



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ

Yapısal Programlamaya Giriş
(BLM 1012)
DÖNEM PROJESİ

20011504 – Mehmet Talha Duman

talha.duman@std.yildiz.edu.tr

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

1. Giriş

Bu çalışma, $N \times N$ boyutundaki bir matris üzerinde sayı eşleştirme oyununun algoritmik çözümünü sunmayı hedeflemektedir. Örneğin, $N=5$ için, matriste 1 ile 5 arasındaki sayılar çiftler halinde yer almakta ve bu matris üzerinde belirli kurallara göre eşleştirme yapılması gerekmektedir. Bu rapor, matris üzerindeki sayıları belirtilen kurallara uygun şekilde eşleştirmek için kullanılacak yöntemleri ve elde edilen sonuçları detaylı bir şekilde sunmaktadır.

2. Yöntem

Raporun ikinci kısmı olan yöntem kısmında her bir fonksiyonun detaylı açıklaması yapılacaktır ve oyuna dair görüntüler yer alacaktır.

1. randomMatrix (): "randomMatrix" fonksiyonu, belirli bir boyuttaki matris üzerinde rastgele sayıları yerleştirmek için kullanılan bir algoritmayı içermektedir. Fonksiyon, verilen boyuta bağlı olarak matrisin boyutunu ve matrisin kendisini parametre olarak alır. İlk olarak, matrisin tüm elemanları "resetMatrix" fonksiyonu yardımıyla sıfıra eşitlenir, böylece matrisin başlangıç durumu temizlenir. Ardından, iki döngü kullanılarak her bir sayı iki kez yerleştirilir. İlk döngü, 1'den başlayarak belirtilen boyuta kadar olan sayıları matris üzerine yerleştirir. İkinci döngü ise her bir sayıyı iki kez yerleştirmek için kullanılır. Her bir sayı için, rastgele iki indis oluşturulur ve bu indisler matrisin satır ve sütunlarını belirlemek için kullanılır. Eğer belirlenen indislere karşılık gelen matris elemanı boş ise (0), sayı bu indislere yerleştirilir. Ancak, eğer belirlenen indislere karşılık gelen matris elemanı dolu ise, yeni bir boş konum bulana kadar yeni rastgele indisler oluşturulur. Bu işlem, tüm sayılar yerleştirilene kadar devam eder. Sonuç olarak, matris üzerinde rastgele yerleştirilen sayıları içeren bir matris elde edilir. Bu fonksiyon, matris üzerinde rastgele sayıları yerleştirme işlemini gerçekleştirmek için bir algoritma sunmaktadır.

```
1.Rasgele Matris Olustur
2.Dosyadan Matris Olustur
3.Kullanıcıların Skorlarını Göster
Çıkmak için 4'e basınız.
1
Kullanıcı adınızı giriniz. Mehmet
1.Manuel Modda Oyna
2.Ötö Modda Oyna
3.Ana Menüye Dön
1
Lütfen matris boyutunu giriniz. 4

-----
4 |   |   | 2 |
-----
|   |   |   |
-----
1 | 3 | 1 | 3 |
-----
2 |   | 4 |   |
```

