



Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM1012

Yapısal Programlamaya Giriş

Gr:2

Öğr. Gör. Dr. Ahmet ELBİR

Dönem Projesi

İsim: Bora İLCİ

No: 21011035

E-posta: bora.ilci@std.yildiz.edu.tr

Proje Konusu

Bu projede bir adet $N \times N$ boyutlu matrisin üzerinde bir renk eşleştirme oyunu tasarlanmıştır. Kullanıcıdan alınacak N değerine bağlı olarak oyun tahtasındaki renk sayısı ve oyun alanının boyutu belirlenip her renk çiftler halinde matrise yerleştirilir. Oyunu oynamak ve tamamlamak için bazı kurallar bulunmaktadır :

1. Sadece **aynı renkler** birbiriyle eşleşmelidir.
2. Birbiriyle **kesişmeyen** N farklı yol elde edilmelidir.
3. Oyunu tamamlamak için boş hücre **kalmamalıdır**.

Aşağıda örnek bir oyun alanı ve doğru çözümü gösterilmiştir:

Original matrix :				
Blue				Light Blue
Green		Light Blue	Red	
		Blue	Green	
Purple			Purple	Red
Final matrix :				
Blue	Blue	Light Blue	Light Blue	Light Blue
Green	Blue	Light Blue	Red	Red
Green	Blue	Blue	Green	Red
Green	Green	Green	Green	Red
Purple	Purple	Purple	Purple	Red

Örnek bir matris ve çözülmüş hali

Ana Menü Tanıtımı

Program ilk çalıştırıldığında kullanıcı ana menü ile karşılaşır.

Menüde 4 adet seçenek bulunmaktadır. 1. ve 2. seçenekler oyunla alakalı matris türünü seçmek için, 3. seçenek kullanıcı skorlarını görüntülemek için, 4. seçenek ise programdan çıkış yapmak için seçilmelidir.

```
###MAIN MENU###

1 -> Generate a Random Matrix
2 -> Create a Matrix From a File
3 -> Show User Scores
4 -> Exit
```

Oyuna giriş yapmadan önce matrisin nasıl oluşturulacağı seçilir. Daha sonra kullanıcıdan bir kullanıcı ismi girmesi istenir. Maksimum 15 adet kullanıcıya kadar kullanıcı eklenebilir. Eğer önceden kaydedilmiş bir kullanıcı adı tekrar girilirse, varolan kullanıcı adı üzerinden oyuna geçilir ve oynanan oyunların puanları aynı kullanıcı adına yazılır.

Programda matris oluşturmak için iki adet yol bulunmaktadır.

1. Rastgele Matris Oluşturma

Önce kullanıcıdan oyun tahtasının boyutunu ve renk sayısını belirlemek için N değeri istenir. Daha sonra N değerine göre matris üzerine yerleştirilecek renk çiftlerinin koordinatları rastgele olacak şekilde program tarafından belirlenir ve oyun tahtasına aktarılır.

```
Enter player name (maximum 15 characters): Mike

Enter the dimension of the matrix.
Dimension : 5

-----
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
-----

1 -> Play in Manuel Mode
2 -> Play in Automatic Mode
3 -> Return to Main Menu
```

Rastgele matris oluşturma

2. Dosyadan Matris Oluřturma

Önce kullanıcıdan oyun tahtasının boyutunu ve renk sayısını belirlemek için N değeri istenir. Program, halihazırda oluşturulmuş bir dosyadan renk çiftlerinin koordinatlarını alır ve oyun tahtasını bu dosyayı baz alarak oluşturur.

```
Enter player name (maximum 15 characters): Harvey
Enter the dimension of the matrix.
Dimension : 5

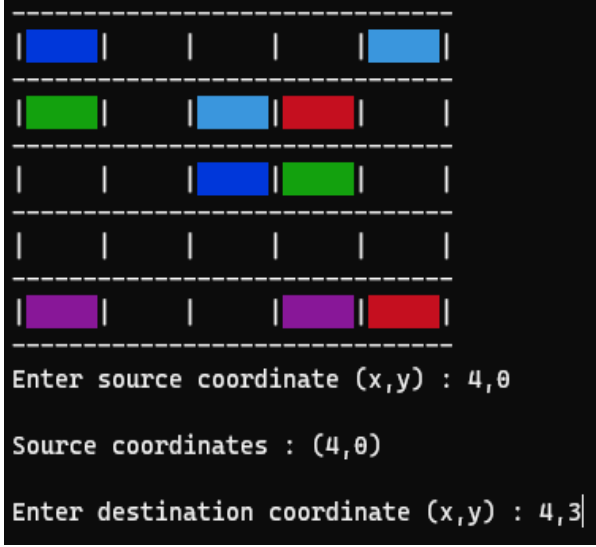
Enter the name of the file.
File name : 5x5_test_case1.txt
-----
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
-----
1 -> Play in Manuel Mode
2 -> Play in Automatic Mode
3 -> Return to Main Menu
```

