



INTRODUCTION TO COMPUTER SCIENCE

- SEMESTER PROJECT -

GROUP 1

Instructor : Mehmet Amaç Güvensan

İrem ATILGAN

17061036

İrem Atılgan

06.01.2018

OYUNA GENEL BAKIŞ

- 2048 Oyunu, özünde, toplama işlemleri üzerine kurulu bir oyundur. Oyundaki amacımız elimizdeki iki sayıyla başlayıp 2048'e ulaşana dek sayıları toplamaktır. Ancak oyun, 2048 sayısına ulaşmasak bile, hiçbir yan yana ve üst üste sayı toplanamayınca biter.
- Benim kodladığım oyunda, orijinal 2048 oyununda bulunan tüm özellikler mevcuttur; bununla birlikte farklılık katabilmek için eklediğim birkaç küçük özellik vardır.
- Ödevde istenildiği üzere,
 - Üretilen her yeni sayının başında "*" işareti vardır.
 - Oyunda istendiği zaman "R" tuşuna basılarak bir önceki hamleye gidilebilir.
 - Sayılardan biri 2048'e ulaştığında veya toplanabilecek sayı kalmadığında oyun biter.
 - Oyun sırasında "E" ye basıldığında oyun biter, "X" e basıldığında ise yeniden başlar.
 - Her el skor yenilenir ve maksimum skor yazar.
- Bunun haricinde ek olarak,
 - Oyun açıldığında ve oyun sırasında menüye girilebilir.
 - Menüden skor tablosuna veya kurallara bakılabilir, oyundan çıkılabilir ve oyuna devam edilebilir.
 - Oyun her bitişinde oyuncunun ismini ister ve kaydettiği skor ve isimleri skor tablosuna yazar.
 - Oyun sırasında her 10 ve üzeri sayıda skorumuz arttığında, bir hücreyi silme hakkı elde ederiz (Joker)

Oyunum ilk olarak iki parçaya ayrılıyor, bu kısımları ayrıntılarıyla açıklayacağım:

- **OYUN**
- **MENÜ**

OYUN

		4	
	*4		

SCORE = 0

MAX SCORE = 0

JOKER = 0

YENIDEN BASLAT = X

GERI DON = R

OYUNU BITIR = E

Şekil 1 : Oyun, iki sayının rastgele bir hücreye 4 ile 2 arasında bir değer verilmesiyle başlar

- Oyunda kullandığım tabloyu yazdırmak için print_tablo adında yeni bir fonksiyon oluşturdum
- Tabloda yeni üretilen sayıların konumu a (satır) ve b (sütun) ile ifade ediliyor. Üretilen her yeni sayının başına "*" getirebilmek için döngü sırasında m ve n'nin, a ve b'ye eşit olup olmadığına bakıyor, ona göre sayıyı yazdırıyorum. Eğer öyle değilse ve sıfırdan farklıysa, sayıyı normal şekilde yazdırıyor, sayı sıfırsa yazdırmıyorum.


```

int main()
{
    system("color 06");
    int tablo[4][4], eskitablo[4][4], lastscore[50];
    int i,j,tmp,score,maxscore,a,b,cevap,sign,count,sum,poprandom,key,oldscore,joker,jsatir,jsutun;
    char ch;//oyun içindeki hareketlerimizin algılanması içi ch değişkeni oluşturdum, yapılan hareketlerin değeri buraya gidecek
    char isim_listesi[50][50];//oyun bittiğinde isimleri almamız için oluşturduğumuz matris
    j = 0;
    tmp = 0;//sürüklemeler sırasında geçici olarak j'yi atayacağımız değişken (ileride daha ayrıntılı açıklayacağım)
    sum = 0;score = 0;maxscore = 0;
    count = 0;//oyunu bitiren oyuncuları saymak için count değişkeni oluşturdum
    sign = 0;//oyunun devam edip etmeyeceğini belirlemek için sign değişkeni
    poprandom = 0;//hangi durumlarda random sayı üretileceğini belirlemek için poprandom
    joker = 0;
    srand(time(NULL));//rastgele sayı üretebilmesi ve bu rastgele sayının sürekli değişebilmesi için

```

Şekil 4 : Değişkenleri tanımladığımız bölüm

```

for(i = 0; i < 50; i++)
{
    lastscore[i] = 0;//skor listesinin tamamına 0 değeri veriyorum
}

for(i = 0; i < 4; i++)
{
    for(j = 0; j < 4; j++)
    {
        tablo[i][j] = 0;//tabloyu sıfırlıyorum
    }
}

for(i = 0; i < 2; i++)
{
    //rastgele satır ve sütunlar belirliyorum
    a = rand() % 4;
    b = rand() % 4;

    //belirlenen satır ve sütundaki sayı sıfırdan farklı olana dek a ve b yi değiştiriyorum
    while(tablo[a][b] != 0)
    {
        a = rand() % 4;
        b = rand() % 4;
    }
    //tablodaki hücreye oyunun kurallarına göre ya 4 ya da 2 değeri atanıyor
    tablo[a][b] = rand() % 3 + 2;//2 ile 4 arasında rastgele sayı üretmesi için
    while(tablo[a][b] % 2 != 0)//rastgele atanan değer 3 olmaması için
    {
        tablo[a][b] = rand() % 3 + 2;
    }
}

