

## VERİ YAPILARI

### 2021 – 2022 Bahar Dönemi

### Ödev - 1

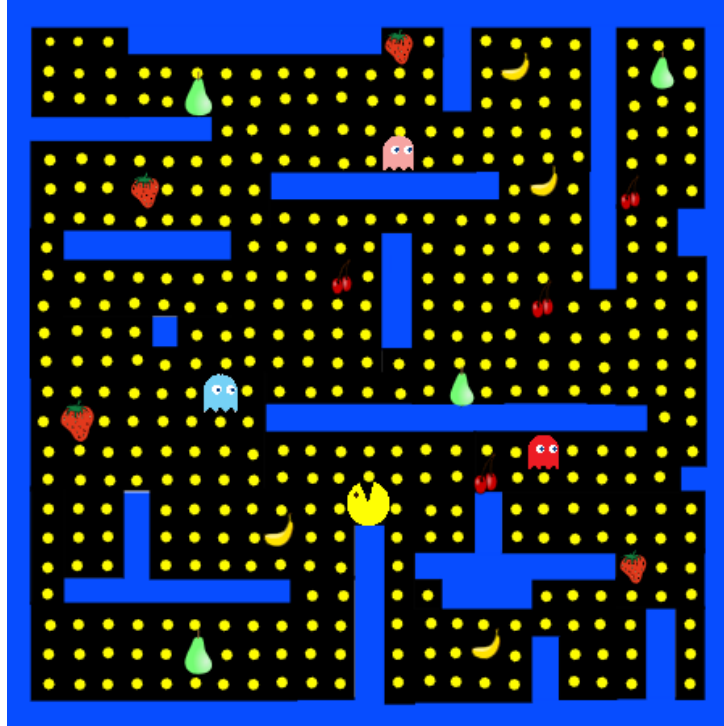
**Ödev Duyurulma Tarihi :** 20 Nisan 2022 Çarşamba, Saat 16:00

**Ödev Teslim Tarihi :** 27 Nisan 2022 Çarşamba, Saat 23:59

#### 1. TANIMLAMA

Bu ödevde PAC-MAN isimli oyunu simüle eden C++ kodu yazmanız istenmektedir. Oyunda PAC-MAN bir labirent içerisinde hareket ederek diskleri bitirmeye çalışmaktadır. Hedefi hayaletlerden kaçarak tüm diskleri toplamak olan PAC-MAN, tüm diskleri topladığında oyunu tamamlar. Kodu yazarken sınıf yapısı, yığın ve kuyruk yapılarını kullanınız (yığın (stack) ve kuyruk (queue) yapılarını bağlı liste ile oluşturunuz. Hazır standart C++ kuyruk ve yığın yapılarını kullanmayınız).

- PAC-MAN 600 saniyelik süre ile oyuna başlayacaktır ve süresi bitmeden tüm diskleri toplamalı gerekmektedir.
- Oyun alanında başlangıç noktasından ilerlerken her adımda 1 sn süre harcayacaktır.
- Yol üzerinde süresini arttıracak meyveler ve süresini azaltacak hayaletler bulunmaktadır.
- 3 kez hayaletle çarpışan PAC-MAN yanacak ve oyun bitecektir.
- Süresi bitmeden tüm diskleri toplayamayan PAC-MAN yanacak ve oyun bitecektir.
- Süresi bitmeden tüm diskleri toplayan PAC-MAN oyunu kazanacaktır.



Şekil 1. PAC-MAN Oyun Alanı

## 2. PROBLEM

Oyun alanı bilgisi **pacman.txt** dosyasında dijital olarak verilmiştir. Problem adımları;

