

1. Problem Tanımı:

Tetris oyunu, düşen blokları yatay bir şekilde bir araya getirerek tam dolu satırlar oluşturmayı amaçlayan bir oyun türüdür. Oyuncunun hedefi, dolu satırları silerek puan kazanmak ve yeni blokların düşmesini sağlamaktır.

2. Çözüm:

2.1 Menü

İlk olarak kullanıcıyı aşağıdaki gibi bir menü karşılar:

```
1. Oyuna Basla
2. Kurallari ogren
3. Oyundan cik
Seciminizi yapin:|
```

```
int menu(){
    if(oyunSayisi==0){
        SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN);
        printf("1. Oyuna Basla\n");
        SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_RED);
    }else{
        printf("Son Oyunun Puani:%d      Rekor:%d\n\n",puan,maxPuan);

        SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN);
        printf("1. Oyuna Tekrar Basla\n");
        SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN);
    }

    SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN);
    printf("2. Kurallari ogren\n");
    SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_RED);

    SetConsoleTextAttribute(hConsole, FOREGROUND_RED);
    printf("3. Oyundan cik\n");
    SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_RED);

    printf("Seciminizi yapin:");
    int secim;
    scanf("%d",&secim);
    if(secim==1){
```

Eğer kullanıcı oyuna başlamak için '1'e basarsa sırasıyla aşağıdaki işlemler çalışır.

```
if(secim==1){
    puan=0;
    system("cls");
    printf("\nSatir sayisini giriniz:\n");
    scanf("%d",&ROW);
    printf("Sutun sayisini giriniz:\n");
    scanf("%d",&COL);

    for(i=2;i>=0;i--){
        system("cls");
        printf("\n%d\n",i+1);
        printf("Oyun basliyoor...");
        sleep(1);
    }

    for(i=0; i < ROW; i++){
```

```

    for(i=0; i < ROW; i++){
        for(j=0; j < COL; j++){
            gameBoard[i][j] = ' ';
        }
    }
    while(isGameOver()){
        system("cls");
        random();
        rotate();
        location();
        deleteRow();
    }
    for(i=0; i < ROW; i++){
        for(j=0; j < COL; j++){
            gameBoard[i][j] = ' ';
        }
    }
}
} else if(secim==3){

```

2.2 Oyuna Başlama

İlk olarak kullanıcıya oynamak istediği oyun tahtasının boyutu sorulur. System("cls") kullanarak her adımda ekranı sildirdim, böylece oyun tahtasının sabit kalmasını sağladım.

2.3 Random Şekil Atama

Sonra random bir şekil üretilerek (rand() fonksiyonundan yararlandım) kullanıcının ekranına oyun tahtası ve şekil verilir:

```

void random(){
    // Random bir şekil üretiyor.
    satir = rand() % 5;
    sutun = rand() % 4;

    // GAMEBOARD
    printf(" ");
    for (j = 0; j < COL; j++) {
        printf("_");
    }
    SetConsoleTextAttribute(hConsole, FOREGROUND_RED | FOREGROUND_GREEN);
    printf("      Puan: %d      Rekor: %d",puan,maxPuan);
    SetConsoleTextAttribute(hConsole, FOREGROUND_GREEN | FOREGROUND_BLUE | FOREGROUND_RED);
    printf("\n");

    for (i = 0; i < ROW; i++) {
        printf("|");
        for (j = 0; j < COL; j++) {
            printf("%c", gameBoard[i][j]);
        }
        printf("|\\n");
    }

    printf(" ");
    for (j = 0; j < COL; j++) {
        printf("%d", (j+1)%10);
    }
    printf("\\n\\n");

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            printf("%c", shapes[satir][sutun][i][j]);
        }
        printf("\\n");
    }
}

```

Çok sayıda şekil bulunmadığı için, şekilleri ve dönmüş hallerini manuel olarak tanımlamayı tercih ettim. Bu tanımlamaları bir çok boyutlu matris formatında gerçekleştirdim. Her satır, farklı bir şekli temsil ederken, her sütun da o şeklin farklı dönmüş hallerini içermektedir.

Random bir şekil vermek için de rand() fonksiyonu ile random bir satır ve sütun değeri alınarak oraya denk gelen şekil ekrana verilir.

```
Puan: 0      Rekor: 0

12345678901234567890
X
X
XX
Saga dondurmek icin "d", sola dondurmek icin "a", dondurmek istemiyorsanız "s" tusuna basin:|
```

2.4 Şekli Döndürme

Kullanıcı “d” tuşuna basarak şekli istediği kadar sağa “a” tuşuna basarak ise istediği kadar sola döndürebilir. Eğer ki kullanıcı şekli yeteri kadar döndüğünü düşünürse ya da hiç döndürmek istemezse “s” tuşuna basarak x koordinatını gireceği aşamaya geçer.

```
void rotate(){          // Şekilleri döndürüyor.
    int devamEt = 1;
    while (devamEt) {
        printf("Saga dondurmek icin \"d\", sola dondurmek icin \"a\", dondurmek istemiyorsanız \"s\" tusuna basin:");
        char islem;
        scanf(" %c", &islem);

        if (islem == 's') {
            devamEt = 0;
        } else if (islem == 'd') {
            sutun++;
            if (sutun > 3) {
                sutun = sutun - 4;
            }
        } else if (islem == 'a') {
            sutun--;
            if (sutun < 0) {
                sutun = sutun + 4;
            }
        }
    }
    system("cls");
    // GAMEBOARD
```

2.5 Şekli Yerleştirme

```
printf("Seklin sol ust kosesini yerlestirmek istediginiz x koordinatini giriniz:");
int koordinat;
scanf("%d",&koordinat);
```

Kullanıcıdan x koordinatı alındıktan sonra şekil gidebileceği en aşağı satıra animasyonlu bir şekilde düşer.

