



EGE ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

NESNEYE DAYALI PROGRAMLAMA

OKAN KAYA 05110000094

IRIT MAULANA SAPTA 05120000816

OĞUZHAN SALTİK 05110000010

Öğretim Üyesi :

Yar.Doçent Rıza Cenk Erdur

İçindekiler

NESNEYE DAYALI PROGRAMLAMA.....	
Projenin Tanımı	
Kullanılan Tasarım Desenleri	
FACTORY	
STRATEGY	
PROTOTYPE.....	
COMMAND I	
COMMAND II	
Programın UML Diagramı:	
Kullanıcı Kataloğu	

Projenin Tanımı

Rakam yerleştirme olarak da bilinen SUDOKU, standart olarak 9x9 boyutlarında bir diyagramda

çözülen ve her satır, her sütun ve her 3x3'lük karede 1'den 9'a rakamların birer kez yer alması gereken

bir zeka oyunu türüdür

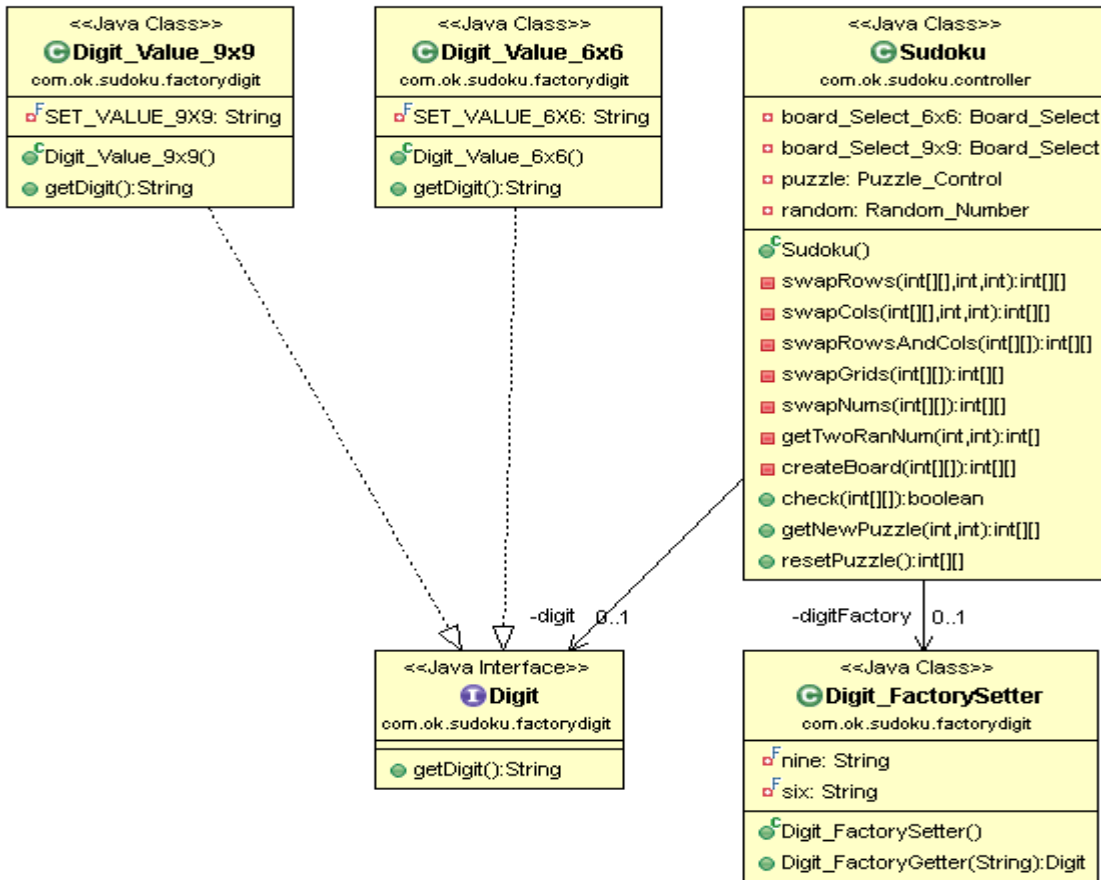
Proje kapsamında, SUDOKU bulmacaları üreterek manuel olarak çözmesi için kullanıcıya sunan bir

masaüstü uygulaması geliştirmeniz istenmektedir.

Kullanılan Tasarım Desenleri

FACTORY

Fabrika yordam tasarım deseni, nesne yaratma sorumluluğunun bir yordama verilmesidir. Yaratılan nesne, bir sınıf hiyerarşisindeki alt sınıflardan biridir. Hangi alt sınıfın yaratılacağı kararı fabrika yordam içinde verilir. Bu yordam ile belirli bir sınıf hiyerarşisindeki alt nesnelerden birinin yaratma sorumluluğu belirli bir arayüze verilerek sistemden soyutlanmış olur. Böylece nesneleri yaratma kodlarında, kod tekrarları önlenmiş olur. Sistem içinde sınıfların yaratılacağı yer tek olduğu için, ilgili mantıklar tek bir yerde toplanabilir.

