Proje Tanımı: Bu projede orijinal Reversi oyununun üç kişilik versiyonu olan Triversi oyunu için bir program hazırlanması beklenmektedir. Orijinal Reversi oyunundan farklı olarak;

• Kullanıcıların taşları "kırmızı", "sarı" ve "mavi" renklerdedir ve taşlar sınırsızdır.

• Oyun tahtasının boyutu sabit değildir, dinamik olarak kullanıcıdan alınan değere göre NxN ebatlarında oluşturulmaktadır

• Oyun,1. oyuncunun kırmızı taşını oyun tahtasının ortasına en yakın koordinata koyması ile başlar.

• Sırasıyla 2. oyuncu sarı, 3.oyuncu mavi taşını tahtaya diğer taşlara yatay/dikey/çapraz olarak koyar ve oyun bu şekilde devam eder.

• Renk değişimi için, yatay/dikey/çapraz sıranın bir açık ucuna koyulan tas ile o sıradaki aynı renkteki diğer en yakın taşın arasındaki diğer renklerin tamamı, koyulan taşın rengine döner.

Açıklama:

• Programınız çalıştırıldığında ilk olarak oyun tahtasının boyutu sorulmalıdır. Ekranda taşma olmaması için Nmax=23 olarak sınırlandırabilirsiniz.

• Her turda sıradaki oyuncudan taşını koyacağı koordinat alınmalı, taş oyun tahtasında gösterilmeli ve renk değişimleri ile tahtanın son hali yazdırılmalıdır,

• Oyun tahtasında boş göz kalmadığında oyun biter.

• Oyun sonunda her renkten kaçar taş olduğu ve kazanan oyuncu ekrana yazılmalıdır.

Temel Bileşenler ve Değişkenler

```
int N, i, a, b, s, j, k, c, d, birinci, ikinci, üçüncü;
int tmp[SIZE];
char mtr[SIZE][SIZE];
```

Değişkenlerin Görevleri:

• N:Dinamik tahta boyutu

• mtr[][]:Oyun tahtası matrisi

• tmp[] :Yon kontrol dizisi (8 yön için)

• birinci, ikinci, üçüncü :Oyuncu skorları

• Diğer değişkenler: Döngü ve koordinat kontrolü için yardımcı değişkenler

Oyun Tahtasının Oluşturulması ve İlk Kurulum

```
printf("lutfen kare matrisin boyutunu yazınız:");
scanf("%d", &N);
for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        mtr[i][j] = '_';
    }
}
```

Bu bölümde oyunun temelini oluşturan dinamik tahta yapısını kuruyorum. Özellikle:

1. Kullanıc1dan tahta boyutunu alıyorum

2. Bu boyuta göre dinamik bir matris oluşturuyorum

3. Başlangıçta tüm hücreleri boş olarak işaretliyorum

Oyun Mantığı ve Hareket Sistemi Başlangiç Hamlesi

```
mtr[N / 2][N / 2] = 'K';
```

İlk hamle özel bir önem taşıyor çünkü:

• Oyun dengesi için merkeze yakın başlama zorunluluğu var

• ilk taşın konumu sonraki hamleler için referans noktası oluşturuyor

Hamle Doğrulama Sistemi

Geliştirdiğim sistem üç katmanlı bir kontrol mekanizması içeriyor:

1. Sınır Kontrolü

```
if(a < 0 || a >= N || b < 0 || b >= N)
```

• Tahtanın fiziksel sınırlarını kontrol ediyor

• Geçersiz koordinatları anında tespit ediyor

2.Boşluk Kontrolü

```
while(mtr[a][b] != '_')
```

• Seçilen konumun müsait olup olmadığını kontrol ediyor

• Üst üste taş konulmasını engelliyor

2. Bitişiklik Kontrolü

```
while((mtr[a + 1][b] == '_' || ...) && ...)
```

• Yeni taşın mevcut taşlara bitişik olmasını sağlıyor

• 8 yönde kontrol yaparak geçerli hamleleri belirliyor

Renk Değişim Sistemi Oyunun en karmaşık kısmı olan renk değişim sistemi için geliştirdiğim algoritma:

```
while ((a + s - 1 < N || b + s - 1 < N || a - s + 1 >= 0 || b - s + 1 >= 0))
```

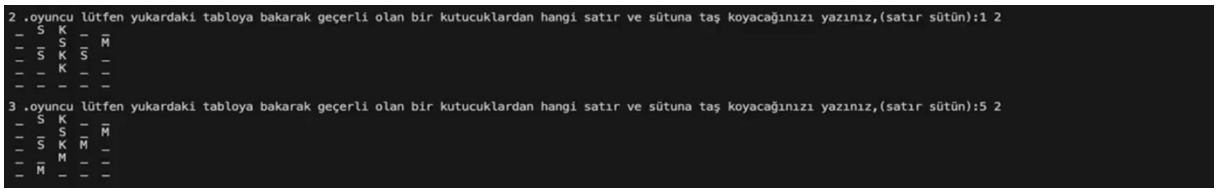
Bu sistem:

1. Her hamleden sonra 8 yönü kontrol ediyor

2. Aynı renkte taş bulunca aradaki taşları değiştiriyor

3. Çoklu yönde eşzamanlı değişimlere izin veriyor

Renk Değişim Örneği:



Oyun Sonu Mekanizması

```
for (i = 0; i < N; i++)
{
    for (j = 0; j < N; j++)
    {
        if (mtr[i][j] == 'K') birinci++;
        // ...
    }
}
```

Oyun sonunda:

• Her renkteki taşlar sayılıyor

• En çok taşa sahip oyuncu belirleniyor

• Beraberlik durumları kontrol ediliyor

Karşılaşılan Teknik Zorluklar ve Çözümler

1. Çoklu Yön Kontrolü

• Sorun: 8 farklı yönün eşzamanlı kontrolü

• Çözüm: Modüler bir kontrol sistemi ve geçici dizi kullanımı

3. Taş Çevirme Mantığı

• Sorun: Birden fazla yönde taş çevirme gerekliliği

• Çözüm: Her yön için bağımsız kontrol ve işlem yapan algoritma

3. Dinamik Tahta Yönetimi

• Sorun: Farklı boyutlarda tahtaların yönetimi

• Çözüm: Esnek sınır kontrolleri ve dinamik bellek yönetimi

https://drive.google.com/file/d/1tAPS0ZNNiLRbc-V9hBYqOiPQAsKvvSkV/view?usp=sharing

İsmail Orhan

23011081

Dönem Projesi