

Yusuf Emir ALTINKAYNAK Tetris Çalışması Analizi

Problem Tanımı: Tetris, birinci şahıs perspektifinden oynanan bir bulmaca oyunudur. Oyun, dört karelik bloklardan oluşan şekillerin, ekranın üst kısmından aşağı doğru düşerek bir yatay düzlemde sıralanmasına dayanır. Oyuncunun amacı, düşen blokları bir araya getirerek yatay düzlemlerde tam dolu satırlar oluşturmasıdır. Ben ise bu C kodunda kullanıcıdan alınan oyun tahtasının rastgele üretilen şekillerle uygun bir şekilde doldurulmasını, tam doldurulan her satırda satırların kırılarak puanların arttırılmasını ve en yüksek puanlarının tutulmasını sağlamaya çalıştım.

Programın genel akışını şu şekilde özetleyebiliriz:

- 1- Kullanıcıya ana menü seçenekleri sunulur.
- 2- Kullanıcı, oyuna başlamak veya çıkmak için bir seçenek seçer.
- 3- Oyuna başlandığında, kullanıcı tablonun boyutunu ve enini belirler.
- 4- Oyuncuya rastgele seçilen bir Tetris şekli gösterilir ve kullanıcıya bu şekli döndürme veya bırakma seçenekleri sunulur.
- 5- Kullanıcı, şekli istediği konumda bıraktığında, program bu şekli tabloya yerleştirir ve dolu satırları temizler.
- 6- Oyun devam ederken puan hesaplanır ve en yüksek skor güncellenir.
- 7- Oyunun bitip bitmediği kontrol edilir ve kullanıcıya ana menüye dönmek veya çıkmak isteyip istemediği sorulur.

Çözüme ve Algoritmaya ait detaylı rapor:

Chapter-1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #define SIZE 3
5 int main ()
6 {
7     // MADE BY YEA
8     srand(time(NULL));
9
10    char turn1,nick[50] = {" "},nickbest[50] = {" "};
11    int boy,en,i,j,r=0,turn=0,temp,kord,z=0,randIndex,row,flag,sayac,control=0,control2=0,yusuf1,yusuf,control3,control4,point=0,start,control5,choice,high=0,satir,sutun,sayac2;
12    char tempa[SIZE][SIZE], tempa2[SIZE][SIZE];
13
14 }
```

İlk olarak gerekli kütüphaneleri import ederek başladım. Rastgelelik için süreyi kullandıktan sonra değişkenlerimi oluşturarak kullanacağım dizileri oluşturdum.

Chapter-2

```
16 while (control5 == 0) { // Oyuna başlayıp bitirme
17
18     if (point >= high) { // high score kontrolü
19         high = point;
20         for (i = 0; i < 50; i++){
21             nickbest[i] = nick[i];
22         }
23     }
24     point = 0;
25
26     printf(" *****ANA MENU***** ");
27     printf("\nHigh Score: %s %d ",nickbest,high); // Ana menü
28     printf("\nAna Menuye hos geldiniz oyuna hos geldin baslamak için '1' yazın çıkmak için '2' yazınız: ");
29     scanf("%d",&start);
30 }
```

Tüm oyunumu bir döngü içerisine alarak birden fazla kez oynanmasını sağladım. Her oyun başı en yüksek skoru ve bu skoru yapan kişiyi güncelledim. Ve her oyun başındaki puan sayacı sıfırladım. En yüksek skoru yazdıran ve yeni oyuna başlayıp başlamak istemediğini soran bir ana menü tasarladım.

Chapter-3