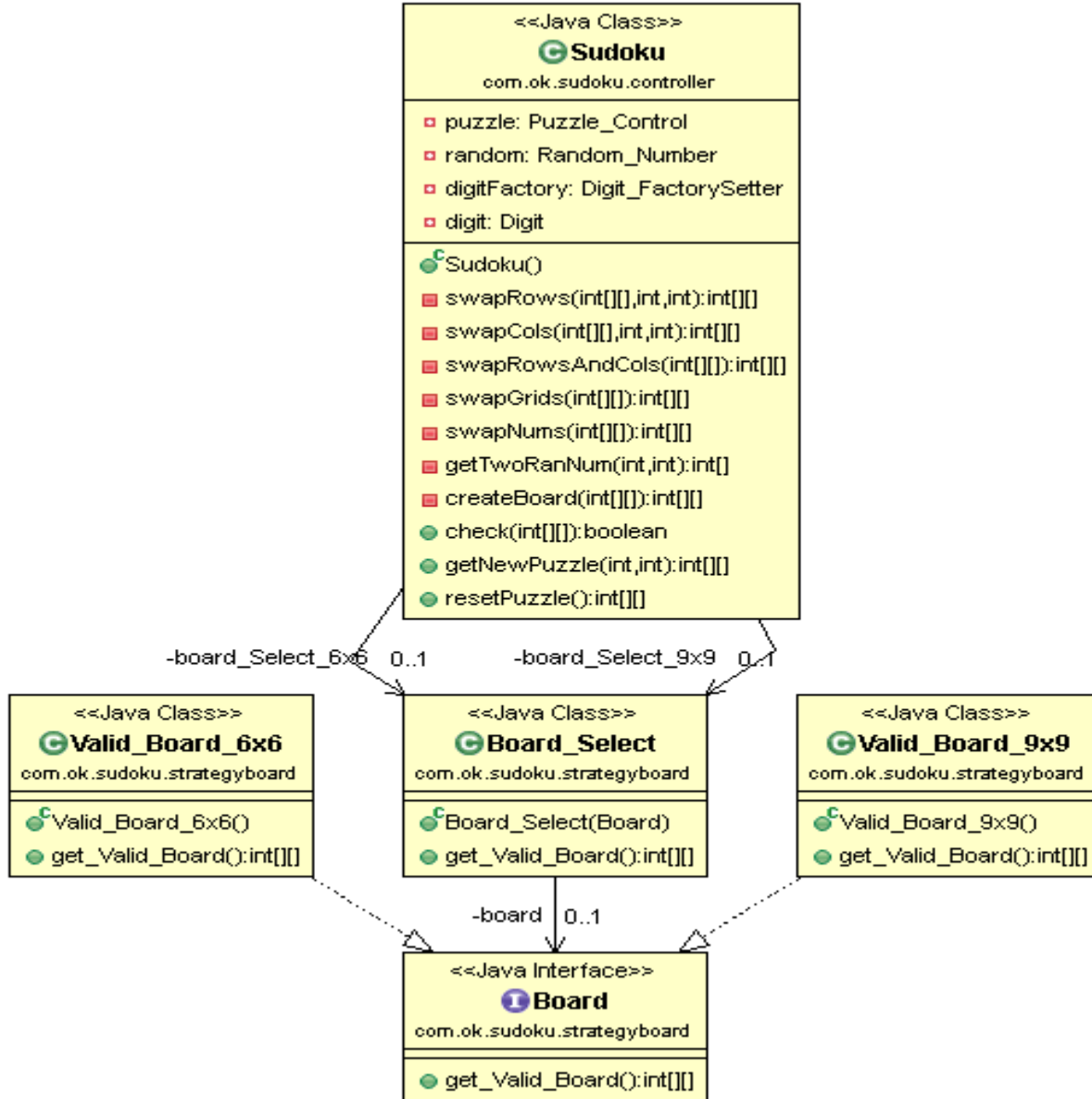


Programda kullanıcıya 6 ve 9 olmak üzere iki çeşit oyun türü sunduk . Sudoku Classında kullanıcının secemine göre 6 ve 9 lu sudoku oyunu seçilecektir. Bunun için Digit interface Class oluşturduk ve bu class ı implements eden iki Class oluşturduk .

En son olarak bu classların oluşturulmasından sorumlu Digit_FactorySetter Classını oluşturduk. Factory Pattern sayesinde gereksiz kod tekrarından kurtulmuş olduk ve tek bir Classta ilgili Digit verilerini üretmiş olduk.

