



ALGORİTMA VE PROGRAMLAMA - 1

Güz Dönemi 2. Proje Raporu



Kilit Oyunu

Ali Gökalp Karakuş 05210000198

Selanay Akbaba 05210000248

05 OCAK 2023
EGE ÜNİVERSİTESİ

İÇİNDEKİLER

Programcı Katalođu.....	2
Test Katalođu.....	7
Kaynakça.....	14



Programcı Kataloğu:

Kilit oyunu için hazırladığım algorithmada kullandığım listeler:

```
293 satir = ["A", "B", "C", "D", "E", "F", "G", "H"]
294 sutun = [[" " for a in range(kaca_kac)] for b in range(kaca_kac)]
```

Bu listelerden “satir” adlı liste tabloyu oluştururken en üstte ve en altta harfleri yazdırmak için kullandık. Aynı zamanda kullanıcının konum değişimi için verdiği bilgilerde girdiği harflerin indeksini alırken de kullandık.

“sutun” listesi ise çift boyutlu bir listedir. Bu listeyi tabloyu yazdırırken ve taşları hareket ettirmede kullandık.

Kilit oyunu için hazırladığımız 4 fonksiyon var. Bu fonksiyonlardan ilki “karakter_secimi”:

```
1 def karakter_secimi(sutun, kaca_kac, satir):
2     global birinci_oyuncu, ikinci_oyuncu
3     hata1 = 'e'
4     hata2 = 'e'
5     birinci_oyuncu = input("1. oyuncuyu temsil etmek için karakter giriniz:") # birinci oyuncunun girişi ve kontrolleri
6     while hata1 == 'e':
7         if len(birinci_oyuncu) > 1 or len(birinci_oyuncu) == 0:
8             birinci_oyuncu = input("1. oyuncuyu temsil etmek için bir karakter giriniz:")
9             hata1 = 'e'
10        elif birinci_oyuncu == " ":
11            birinci_oyuncu = input("1. oyuncuyu temsil etmek için boşluk karakterinden başka karakter giriniz:")
12            hata1 = 'e'
13        else:
14            hata1 = 'h'
15    ikinci_oyuncu = input("2. oyuncuyu temsil etmek için karakter giriniz:") # ikinci oyunun girişi ve kontrolleri
16    while hata2 == 'e':
17        if len(ikinci_oyuncu) > 1 or len(ikinci_oyuncu) == 0:
18            ikinci_oyuncu = input("2. oyuncuyu temsil etmek için bir karakter giriniz:")
19            hata2 = 'e'
20        elif ikinci_oyuncu == " ":
21            ikinci_oyuncu = input("2. oyuncuyu temsil etmek için boşluk karakterinden başka karakter giriniz:")
22            hata2 = 'e'
23        elif birinci_oyuncu == ikinci_oyuncu:
24            print("Oyuncu taşları aynı olamaz.")
25            ikinci_oyuncu = input("2. oyuncuyu temsil etmek için bir karakter giriniz:")
26            hata2 = 'e'
27        else:
28            hata2 = 'h'
29    if hata1 == 'h' and hata2 == 'h': # kullanıcı girişleri hatasızsa
30        birinci_oyuncu = birinci_oyuncu.upper()
31        ikinci_oyuncu = ikinci_oyuncu.upper()
32        print(f"Birinci oyuncunun seçtiği karakter: {birinci_oyuncu}")
33        print(f"İkinci oyuncunun seçtiği karakter: {ikinci_oyuncu}")
34        for index in range(kaca_kac): # ilk ve son satırlara karakterleri yazdırma
35            sutun[kaca_kac - 1][index] = birinci_oyuncu
36            sutun[0][index] = ikinci_oyuncu
37        print()
38        tablo(kaca_kac, sutun, satir)
39        print()
40
```

“karakter_secimi” fonksiyonu birinci ve ikinci oyuncunun girdiği karakterlerin uygunluğunu karakter sayısının 1 olup olmadığını, ya da karakter olarak boşluk seçilip seçilmediğini kontrol etmektedir. Kontrolün sonunda girilen karakterlerin büyük harf olmaması durumunu engellemek için upper() metodunu kullandık. Fonksiyonun sonunda ise kullanıcıların girdiği

karakterleri oluşacak olan tablonun en alt ve en üst kısımlarına yazdırarak oyun alanımızı hazır hale getiriyoruz.