Şekil 1: Rastgele matris oluşturma menüsü görüntüsü

2. move (): "move" fonksiyonu, belirli bir matris üzerinde bir sayının belirli bir konuma yerleştirilmesini sağlayan bir işlevdir. Fonksiyon, hedef konumun satır ve sütun indislerini (ind1 ve ind2), yerleştirilecek sayıyı (number), matrisin kendisini (matris) ve matrisin boyutunu (boyut) parametre olarak alır. İlk olarak, fonksiyon, hedef konumun boş olup olmadığını kontrol eder. Eğer hedef konum (matris[ind1][ind2]) boş (0) ise, sayı bu konuma yerleştirilir (matris[ind1][ind2] = number). Ancak, eğer hedef konum matrisin boyutunu aşıyorsa, yani ind1 veya ind2 değerleri boyut değerinden büyük veya eşitse, bir hata mesajı gösterilir. Bu durumda, "Girdiğiniz indisler matris boyutunu aşıyor, lütfen düzeltin." şeklinde bir mesaj ekrana yazdırılır. Son olarak, eğer hedef konum dolu ise, yani matris[ind1][ind2] değeri 0 değilse, bir hata mesajı gösterilir. Bu durumda, "Hareket etmek istediğiniz alan dolu." şeklinde bir mesaj ekrana yazdırılır. Bu fonksiyon, belirtilen koşullara göre bir sayının matris üzerinde belirli bir konuma yerleştirilmesini sağlamaktadır. Fonksiyon, hedef konumun uygunluğunu kontrol ederek işlemi gerçekleştirir ve gerekli hata mesajlarını kullanıcıya bildirir.

```
1.Rasgele Matris Olustur
2.Dosyadan Matris Olustur
3.Kullanıcıların Skorlarını Göster
Çıkmak için 4'e basınız.
1
Kullanıcı adınızı giriniz. Mehmet
1.Manuel Modda Oyna
2.Oto Modda Oyna
3.Ana Menüye Dön
1
Lütfen matris boyutunu giriniz. 4

-----
4 |   |   | 2 |
-----
|   |   |   |
-----
1 | 3 | 1 | 3 |
-----
2 |   | 4 |   |
-----

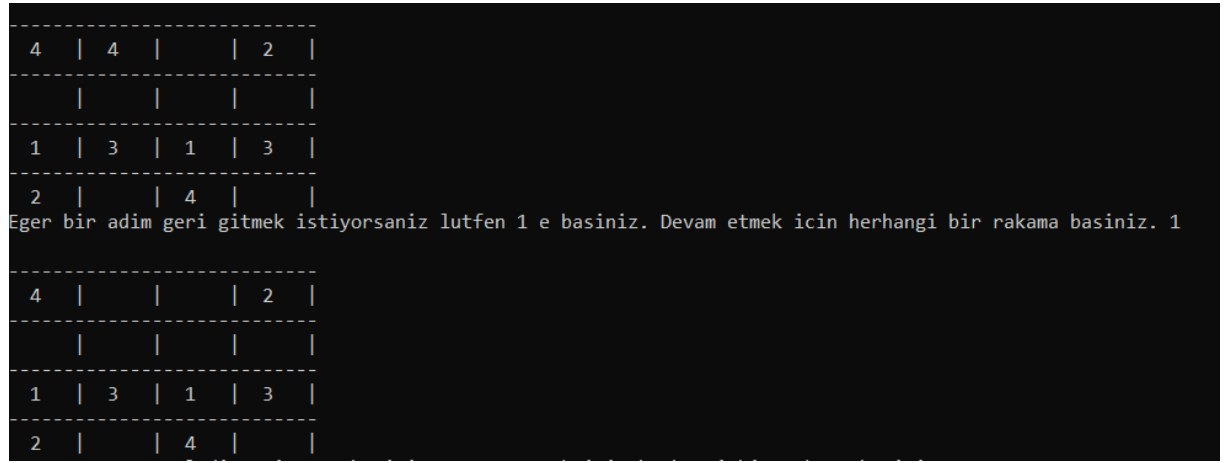
Lütfen birleştirmek istediğiniz noktanın bulunduğu ilk ve ikinci indisi giriniz.
Örnek: Önce satır sonra sütun olacak şekilde (0,1) noktasında ise 0 1 yazınız.
0 1
Lütfen yerleştirmek istediğiniz sayıyı giriniz. 4

-----
4 | 4 |   | 2 |
-----
|   |   |   |
-----
1 | 3 | 1 | 3 |
-----
2 |   | 4 |   |
-----
```