1. PAC-MAN oyun alanında ilerlerken her bir adımda 1 sn lik süre harcar.
2. **pacman.txt** dosyasındaki “9” PAC-MAN ‘i, “0” duvarı, “1” ise yolu temsil etmektedir.
3. **pacman.txt** dosyasındaki “2” meyveleri temsil eder. PAC-MAN herhangi bir meyveyi aldığıda 5 sn süre kazanır ve tüm hayaletler bekleme alanına alınır. Her bir hayaletin bekleme alanında geçireceği süre arasında 5 saniye fark olacaktır (ilk hayalet 5 saniye sonra çıkacak, ikinci hayalet 10 saniye sonra çıkacak, üçüncü hayalet ise 15 saniye sonra bekleme alanından çıkacaktır). Çıkan hayaletler bekleme alanına girmeden önceki konum ve hareket yönünde yeniden oluşacaklardır.
4. PAC-MAN’ın toplamış olduğu her 10 disk için daha önce topladığı meyveler, toplanma sırasıyla yeniden eski konumunda oluşacaktır. Toplanan meyve yeniden oluşuncaya kadar bulunduğu konumda disk varmış gibi davranılır. 10 disk toplanınca disk yerine meyve oluşturulur. PAC-MAN 10 disk toplayana kadar hiç meyve toplamamışsa herhangi bir işlem yapılmayacaktır.
5. **pacman.txt** dosyasındaki “3”, “4” ve “5” hayaletleri temsil eder ve hayaletlerin süre sorunu yoktur. PAC-MAN’ın her bir hayalet ile çarpışması durumunda, PAC-MAN 10 sn süre kaybeder.
6. Hayaletler bulundukları konumda sürekli hareket halinde olacaktır:
  - a. 3 nolu hayalet ilk bulunduğu konumdan önce sağa doğru hareket edecek duvar ile karşılaştıktan sonra bu defa da sola doğru hareket edecektir.
  - b. 4 nolu hayalet ilk bulunduğu konumdan önce yukarı doğru hareket edecek duvar ile karşılaştıktan sonra bu defa da aşağı doğru hareket edecektir.
  - c. 5 nolu hayalet ilk bulunduğu konumdan önce sola doğru hareket edecek duvar ile karşılaştıktan sonra bu defa da sağa doğru hareket edecektir.Hayaletler aynı konumda bulunabilirler (kesişme, çarpışma durumları sorun teşkil etmeyecektir).
7. PAC-MAN’ın başlangıç noktası **pacman.txt** dosyasının tüm değerleri taranarak bulunacaktır. Örneğin; bu oyun alanında başlangıç (17,12) noktasıdır, Başlangıç noktasının belirlenmesi amacıyla yapılan tarama işleminde herhangi bir süre harcanmayacaktır.
8. PAC-MAN her adımda ilk önce yukarı yönde hareket edebiliyorsa yukarı, yukarıda duvar varsa ve sağ açıksa sağ taraftan devam edecek, sağda duvar varsa aşağıyı kontrol edecektir. Aşağı taraf açıksa aşağı yönlü hareket edecek, aşağıda duvar varsa sol tarafı kontrol edecektir. Sol taraf açıksa sola doğru devam edecektir. Attığı **her adım için** bu yaklaşım izlenecektir.

**PAC-MAN’ın takip edeceği yolda disk olmaması halinde;**  
**İlk durumda disk olmayan yön duvar gibi düşünülür ve PAC-MAN diğer yönleri kontrol eder. Eğer diğer yönlerin herhangi birinde duvar yok ve disk varsa o yönde hareket eder. Aksi halde sadece duvar ve disk olmayan yollar arasında kalırsa disk olmayan yolu tercih edip o yönde hareket eder.**
9. PAC-MAN yol üzerinde ilerlerken toplamda 3. kez hayalete çarptığı zaman PAC-MAN yanar ve süresi kalmış olsa bile oyun biter.

10. Tüm diskleri toplayamadan süresi biten PAC-MAN yanar ve oyun biter.  
 11. Süresi bitmeden tüm diskleri toplayan PAC-MAN oyunu kazanır.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	1	1	1	0	0	0	0	0	0	0	0	0	2	1	0	1	1	1	1	0	1	1	1	1	0
2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	2	1	1	0	1	2	1	1	0
3	0	1	1	1	1	1	2	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0
4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0
5	0	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	0	1	1	1	1	0
6	0	1	1	1	2	1	1	1	1	0	0	0	0	0	0	0	0	1	2	1	0	2	1	1	1	0
7	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	
8	0	1	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0	
9	0	1	1	1	1	1	1	1	1	1	1	2	1	0	1	1	1	1	1	1	0	1	1	1	1	0
10	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	2	1	1	1	1	1	1	0
11	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0
12	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
13	0	1	1	1	1	1	1	4	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	0
14	0	1	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
15	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1	1	1	1	1	0
16	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	0	0	
17	0	1	1	1	0	1	1	1	1	1	1	1	9	1	1	1	0	1	1	1	1	1	1	1	1	0
18	0	1	1	1	0	1	1	1	1	2	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	0
19	0	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	2	1	1	1	0
20	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	1	1	1	1	1	0
21	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	
22	0	1	1	1	1	1	2	1	1	1	1	1	0	1	1	1	2	1	0	1	1	1	0	1	0	
23	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Şekil 2. Oyun Alanının Dijital Gösterimi

### 3. OLUŞTURULACAK SINIFLAR

(Kodu yazarken ihtiyaç durumunda daha fazla sınıf ve fonksiyon oluşturulabilir.)