```
32 if (start == 1){ // oyuna girip girmeme
33
34     printf("\nNickinizi yazınız: ");
35     scanf("%s",&nick);
36
37     printf("\nTablonuzun boy uzunlugunu giriniz: ");
38     scanf("%d", &boy);
39
40     printf("\nTablonuzun en uzunlugunu giriniz: ");
41     scanf("%d", &en);
42
43     char table[boy][en];
44
45     for (i = 0; i < boy; i++) { //tablyu oluşturma
46         for (j = 0; j < en; j++) {
47             table[i][j] = '.';
48         }
49     }
50
51     for (i = 0; i < boy; i++) { //tablyu yazdırma
52         for (j = 0; j < en; j++) {
53             printf("%4c", table[i][j]);
54         }
55         printf("\n");
56     }
57
58
59 }
```

Oyuna girdikten sonra oyuncudan nickini ve oynamak istediği oyun tablasının boyutlarını aldım. Bu tabloya uygun bir dizi oluşturarak elemanlarını '.' Olarak doldurdum ve tabloyu ekrana yazdırdım.

Chapter-4

```
60 r = 0;
61 while (r==0){ // oyunu başlatma
62
63     char tet[7][SIZE][SIZE] = {
64     {
65         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
66         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
67         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
68     },
69     {
70         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
71         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
72         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
73     },
74     {
75         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
76         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
77         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
78     },
79     {
80         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
81         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
82         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
83     },
84     {
85         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
86         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
87         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
88     },
89     {
90         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
91         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
92         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
93     },
94     {
95         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
96         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
97         {'x', 'x', 'x', 'x', 'x', 'x', 'x'},
98     }
99     };
100
101     randIndex = rand() % 7; // rastgele sayı üretme
102
103     printf("\nŞeklinizi:\n"); // şekli yazdırma
104     for(i=0; i<SIZE; i++){
105         for(j=0; j<SIZE; j++){
106             printf("%4c", tet[randIndex][i][j]);
107         }
```

Burada ise oyun içindeki her turu oluşturan bir döngü yaptım ve en üst satıra bir şekil gelene kadar dönmesini sağladım. Sonrasında ise her tur şekil matrislerini vererek şekillerin döndürülmüş halde kalmamasını sağladım. Ardından time.h kütüphanesi kullanarak rastgele bir sayı ürettim ve 3 boyutlu matrisimden o sayıya göre rastgele bir matris seçmesini sağladım ve for döngüsü içerisinde seçilen şekli ekrana yazdırdım.

Chapter-5

```
111 while (turn == 0) { // şekli döndürme
112     printf("\nŞeklinizi bir kez sağa döndürmek için r yazınız, döndürme işlemini bitirmek için k yazınız: ");
113     scanf(" %c", &turn1);
114
115     if (turn1 == 'r') {
116         for (i = 0; i < SIZE; i++) { // matris kopyalama işlemi
117             for (j = 0; j < SIZE; j++) {
118                 temparr[i][j] = tet[randIndex][i][j];
119             }
120         }
121         for (i = 0; i < SIZE; i++) { // 90 derece sağa döndürme işlemi
122             for (j = 0; j < SIZE; j++) {
123                 tet[randIndex][i][j] = temparr[SIZE - 1 - j][i];
124             }
125         }
126         printf("Score: %d\n", point);
127         printf("\nŞeklinizi:\n"); // şekli yazdırma
128         for(i=0; i<SIZE; i++){
129             for(j=0; j<SIZE; j++){
130                 printf("%4c", tet[randIndex][i][j]);
131             }
132         }
133     }
134     else if (turn1 == 'k') {
135         turn = 1;
136     }
137 }
138
139 }
140
141 }
```

Burada da kullanıcının istediği kadar şekli 90 derece sağa döndürmesini sağlayan ve istediği zaman sonlandırabileceği bir dizi işlem tanımladım ve yazdırdım.

Chapter-6