Şekil 2: move fonksiyonu yardımıyla oynanan bir adımın görüntüsü

3. undo (): "undo" fonksiyonu, belirli bir matris üzerinde belirli bir konumdaki sayının geri alınmasını sağlayan bir işlevdir. Fonksiyon, hedef konumun satır ve sütun indislerini (ind1 ve ind2) ve matrisin kendisini (matris) parametre olarak alır. Fonksiyon, öncelikle hedef konumdaki elemanın boş olup olmadığını kontrol eder. Eğer hedef konumdaki eleman (matris[ind1][ind2]) 0 değilse, yani dolu ise, o konumdaki sayı geri alınmak istenmektedir. Bu durumda, hedef konumun değeri 0'a eşitlenir ve sayı geri alınmış olur (matris[ind1][ind2] = 0). Bu fonksiyon, kullanıcının isteği doğrultusunda bir matrisin belirli bir konumundaki sayının geri alınmasını sağlamaktadır. Geri alınacak konumdaki elemanın boş olup olmadığı kontrol edilir ve dolu ise sayı geri alınır. Bu işlem, matris üzerindeki değişiklikleri geri

alma yeteneği sağlamaktadır. (Not: Oyunun kodlanması sırasında ind1 ve ind2 "move" fonksiyonunda alınan indislerdir, bu sayede oyunda bir önceki adıma gitmek için ayrıca indis girmemiz gerekmez.)



Şekil 3: undo fonksiyonu yardımıyla bir adım geri alma görüntüsü

4. playManual (): "playManual" fonksiyonu, kullanıcının manuel olarak sayıları eşleştirdiği bir oyun sürecini gerçekleştiren bir işlevdir. Fonksiyon, matrisin kendisini (matris) ve matrisin boyutunu (boyut) parametre olarak alır. İlk olarak, "bitis" ve "undoCounter" değişkenleri tanımlanır ve "bitis" değeri 1 olarak atanır. Bu değişkenler, oyun döngüsünü kontrol etmek ve geri alma adımlarını saymak için kullanılır. Daha sonra, bir while döngüsü başlatılır. Döngü, "bitis" değeri 0 olmadığı sürece devam eder. Bu, kullanıcının oyunu tamamladığında veya çıkış yapmak istediğinde döngünün sona ermesini sağlar. Döngü içinde, kullanıcıya matrisin güncel durumunu gösteren "drawBoard" fonksiyonu çağrılır. Ardından kullanıcıdan, birleştirmek istediği sayının konumunu ve yerleştirmek istediği sayıyı girmesi istenir. İlk olarak, sayının konumunu temsil eden "ind1" ve "ind2" değişkenleri alınır. Kullanıcıdan sayıyı temsil eden "number" değişkeni alınır. Bu bilgilerle "move" fonksiyonu çağrılır ve sayı belirtilen konuma yerleştirilir. Yeniden "drawBoard" fonksiyonu çağrılarak matrisin güncel durumu kullanıcıya gösterilir. Kullanıcıya geri alma işlemi yapmak isteyip istemediği sorulur. Eğer 1'e basarsa, "undoCounter" artırılır, "undo" fonksiyonu çağrılarak son yerleştirdiği sayı geri alınır ve matrisin güncel durumu gösterilir. Son olarak, kullanıcının oyunu tamamlamak isteyip istemediği sorulur. Eğer 0'a basarsa, döngü sona erer ve oyun tamamlanır. Aksi halde, döngü devam eder ve kullanıcıdan yeni bir adım atması beklenir. Fonksiyon, geri alınan adımların sayısını temsil eden "undoCounter" değeriyle tamamlanır ve bu değer geri döndürülür. Bu fonksiyon, kullanıcının manuel olarak sayıları eşleştirdiği oyun sürecini yürüten bir algoritma sunar. Kullanıcıya matrisin durumu gösterilir, sayıların yerleştirilmesi ve geri alma adımları kullanıcıdan alınır ve oyun tamamlanana kadar sürekli olarak güncel durum gösterilir.