#### Main.cpp:

oyunAlanıOku() : pacman.txt dosyasından oyun alanı bilgileri okunacaktır (Bakınız Tablo-1).

sureHesapla(): PAC-MAN her bir hareketinde 1 sn harcar. Ancak, oyun alanı üzerinde her meyveyi topladığında mevcut süresine 5 sn eklenir. Ayrıca oyun alanı üzerinde hayaletlere çarptığında ise mevcut süresinden 10 sn düşürülür.

diskTopla(): PAC-MAN 600 sn ile oyun alanına başlangıç noktasından giriş yapar. Her bir birimlik ilerleme için 1 sn harcanır ve bir disk toplanmış olur. Süresi bitene kadar hareketine devam eder. Süresi bittiğinde PAC-MAN tüm diskleri toplamamış ise oyun biter. PAC-MAN'ın yol üzerindeki her adım bilgisi yazılmalıdır (duvar ile karşılaşp tekrar geri dönülen yollar dâhil). Daha önce geçilen noktada disk bulunmaz.

meyveOlustur(): PAC-MAN'ın toplamış olduğu her 10 disk için daha önce topladığı meyveleri, toplanma sırasıyla yeniden eski konumunda oluşturur. 10 disk toplayana kadar hiç meyve toplamamışsa herhangi bir işlem yapılmaz.

hayaletBeklet(): PAC-MAN meyveyi topladığında oyun alanındaki tüm hayaletler beklenme alınır. Her bir hayaletin bekleme alanında geçireceği süre arasında 5 saniye fark olacaktır (ilk hayalet 5 saniye sonra çıkacak, ikinci hayalet 10 saniye sonra çıkacak, üçüncü hayalet ise 15 saniye sonra bekleme alanından çıkacaktır). Çıkan hayaletler bekleme alanına girmeden önceki konum ve hareket yönünde yeniden oluşacaklardır. Bekleme alanında hayalet varken PAC-MAN yeni bir meyve toplarsa, bekleme alanına alınacak hayaletin bekleme süresi bekleme alanına en son alınan hayaletin bekleme süresinden 5 saniye daha fazla olacaktır. Örneğin, bekleme alanında 3 saniye süresi kalmış olan bir tane hayalet olsun. PAC-MAN yeniden meyve alırsa bekleme alanına gelen hayaletlerin bekleme süresi şu şekilde hesaplanacaktır: 2. hayalet için:  $3+5=8$  saniye, 3. hayalet için:  $8+5=13$  saniye olacak.

#### 4. GİRDİ VERİ DOSYALARI

**pacman.txt** dosyasının içeriği Tablo 1’de gösterilmiştir.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	2	1	0	1	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	2	1	1	0	1	2	1	0
0	1	1	1	1	1	2	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	0	1	1	1	0
0	1	1	1	2	1	1	1	1	0	0	0	0	0	0	0	0	1	2	1	0	2	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
0	1	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	2	1	0	1	1	1	1	1	1	0	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	2	1	1	1	1	0
0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	4	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	0
0	1	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	0	0
0	1	1	1	0	1	1	1	1	1	1	1	9	1	1	1	0	1	1	1	1	1	1	1	0
0	1	1	1	0	1	1	1	1	2	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0
0	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	2	1	1	0
0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0
0	1	1	1	1	1	2	1	1	1	1	1	0	1	1	1	2	1	0	1	1	1	0	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 1. *pacman.txt*

## 5. YAPILACAKLAR

1. **pacman.txt** dosyasından oyun alanı bilgileri okunacaktır.
2. Oyun alanında her bir meyve ve hayalet ile karşılaşma konum bilgisi **diskTopla()** fonksiyonu ile **meyveKonum.txt** ve **hayaletKonum.txt** dosyalarına yazdırılacaktır.
3. PAC-MAN'ın bulunduğu nokta ve mevcut süresi aşağıda gösterilen formatta **pacmanHareket.txt** dosyasına yazdırılacaktır (örneğin; **(1,2-20)**; konum bilgisi (1,2), süre 20 sn olarak gösterilmektedir) .
4. Süresi bittiğinde ve tüm diskler toplanmamış ise veya 3 kez hayalet ile çarpışınca oyun bitecektir. Bu bilgiler de **pacmanHareket.txt** dosyasında verilmelidir.