```
142 for ( i = 0; i < SIZE; i++) { // matris kopyalama işlemi
143     for ( j = 0; j < SIZE; j++) {
144         tetpar2[i][j] = tet[randIndex][i][j];
145     }
146 }
147
148 yusuf=0;
149 yusuf1=0;
150
151 while (yusuf<2) { // matrisin sütunlarını uygun hale getirme işlemi
152     control=0;
153
154     for (i = 0; i < 3; i++){
155         if (tet[randIndex][i][0] == ' '){
156             control++;
157         }
158     }
159
160     if (control==3){
161         for ( j = 0; j < 3; j++) {
162             tet[randIndex][j][2] = ' ';
163         }
164         for ( j = 0; j < 3; j++) {
165             tet[randIndex][j][0] = tetpar2[j][1];
166         }
167         for ( j = 0; j < 3; j++) {
168             tet[randIndex][j][1] = tetpar2[j][2];
169         }
170     }
171
172     for ( i = 0; i < SIZE; i++) { // matris kopyalama işlemi
173         for ( j = 0; j < SIZE; j++) {
174             tetpar2[i][j] = tet[randIndex][i][j];
175         }
176     }
177
178     yusuf1++;
179 }
180
181 }
182
183 while (yusuf<2) { // matrisin satırlarını yerleşime uygun hale getirme işlemi
184     control2=0;
185
186     for (i = 0; i < 3; i++){
187         if (tet[randIndex][2][i] == ' '){
188             control2++;
189         }
190     }
191
192     if (control2==3){
193         for ( j = 0; j < 3; j++) {
194             tet[randIndex][0][j] = ' ';
195         }
196         for ( j = 0; j < 3; j++) {
197             tet[randIndex][1][j] = tetpar2[0][j];
198         }
199         for ( j = 0; j < 3; j++) {
200             tet[randIndex][2][j] = tetpar2[1][j];
201         }
202     }
203
204 }
```

Burada ise 3x3 matrislerin döndürüldükten sonraki yerleştirme işleminde çıkan ve matrislerin yerleştirilirken istediğimiz yere değil de solunda ve aşağısında oluşan boşluklar yüzünden birkaç satır üste veya sağa yerleştiren bir sorunla karşılaştım. Bu sorunları çözmek için geçici matrisler oluşturdum. Sonrasında ise matrisleri 2 şer defa kontrol ettirdim ve altında veya solunda boşluk varsa matrisleri birer adım kaydırarak yeni oluşan kısımları boşluk ile doldurdum. Böylece matrislerin altında ve solunda boşluk kalmamasını sağladım. Bu sayede ne kadar döndürürsek döndürelim matrisler sonunda sol alt köşeye yerleşebiliyor. Sol alt köşeye yerleşen matrisler de tabloya yerleştirme işleminde hatasız çalışabiliyordu.

Chapter-7

```
216 turn=0,row=0,flag=0,turn1='w',sayac=0,control3 = 1;
217 printf("Kordinatları giriniz: ");
218 scanf("%d", &kord);
219 row=2,i=0,j=0;
220
221
222 for(i = SIZE; 0 < i; i--){ // şekil sayacı
223     for(j = SIZE; 0 < j; j--){
224         if (tet[randIndex][i-1][j-1] == 'x'){
225             sayac++;
226         }
227     }
228 }
229
230
231 while (control3 == 1){ // şeklin yerleşebileceği en uygun satırı bulma işlemi
232     flag = 0;
233     for(i = SIZE; 0 < i; i--){
234         for(j = SIZE; 0 < j; j--){
235             if (table[row+1-3][kord-2+j] == '.' && tet[randIndex][i-1][j-1] == 'x') {
236                 flag++;
237             }
238         }
239     }
240     if (sayac == flag && row != boy){
241         row++;
242     } else {
243         control3 = 0;
244     }
245     row--;
246 }
247
248
249 for(i = SIZE; 0 < i; i--){ // şekli yerleştirme işlemi
250     for(j = SIZE; 0 < j; j--){
251         if (table[row+1-3][kord-2+j] == '.' && tet[randIndex][i-1][j-1] == 'x') {
252             table[row+1-3][kord-2+j] = 'x';
253         }
254     }
255 }
256 printf("Score: %d\n",point);
257 for (i = 0; i < boy; i++){ // tabloyu yazdırma
258     for (j = 0; j < en; j++){
259         printf("%4c", table[i][j]);
260     }
261     printf("\n");
262 }
263
264 }
```

Sonrasında ise ilk olarak kullanıcıdan bir kordinat değeri istedim. Ondan sonra rastgele şekil matrisimde kaç parça kullandığımı sayan bir döngü oluşturdum ve bunu tuttum. Sonrasında flag yöntemini kullanarak üst tarafta saydığım sayaç ile üstten kontrol edip gelerek matrisin yerleşebileceği en uygun satırı buldum. Ondan sonra şekil ve tablomun yerlerini noktası noktasına kontrol ederek uygun yerin doldurulmasını sağladım. Sonrasında ise şekli yerleştirdiğim tabloyu yazdırdım.

Chapter-8