5. readFromFile (): "readFromFile" fonksiyonu, bir dosyadan veri okuyarak matrisin ilgili konumlarına yerleştiren bir işlevdir. Fonksiyon, belirtilen dosya adını kullanarak dosyayı açar ve dosyanın sonuna kadar bir döngü içinde satır satır veri okur. Her bir satırda, okunan değerler matrisin ilgili konumuna yerleştirilir. Fonksiyon, dosya açma işlemi başarısız olursa bir hata mesajı gösterir ve fonksiyon sonlanır. Bu fonksiyon, dosyadan veri okuma ve matris güncelleme işlemlerini gerçekleştiren bir algoritma sunar.

```
1.Rasgele Matris Olustur
2.Dosyadan Matris Olustur
3.Kullanıcıların Skorlarını Göster
Çıkmak için 4'e basınız.
2
Kullanıcı adınızı giriniz. Talha
1.Manuel Modda Oyna
2.Oto Modda Oyna
3.Ana Menüye Don
1
Dosya Adını Giriniz: Data.txt

-----
1 |   |   |   |   |
-----
2 | 3 |   | 3 | 1 |
-----
  | 4 |   |   |   |
-----
  |   |   | 2 | 4 |
-----
5 |   |   |   | 5 |

Lütfen birleştirmek istediğiniz noktanın bulunduğu ilk ve ikinci indisi giriniz.
Örnek: Önce satır sonra sütun olacak şekilde (0,1) noktasında ise 0 1 yazınız. 0 1
Lütfen yerleştirmek istediğiniz sayıyı giriniz. 1

-----
1 | 1 |   |   |   |
-----
2 | 3 |   | 3 | 1 |
-----
  | 4 |   |   |   |
-----
  |   |   | 2 | 4 |
-----
5 |   |   |   | 5 |
Eğer bir adım geri gitmek istiyorsanız lütfen 1'e basınız. Devam etmek için herhangi bir rakama basınız. 2
Eğer oyunu tamamladıysanız 0'a basınız. Devam etmek için herhangi bir rakama basınız. 2

-----
1 | 1 |   |   |   |
-----
2 | 3 |   | 3 | 1 |
-----
  | 4 |   |   |   |
-----
  |   |   | 2 | 4 |
-----
5 |   |   |   | 5 |
```

Şekil 4: Dosyadan okuma menüsü görüntüsü

6. drawBoard (): "drawBoard" fonksiyonu, matrisin mevcut durumunu ekranda görsel olarak temsil eden bir işlevdir. Fonksiyon, matrisin boyutunu ve elemanlarını gezerek, dolu ve boş konumları belirli bir formatta ekrana yazdırır. Bu şekilde kullanıcıya matrisin güncel durumu gösterilir.

7. matrixSize (): "matrixSize" fonksiyonu, belirtilen dosyanın içindeki satır sayısını kullanarak matrisin boyutunu hesaplayan bir işlevdir. Fonksiyon, dosya adını temsil eden bir karakter dizisi parametresi alır. İlk olarak, dosyanın açılması kontrol edilir ve dosya başarıyla açılamazsa bir hata mesajı verilir. Ardından, dosyanın her bir karakteri tek tek okunur ve satır sonu karakteri olan '\n' ile karşılaşıldığında satır sayısı bir artırılır. Bu şekilde, matrisin bir satırını temsil eden çift sayıda satırın olduğu tespit edilir. Döngü tamamlandığında, dosya kapatılır ve hesaplanan satır sayısı 2'ye bölünerek matrisin boyutu elde edilir. Son olarak, boyut değeri fonksiyon tarafından döndürülür. "matrixSize" fonksiyonu, dosya açma, karakter okuma ve satır sayısı hesaplama işlemlerini içeren basit bir algoritma sunar. (Not: Bu fonksiyon dosyadan okuma modunda gerekli olan boyut bilgisini sağlamak için yazılmıştır.)