2. fonksiyonumuz “hareketlendirme” adlı fonksiyon. Bu fonksiyonda kullanıcıların girdiği konum değişimi bilgisinin uygunluğu kontrol ediliyor uygun değilse girişler tekrar isteniyor. Doğru giriş sonucunda oluşabilecek durumlar değerlendirilerek oyun tablomuz tekrar şekilleniyor.

```
42 def hareketlendirme(sutun, kaca_kac, satir, kalan_1, kalan_2):
43     oyuncu = 2
44     global birinci_oyuncu, ikinci_oyuncu, hamle_yapan
45     while True:
46         try:
47             while kalan_1 > 1 and kalan_2 > 1: # kalan taş sayısına göre bitiş
48                 hatasiz_hamle = 'e'
49                 if oyuncu % 2 == 0: # çift veya tek sayı olmasına göre hamle yaptırma
50                     hamle_yapan = birinci_oyuncu
51                     bekleyen = ikinci_oyuncu
52                 else:
53                     hamle_yapan = ikinci_oyuncu
54                     bekleyen = birinci_oyuncu
55                 print()
56                 konum_degisim = input(
57                     f"Oyuncu {hamle_yapan}, Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef "
58                     f"konumunu giriniz (ör:1A 2A):") # gidilecek konum için giriş
59                 while len(konum_degisim) > 5 or len(konum_degisim) < 5: # 5 karaktere ihtiyaç olduğundan kontrol
60                     konum_degisim = input(
61                         f"Oyuncu {hamle_yapan}, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın "
62                         f"konumunu ve hedef konumunu tekrar giriniz (ör:1A 2A):")
63                 hedef_liste = list(konum_degisim) # girilen veriyi listeye dönüştürme
64                 hedef_liste[1] = hedef_liste[1].upper() # hareket ettirmede küçük harf girilmesine karşı önlem
65                 hedef_liste[4] = hedef_liste[4].upper()
66                 print()
67                 while hatasiz_hamle == 'e': # hamle hatalarının kontrolü
68                     if sutun[int(hedef_liste[0]) - 1][satir.index(str(hedef_liste[1]))] == " ":
69                         print("Belirtmiş olduğunuz ilk konumda taşınız yok.")
70                         hatasiz_hamle = 'e'
71                     elif hedef_liste[2] != " ":
72                         print("Boşluk karakteri olacak şekilde örneğin 1A 5A olarak giriniz.")
73                         hatasiz_hamle = 'e'
74                     elif hedef_liste[1] != hedef_liste[4] and hedef_liste[0] != hedef_liste[3]:
75                         print("Lütfen bulunduğunuz satır ve sütun boyunca hareket ettirin.")
76                         hatasiz_hamle = 'e'
77                     elif sutun[int(hedef_liste[0]) - 1][satir.index(str(hedef_liste[1]))] != hamle_yapan:
78                         print("Size ait olmayan bir taşı hareket ettirmeye çalıştınız.")
79                         hatasiz_hamle = 'e'
80                     elif sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4]))] != " ":
81                         print("Belirtmiş olduğunuz son konum dolu.")
```



```

81         print("Size ait olmayan bir taşı hareket ettirmeye çalıştınız.")
82         hatasiz_hamle = 'e'
83     elif hedef_liste[2] != " ":
84         print("Boşluk karakteri olacak şekilde örneğin 1A 5A olarak giriniz.")
85     else:
86         hatasiz_hamle = 'yok'
87     if hatasiz_hamle == 'e':
88         konum_degisim = input(
89             f"Oyuncu {hamle_yapan}, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi "
90             f"taşınızın konumunu ve hedef konumunu giriniz:")
91         while len(konum_degisim) > 5 or len(konum_degisim) < 5:
92             konum_degisim = input(
93                 f"Oyuncu {hamle_yapan}, hatalı giriş. Lütfen hareket ettirmek istediğiniz "
94                 f"kendi taşınızın konumunu ve hedef konumunu tekrar giriniz (ör:1A 2A):")
95         hedef_liste = list(konum_degisim)
96         hedef_liste[1] = hedef_liste[1].upper()
97         hedef_liste[4] = hedef_liste[4].upper()
98         print()
99     while hatasiz_hamle == 'yok': # Üstünden atlamaya engel olmak için yazılan kısım
100     if sutun[int(hedef_liste[0]) - 1] == sutun[int(hedef_liste[3]) - 1] and \
101         satir.index(str(hedef_liste[1])) > satir.index(str(hedef_liste[4])):
102         for i in range(satir.index(str(hedef_liste[4])), satir.index(str(hedef_liste[1]))):
103             if sutun[int(hedef_liste[0]) - 1][i] != " ":
104                 print("Belirtmiş olduğunuz konuma giden yol dolu.")
105                 hatasiz_hamle = 'e'
106                 break
107         else:
108             hatasiz_hamle = 'yok'
109     elif sutun[int(hedef_liste[0]) - 1] == sutun[int(hedef_liste[3]) - 1] and \
110         satir.index(str(hedef_liste[1])) < satir.index(str(hedef_liste[4])):
111         for i in range(satir.index(str(hedef_liste[1])), satir.index(str(hedef_liste[4]))):
112             if sutun[int(hedef_liste[0]) - 1][i + 1] != " ":
113                 print("Belirtmiş olduğunuz konuma giden yol dolu.")
114                 hatasiz_hamle = 'e'
115                 break
116         else:
117             hatasiz_hamle = 'yok'
118     elif satir.index(str(hedef_liste[1])) == satir.index(str(hedef_liste[4])) and \
119         int(hedef_liste[0]) < int(hedef_liste[3]):
120         for i in range(int(hedef_liste[0]) - 1, int(hedef_liste[3]) - 1):