```
266 sayac2 = 0; // dolu satırları kırıp puan sayma işlemi
267 while (sayac2 < boy) {
268
269     for (i = 0; i < boy; i++){
270         control4 = 0;
271         if (table[i][0] == 'x'){
272             for (j = 0; j < en; j++){
273                 if (table[i][j] == 'x'){
274                     control4++;
275                     satir = i, sutun = en;
276                 }
277             } if (control4 == en){
278                 for (i = i; i > 0; i--) {
279                     for (j = 0; j < en; j++) {
280                         table[i][j] = table[i - 1][j];
281                     }
282                 }
283
284                 for (j = 0; j < en; j++) {
285                     table[0][j] = '.';
286                 }
287                 point++; // puanı arttırma işlemi
288
289                 printf("Score: %d\n",point);
290                 for (i = 0; i < boy; i++) { //tablyu yazdırma
291                     for (j = 0; j < en; j++) {
292                         printf("%4c", table[i][j]);
293                     }
294                     printf("\n");
295                 }
296             }
297         }
298     }
299     sayac2++;
300 }
301
302
303
304
305
306 z=0; // en üstte gelip bitiş kontrol etme işlemi
307 while (z < en){
308     if (table[0][z]=='x'){
309         r=1;
310     }
311     z++;
312 }
313
314
315 if (r == 1) { // en üstte şekil varsa bitirme işlemi
316
317     printf("\nOyun bitmiştir, sokorunuz: %d ana menüye donmek için 1 cikmak için 2 yazınız: ",point);
318     scanf ("%d", &choice );
319 }
320
321 }
322
```

Sonrasında ise dolu satırları kırmak için her seferinde tüm satırları kontrol eden bir döngü oluşturdum. Bu döngü içerisinde başka bir döngü oluşturarak ilk sütununda şekil olan satırları bulup satırların devamında kaç 'x' olduğunu sayan ve bunu tutan bir döngü oluşturdum. Bundan sonra tuttuğum 'x' sayacı eğer sütun sayısına eşitse yani satır full bir şekilde doluyorsa o satırı kırıp o satırın üstündeki satırları bir alt satıra kopyalayan bir döngü yaptım. Son olarak en üst satırı yine '.' ile doldurarak her kırıdığım satırı puan olarak tuttum. Ve her satır kırıldığında yeni tablomuzu yazdırarak karşılaştırma yapmamızı sağladım. Bu sayede birden fazla satırda da doluyorsa hepsini aynı anda kırıp eski ve yeni tablolar arasında karşılaştırma yaparak teker teker skorun arttığını görmemizi sağladım.

En altta ise en üst satırda bir blok var mı diye kontrol ettirdim ve varsa oyunu bitirmesini sağladım. Oyun bitince ise kullanıcıya ana menüye dönmek mi istediğini yoksa direkt çıkmak mı istediğini sorarak kullanıcıyı istediği zaman ayrılmasını sağladım.

Chapter-9

```
324 } else { // oyundan çıkma
325     control5 = 1;
326 }
327
328
329 if (choice == 2) {
330     control5 = 1;
331 }
332 }
333
334 printf ("\nGirdiginiz icin tesekkurler");
335
336 return 0;
337 }
```

Son olarak kullanıcıdan aldığım cevaba göre ana döngüden çıkan ve bitişinde bir mesaj göstermesini sağlayarak main fonksiyonumu 0 döndürerek oyunu bitirdim.

Test Videosu linki:

https://drive.google.com/file/d/1Tegl_hTjOnqB635C_CnBtL9L05XEBu0q/view?usp=sharing

22011057 // Yusuf Emir ALTINKAYNAK

Göksel Biricik