Dosyadan matris oluřturma

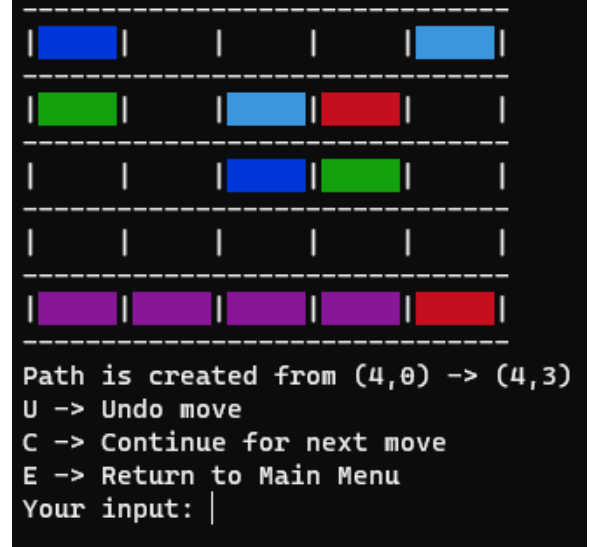
Oyun tahtası oluşturulduktan sonra kullanıcıyı bir menü karşılar. Bu menüde kullanıcıya üç adet seçenek sunulmuştur. Eğer oyun kullanıcı tarafından manuel şekilde oynanmak isteniyorsa 1. seçenek, programın otomatik olarak oynaması isteniyorsa 2. seçenek, ana menüye dönmek için 3. seçenek seçilmelidir.

Manuel Oyun Modu

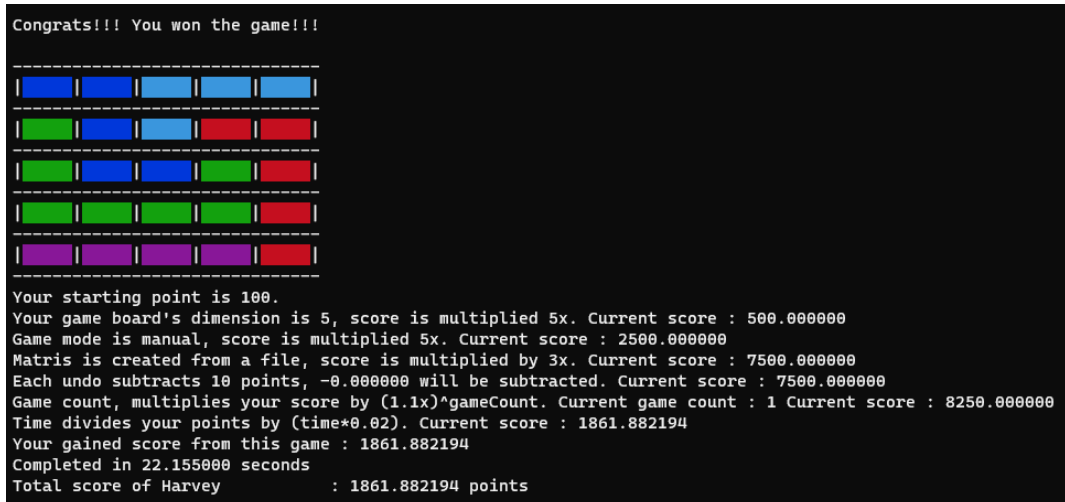
Manuel modda kullanıcıdan bir adet başlangıç koordinatı ve bir adet varış koordinatı istenir. Kullanıcı sadece dikey veya yatay olacak şekilde tek bir doğrultuda gidebilir. Çapraz gitmek istemesi durumunda tekrar koordinat alma ekranına geri dönecektir. Ayrıca oyun alanının dışında bir koordinat girilirse de tekrar koordinat alınır. Girilen koordinatlar uygunsa, başlangıç hücresi ile varış hücresi arasındaki tüm hücreler, başlangıç hücresinin rengi ile boyanır ve yol oluşturulmuş olur.