```

Şekil 3 : Rastgele Sayı Ürettiğimiz Kısım

//OYUN BÖLÜMÜ

```
while(sign == 0)//OYUN sign 1 olana dek devam edecek
{
    if(score > maxscore) maxscore = score;

    poprandom = 0;//her el random sayı üretip üretemeyeceğimizi anlamak için kullandığımız poprandom'u sıfırlıyorum

    print_tablo(tablo,score,maxscore,a,b,joker);

    ch = getch();
    if(ch == -32 || ch == 0)//Bazı bilgisayarlarda ilk değer sıfır olabiliyor
    {
        //-32'den sonra gelen kodu çekiyoruz
        ch = getch();
    }
    if(ch == 'r')//oyuncu r'ye bastığında eski hamleye geri dönebilmek için
    {
        for(i = 0; i < 4; i++)
        {
            for(j = 0; j < 4; j++)
            {
                tablo[i][j] = eskitablo[i][j];//kaydettiğimiz eski tabloyu, tabloma atıyorum
            }
        }
        if(score != oldscore)//geçen el skorda bir değişim olduysa
        {
            score = oldscore;//eski skora geri dön
            joker--;//kazandığımız joker hakkını sil
        }
    }
}
```

Şekil 5 : Oyun Kodlarına Giriş

Bu kısımda oyunun devamlılığının sign'a bağlı olduğunu belirterek başlıyoruz. Sign'ı:

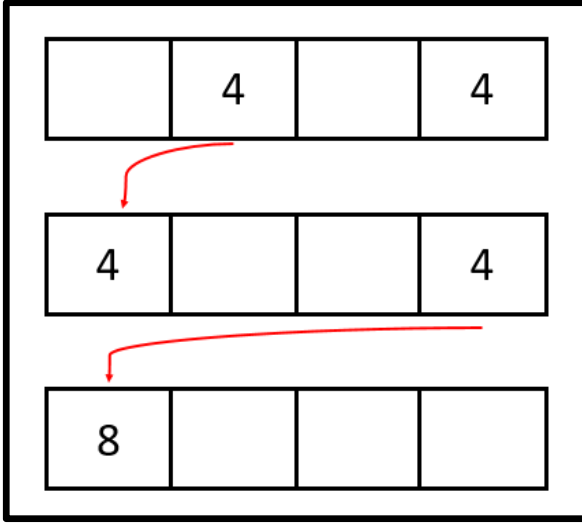
- Oyunu bitirmek istediğimizde,
- Tablodaki tüm hücreler dolduğunda ve toplanabilir sayı kalmadığında ve
- Tablodaki sayılardan biri 2048'e ulaştığında 1 yapıyoruz (sign = 1).

“poprandom” değişkenini ise, o el hareket ettiysek veya iki sayıyı topladıysak değerini 1 yapıyoruz. Bu değişken sayesinde yeni bir sayı üretilip üretilmemesi gerektiğini anlayabiliyoruz.

8			
2	*4		

Şekil 6 : poprandom'un değerinin sıfır olması gerektiği durumlar

1. OYUNUN MANTIĞI



Şekil 8 : Sola kaydırma örneği

Herhangi bir yöne sayıları sürükleyebilmek için nasıl bir yol izlediğimi Şekil 2'deki sola kaydırma örneğinden yola çıkarak açıklayacağım :

- Her yeni satırda sayıları, soldan başlayarak, sağa doğru tarıyoruz. Bulduğumuz sıfırdan farklı ilk sayı ise duruyoruz.
- Bundan sonra yapacağımız şey, durduğumuz hücrenin solunda kalan hücreleri sıfırdan farklı değeri olana dek kontrol etmek. Bunu sağlayabilmek için j'nin değerini azaltmak yerine tmp adlı bir değişkene j değerini atıyor ve onu kullanarak solda kalan hücreleri kontrol ediyorum. Bu sayede işim bittiğinde j, kaldığı yerden devam edebiliyor.

```
if(ch == left)//sola kaydırmak istiyorsak
{
    for(i = 0; i < 4; i++)
    {
        sum = 0;//kaydırma sırasında, daha önce toplama yapılıp yapılmadığını tespit etmek için "sum" değişkenini her satırda 0 yapıyoruz
        for(j = 1; j < 4; j++)
        {
            if(tablo[i][j] != 0)
            {
                //hücre sıfırdan farklı olursa j'de oynama yapamayacağımızdan tmp adlı değişkene j'nin değerini atıyorum
                tmp = j;

                //geldiğimiz hücrenin gerisindeki hücre 0'dan farklı olana dek geri gidiyorum
                while((tablo[i][tmp-1] == 0) && (tmp >= 1))
                {
                    tmp--;
                }
                if(tmp != j) poprandom = 1; /*tmp'de değişiklik olmuşsa hücremizde hareketlilik olacaktır,
                bu da yeni sayı üretebilmemiz için yeterlidir*/

                if(tmp == 0)//gerideki hücrelerin tamamı sıfır ise
                {
                    tablo[i][0] = tablo[i][j]; //ilk hücreye kaldığımız hücrenin değerini ata
                    tablo[i][j] = 0; //kaldığımız hücrenin değerini sıfırla
                }
                else if(tablo[i][tmp-1] == tablo[i][j])
                {
                    if(tmp == j) poprandom = 1; //sayılar birbirine eşitse ve hücremizin tam arkasındaki hücreyse yeni sayı üretilebilir
                }
            }
        }
    }
}
```