8. getNearestEqualNumber (): "getNearestEqualNumber" fonksiyonu, verilen bir konumun (satır ve sütun indeksi) içinde bulunduğu matris üzerinde en yakın eşit sayıyı bulan bir işlevdir. Fonksiyon, satır indeksi (rowIndex), sütun indeksi (colIndex), matris (matrix) ve matrisin boyutu (boyut) parametre olarak alır. İlk olarak, işlem için gerekli değişkenler tanımlanır. "number" değişkeni, en yakın eşit sayıyı temsil eder ve varsayılan olarak 0 olarak atanır. "minDistance" değişkeni, en küçük mesafeyi temsil eder ve başlangıçta boyutun iki katı olarak atanır. İki adet for döngüsü kullanılarak matrisin her bir elemanı gezilir. Her elemanın değeri kontrol edilir ve eşitlik durumu kontrol edilir. Eğer eleman 0 değilse ve modülü aynı ise, bir mesafe hesaplanır. Mesafe, elemanın konumu ile verilen konum arasındaki satır ve sütun farklarının toplamıdır. Eğer hesaplanan mesafe, şu ana kadar kaydedilen en küçük mesafeden daha küçükse, mesafe ve sayı güncellenir. Eğer "number" hala 0 ise, yani eşit bir sayı bulunamamışsa, ikinci bir döngü kullanılarak matriste bulunmayan bir sayı seçilir. Bu sayı, verilen konumda bulunan sayıdan farklı olmalıdır. Sonuç olarak, "number" değeri döndürülür. "getNearestEqualNumber" fonksiyonu, matris üzerinde gezinme, sayı kontrolü ve mesafe hesaplama gibi işlemleri gerçekleştirerek, verilen bir konumun en yakın eşit sayısını bulur. Fonksiyon, matrisin boyutuna ve elemanlarının durumuna bağlı olarak uygun bir algoritma sunar. (Not: Maalesef bu fonksiyon doğru bir şekilde çalışmıyor.)

9. playAuto (): "playAuto" fonksiyonu, verilen bir matris üzerinde otomatik olarak oyun oynayan bir işlevdir. Matrisi ve boyutunu parametre olarak alır. Fonksiyon, öncelikle matrisin başlangıç durumunu "drawBoard" fonksiyonunu kullanarak ekrana yansıtır. Daha sonra, iç içe döngüler kullanarak matrisin her bir hücreğini dolaşır. Bir hücrenin değeri 0 ise (boş bir hücre), fonksiyon, "getNearestEqualNumber" fonksiyonunu kullanarak en yakın eşit sayıyı bulmaya çalışır. Eğer 0 olmayan bir sayı bulunursa (eşleşen bir sayı mevcutsa), "move" fonksiyonu çağrılarak sayı boş hücreye yerleştirilir. Güncellenen matris durumu ise "drawBoard" fonksiyonuyla ekranda gösterilir, böylece otomatik oyun ilerlemesi takip edilebilir. Her bir hücre dolaşarak boş hücreler, mevcut kurallara uygun olarak en yakın eşit sayılarla doldurulmaya devam edilir. "playAuto" fonksiyonu, "getNearestEqualNumber", "move" ve "drawBoard" fonksiyonlarını kullanarak otomatik bir oyun deneyimi sunar. Otomatik oyun ilerledikçe matrisin ara durumları ekranda gösterilir.

```
1.Rasgele Matris Olustur
2.Dosyadan Matris Olustur
3.Kullaniciarin Skorlarini Goster
Cikmak icin 4'e basiniz.
2
Kullanici adinizi giriniz. Mehmet
1.Manuel Modda Oyna
2.Oto Modda Oyna
3.Ana Menuye Don
2
Dosya Adini Giriniz: Data.txt

-----
1 |  |  |  |  |  |
-----
2 | 3 |  |  | 3 | 1 |
-----
  | 4 |  |  |  |  |
-----
  |  |  |  | 2 | 4 |
-----
5 |  |  |  |  | 5 |
-----
```

Şekil 5: Otomatik mod menü görüntüsü

1	5	5	5	5
2	3	5	3	1
5	4	5	5	5
5	5	5	2	4
5	5	5	5	5

Şekil 6: playAuto fonksiyonu sonucunda oluşan çıktı

10. calculateScore (): "calculateScore" fonksiyonu, oyunun sonucunda bir puan hesaplamak için kullanılır. Fonksiyon, rastgele matris oluşturma sayısını (randomMatrixCounter), geri alma işlemi sayısını (undoCounter), manuel oynama sayısını (manualCounter), geçen süreyi (sure) ve matris boyutunu (boyut) parametre olarak alır. Puan hesaplama işlemi şu adımlarla gerçekleştirilir:

1. Rastgele matris oluşturma sayısının iki katı ($2 * \text{randomMatrixCounter}$) puan olarak eklenir.
2. Geri alma işlemi sayısı (undoCounter) puan olarak çıkarılır.
3. Manuel oynama sayısı (manualCounter) dört katı ($4 * \text{manualCounter}$) puan olarak eklenir.
4. Geçen süre (sure) 10'a bölünerek puan azaltması yapılır.
5. Matris boyutu (boyut) puan olarak eklenir.