STRATEGY

Bir işlem için farklı yöntemlerin uygulanabilir olduğu durumlarda, bu yöntemi kullanacak olan nesne, hangi yöntemin uygulanacağını seçer. Çünkü bu içerik nesnesi, yöntemleri belirleyen üst sınıfı içerir. Farklı yöntem veya strateji alt sınıfları da, bu üst sınıftan türerler. Bu tasarım deseniyle, yöntemin nasıl uygulanması gerektiği ile ilgili detaylar, bu yöntemi kullanacak nesneden ayrılmış olur. Ayrıca bu tasarım deseninin kullanılmasıyla, kod uzun "if/else" veya "switch" ifadelerinden ayıklanır.



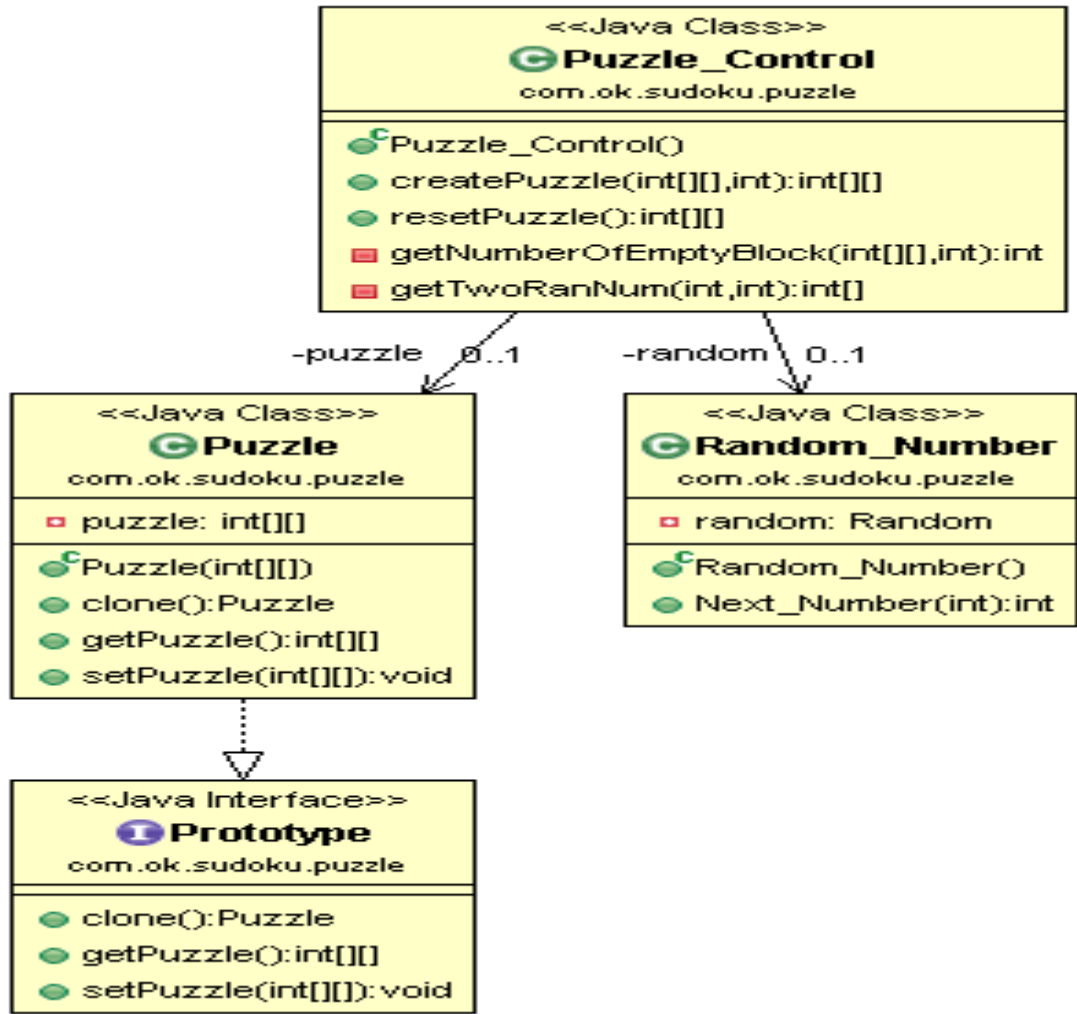
Kullanıcının seçmiş olduğu oyun turune göre Boardları oluşturmak için Strategy Patterni kullandık .Bu sayede gereksiz kontrol birimlerinden kurtulmuş olduk.Ayrıca Board İnterface sini implements etmiş iki Class in Sudoku Clasındaki yaratılma şeklini teke indirgemiş olduk ve gereksiz kod tekrarından kurtulmuş olduk.

PROTOTYPE

Var olan bir nesneden, kopyalama yöntemi ile yeni nesne yaratmak için bu tasarım deseni kullanılır. Nesne yaratmak için "new" operatörü kullanılmaz. Yazılım dillerindeki "clone" gibi, nesne kopyalama yordamlarından faydalanılır.

Örneğin java programlama dilinde, bir nesnenin kopyalanabilmesi için, nesnenin "Cloneable" arayüzünden türemiş olması gereklidir.