Şekil 9 : Sola Kaydırma Kodları

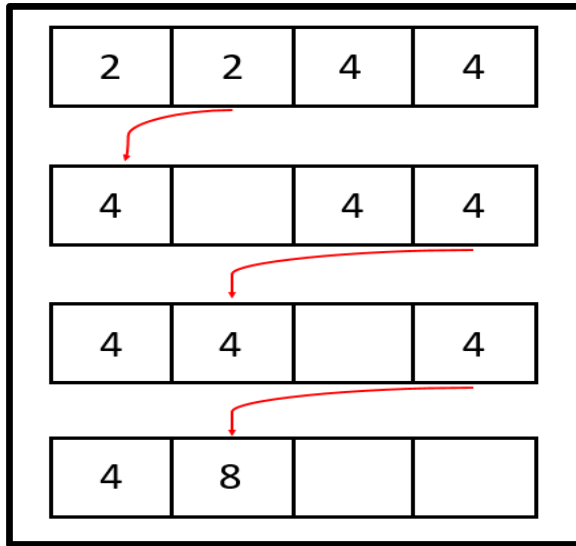
Şekil IX'da görüldüğü gibi tmp, solundaki sayı sıfırdan farklı oluncaya dek bir azalıyor yani sola gitmeye devam ediyor. Eğer solundaki tüm hücreler sıfırsa, en soldaki hücreye taşınıyor. Fakat karşısına sıfırdan farklı değeri olan bir hücre gelirse sayıya bağlı olarak 2 yol izliyor. Hücre,

- Bulunduğumuz hücreye eşitse
- Bulunduğumuz hücreden farklıysa

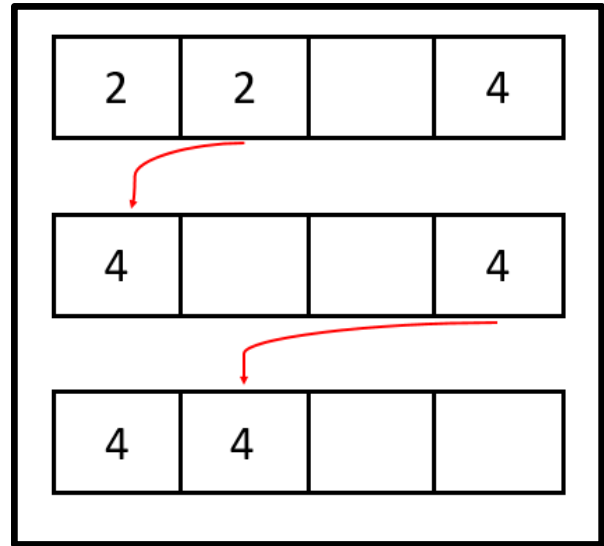
Bulunduğumuz hücreye eşitse, toplamadan önce toplamaya izni olup olmadığına bakıyor ve bunun için sum değişkenini kullanıyor (sum değişkenini daha sonra açıklayacağım). $sum = 1$ ise toplamıyor ve ulaştığımız hücrenin hemen sağındaki hücreye yerleşiyor; $sum = 0$ ise sayıları topluyor.

Bulunduğumuz hücreye eşit değilse, hücremiz, ulaştığımız sayının hemen sağındaki hücreye yerleşiyor.

SUM :



Şekil 10 : sum'ın Kullanılma Nedeni



Şekil 11 : sum'ın Kullanılma Nedeni 2

Şekil X ve XI'deki durumlarla karşılaşıldığında oyun verilen metodu izliyor. Eğer karşılaşılan her iki eşit sayıyı toplasaydık yukarıdaki durumu elde edemezdik. Bu yüzden sum değişkeni oluşturdum ve oyunda yapılan ilk toplama işleminden sonra sum'ı 1 yaparak birdahaki iki eşit sayının toplanmasına izin vermedim. Ancak Şekil X'daki gibi toplama işlemine izin verilmediyse, daha sonraki toplamanın gerçekleşebilmesi için sum 1 ise 0 yapıyoruz.

```

//iki sayı eşit olabilir ancak kendilerinden önce başka iki sayı toplanmışsa o sayıları toplayamayız
//bunu anlayabilmek için sum değişkeni oluşturdum

if(sum == 1)//daha önce toplama yapılmışsa
{
    tablo[i][tmp] = tablo[i][j]; //değeri eşit olunan hücreden bir sonraki hücreye yerleş
    tablo[i][j] = 0; //bulunduğumuz hücreyi sıfırla
    sum = 0;
}
else if(sum == 0)//toplama yapılmamışsa
{
    tablo[i][tmp-1] = tablo[i][j] + tablo[i][j];
    score = score + tablo[i][tmp-1];
    tablo[i][j] = 0;
    sum = 1; //sum'ı toplama yapıldığına dair işaretliyorum
}
}

```

Şekil 12 : sum Mantığı (Sola Kaydırma)

			2
		*2	2
		2	4

		4	4
			4
	*4		

Şekil 13 : Oyun İçerisinde Yukarı Kaydırma Sırasında Uygulanan Sum Örneği

```

//ulaştığımız hücrenin değeri bulunduğumuz hücredekenden farklıysa ve hemen gerimizdeki hücre değilse
else if(tablo[i][tmp-1] != tablo[i][j] && (tmp != j))
{
    tablo[i][tmp] = tablo[i][j]; //ulaştığımız hücreden sonraki boş hücreye yerleş
    sum = 0;
    tablo[i][j] = 0; //kaldığımız hücreyi sıfırla
}