Bu adımlar sonucunda hesaplanan puan, "score" değişkenine atanır ve fonksiyon tarafından döndürülür. "calculateScore" fonksiyonu, oyuncunun performansını ve oyun sürecini değerlendirerek bir puan ortaya çıkarmak için kullanılır.

```
1.Rasgele Matris Olustur
2.Dosyadan Matris Olustur
3.Kullanıcıların Skorlarını Göster
Çıkmak için 4'e basınız.
3
Mehmet adlı kullanıcının skoru: 7
Talha adlı kullanıcının skoru: 10
Duman adlı kullanıcının skoru: 10
```

Şekil 7: Kullanıcı skorlarının görüntüsü

11. `resetMatrix()`: "resetMatrix" fonksiyonu, verilen bir matrisin tüm elemanlarını sıfırlayan bir işlevdir. Matrisi ve boyutunu parametre olarak alır. İç içe döngüler kullanarak matrisin her bir elemanını dolaşır. Bu işlem, matrisin tüm elemanlarının sıfırlanması için gerçekleştirilir. "resetMatrix" fonksiyonu, bir matrisin başlangıç durumuna dönmek veya matrisi sıfırlamak için kullanılır. Matrisin elemanları yeniden kullanıma hazır hale getirilir, böylece yeni bir oyun veya işlem yapılabilir.

12. `main()`: "main" fonksiyonu, oyunun ana kontrol akışını sağlayan ve kullanıcının menüler arasında geçiş yapmasını, oyun modlarını seçmesini ve oyunun ilerleyişini yöneten bir işlevdir. Fonksiyon başında gerekli değişkenler ve matris tanımlamaları yapılmıştır. Daha sonra bir döngü oluşturulmuştur, bu döngü oyunun ana menüsünde kullanıcının seçimlerine göre işlem yapmaktadır. Kullanıcıya menü seçenekleri sunulur ve seçimine göre ilgili işlemler gerçekleştirilir. İlgili seçeneklere göre rastgele matris oluşturma, dosyadan matris okuma veya kullanıcı skorlarını gösterme işlemleri gerçekleştirilebilir. Rastgele matris oluşturma seçeneği seçildiğinde kullanıcıdan matris boyutu alınır ve "randomMatrix" fonksiyonuyla rastgele bir matris oluşturulur. Ardından kullanıcıya manuel veya otomatik oyun modunu seçme seçeneği sunulur. Kullanıcının seçimine göre ilgili fonksiyonlar çağrılır ve oyun ilerler. Dosyadan matris okuma seçeneği seçildiğinde kullanıcıdan dosya adı alınır ve "readFromFile" fonksiyonuyla dosyadan matris okunur. Yine kullanıcıya manuel veya otomatik oyun modu seçeneği sunulur, seçimine göre ilgili fonksiyonlar çağrılır ve oyun ilerler. Kullanıcı skorlarını gösterme seçeneği seçildiğinde, kaydedilen kullanıcı adları ve skorlar ekrana yazdırılır. Çıkış seçeneği seçildiğinde döngü sonlandırılır ve program sonlanır. Ayrıca oyunun ilerleyişi sırasında kullanıcı adları, skorlar, hamle sayıları gibi veriler değişkenlerde saklanır ve "calculateScore" fonksiyonuyla oyunun sonunda bir skor hesaplanır. Bu şekilde "main" fonksiyonu, oyunun akışını yöneten ve kullanıcının seçimlerine göre ilgili fonksiyonları çağıran bir yapıya sahiptir.