```

```

120         for i in range(int(hedef_liste[0]) - 1, int(hedef_liste[3]) - 1):
121             if sutun[i + 1][satir.index(str(hedef_liste[1]))] != " ":
122                 print("Belirtmiş olduğunuz konuma giden yol dolu.")
123                 hatasiz_hamle = 'e'
124                 break
125         else:
126             hatasiz_hamle = 'yok'
127     elif satir.index(str(hedef_liste[1])) == satir.index(str(hedef_liste[4])) and \
128         int(hedef_liste[0]) > int(hedef_liste[3]):
129         for i in range(int(hedef_liste[3]), int(hedef_liste[0])):
130             if sutun[i - 1][satir.index(str(hedef_liste[1]))] != " ":
131                 print("Belirtmiş olduğunuz konuma giden yol dolu.")
132                 hatasiz_hamle = 'e'
133                 break
134         else:
135             hatasiz_hamle = 'yok'
136     else:
137         break
138     if hatasiz_hamle == 'e':
139         konum_degisim = input(
140             f"Oyuncu {hamle_yapan}, lütfen hareket ettirmek istediğiniz kendi "
141             f"taşınızın konumunu ve hedef konumunu giriniz:")
142         while len(konum_degisim) > 5 or len(konum_degisim) < 5:
143             konum_degisim = input(
144                 f"Oyuncu {hamle_yapan}, hatalı giriş. Lütfen hareket ettirmek istediğiniz "
145                 f"kendi taşınızın konumunu ve hedef konumunu tekrar giriniz (ör:1A 2A):")
146         hedef_liste = list(konum_degisim)
147         hedef_liste[1] = hedef_liste[1].upper()
148         hedef_liste[4] = hedef_liste[4].upper()
149         print()
150     else:
151         break
152     while hatasiz_hamle == 'yok': # eğer hata yoksa harekete ettirme kısmı
153         sutun[int(hedef_liste[0]) - 1][satir.index(str(hedef_liste[1]))] = " "
154         sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4]))] = hamle_yapan
155         if kaca_kac - 1 >= int(hedef_liste[3]) + 1: # sütundaki kısırtma kontrolü
156             if sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4]))] == \
157                 sutun[int(hedef_liste[3]) + 1][satir.index(str(hedef_liste[4]))] == \
158                 hamle_yapan and sutun[int(hedef_liste[3])][
159                     satir.index(str(hedef_liste[4]))] == bekleyen:

```

```

159         satir.index(str(hedef_liste[4])) == bekleyen:
160     sutun[int(hedef_liste[3])[satir.index(str(hedef_liste[4]))] = " "
161     print(f"{bekleyen} taşı {int(hedef_liste[3]) + 1}{hedef_liste[4]} konumundan çıkarıldı.")
162     if hamle_yapan == birinci_oyuncu:
163         kalan_2 -= 1
164     else:
165         kalan_1 -= 1
166
167     if int(hedef_liste[3]) - 3 >= 0: # sütundaki kısırtma kontrolü
168         if sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4]))] == \
169             sutun[int(hedef_liste[3]) - 3][
170                 satir.index(str(hedef_liste[4]))] == hamle_yapan and sutun[int(hedef_liste[3]) - 2][
171                     satir.index(str(hedef_liste[4]))] == bekleyen:
172             sutun[int(hedef_liste[3]) - 2][satir.index(str(hedef_liste[4]))] = " "
173             print(f"{bekleyen} taşı {int(hedef_liste[3]) - 1}{hedef_liste[4]} konumundan çıkarıldı.")
174             if hamle_yapan == birinci_oyuncu:
175                 kalan_2 -= 1
176             else:
177                 kalan_1 -= 1
178
179     if kaca_kac - 1 >= satir.index(str(hedef_liste[4])) + 2: # satırdaki kısırtma kontrolü
180         if sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4]))] == \
181             sutun[int(hedef_liste[3]) - 1][
182                 satir.index(str(hedef_liste[4])) + 2] == hamle_yapan and \
183                 sutun[int(hedef_liste[3]) - 1][
184                     satir.index(str(hedef_liste[4])) + 1] == bekleyen:
185             sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4])) + 1] = " "
186             print(f"{bekleyen} taşı {int(hedef_liste[3])}"
187                   f"{satir[satir.index(str(hedef_liste[4])) + 1]} konumundan çıkarıldı.")
188             if hamle_yapan == birinci_oyuncu:
189                 kalan_2 -= 1
190             else:
191                 kalan_1 -= 1
192
193     if satir.index(str(hedef_liste[4])) - 2 >= 0: # satırdaki kısırtma kontrolü
194         if sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4]))] == \
195             sutun[int(hedef_liste[3]) - 1][
196                 satir.index(str(hedef_liste[4])) - 2] == hamle_yapan and \
197                 sutun[int(hedef_liste[3]) - 1][
198                     satir.index(str(hedef_liste[4])) - 1] == bekleyen:
199             satir.index(str(hedef_liste[4])) - 1] == bekleyen:
200             sutun[int(hedef_liste[3]) - 1][satir.index(str(hedef_liste[4])) - 1] = " "
201             print(f"{bekleyen} taşı {int(hedef_liste[3])}"
202                   f"{satir[satir.index(str(hedef_liste[4])) - 1]} konumundan çıkarıldı.")
203             if hamle_yapan == birinci_oyuncu:
204                 kalan_2 -= 1
205             else:
206                 kalan_1 -= 1
207
208     if sutun[0][0] == bekleyen and sutun[0][1] == hamle_yapan and sutun[1][0] == hamle_yapan:
209         # sol üst köşedeki kısırtma
210         sutun[0][0] = " "
211         print(f"{bekleyen} taşı 1A konumundan çıkarıldı")
212         if hamle_yapan == birinci_oyuncu:
213             kalan_2 -= 1
214         else:
215             kalan_1 -= 1
216
217     elif sutun[kaca_kac - 1][0] == bekleyen and sutun[kaca_kac - 2][0] == \
218         hamle_yapan and sutun[kaca_kac - 1][1] == hamle_yapan: # sol alt köşedeki kısırtma
219         sutun[kaca_kac - 1][0] = " "
220         print(f"{bekleyen} taşı {kaca_kac}A konumundan çıkarıldı.")
221         if hamle_yapan == birinci_oyuncu:
222             kalan_2 -= 1
223         else:
224             kalan_1 -= 1
225
226     elif sutun[0][kaca_kac - 1] == bekleyen and \
227         sutun[0][kaca_kac - 2] == hamle_yapan and \
228         sutun[1][kaca_kac - 1] == hamle_yapan: # sağ üst köşedeki kısırtma
229         sutun[0][kaca_kac - 1] = " "
230         print(f"{bekleyen} taşı 1{satir[kaca_kac - 1]} konumundan çıkarıldı.")
231         if hamle_yapan == birinci_oyuncu:
232             kalan_2 -= 1
233         else:
234             kalan_1 -= 1
235
236     elif sutun[kaca_kac - 1][kaca_kac - 1] == bekleyen and \
237         sutun[kaca_kac - 1][kaca_kac - 2] == hamle_yapan and \
238         sutun[kaca_kac - 2][kaca_kac - 1] == hamle_yapan: # sağ alt köşedeki kısırtma