```

Şekil 14 : Sola Kaydırma Son Bölümü

Sola kaydırma sırasında kullandığım mantık tmp, i ve j'lerin haricinde diğer yönlerde de aynıdır. Bu nedenle onların sadece source kodunu paylaşarak devam edeceğim.

```

else if(ch == right) //saga kaydirdigimizda (Aynı Yöntemi Uyguladım)
{
    for(i = 0; i < 4; i++)
    {
        sum = 0;

        for(j = 3; j >= 1; j--)
        {
            if(tablo[i][j-1] != 0) //bulundugumuz hucredeki eleman sifirdan farkli olana dek geri git
            {
                tmp = j;

                while((tmp <= 3) && (tablo[i][tmp] == 0))
                {
                    tmp++;
                }

                if(tmp != j) poprandom = 1; //bir gerimizde olmayan hucre sifirdan farkliysa yeni sayi uretilebilir
                //bu kosulun amaci, sayimizin hareket edip etmeyecegini anlamaktir

                //sayimiz en saga kadar gidebiliyorsa
                if(tmp == 4)
                {
                    tablo[i][3] = tablo[i][j-1];
                    tablo[i][j-1] = 0;
                }
                else //ulastigimiz sayi hucredeki sayidan farkliysa ya da hucreye esitse
                {
                    if(sum == 1 || (tablo[i][tmp] != tablo[i][j-1])) //iki hucreyi toplamamaliysak veya hucreler birbirine esit degilse
                    {
                        sum = 0;
                        tablo[i][tmp-1] = tablo[i][j-1];
                        if(tmp != j) //hucre hareket etmisse yani sifirdan farkli olan hemen bir sagimizdaki hucre degilse
                        {
                            tablo[i][j-1] = 0;
                        }
                    }
                    else //hucreler birbirine esitse ve toplanabilirse
                    {
                        if(tmp == j) poprandom = 1;
                        tablo[i][tmp] = tablo[i][j-1] * 2;
                        score = score + tablo[i][tmp];
                        tablo[i][j-1] = 0;
                        sum = 1; //bundan sonra esit olan hucreler toplanamayacak
                    }
                }
            }
        }
    }
}

```

Şekil 15 : Sağa Kaydırma

```

else if(ch == up)//yukari kaydirdigimizda
{
    for(j = 0; j < 4; j++)
    {
        sum = 0;

        for(i = 1; i < 4; i++)
        {
            if(tablo[i][j] != 0)//yukaridan asagiya dogru sifirdan farkli olan ilk hucreyi bulalim
            {
                tmp = i;

                //tmp en asagidaki hucreye erisinceye dek veya sifirdan farkli, bir sayiyla karsilasana dek asagi inecek
                while(tmp >= 1 && (tablo[tmp-1][j] == 0))
                {
                    tmp--;
                }
                if(tmp != i) {poprandom = 1;}//sayinin hareket edip edemeyecegine gore rastgele sayi uretecegiz

                if(tmp == 0)
                {
                    tablo[0][j] = tablo[i][j];
                    tablo[i][j] = 0;
                }
                else
                {
                    if(sum == 1 || (tablo[tmp-1][j] != tablo[i][j]))//buldugumuz hucre, hucremize esit degilse veya toplama iznimiz yoksa
                    {
                        sum = 0;
                        tablo[tmp][j] = tablo[i][j];//buldugumuz hucrenin bir altindaki hucreye yerles
                        if(tmp != i)//hucre hareket etmisse yani sifirdan farkli olan hemen bir ustumuzdeki hucre degilse
                        {tablo[i][j] = 0;}//bulundugumuz hucreyi sifirla
                    }
                    else
                    {
                        if(tmp == i) {poprandom = 1;}//toplama islemi yapildiyse yeni sayi uretilebilir
                        tablo[tmp-1][j] = tablo[i][j] * 2;
                        score = score + tablo[tmp-1][j];
                        tablo[i][j] = 0;
                        sum = 1;//bundan sonraki hucreler esitse toplama islemine izin verme
                    }
                }
            }
        }
    }
}
}
}

```

Şekil 16 : Yukarı Kaydırma

```

else if(ch == down)//asagi kaydirdigimizda
{
    for(j = 0; j < 4; j++)
    {
        sum = 0;

        for(i = 3; i >= 0; i--)
        {
            if(tablo[i][j] != 0 && i != 3)//asagidan yukariya giderken, ustumuzdeki bir hucre sifirdan farkliysa
            {
                tmp = i;

                while(tmp < 4 && (tablo[tmp+1][j] == 0))//hucre sifir olmayana dek veya en uste ulasana dek yukari cik
                {
                    tmp++;
                }
                if(tmp != i) poprandom = 1;//hareket edilmisse rastgele sayi uret

                if(tmp == 4)//en ustteki hucreye kadar, tum hucreler sifirsa
                {
                    tablo[3][j] = tablo[i][j];//en ustteki hucreye yerles
                    tablo[i][j] = 0;
                }
            }
            else
            {
                if(sum == 1 || (tablo[tmp+1][j] != tablo[i][j]))//buldugumuz hucre, hucremize esit degilse veya toplama iznimiz yoksa
                {
                    sum = 0;
                    tablo[tmp][j] = tablo[i][j];
                    if(tmp != i)//hucre hareket etmisse yani sifirdan farkli olan hemen bir altimizdaki hucre degilse
                    {tablo[i][j] = 0;}
                }
                else
                {
                    if(tmp == i) poprandom = 1;//sayilar birbirine esitse ve toplanabiliyorsa yeni sayi uretilebilir
                    tablo[tmp+1][j] = tablo[i][j] * 2;
                    score = score + tablo[tmp+1][j];
                    tablo[i][j] = 0;
                    sum = 1;//bundan sonraki toplamaya izin yok
                }
            }
        }
    }
}
}
}

```

Şekil 17 : Aşağı Kaydırma

2. OYUNDAN ÇIKIŞ, OYUNU YENİDEN BAŞLATMA VE OYUN SONU

Oyun sırasında, daha önce de söylediğim üzere, X tuşuna basıldığında yeniden başlatabiliyor ve E tuşuna bastığında oyunu bitirebiliyor.

```
else if(ch == 'x' || ch == 'X')//Oyunu yeniden başlatmak istersek
{
    for(i = 0; i < 4; i++)
    {
        for(j = 0; j < 4; j++)
        {
            tablo[i][j] = 0;//tablodaki tüm değerleri sıfırla
            eskitablo[i][j] = 0;
        }
    }
    //yeniden random sayı üret
    a = rand() % 4;
    b = rand() % 4;

    while(tablo[a][b] != 0)
    {
        a = rand() % 4;
        b = rand() % 4;
    }
    joker = 0;
    oldscore = 0;
    //skoru sıfırla
    score = 0;

    tablo[a][b] = rand() % 3 + 2;
    while(tablo[a][b] % 2 != 0)
    {
        tablo[a][b] = rand() % 3 + 2;
    }
}
else if(ch == 'e' || ch == 'E')//oyunu bitirmek istersek
{
    sign = 1;
}
```