**Çıktı dosyaları:** *(çıktı örnek olarak verilmiştir)*

### **meyveKonum.txt**

Meyve1 → (2,22)  
Meyve2 → (6,21)  
Meyve1 yeniden oluşturuldu.  
Meyve3 → (2,12)  
Meyve2 yeniden oluşturuldu.  
Meyve4 → (10,18)  
..  
Meyve15 → (22,6)

### **hayaletKonum.txt**

Hayalet1 → (15,18)  
Hayalet1 bekleme durumunda  
Hayalet2 → (5,13)  
Hayalet2 bekleme durumunda  
Hayalet3 → (13,7)  
Hayalet3 bekleme durumunda

### **pacmanHareket.txt**

(16,12 – 59)  
(15,12 – 58)  
(15, 13 – 57)  
(15,14 – 56)  
(15,16 – 55)  
(15,17 – 54)  
(15,18 – 44)  
(15,19 – 43)  
...  
(22,3 – 0) Süre bitti.

Topladığı disk sayısı: \*\*\*\*  
Hayaletle çarpışma sayısı: \*\*\*\*

## 6. DİKKAT EDİLMESİ GEREKENLER

1. Ödevleriniz değerlendirilirken elimizdeki farklı test dosyaları kullanılacaktır. Bu test dosyaları sizinle paylaştığımız örnek “\*.txt” dosyaları ile aynı formatta olacaktır. Bu yüzden kodunuzun farklı boyutlarda veri alıp veri yazma işlemine izin verdiğinden emin olunuz.

2. Ödev C++ programlama diliyle yazılacaktır (C++ dili, C dilini desteklediğinden kodunuzun çalışıyor olması C++’a özgü hazırlanmış anlamına gelmez. C++ dilinde hazır **STL kütüphanelerini** kullanmayınız. Kullandığınız takdirde ödev değerlendirmeye alınmayacaktır. C++ diline özgü olmasına dikkat edin, nesne kullanımı, bellekte yer ayırma gibi işlemlerde dikkatli olun, malloc yerine new, free yerine delete gibi).

3. Ödevi hazırlarken doğru veri yapısını ve doğru algoritmayı kullandığınızdan emin olun!

4. Ödevleriniz Haruzem üzerinden (ue.harran.edu.tr) yüklenecektir.

Örneğin, ödevi gönderen kişi Ayşe Şeker olsun ve öğrenci numarası 123145344 olsun.

Öncelikle, ödev dosya ismi **123145344\_Ayse\_Seker\_odev1.zip** şeklinde olacaktır. Yani göndereceğiniz ödev dosyasını, kendi öğrenci numaranız ve ad soyadınıza göre örnekte gösterildiği gibi adlandırın (dosyayı isimlendirirken Türkçe karakter kullanmayınız, boşluk bırakmayınız, ayırmayı alt çizgiler kullanarak yapınız).

5. Ödevlerinizi en geç teslim tarihi saatine (ilk sayfada) kadar göndermiş olduğunuzdan emin olunuz. **Ödev teslim tarih saatinden sonra gönderilen ödevler değerlendirilmeye alınmayacaktır!**

6. Bu ödevin, genel dönem puanına etkisi **20 puandır**. Kopya ödev, İnternette hazır kopyala yapıştır ödev teslim etmeyiniz. Bu tür durumların tespitinde öğrencinin **disiplin kuruluna sevk**i yapılacaktır. O yüzden ödevinizi kendiniz yapınız!

7. Göndereceğiniz kodun **Linux konsol (terminal)** ortamında çalışıyor olduğuna dikkat ediniz.

8. Ödevi Gönderirken sadece “\*.cpp”, “\*.h” dosyalarınızı gönderiniz. Ödevi gönderirken **proje halinde göndermeyiniz. Aksi Takdirde ödev değerlendirmeye alınmayacaktır. Ayrıca ödevi gönderirken lütfen “\*.exe” dosyası göndermeyiniz.**

9. Ödev ile ilgili sorularınızı [handangumus@harran.edu.tr](mailto:handangumus@harran.edu.tr) ve [songulakdag@harran.edu.tr](mailto:songulakdag@harran.edu.tr) mail adreslerinden hocalarımıza sorabilirsiniz.