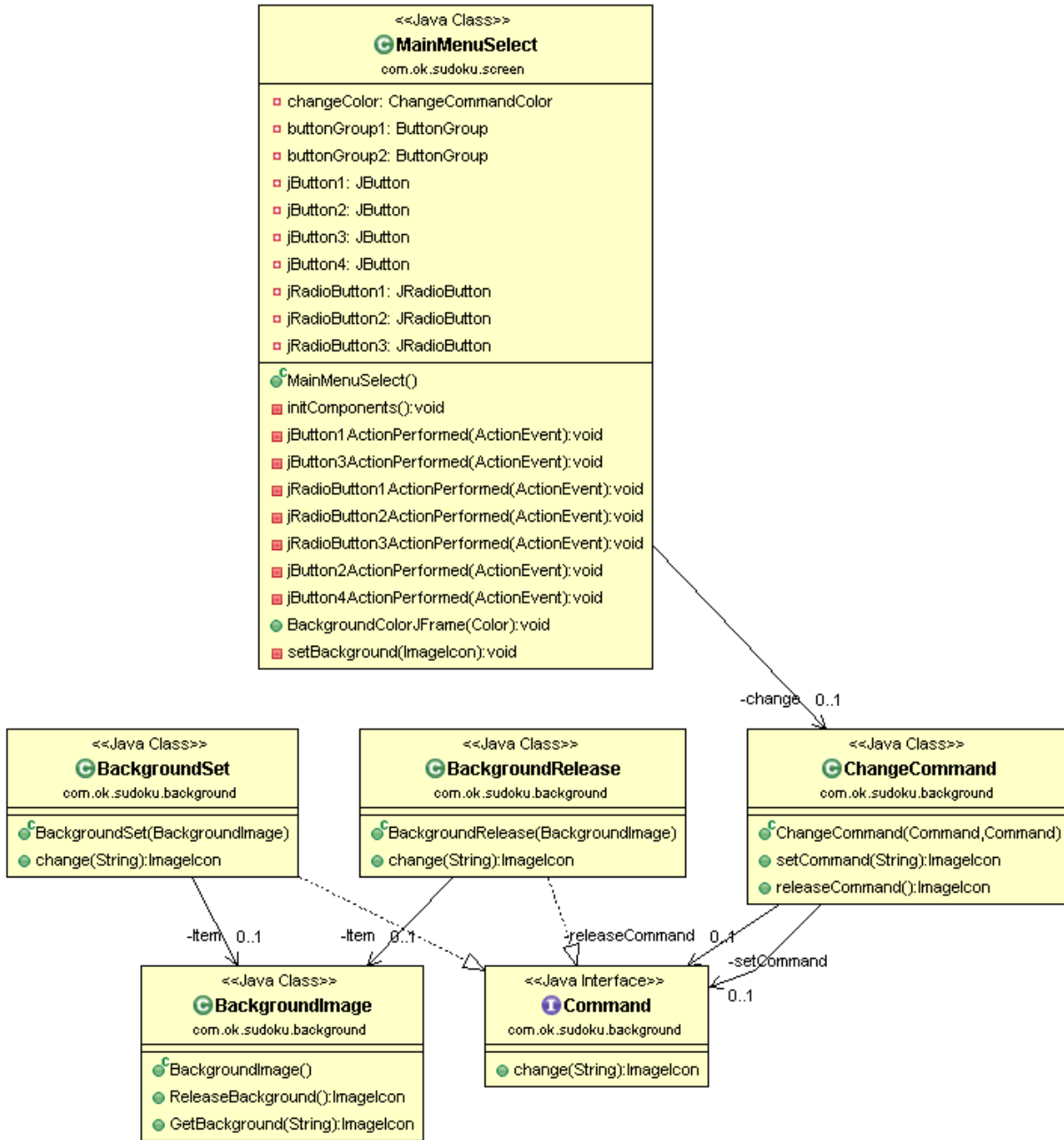
Bazı nesneler çok büyük olup, çok kaynak harcıyor olabilir. Bu nesnelerin yeniden yaratılması yerine, kolonlanması performans ve kaynak kullanımı açısından tasarruf edilmesine yarar. Bu durumlarda "Kopya Nesne" tasarım deseni kullanılabilir.



Prototype Patternin kullanılma sebebi Puzzle nesnesinin oyun içinde birçok defa kopyasının yaratılması ve bu yüzden gereksiz kaynak kullanımına sebep olmasıdır. Bu durumu onlemek için Prototype interfacesini implements eden Puzzle nesnesi içerisinde clone metodu kullandık . Bu sayede gereksiz kod tekrarı ve fazla kaynak kullanımından kurtulmuş olduk.

