



Öğrencinin;

ADI: Ebrar
SOYADI: ŞAHİN
NO: 1321221005
BÖLÜM: Bilgisayar Mühendisliği

Dersin;

ADI: İleri Programlama Teknikleri
EĞİTMEN: Prof. Dr. Mehmet Akif Eyler

Projenin Amacı:

Sayıları küçükten büyüğe sıralamak.

Kural: Sadece 0'ın yanındaki sayılar hareket edebilir.

Projede Yer Alan Ekranlar

- Ana Ekran



Oyunu açan kişi bu ekrandan istediği seviyeyi seçiyor ve oynama başlıyor.

Bu ekranda nelerden faydalanıldı?

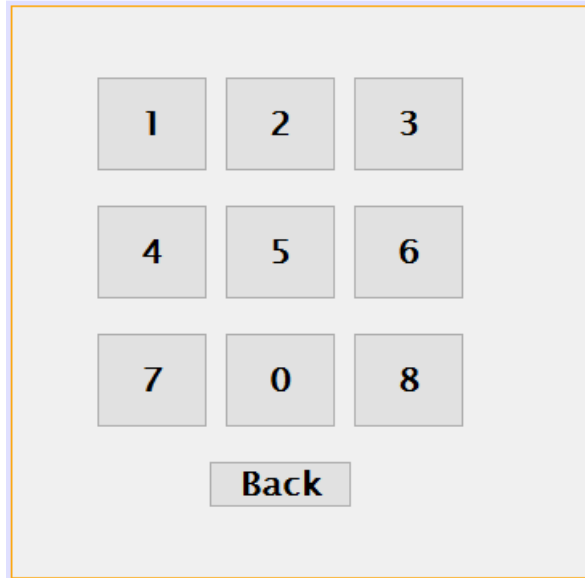
Swing API

java.io.BufferedWriter

java.io.File

java.io.FileWriter

- Easy Ekranı



Bu sayfada kişi 3x3 lük butonları sıraya koymaya çalışıyor. Sayfaya butonlar 0' dan 8' e kadar *randomize2()* metodu ile karıştırılıp getiriliyor. Oyunun bitip bitmediği *finish()* metodu ile kontrol ediliyor.

Bu ekranda nelerden faydalanıldı?

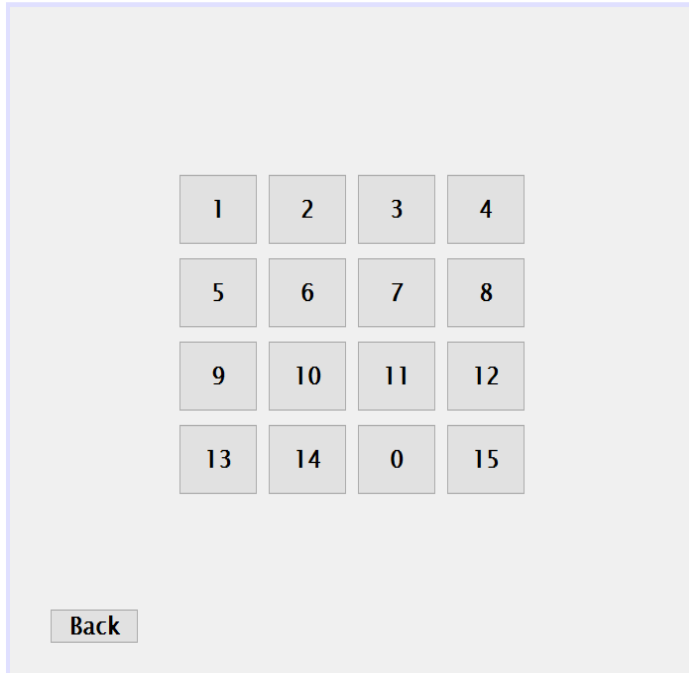
IGame (Interface)→randomize2(), finish()

Swing API

java.util.ArrayList

java.util.Random

- **Medium Ekranı**



Medium radio button seçilirse;

Bu ekranda kişi 4x4 lük butonları sıraya dizmeye çalışıyor. Sayfaya butonlar 0' dan 15' e kadar *randomize2()* metodu ile karıştırılıp getiriliyor. Oyunun bitip bitmediği *finish()* metodu ile kontrol ediliyor.