Şekil 18 : Oyunu Yeniden Başlatma ve Oyunu Bitirme Kodları

Oyunun sonlanıp sonlanmayacağını anlamak için iki kontrol yapıyoruz :

- 1- Herhangi bir sayı 2048'e ulaştı mı?
- 2- Tablo tamamen doldu mu? Dolduysa hala toplanabilecek sayılar kaldı mı?

```

i = 0;
while(i < 4 && tablo[i][j] != 2048)//tablodaki elemanlardan biri 2048 mi diye kontrol ediyoruz
{
    j = 0;
    while(j < 4 && tablo[i][j] != 2048 )
    {
        {j++;}
        if(j == 4)
        {
            i++;
        }
    }
}
if(tablo[i][j] == 2048) sign = 1;//2048 olursa oyunu bitir

```

Şekil 20 : Tablodaki Herhangi Bir Sayı 2048'e Ulaştı mı ?

//sayılardan biri 2048 ise tabloyu kontrol etmemize gerek yok

```

if(sign == 0){
    i = 0; j = 4;
    while(i < 4 && j == 4)//tablodaki tüm elemanlar 0'dan farklı mı diye kontrol ediyoruz
    {
        j = 0;
        while(j < 4 && tablo[i][j] != 0 )
        {
            {j++;}
            i++;
        }
    }

    if(j == 4) //j = 4 ise tablodaki tüm elemanlar sıfırdan farklıdır
    {
        i = 0; j = 3;
        while(i < 4 && j == 3)//yataydaki elemanların toplanıp toplanamayacağını kontrol ediyoruz
        {
            j = 0;
            while(j < 3 && tablo[i][j] != tablo[i][j+1])//tablodaki arka arkaya gelen iki sayı esit olmadığı surece dongu devam eder
            {
                {j++;}
                i++;
            }
        }

        if(j == 3)//yataydaki elemanlar toplanamıyorsa dikeydeki elemanların toplanabilirliğine bakıyoruz
        {
            j = 0; i = 3;
            while(j < 4 && i == 3)
            {
                i = 0;
                while(i < 3 && tablo[i][j] != tablo[i+1][j])//dikeyde ust uste elemanlar birbirine esit olmadığı surece
                {
                    {i++;}
                    j++;
                }
            }

            if(j == 3)//yataydaki elemanlar toplanamıyorsa dikeydeki elemanların toplanabilirliğine bakıyoruz
            {
                j = 0; i = 3;
                while(j < 4 && i == 3)
                {
                    i = 0;
                    while(i < 3 && tablo[i][j] != tablo[i+1][j])//dikeyde ust uste elemanlar birbirine esit olmadığı surece
                    {
                        {i++;}
                        j++;
                    }
                }

                if( i == 3 )//dikeydeki elemanlar da toplanamıyorsa oyunun bittigini belirtiyoruz
                {
                    sign = 1;//bu yuzden sign'ı sıfırdan bire cevriyoruz
                }
                else
                    system("CLS");
            }
        }
        else
            system("CLS");
    }
}

```

Şekil 19 : Tabloyu Kontrol Ediyoruz

Her el sonunda yeni bir random sayı üretip üretemeyeceğimizi kontrol ediyorum :

```
else if((ch != 'r' || ch != 'R') && poprandom == 1)//eski tabloya geri donmediyse ve random sayi uretmemize izin verildiyse
{
    a = rand() % 4;
    b = rand() % 4;

    while(tablo[a][b] != 0)
    {
        a = rand() % 4;
        b = rand() % 4;
    }

    tablo[a][b] = rand() % 3 + 2;
    while(tablo[a][b] % 2 != 0)
    {
        tablo[a][b] = rand() % 3 + 2;
    }

    poprandom = 0;

    system("CLS");
}
else//sayi uretemiyorsak ya da R'ye basilmisla
    system("CLS");
```

Şekil 21 : Random Sayı Üretme

Oyunun bittiğini anlamak için sign'ın 1 olup olmadığına bakıyoruz. Bununla birlikte menü üzerinden çıkmak istediğimizde scoreboard'a isim vermek zorunda kalmamak için while'ın içine **key != 0** yazdım.

```
if(sign == 1 && key != 0)//oyun bittiyse
{
    system("CLS");

    print_tablo(tablo,score,maxscore,a,b,joker);//tablonun son halini ve skorlari yazdir

    printf("\n\t\t===== GAME OVER =====\n\n");

    printf("\t\tLutfen isminizi giriniz : ");scanf("%s",isim_listesi[count]);//oynayan kisinin ismini alip listenin ilk satirina kaydediyoruz

    lastscore[count] = score;//oynayan kisinin skorunu ayri bir diziye kaydediyoruz
    count++;//oyun yeniden oynanmak isterse listenin 2. elemanina geciyoruz

    printf("\t\tOynayanlarin skorlarini gormek isterseniz 1'i\n\t\t");
    printf("Oyundan cikmak isterseniz 2'yi\n\t\tOyunu yeniden baslatmak isterseniz 3'u tuslayiniz\n\n");
    scanf("%d",&cevap);
    if(cevap == 1)//scoreboard'u aciyoruz
    {
        //printf("\n\n")
        system("CLS");

        print_liste(isim_listesi,lastscore,count);//scoreboard' yazdirma fonksiyonu

        printf("\n\n\t\tOYUNA DONUS = X\n\t\tOYUNDAN CIK = E\n");
```