```

```

237         sutun[kaca_kac - 2][kaca_kac - 1] == hamle_yapan: # sağ alt kösedeki kısırtma
238         sutun[kaca_kac - 1][kaca_kac - 1] = " "
239         print(f"{bekleyen} taşı {kaca_kac} {satir[kaca_kac - 1]} konumundan çıkarıldı.")
240         if hamle_yapan == birinci_oyuncu:
241             kalan_2 -= 1
242         else:
243             kalan_1 -= 1
244         tablo(kaca_kac, sutun, satir)
245         print()
246         oyuncu += 1 # tek çift sayı kontrolü için
247         break
248     print()
249     if kalan_1 > kalan_2: # son hamle yapan oyunu kazanacağı için kazananı gösterme
250         print(f"Oyuncu {hamle_yapan} oyunu kazandı.")
251     else:
252         print(f"Oyuncu {hamle_yapan} oyunu kazandı.")
253     print("Oyun bitti..")
254     print()
255     break
256
257 except IndexError:
258     print("Index hatası. Oyun alanının dışında yer seçildi.")
259 except ValueError:
260     print("Hatalı giriş tekrar giriniz (ör:1A 3A).")
261
262

```

3. fonksiyonumuz "tablo" her girilen verinin doğru olması sonucunda oyun alanını yazdırmaya yarıyor. Bu fonksiyonda önce "satir" listesi kullanılarak en üste harfler sıralanıyor. Ardından çift boyutlu liste "sutun" listesi yazdırılıyor. Son olarak tekrar "satir" listesi harfleri tekrar sıralıyor.

```

263 def tablo(kaca_kac, sutun, satir):
264     for x in range(kaca_kac): # Üste harfleri yazdırma
265         print(" ", end="")
266         print(f"{satir[x]}", end="")
267     print("")
268     print(" -", end="")
269     print("----" * kaca_kac)
270     for y in range(kaca_kac): # çift boyutlu listeyi yazdırma
271         print(f"{y + 1}| ", end="")
272         for sayi in range(kaca_kac):
273             print(f"{sutun[y][sayi]}", end=" | ")
274         print(f"{y + 1}")
275         print(" -", end="")
276         print("----" * kaca_kac)
277     for x in range(kaca_kac): # alta harfleri yazdırma
278         print(" ", end="")
279         print(f"{satir[x]}", end="")