Hard radio button seçilirse;

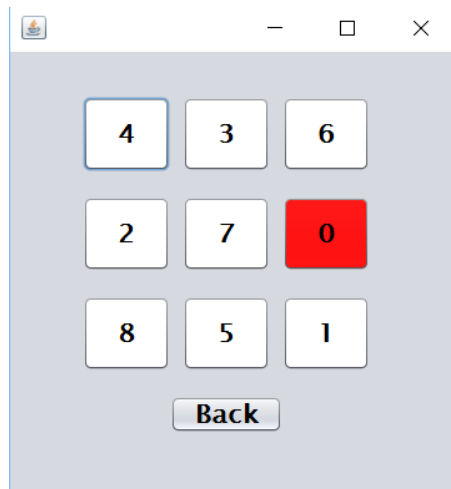
Yine karşımıza medium ekranı geliyor ama bu sefer kişi aynı zamanda zamana karşı yarışıyor. Hard radio button'ı seçildiği zaman ana ekran'da bulunan static bir değişken olan *hardIsClicked*' ın değeri 1 arttırılıyor ve Medium sınıfında bir kontrolle Timer aktif hale getiriliyor.

- *returnButtons()* metodu panelde bulunan metodların getirilmesine yarıyor.

Bu ekranda nelerden faydalanıldı?

java.awt.Component
java.awt.event.ActionEvent
java.util.ArrayList
java.util.Random
javax.swing.Timer
Swing API
IGame (Interface) → *randomize2()*, *finish()*

- **AUTO Butonu**



Auto butonuna basılınca gelen ekrandaki butonları kendisi otomatik olarak sıraya diziyo. Projenin bu kısmında ilk önce 2 saniyede bir (Thread ile) 0' dan 8' e kadar bir değer üretiriyordum ve program *finish()*(o zaman finish metodunun türü boolean idi.) metodundan true değeri dönene kadar devam ediyordu. Fakat bu yol çok uzun sürüyordu. Biraz araştırmayla **A*Search** algoritması ile bu tarz problemlerin kolaylıkla çözüldüğünü buldum. Bu algoritmayı projeme uyguladım. Çözüm, gelen matrise göre uzun ya da kısa olabiliyordu. Her değişiklik olduğunda ortaya çıkan matrisi ilk önce bir tane *jTextArea*' ya sonra bir text file' a yazdırdım. **A*Search** algoritması işini bitirince yazdığım **myThread** adlı sınıfta yazılan text

file'dan satır satır okuma yaparak her satırda gelenleri 2 saniye aralıklarla ekranda gösterdim. Daha belirgin olması için 0' ın arka planını kırmızı hale getirdim.

Nelerden faydalanıldı?

java.awt.Color
java.io.BufferedReader
java.io.File
java.io.FileReader
java.util.ArrayList
java.util.LinkedList
java.util.Queue
java.util.Stack
Swing API

EK AÇIKLAMALAR

1. Bu projenin temeli birinci sınıfta verilen bir ödevi kapsamaktadır. O zamanki ödev kapsamında proje şunları içeriyordu:
 - 3x3'lük butonlar
 - Toplam kaç defa butonlara tıklandı
 - Bitti mi kontrolü
 - Butonların üzerindeki sayılarda rastgele gelmiyordu.

Yukarıdekiler hariç geriye kalanlar İleri Programlama Teknikleri dersi dönem ödevi için yapılmıştır.

2. Projeden elde edilecek .jar uzantısının 100 KB' ı geçmemesi şartından ötürü arka plan resimleri kaldırılmıştır. (Aksi halde 1000KB'ı bulmuştu.)
3. *Randomize2()* metodları sunum için ve kontrol daha kolay olsun diye yorum satırı haline getirilmiştir. Nedeni: Oyunu oynadım ve kısa sürede bitiremedim.