Oyun bittikten sonra isim_listesi matrisimize oyuncunun ismini ve skorunu alıyoruz. Ardından oyuncu isterse oyunu yeniden başlatabiliyor, scoreboard'a bakabiliyor ya da oyundan çıkabiliyoruz.


```

ch = getch();
if(ch == -32)
{
    ch = getch();
}

if(ch == 'x')//scoreboard uzerinden oyunu yeniden baslatmak istersek
{
    system("CLS");

    for(i = 0; i < 4; i++)//tum elemanlari sifirliyoruz
    {
        for(j = 0; j < 4; j++)
        {
            tablo[i][j] = 0;
            eskitablo[i][j] = 0;
        }
    }
    //skoru ve sign'i sifirliyoruz ki oyun devam edebilsin
    score = 0;
    oldscore = 0;
    sign = 0;
    joker = 0;
    //baslangic olarak yine bos iki yere rastgele 2 ya da 4 sayisi atiyoruz
    for(i = 0; i < 2; i++)
    {
        a = rand() % 4;
        b = rand() % 4;

        while(tablo[a][b] != 0)
        {
            a = rand() % 4;
            b = rand() % 4;
        }

        tablo[a][b] = rand() % 3 + 2;
        while(tablo[a][b] % 2 != 0)
        {
            tablo[a][b] = rand() % 3 + 2;
        }
    }
}
}

```

```

else if(cevap == 3)//oyun bittikten sonra direkt yeniden baslatmak istersek
{
    for(i = 0; i < 4; i++)
    {
        for(j = 0; j < 4; j++)
        {
            tablo[i][j] = 0;
            eskitablo[i][j] = 0;
        }
    }
    joker = 0;
    oldscore = 0;
    score = 0;
    sign = 0;
    for(i = 0; i < 2; i++)
    {
        a = rand() % 4;
        b = rand() % 4;

        while(tablo[a][b] != 0)
        {
            a = rand() % 4;
            b = rand() % 4;
        }

        tablo[a][b] = rand() % 3 + 2;
        while(tablo[a][b] % 2 != 0)
        {
            tablo[a][b] = rand() % 3 + 2;
        }
    }
}
else
    system("CLS");
}

if(score != oldscore) joker++;//skorda deęişim olmuřsa joker'i arttır

```

Şekil 22 : Kodun Sonu, Oyun sign'a Bağlı Olarak Devam Eder

```

//SCOREBOARD'DA KULLANILACAK ISIM VE SKOR LİSTESİNİ GOSTEREN FONKSİYONU YAZDIM
void print_liste(char (*isim_listesi)[50], int *lastscore,int count)
{
    int i,j;
    system("CLS");//ekrani temizlemek icin kullandim
    i = 0; j = 0;

    printf("\n\n\t\t=====\\n");
    printf("\t\t=====SCOREBOARD=====\\n");
    printf("\t\t=====\\n");
    while(i < count)//kaydettigimiz skorlari yazdirmak icin
    {
        printf("\t\t");
        printf("%s",isim_listesi[i]);//isim_listesi matrisinin her satirini, yani isimleri yazdiriyoruz
        printf("\t%d",lastscore[i]);//her oynayanin ismiyle beraber ulastigi skoru da yazdiriyoruz
        printf("\\n");
        printf("\t\t-----\\n");

        i++;
    }
}

