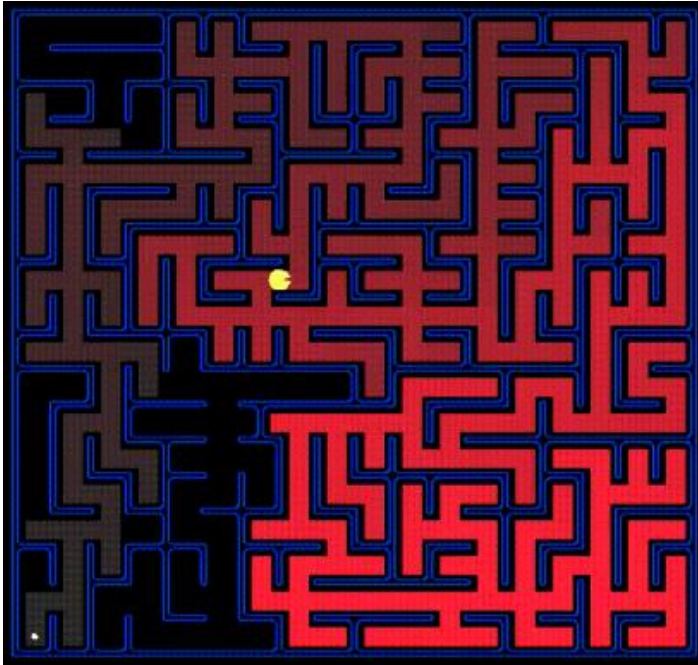
*Breadth First Search:* BFS, labirent içinde geniş bir alana yayılan bir arama yapar. Pacman, bir yönde ilerlerken diğer yönlerde de paralel olarak hareket eder. BFS, en kısa yolu garanti eder, ancak derinleme stratejisine göre daha fazla kaynak kullanabilir.

### Uniform Cost Search



Uniform Cost Search: UCS, maliyeti en aza indirerek labirent içinde bir yol bulmaya çalışır. Pacman, her adımda maliyeti en düşük olan yöne hareket eder. UCS, maliyet hassasiyeti nedeniyle daha optimize edilmiş yollar bulabilir, ancak bazen hedefe ulaşmak için daha fazla adım atabilir.

### A\* search



A\* search: A\*, her adımda hem maliyeti hem de hedefe olan tahmini mesafeyi dikkate alarak bir yol bulmaya çalışır. Pacman, hedefe en kısa maliyetli ve en kısa tahmini mesafeyle ulaşmaya çalışır. A\*, hem optimizasyon hem de bilgi tabanlı bir strateji olduğu için genellikle en etkili arama algoritmalarından biridir.

### Tüm Köşeler



Tüm Köşeler: Bu algoritma, labirent içindeki tüm köşeleri ziyaret etmeyi amaçlar. Pacman, tüm köşeleri ziyaret etmeye odaklanarak bir yol çizer. Köşelerin ziyaret edilmesi, bazı durumlarda daha uzun yolların seçilmesine neden olabilir ancak oyun stratejisini çeşitlendirebilir.

Her algoritmanın avantajları ve dezavantajları bulunmaktadır. Örneğin, kısa vadeli hedefler için DFS, en kısa yolu garanti etmek için BFS veya UCS, daha sofistike bir strateji için A\* tercih edilebilir. All the corners algoritması ise oyun içinde belirli hedefleri takip etmek üzere tasarlanmış bir strateji sunar.

## **Bölüm 5**

### Kaynaklar

[http://ai.berkeley.edu/project\\_overview.html](http://ai.berkeley.edu/project_overview.html)

<http://ai.berkeley.edu/multiagent.html>

<https://inst.eecs.berkeley.edu/~cs188/sp19/project1.html>

<https://inst.eecs.berkeley.edu/~cs188/sp20/project2/>