```


Son fonksiyonumuz ise main() fonksiyonu. Bu kısımda diğer fonksiyonlar kullanılıyor, oyun alanının kaç kaç olması istendiği soruluyor, ek olarak oyuna devam edip edilmeyeceği sorgulanıyor.

```
282 def main():
283     print("    KİLİT OYUNUNA HOŞ GELDİNİZ")
284     print("    -----")
285     MIN_TABLO = 4
286     MAX_TABLO = 8
287     devam = 'e'
288     while devam == 'e' or devam == 'E':
289         try:
290             kaca_kac = int(input("Oyun alanının satır/sütun sayısını giriniz(4-8):"))
291             while not MIN_TABLO <= kaca_kac <= MAX_TABLO:
292                 kaca_kac = int(input("Oyun alanının satır/sütun tekrar sayısını giriniz(4-8):"))
293             satir = ["A", "B", "C", "D", "E", "F", "G", "H"]
294             sutun = [{" " for a in range(kaca_kac)} for b in range(kaca_kac)]
295             kalan_1 = kaca_kac
296             kalan_2 = kaca_kac
297             karakter_secimi(sutun, kaca_kac, satir)
298             hareketlendirme(sutun, kaca_kac, satir, kalan_1, kalan_2)
299             devam = input("Tekrar oynamak istiyor musunuz? (e/E/h/H):")
300             while devam not in ['e', 'E', 'h', 'H']:
301                 print("Hatalı giriş, lütfen e/E/h/H karakterlerden birini giriniz.")
302                 devam = input("Tekrar oynamak istiyor musunuz? (e/E/h/H):")
303             if devam in ['h', 'H']:
304                 print("Oynadığınız için teşekkür eder yine bekleriz...")
305         except ValueError:
306             print("Integer yani sayı girilmeli.")
307
308
309     main()
310
```

Test Kataloğu:

Oyunumuzu örneklerle ilerleterek deneyelim. Öncelikle oyun alanı için istenen bilgileri girelim ve uygun bilgiler olup olmadığını kontrol ettirelim.

```
KİLİT OYUNUNA HOŞ GELDİNİZ
-----
Oyun alanının satır/sütun sayısını giriniz(4-8):f
Integer yani sayı girilmeli.
Oyun alanının satır/sütun sayısını giriniz(4-8):g
Integer yani sayı girilmeli.
Oyun alanının satır/sütun sayısını giriniz(4-8):x
Integer yani sayı girilmeli.
Oyun alanının satır/sütun sayısını giriniz(4-8):4
1. oyuncuyu temsil etmek için karakter giriniz:x
2. oyuncuyu temsil etmek için karakter giriniz:y
Birinci oyuncunun seçtiği karakter: X
İkinci oyuncunun seçtiği karakter: Y

  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|  |  |  |  | 2
-----
3|  |  |  |  | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D
```

Görüldüğü üzere oyun alanı için sayı isteniyor ve uygun veri girilene kadar bekleniyor. Uygun veri girilince oyuncular için karakter bilgileri isteniyor ve son olarak oyun alanımız çıktı olarak veriliyor.

Oyuncuların girdiği karakterlerin de uygunluğu test edilmelidir. Boşluk karakteri ya da hiç karakter girilmediği durumlar gözetilerek tekrar girmeleri istenmelidir. Ayrıca kullanıcıların aynı karakterleri girip girmedikleri de kontrol edilmelidir.

```
KİLİT OYUNUNA HOŞ GELDİNİZ
-----
Oyun alanının satır/sütun sayısını giriniz(4-8):4
1. oyuncuyu temsil etmek için karakter giriniz:
1. oyuncuyu temsil etmek için bir karakter giriniz:
1. oyuncuyu temsil etmek için boşluk karakterinden başka karakter giriniz:x
2. oyuncuyu temsil etmek için karakter giriniz:
2. oyuncuyu temsil etmek için bir karakter giriniz:
2. oyuncuyu temsil etmek için boşluk karakterinden başka karakter giriniz:x
Oyuncu taşları aynı olamaz.
2. oyuncuyu temsil etmek için bir karakter giriniz:y
Birinci oyuncunun seçtiği karakter: X
İkinci oyuncunun seçtiği karakter: Y

  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|  |  |  |  | 2
-----
3|  |  |  |  | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D
```

Oyuncuların girişlerine göre uygun karakterler istenmektedir.