```

Şekil 23 : Scoreboard Yazdıran Fonksiyon

8	4	16	4
4	16	4	
	4		
*2			

SCORE = 104

MAX SCORE = 104

JOKER = 3

YENIDEN BASLAT = X

GERI DON = R

OYUNU BITIR = E

===== GAME OVER =====

Lutfen isminizi giriniz : irem

4	32	8	2
16	4	16	4
4	32	4	2
8	*4	8	4

SCORE = 308

MAX SCORE = 308

JOKER = 19

YENIDEN BASLAT = X

GERI DON = R

OYUNU BITIR = E

===== GAME OVER =====

Lutfen isminizi giriniz : _

Şekil 26 : Oyunu Bitirmek için E'ye Bastığımızda

Şekil 25 : Tablodaki Tüm Hücreler Doluysa ve Sayılar Toplanamıyorsa

2048			
2		*4	
4			

SCORE = 2048

MAX SCORE = 0

JOKER = 0

YENIDEN BASLAT = X

GERI DON = R

OYUNU BITIR = E

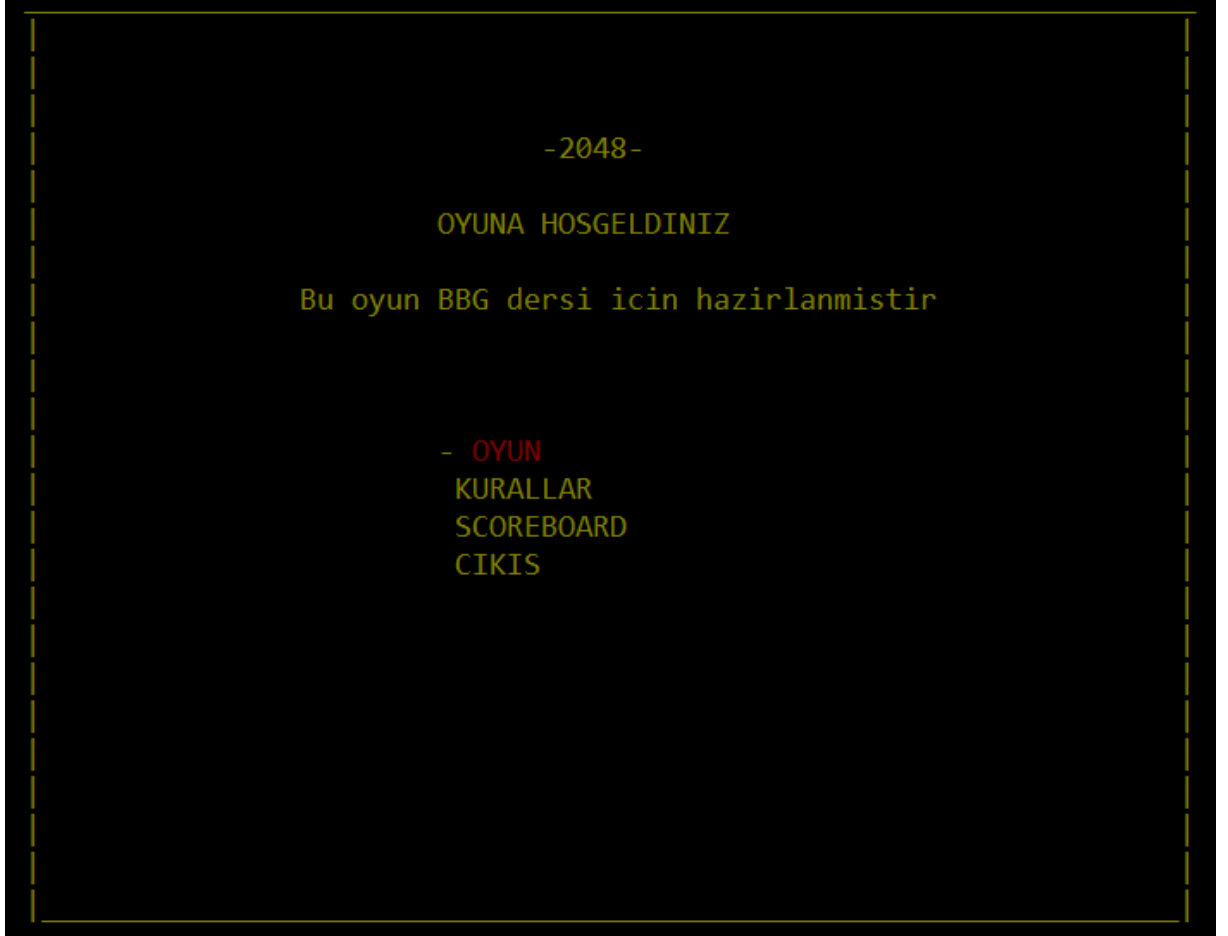
===== GAME OVER =====

Lutfen isminizi giriniz :

Şekil 24 : Sayılardan biri 2048'e ulaştıysa

MENÜ

Menü, yine orijinal oyunda var olmayan seçeneklerden biriydi. Oyunu açtığınızda karşınıza çıkar ve oyun esnasında ESC tuşuna bastığınızda açılır.



Şekil 27 : Menü - Menü üzerinden kurallara ve scoreboard'a erişebilir ya da oyundan çıkabilirsiniz.

[illegible]

[illegible]

Şekil 28 : Menü Fonksiyonu

Şekil 28'deki menü fonksiyonunda, key'in her haline göre bir kare gösterilir. Bu sayede menü üzerine hangi seçeneğe geldiğimizi görebiliriz.

//MENÜ BÖLÜMÜ

```
key = 3; //en ust secenekten baslasin
```

```
print_menu(ch,key);//menüyü yazdır
```

```
ch = getch();
if(ch == -32 || ch == 0)//ilk taramada -32(her bilgisayarda farklı olabilir), ikincide ise istediğim değeri alabildiğim için if yerleştirdim
{
    ch = getch();
}

/*
KEY = 3 -> OYUNA GEÇ
KEY = 2 -> KURALLARA BAK
KEY = 1 -> SCOREBOARD'A BAK
KEY = 0 -> OYUNDAN ÇIK
*/
```

Şekil 29 : Key'lerin Temsil Ettiği Seçenekler

```

while(ch != enter || (key != 0 && key != 3))//ÇIKIŞ ve OYUN seçeneğine gelip ENTER'a basmadığı sürece menüde kal
{
    ch = getch();
    if(ch == -32)
    {
        ch = getch();
    }

    if(ch == up && key < 3)//yukarı ok tuşuna bastığımızda, en üst seçenekte değilsek key'i bir arttırıyorum
    {
        key = key + 1;//bu sayede diğer seçeneklere geçebiliyoruz
    }
    else if(ch == down && key > 0)//aşağı ok tuşuna bastığımızda, en aşağıdaki seçeneğe gelmediyse
    {
        key = key - 1;//key'i bir azaltıyorum
    }

    print_menu(ch,key);

    if(ch == enter)//ENTER'a basarsak
    {
        //OYUN KURALLARI seçeneğine girersek
        if(key == 2)
        {
            system("CLS");

            print_rules();//kuralları yazdır

            while(ch != 'r' && ch != 'R')//menüye geri dönmek için 'r' ye basmadığımız sürece bu seçenekte kal
            {
                ch = getch();
                if(ch == -32)
                {
                    ch = getch();
                }
            }
        }
        //SCOREBOARD'A BAKMAK İSTERSEK
        else if(key == 1)
        {
            system("CLS");
            print_liste(isim_listesi,lastscore,count);
            printf("\n\n\t\tMENUYE DON = R\n");
            while(ch != 'r' && ch != 'R')
            {
                ch = getch();
                if(ch == -32)
                {
                    ch = getch();
                }
            }
        }
        //OYUNDAN ÇIKMAK İSTERSEK
        else if(key == 0)
        {
            sign = 1;
        }
        //OYUNA GİRMEK İSTERSEK
        else
        {
            system("CLS");
        }
    }
}
system("CLS");