COMMAND I

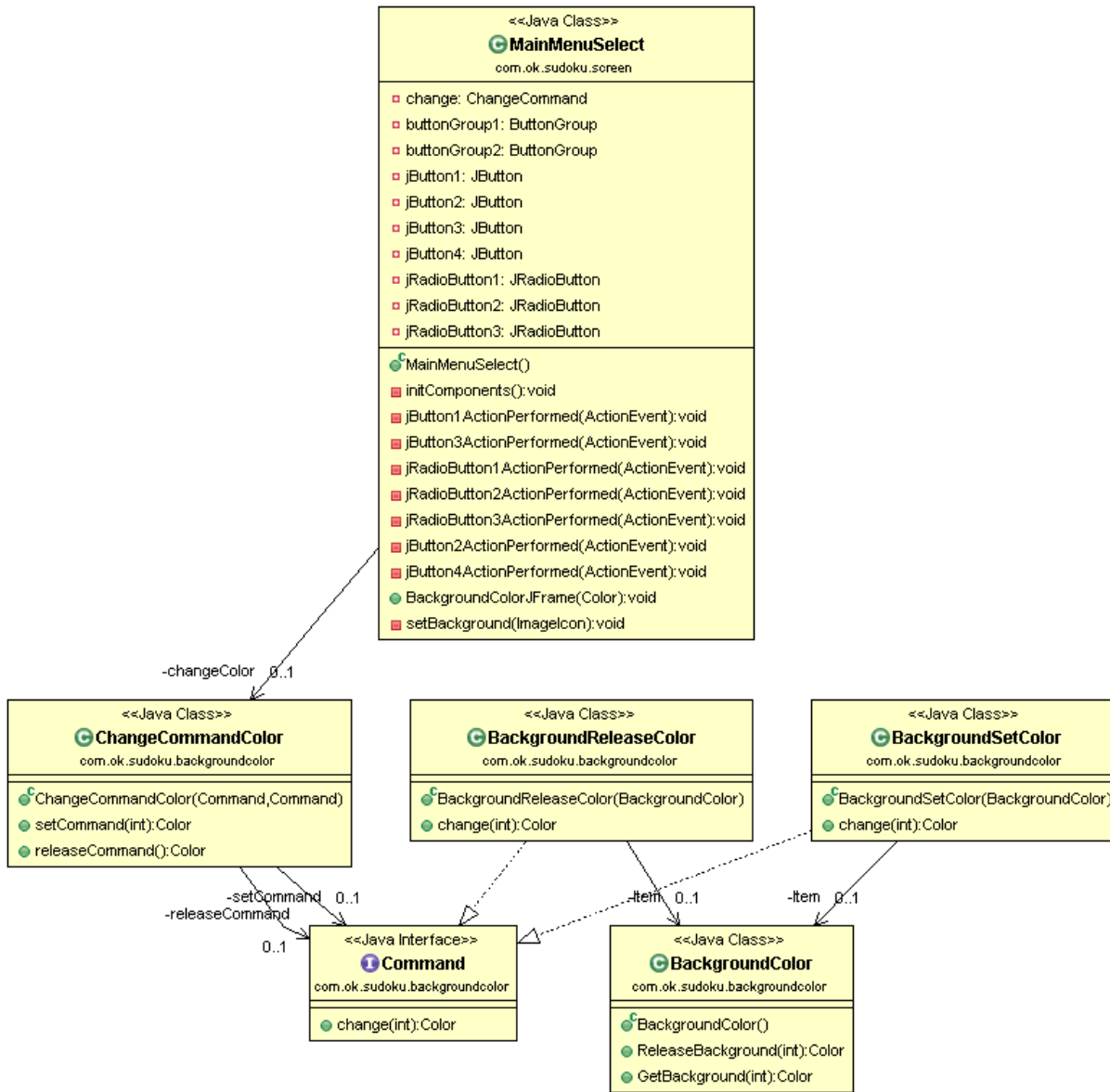
Bu tasarım deseni, nesneler bir işlevi ve bu işlev için gerekli değişirgeleri içerirler. Dışarıdan bu nesneler tetiklenerek bazı işlemler gerçekleştirilir. Bu tasarım deseniyle, işlemi tetikleyecek nesneler ile işlemi yapan nesneler birbirlerinden ayrılmış olur. İşlemi yapacak nesneler birden fazla olabilir ve bunlar koleksiyonlarda saklanabilir. Bu nesnelerin hepsi birden sırayla çalıştırılabilir, böylece istenen fonksiyon, birden fazla işlevin çalışmasıyla gerçekleştirilmiş olur. Yeni işlevler istendiğinde, yeni komut nesneleri de kolayca eklenebilir.



Kullanıcının programın arkaplan resmini değiştirmesi için Command patternini kullandık bu sayede işlemi tetikleyecek nesneler ile işlemi yapacak nesneleri bir birinden ayırmış olduk.