Kullanıcı sütun sayısından fazla bir x koordinatı girerse o şekli gidebileceği en sağ koordinata yerleştirdim.

```
// kullanıcı sütun sayısından daha büyük bir değer girerse en sona, daha küçük bir değer girerse en sağa yaslayacağım.
if(koordinat > COL){
    koordinat = COL;
}else if(koordinat < 1){
    koordinat = 1;
}

//şekillerin fazlasının dışarıya taşmasını engelledim.
if(koordinat == (COL-2) && genislik>3){
    koordinat = koordinat - genislik + 3;
}else if(koordinat==(COL-1) && genislik>2){
    koordinat = koordinat - genislik + 2;
}else if(koordinat==COL && genislik>1){
    koordinat = koordinat - genislik + 1;
}
```

Şeklin gidebileceği en aşağı satırı belirlemek için yukarıdan aşağıya doğru düşerken her adımda sorunsuz bir şekilde yerleşip yerleşemediği kontrol edilir. Eğer çakışma olursa, bir önceki adıma geri dönerek şekil oraya yerleştirilir. Eğer hiç çakışma olmazsa en aşağıya yerleştirilir.

```
int control=1,a=-1, enAlt=0;
while(control==1 && enAlt<ROW){
    a++;
    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            if(shapes[satir][sutun][i][j] == 'X' && gameBoard[a+i][koordinat-1+j] == 'X'){
                control=0;
            }
        }
    }

    for(i=0;i<4;i++){
        for(j=0;j<4;j++){
            if(shapes[satir][sutun][i][j] == 'X'){
                gameBoard[a-1+i][koordinat-1+j] = 'X';
            }
        }
    }

    enAlt=a+uzunluk-1;
}
a--;

for(i=0;i<4;i++){
    for(j=0;j<4;j++){
        if(shapes[satir][sutun][i][j] == 'X'){
            gameBoard[a+i][koordinat-1+j] = 'X';
        }
    }
}

puan++;
```

Aşağıya indiği her adımda şekli tabloya yazdırıp 0,08 saniye dondurarak (usleep(80000) ile) aşağıya düşme animasyonu elde ederek oyunu daha gerçekçi ve zevkli hale getirdim :)

Kullanıcı yerleştirebildiği her bir şekil için 1 puan kazanır.

2.6 Dolu Satırı Silme

Eğer kullanıcı herhangi bir satırı parçalar ile doldurabilirse, o satır 0.5 saniyelğine kırmızı olur ve sonra o satır kırılarak üstteki tüm parçalar aşağıya düşer. Kullanıcı kırabildiği her bir satır için satır uzunluğu kadar puan kazanır.

```
deleteRow(){ // Satırın tamamı dolduğunda o satırı siliyor.
    i=ROW-1;
    for(i=ROW-1;i>=0;i--){
        int flag=0,j=0;
        while(flag==0 && j < COL){
            if(gameBoard[i][j] != 'X'){
                flag=1;
            }
            j++;
        }

        if(flag == 0){
```

While kullanmamız verimliliği arttırdı.

2.7 Dolu Satırı Silme

Eğer oyun tahtasının en üst hattına herhangi bir parça gelirse o parça kırmızı olur ve oyun sona erer. Oynanan oyunun puanı rekordan daha yüksekse yeni rekor oynanan oyunun puanı olur.

```
int isGameOver(){
    int flag=1,j=0;
    while(flag==1 && j<COL){
        if(gameBoard[0][j]=='X'){
            flag=0;
            oyunSayisi++;
            if(puan>maxPuan){
                maxPuan=puan;
            }
        }
        j++;
    }
}
```

```

-----Puan: 4Rekor: 4
|  XX
|  X
|  X
|XXX
|X
|XX
| X
|XX
| X
| X
12345678901234567890

GAME OVER

Son Oyunun Puanı:4Rekor:4

1. Oyuna Tekrar Basla
2. Kurallari ogren
3. Oyundan cik
Seciminizi yapin:|
```

2.7 Oyuna Tekrar Başlama

Kullanıcı isterse 1 tuşuna basarak oyuna tekrar başlayabilir.

Kullanıcı kuralları öğrenmek isterse 2'ye basar ve sonra tekrardan menüye geri dönebilir.

Kullanıcı oyundan çıkmak isterse 3'e basarak oyundan çıkabilir.

Algoritmaların daha detaylı açıklamalarını videoda kodu göstererek anlatmaya çalıştım.

Video:

https://drive.google.com/file/d/1Jts4L4uq_-WHJCuHgDCpUXclRWxrwDeY/view?usp=drive_link