->



Yol oluşturulduktan sonra kullanıcıya 2 seçenek daha sunulur. Eğer yapmak istediği hamleden memnunsa devam edebilir, eğer istemediği bir hamle yaparsa geri alabilir. Sadece son 15 hamlenin geri dönüşü vardır, 15 hamleden sonra eski hamlelerin üstüne yeni hamleler yazılır. Devam edilmek istenirse oyun yine koordinat alma ekranına geri döner ve matris doldurulana kadar oyun devam eder. Matris doldurulduğunda oyun tahtasının doğru doldurup doldurulmadığı kontrol edilir. Daha sonra kullanıcının oyunu doğru bitirip bitiremediğine göre bir puan hesaplaması yapılır, kullanıcıya oyunu kazanıp kazanmadığına göre iki ayrı ekran gösterilir ve ardından puanı gösterilir. Puan gösterildikten sonra ana menüye dönülür.



Otomatik Oyun Modu

Otomatik moda girildiğinde kullanıcıdan herhangi bir hamle girdisi alınmaz, sadece belirli saniye aralıkları ile kullanıcıya programın yaptığı hamlelerin gösterildiği oyun ekranı sunulur.

NOT: Bu mod programın doğru çözümü bulması halinde son bulur fakat program tüm olasılıkları denediği için veya başka bir tabirle derinlemesine bir arama yaptığı için **arama süresi matrisin boyutuna ve matristeki renk çiftlerinin yerlerine bağlı olduğu için bazı oyun tahtalarında çok kısa sürede sonuç elde edilirken, bazı oyun tahtalarında ise dakikalar sürebilmektedir**. Kullanıcının bunu göz önünde bulundurulması istenir.

```
Enter player name (maximum 15 characters): HarveySpecter

Enter the dimension of the matrix.
Dimension : 5

Enter the name of the file.
File name : 5x5_test_case2.txt

-----
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
-----

1 -> Play in Manuel Mode
2 -> Play in Automatic Mode
3 -> Return to Main Menu
```

```
Original matrix :
-----
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
-----

Final matrix :
-----
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
-----

Your starting point is 100.
Your game board's dimension is 5, score is multiplied 5x. Current score : 500.000000
Game mode is automatic, score is multiplied by 1x. Current score : 500.000000
Matris is created from a file, score is multiplied by 3x. Current score : 1500.000000
Game count, multiplies your score by (1.1x)^gameCount. Current game count : 1 Current score : 1650.000000
Time divides your points by (time*0.02). Current score : 1087.243015
Your gained score from this game : 1087.243015
Completed in 7.588000 seconds
Total score of HarveySpecter : 1087.243015 points
```

Otomatik Oyun Modu Algoritması

Otomatik oyun modu algoritmasının çalışma mantığı aslında **derinlemesine kör arama (Depth First Search)** mantığına dayanmaktadır. Bir başka deyişle program doğru oyun tahtasını bulana kadar tüm olasılıkları dener.

Otomatik moda giriş yapılmadan önce oyun tahtasındaki renk çiftlerinin başlangıç ve bitiş koordinatları kaydedilir. Daha sonra sırası program tarafından belirlenen 1. rengin başlangıç koordinatından başlanarak sırayla o hücrenin tüm komşuları kontrol edilir.

Kontrol şartları birden fazladır:

1. *Fonksiyona gönderilen koordinatlar oyun tahtasının içinde mi?*
2. *Fonksiyona gönderilen koordinatlar oyun tahtasında bakıldığında 0 veya o hangi renk ne ise o renk mi?*

Eğer komşu hücre boş ise (yani 0 ise) fonksiyon, komşu hücrenin koordinatları ile tekrar çağırılır. Eğer renk halihazırda **1. rengin bitiş koordinatlarına ulaşılmadıysa** yeni çağırılan fonksiyonda hücrenin rengi 1. renk yapılır (yani halihazırdaki renk). Daha sonra bu komşu hücrenin de hücrelerine sırayla bakılır ve bu işlem komşulardan biri bitiş koordinatına ulaşana kadar devam eder. komşu hücre ile başlangıç rengi aynı ise 2 adet kontrol daha yapılır:

1. *Bu hücrenin koordinatları bitiş koordinatına eşit mi?*
2. *Bu hücrenin koordinatları başlangıç koordinatına eşit mi?*

Eğer bitiş koordinatlarına eşitse, fonksiyon sıradaki rengin başlangıç koordinatları, ve sıradaki rengin değeri ile tekrar çağırılır ve yukarıda anlatılan işlemlerin aynısı bu renk için de yapılır. Eğer başlangıç koordinatına eşitse 2 ihtimal vardır.