```

Şekil 30 : Oyundan önce açılan menünün source kodu

```

//oyun içerisinde menüye girmek istersek
if(ch == esc)
{
    //Menüde kullandığımız kodlar

    print_menu(ch,key);

    while(ch != enter || (key != 0 && key != 3))
    {
        ch = getch();
        if(ch == -32)
        {
            ch = getch();
        }

        if(ch == up && key < 3)
        {
            key = key + 1;
        }
        else if(ch == down && key > 0)
        {
            key = key - 1;
        }
        print_menu(ch,key);

        if(ch == enter)
        {
            if(key == 2)
            {
                system("CLS");
                print_rules();

                while(ch != 'r' && ch != 'R')
                {
                    ch = getch();
                    if(ch == -32)
                    {
                        ch = getch();
                    }
                }
            }
            else if(key == 1)
            {
                system("CLS");
                print_liste(isim_listesi,lastscore,count);
                printf("\n\n\t\tMENUYE DON = R\n");
                while(ch != 'r' && ch != 'R')
                {
                    ch = getch();
                    if(ch == -32)
                    {
                        ch = getch();
                    }
                }
            }
            else if(key == 0)
            {
                sign = 1;
            }
            else
            {
                system("CLS");
            }
        }
    }
}

```

Şekil 31 : Oyun esnasında menüye girdiğimizde kullanılan source kod


```
//KURALLARI YAZDIRAN FONKSİYON
void print_rules()
{
    printf("\n\n");
    printf("\t\t");printf("=====");printf("\n");
    printf("\n\t\t Oyunun Kurallari :\n\t\t -Saga kaydirmek icin SAG OK TUSUNA..\n");
    printf("\t\t -Sola kaydirmek icin SOL OK TUSUNA..");
    printf("\n\t\t -Yukari kaydirmek icin YUKARI OK TUSUNA\n\t\t -Asagi kaydirmek icin ASAGI OK TUSUNA BASMALISINIZ... ");
    printf("\n\t\t -Oyun esnasinda ESC'ye bastiginizda menu acilir..");
    printf("\n\t\t -Oyun icerisinde yaptiginiz her toplama islemi size 1 JOKER kazandirir");
    printf("\n\t\t -Joker'e 'J' tusuna basarak erisebilir ve istediginiz bir hucreyi silebilirsiniz\n\n");
    printf("\t\t (Eger oyunu yeniden baslatmak istiyorsaniz X'e basiniz)\n");
    printf("\n\n\t\tGERI DON = R\n\n");
    printf("\t\t");printf("=====");printf("\n");
}

```

```
|=====|
|
| Oyunun Kurallari :
| -Saga kaydirmek icin SAG OK TUSUNA..
| -Sola kaydirmek icin SOL OK TUSUNA..
| -Yukari kaydirmek icin YUKARI OK TUSUNA
| -Asagi kaydirmek icin ASAGI OK TUSUNA BASMALISINIZ...
| -Oyun icerisinde yaptiginiz her toplama islemi size 1 JOKER kazandirir
|   -Joker'e 'J' tusuna basarak erisebilir ve istediginiz bir hucreyi silebilirsiniz
|
| (Eger oyunu yeniden baslatmak istiyorsaniz X'e basiniz)
|
|
|   GERI DON = R
|
|=====|

```

Şekil 32 : Kural Fonksiyonu ve Kurallar

SONUÇ

Ayrıntılarıyla açıkladığım üzere 2048 oyunum bu özelliklerden ibarettir. Oyunu orijinal haliyle kıyaslarsak bir eksiği yoktur hatta bazı eklemelerim mevcuttur.

Oyuna çeşitlilik katmak amacıyla joker gibi oyuncuya avantaj veya dezavantaj kazandıracak özellikler eklenebilirdi. Örneğin belirli bir skordan sonra, her el boş hücrelerden birine bomba (mecazi anlamda) konulabilir, sayı o bombanın üzerine gelirse ekstra sayı üretilebilirdi. Bu da oyuncuyu biraz daha zora sokabilirdi. Kazandığımız jokerlerle bombaları da silebilirdik. Elbette bunlar belirli bir zaman verildiğinde uygulanabilecek fikirlerdir.

Oyunumda bununla birlikte skor tablosundaki skorlar sırasıyla kaydedilmiştir. Bunun yerine en yüksek skordan en düşük skora olacak şekilde isimler listelenebilirdi. Ancak skor tablosunu asıl oyundan bağımsız olarak oluşturduğum için bu özelliğini değiştirmek istemedim ve bu eklentimin oyundaki en iyi eklentilerimden biri olduğunu düşünüyorum.

Oyun dışında ve içinde menü kullanmamın, ekranda kalabalık oluşmamasını sağladığı ve akıcı bir görünüm oluşturduğu için daha doğru olduğunu düşünüyorum.

Oyunda C'de bulunan,

- Dizileri
- Matrisleri
- Fonksiyonları
- Pointer'ları ve
- Gömülü Döngüleri kullandım.

Kodlarken amacım aynı zamanda daha iyi bir oyun ambiyansı oluşturmaktı. Amacım ve fikirlerim doğrultusunda kodlamamı bu şekilde bitirdim.

KAYNAKÇA :

Ok tuşlarını nasıl kullanabileceğimi öğrenmek için : <https://stackoverflow.com>

Renk kodlarını öğrenmek için : <https://forum.donanimhaber.com> sitelerinden faydalandım.