Oyun tablosunun farklı boyutlardaki çıktılarına bakalım. Tüm tabloları x ve y karakterleri ile yazdıralım.

4x4

	A	B	C	D	
1	Y	Y	Y	Y	1
2					2
3					3
4	X	X	X	X	4
	A	B	C	D	

5x5

	A	B	C	D	E	
1	Y	Y	Y	Y	Y	1
2						2
3						3
4						4
5	X	X	X	X	X	5
	A	B	C	D	E	

8x8

	A	B	C	D	E	F	G	H	
1	Y	Y	Y	Y	Y	Y	Y	Y	1
2									2
3									3
4									4
5									5
6									6
7									7
8	X	X	X	X	X	X	X	X	8
	A	B	C	D	E	F	G	H	

4x4 tablosunda bir oyun alanı hazırlayalım. Oyuncuların konum değişimleri yaparken girdikleri hatalı verileri ve aldıkları hata raporlarına bakalım.

```
A B C D
-----
1| Y | Y | Y | Y | 1
-----
2|  |  |  |  | 2
-----
3|  |  |  |  | 3
-----
4| X | X | X | X | 4
-----
A B C D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4a 1a

Belirtmiş olduğunuz son konum dolu.
Oyuncu X, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz:
```

X oyuncusu hatalı hareket yaptı ve hata raporunu yazdırdı. Dolu yere oynaması halinde uyarıldı ve tekrar giriş yapması istendi.

Bir diğer hata ise oyuncunun boş bir yerden taşı oynatmak istemesidir. Uyarılır ve tekrar giriş istenir.

```
A B C D
-----
1| Y | Y | Y | Y | 1
-----
2|  |  |  |  | 2
-----
3|  |  |  |  | 3
-----
4| X | X | X | X | 4
-----
A B C D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):3a 2a

Belirtmiş olduğunuz ilk konumda taşınız yok.
Oyuncu X, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz:
```

Hatalardan biri rakibin taşı oynatmaya çalışmaktır. Uyarılır ve tekrar giriş istenir.

```
A B C D
-----
1| Y | Y | Y | Y | 1
-----
2|  |  |  |  | 2
-----
3|  |  |  |  | 3
-----
4| X | X | X | X | 4
-----
A B C D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):1a 3a

Size ait olmayan bir taşı hareket ettirmeye çalıştınız.
Oyuncu X, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz:
```

Kullanıcı sadece satırlar ve sütunlar boyunca hareketlendirmeler yapmalıdır. Eğer tek hamlede çapraz gitmeye çalışırsa uyarılır ve tekrar giriş istenir.

```
  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4a 2b

Lütfen bulunduğunuz satır ve sütun boyunca hareket ettirin.
Oyuncu X, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz:|
```

Giriş istenirken “1A 3A” şeklinde istenmektedir. Koordinatlar arasında bir boşluk karakteri olmalı ki diğer hatalar ile çakışıp yanlış hata raporu döndürmesin.

```
  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4a 2b

Lütfen bulunduğunuz satır ve sütun boyunca hareket ettirin.
Oyuncu X, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz:|
```

```
  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4ab2a

Boşluk karakteri olacak şekilde örneğin 1A 5A olarak giriniz.
Oyuncu X, hatalı giriş. Lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz:|
```

Yukarıda da bahsettiğim gibi kullanıcılardan 1A 3B gibi rakam ve harflerle konum değişimi istendiği için sadece sayılardan oluşan yanlış bir veri girilirse uyarılır.

```
  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):12345
Hatalı giriş tekrar giriniz (ör:1A 3A).
```

Eğer oyuncu oyun alanının dışında bir konum seçerse indeks hatası uyarısı gelir.

```
  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):5a 2a
Index hatası. Oyun alanının dışında yer seçildi.

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):|
```