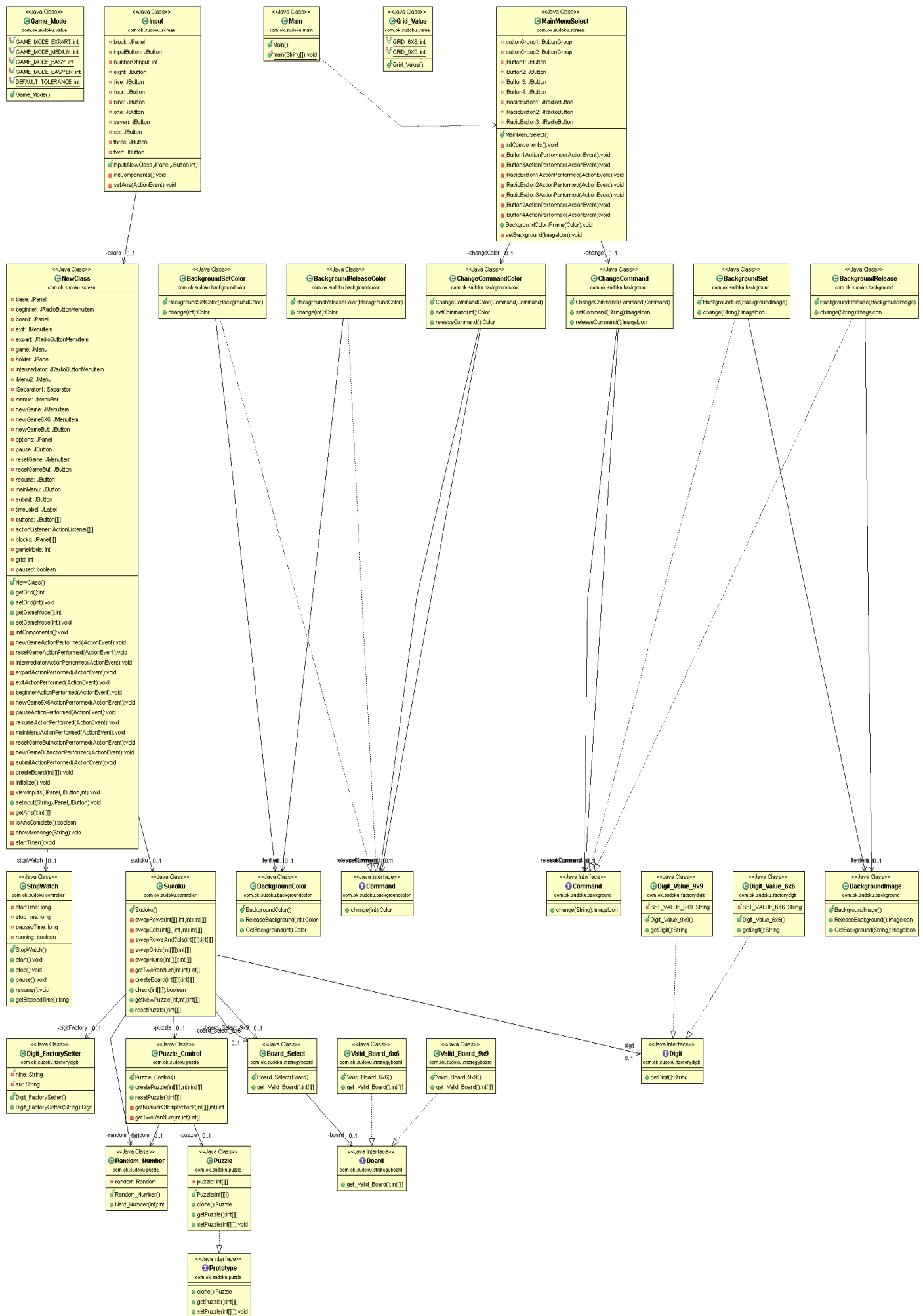
Kullanıcı arkaplan resmini değiştirmesi için mainmenuselect nesnesinden ChangeCommand nesnesini yaratarak istediği resmi silebilir veya değiştirebilmektedir.

COMMAND II



Kullanıcının programın arkaplan rengini değiştirmesi için Command patternini kullandık bu sayede işlemi tetikleyecek nesneler ile işlemi yapacak nesneleri bir birinden ayırmış olduk, Ayrıca işlemi yapacak nesneleri tek bir nesne altında topladık bu sayede bir işlevi sırayla çalışabilir duruma getirdik. Kullanıcı arkaplan resmini değiştirmesi için mainmenuselect nesnesinden ChangeCommandColor nesnesini yaratarak istediği rengi söylebilir veya değiştirebilmektedir.