3.Algoritma Karmaşıklığı

Manuel mod için algoritma karmaşıklığı $O(n^2)$ 'dir, Çünkü kullanıcıdan giriş alırken tüm hücrelere erişilir ve move veya undo fonksiyonları çağrılabilir. Otomatik mod için de algoritma karmaşıklığı $O(n^2)$ 'dir. Çünkü iki for döngüsü ile sıfırdan boyuta kadar döngüde kalınır.

4.Sonuç

Bu çalışmada, NxN boyutundaki matris üzerindeki sayı eşleştirme oyunu için bir algoritma geliştirildi. Manuel modda oyun başarıyla oynanabilmektedir, ancak otomatik modda bazı sorunlar ortaya çıkmaktadır. Otomatik mod, beklenen sonuçları üretme konusunda zorluklar yaşamakta ve hatalar yapmaktadır. Bu sorunların kaynağını ve çözümünü belirlemek için farklı bir algoritma geliştirilmesi düşünülebilir. Özetle, manuel modda oyun başarılı ve doğru sonuçlar elde edilebilirken, otomatik modda ise hatalar meydana gelmektedir. Bu durumun üzerinde çalışarak otomatik modun geliştirilmesi gerekmektedir. Aynı zamanda manuel modda da oyunu daha hızlı oynayabilmeyi sağlayabilecek geliştirmeler yapılmalıdır.

1						
2		3				3 1
		4				
				2		4
5						5

1. Adım

Lutfen birlestirmek istediginiz noktanin bulundu gu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 0
Lutfen yerlestirmek istediginiz sayiyi giriniz. 1

1		1				
2		3				3 1
		4				
				2		4
5						5

2. Adım

Lutfen birlestirmek istediginiz noktanin bulundu gu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 0 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 1

1		1		1			
2		3				3 1	
		4					
				2		4	
5						5	

3. Adım

Lutfen birlestirmek istediginiz noktanin bulundu gu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 0 3
Lutfen yerlestirmek istediginiz sayiyi giriniz. 1

1		1		1		1		
2		3		3		3 1		
		4						
				2		4		
5						5		

4. Adım

Lutfen birlestirmek istediginiz noktanin bulundu gu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 0 4
Lutfen yerlestirmek istediginiz sayiyi giriniz. 1

1		1		1		1		1	
2		3		3		3		3 1	
		4							
				2		4			
5						5			

5. Adım

Lutfen birlestirmek istediginiz noktanin bulundu gu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 1 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 3

1		1		1		1		1	
2		3		3		3		3 1	
		4							
				2		4			
5						5			

6. Adım

Lutfen birlestirmek istediginiz noktanin bulundu gu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 2 0
Lutfen yerlestirmek istediginiz sayiyi giriniz. 2

Şekil 8: Oyunun adımları-1

1	1	1	1	1	1
2	3	3	3	1	1
2	4				
			2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 3 0
Lutfen yerlestirmek istediginiz sayiyi giriniz. 2

7. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4				
2			2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 3 1
Lutfen yerlestirmek istediginiz sayiyi giriniz. 2

8. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4				
2	2		2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 2 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 4

9. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4				
2	2		2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 2 3
Lutfen yerlestirmek istediginiz sayiyi giriniz. 4

10. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4			
2	2	2	2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 1
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

11. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4		
2	2	2	2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

12. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4		
2	2	2	2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 2 4
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

13. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4	4	
2	2	2	2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 1
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

14. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4	4	
2	2	2	2	4	
5	5				5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

15. Adım Son Hali

Şekil 9: Manuel oyunun adımları-2

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4	4	
2	2	2	2	4	
5					5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 1
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

13. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4	4	
2	2	2	2	4	
5	5				5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

14. Adım

1	1	1	1	1	1
2	3	3	3	1	1
2	4	4	4	4	
2	2	2	2	4	
5	5	5	5	5	5

Lutfen birlestirmek istediginiz noktanin bulunduđu ilk ve ikinci indisi giriniz.
Ornek: Once satir sonra sutun olacak sekilde (0,1) noktasinda ise 0 1 yaziniz. 4 2
Lutfen yerlestirmek istediginiz sayiyi giriniz. 5

15. Adım Son Hali

Şekil 10: Manuel oynanan bir oyunun sonucu