Üstteki hatalı durumları yaşamadan doğru verileri girince oyuncular birbirlerinin taşlarını oyundan çıkarmaya çalışacaktır. Bu taş çıkarma durumlarını hatalı veriler girmeksizin göstereyim ve oyundan atılan taşların hangi konumda olduklarını da görelim.

```
  KİLİT OYUNUNA HOŞ GELDİNİZ
-----
Oyun alanının satır/sütun sayısını giriniz(4-8):4
1. oyuncuyu temsil etmek için karakter giriniz:x
2. oyuncuyu temsil etmek için karakter giriniz:y
Birinci oyuncunun seçtiği karakter: X
İkinci oyuncunun seçtiği karakter: Y

  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4b 2b

  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   | X |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X |   | X | X | 4
-----
  A  B  C  D
```


Oyuncu Y, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):1a 2a

	A	B	C	D	
1		Y Y Y	1		
2	Y X			2	
3				3	
4	X	X X	4		

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4a 3d

	A	B	C	D	
1		Y Y Y	1		
2	Y X			2	
3			X	3	
4	X	X		4	

Oyuncu Y, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):1c 2c

X taşı 2B konumundan çıkarıldı.

	A	B	C	D	
1		Y	Y	1	
2	Y	Y		2	
3			X	3	
4	X	X		4	

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4a 3a

	A	B	C	D	
1		Y	Y	1	
2	Y	Y		2	
3	X		X	3	
4		X		4	

Oyuncu Y, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):1b 4b

```
  A   B   C   D
-----
1|   |   |   | Y | 1
-----
2| Y |   | Y |   | 2
-----
3| X |   |   | X | 3
-----
4|   | Y | X |   | 4
-----
  A   B   C   D
```

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4c 3c

```
  A   B   C   D
-----
1|   |   |   | Y | 1
-----
2| Y |   | Y |   | 2
-----
3| X |   | X | X | 3
-----
4|   | Y |   |   | 4
-----
  A   B   C   D
```

Oyuncu Y, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):4b 4a

X taşı 3A konumundan çıkarıldı.

```
  A   B   C   D
-----
1|   |   |   | Y | 1
-----
2| Y |   | Y |   | 2
-----
3|   |   | X | X | 3
-----
4| Y |   |   |   | 4
-----
  A   B   C   D
```

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):3d 4d

```
  A   B   C   D
-----
1|   |   |   | Y | 1
-----
2| Y |   | Y |   | 2
-----
3|   |   | X |   | 3
-----
4| Y |   |   | X | 4
-----
  A   B   C   D
```

Görüldüğü gibi taşlar birbirlerini köşelerde, satırlarda ve sütunlarda kıştırabiliyor ve kıştırılan taş oyundan atılmış oluyor. En sonda devam etmek isteniliyor mu diye sorgulanıyor e/E/h/H karakterlerinden biri girilmediyse tekrarı isteniyor ve h/H karakterleri girilirse oyun sonlanıyor. e/E girişi yapılır ise şöyle bir görünüm ortaya çıkar ve oyuna devam edilir.

```
X taşı 3C konumundan çıkarıldı.
  A  B  C  D
-----
1|   |   |   | Y | 1
-----
2| Y |   | Y |   | 2
-----
3|   |   |   |   | 3
-----
4|   |   | Y | X | 4
-----
  A  B  C  D

Oyuncu Y oyunu kazandı.
Oyun bitti..

Tekrar oynamak istiyor musunuz? (e/E/h/H):e
Oyun alanının satır/sütun sayısını giriniz(4-8):4
1. oyuncuyu temsil etmek için karakter giriniz:x
2. oyuncuyu temsil etmek için karakter giriniz:y
Birinci oyuncunun seçtiği karakter: X
İkinci oyuncunun seçtiği karakter: Y

  A  B  C  D
-----
1| Y | Y | Y | Y | 1
-----
2|   |   |   |   | 2
-----
3|   |   |   |   | 3
-----
4| X | X | X | X | 4
-----
  A  B  C  D

Oyuncu X, lütfen hareket ettirmek istediğiniz kendi taşınızın konumunu ve hedef konumunu giriniz (ör:1A 2A):|
```

Kaynakça:

- 1- <https://tr.depositphotos.com/549384476/stock-illustration-board-games-abstract-concept-vector.html?ref=5215893>