- a. Yeni bir renge geçiş yapıldığında fonksiyon ilk defa çağırılmıştır.
- b. Arama yapılırken başlangıç koordinatına tekrar gelinmiştir.

Eğer ilk defa başlangıç koordinatlarına gelindiye, o koordinat ayrı bir matriste işaretlenir ve fonksiyonun komşularına yine yukarıda anlatıldığı gibi bakılmaya devam edilir. Dört komşuya bakıldıktan sonra ayrı matriste işaretlenen hücre tekrar silinir. Ama **işaretlenen hücre silinmeden önce** arama yapılırken tekrar başlangıç hücresinin koordinatları fonksiyona gönderilirse o hücre **tekrar kontrol edilmez**. Bunun amacı gereksiz kontrolleri engellemektir.

Tüm bu işlemler **en son renk değerine gelinene kadar** devam eder. En son renge gelindiğinde artık ekstra olarak **oyun tahtasının dolup dolmadığı kontrol edilmeye başlanır**. Eğer oyun tahtası dolarsa, **oyunun doğru çözülüp çözülmediği** (bir başka deyişle tüm renklerin birbirine bağlanıp bağlanmadığı) **kontrol edilir**. Bu işlem de bir rengin başlangıç koordinatından bitiş koordinatına gidilip gidilmediğini kontrol eder. Bu işlem de yine yukarıda anlatılan arama mantığı ile yapılır fakat bir adet fark vardır, burada kontrol edilen şart hücrenin 0 olması değil, kontrol edilen renk ile aynı renk olup olmamasıdır. Eğer tüm renkler için bu kontrol doğru yapıldıysa o anki oyun alanı ayrı bir matrise kaydedilir ve değişmesi engellenir. Otomatik oyun fonksiyonunun stackteki tüm işlemleri bittikten sonra çözülmüş oyun tahtası ekrana gösterilir.

Algoritma Karmaşıklığı

Manuel Oyun Modu

Manuel oyun modunda, karmaşıklık $T(N)$ dir, N burada kullanıcının hamle sayısıdır. Kullanıcının hamle sayısına göre oyun çok hızlı da bitebilir, çok uzun da sürebilir.

Otomatik Oyun Modu

Otomatik oyun modunda da her hücre 4 kere kontrol edilecektir fakat bu işlem matris dolana kadar yapılacağı için yukarıdaki gibi tek seferlik bir arama yapılmayacak, aksine daha fazla olasılık denenecektir. Her hücreden 4 farklı hücreye gidebileceğimizi varsayarsak, matrisin boyutunu da N kabul ettiğimiz taktirde hücre sayısı N^2 olacaktır. Olasılıkta farklı olasılıkları çarparak hesapladığımız için $4 * 4 * 4 \dots 4 = 4^{N^2}$ kadar olasılığımız bulunmaktadır. Bu da bizi $T(4^{N^2})$ karmaşıklığa götürür.

Puan Sistemi

Her oyuncu, her oyunun başında o oyun için 100 puan ile başlangıç yapar. Bazı parametrelere göre bu puan artıp azalabilir. Bu parametreler şöyledir:

1. Oyun modunun otomatik/manuel durumu
2. Oyun tahtasının rastgele/dosyadan oluşturma durumu
3. Oyunun daha önce kaç defa oynandığı
4. Oyunun süre uzunluğu
5. Oyun alanının boyutu
6. Oyunun kazanılıp kazanılmadığı (kayıp durumunda %75 az puan alınır)

Aşağıda parametrelere göre puanın nasıl değiştiğinin örneği verilmiştir:

```
Final matrix :
-----
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
-----
Your starting point is 100.
Your game board's dimension is 5, score is multiplied 5x. Current score : 500.000000
Game mode is automatic, score is multiplied by 1x. Current score : 500.000000
Matris is created from a file, score is multiplied by 3x. Current score : 1500.000000
Game count, multiplies your score by (1.1x)^gameCount. Current game count : 1 Current score : 1650.000000
Time divides your points by (time*0.02). Current score : 2140.633108
Your gained score from this game : 2140.633108
Completed in 3.854000 seconds
Total score of Harvey : 2140.633108 points

---Player List---
Player Name : Mike           Score : 18.507747
Player Name : Harvey        Score : 2161.949686
Player Name : Jessica       Score : 573.115665
```