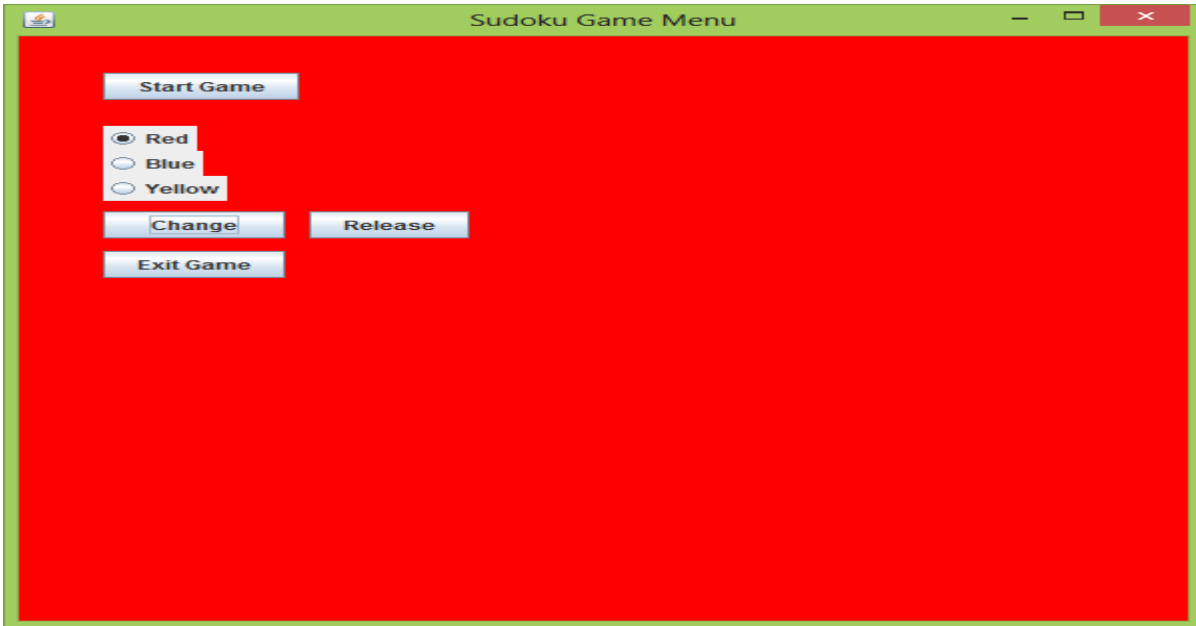
Programın UML Diagramı:



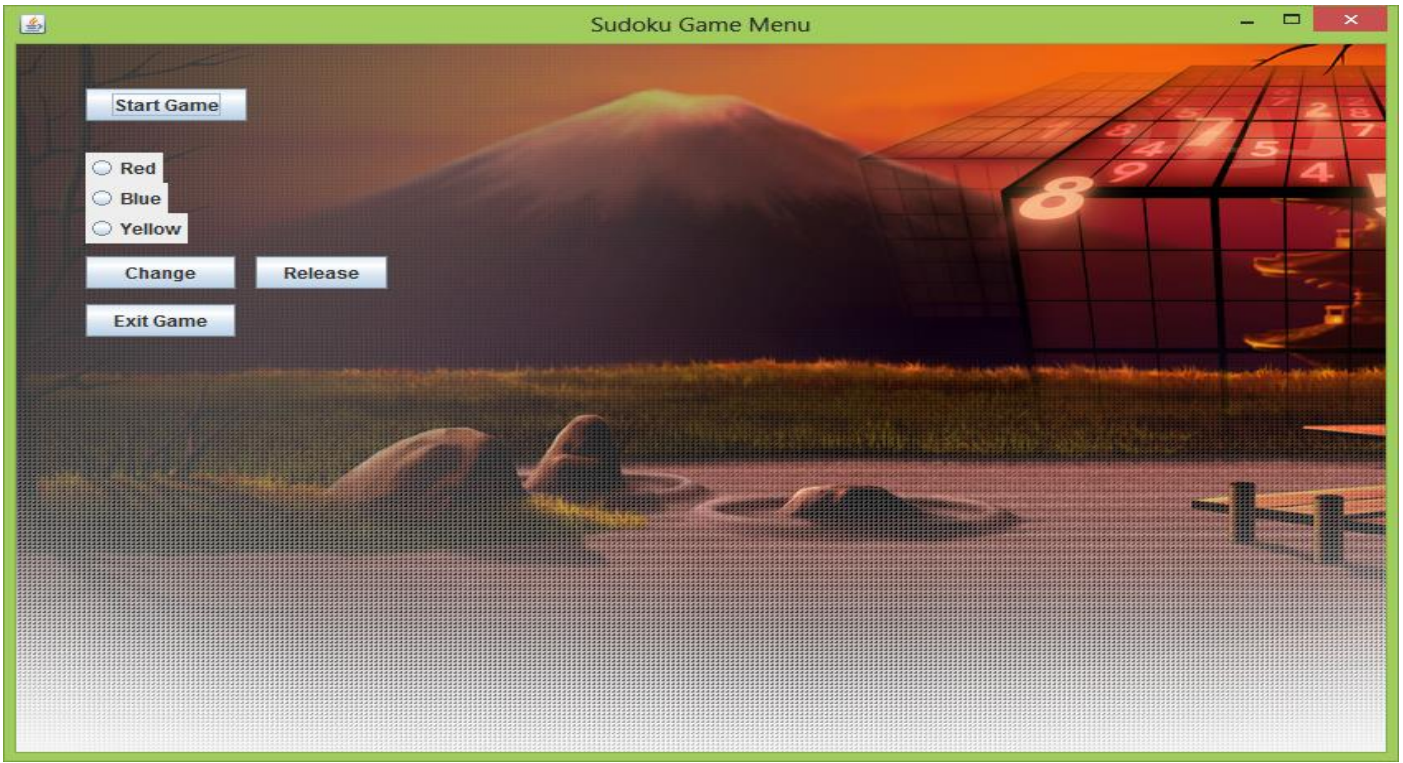
Kullanıcı Katalođu



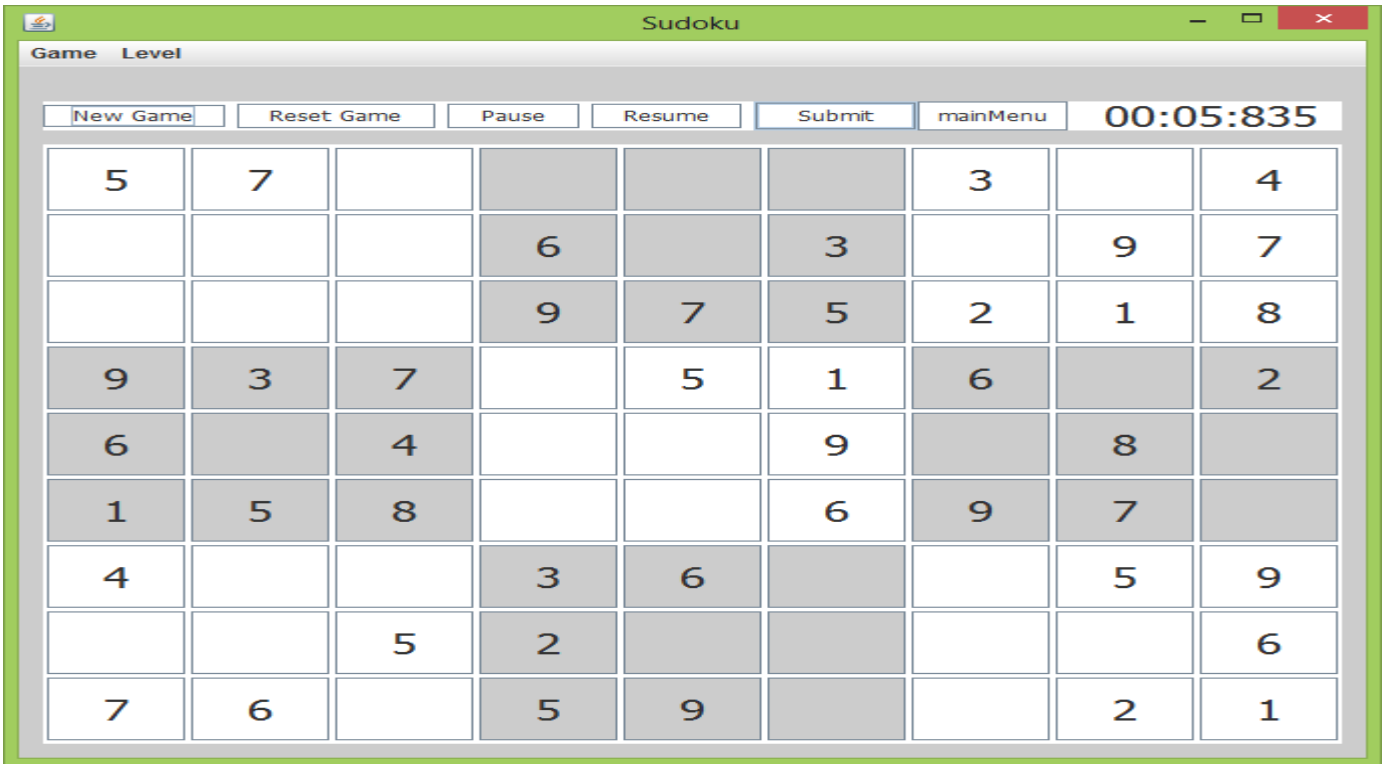
- Kullanıcı programı ilk çalıştırdığında karşısına yukardaki ekran gelmektedir
- Bu ekranda oyuna giriş yapabilme ,arka plan rengini değıştirme ve oyundan çıkmak için butonlar vardır.



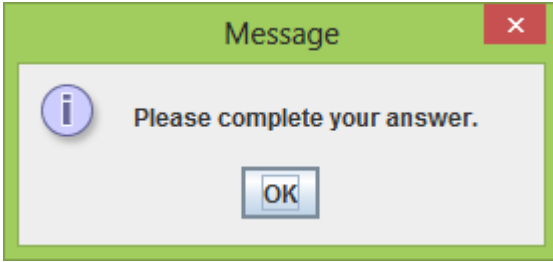
- Kullanıcı seçeneklerde belirtilen renkleri arka plan rengi yapabilmektedir.



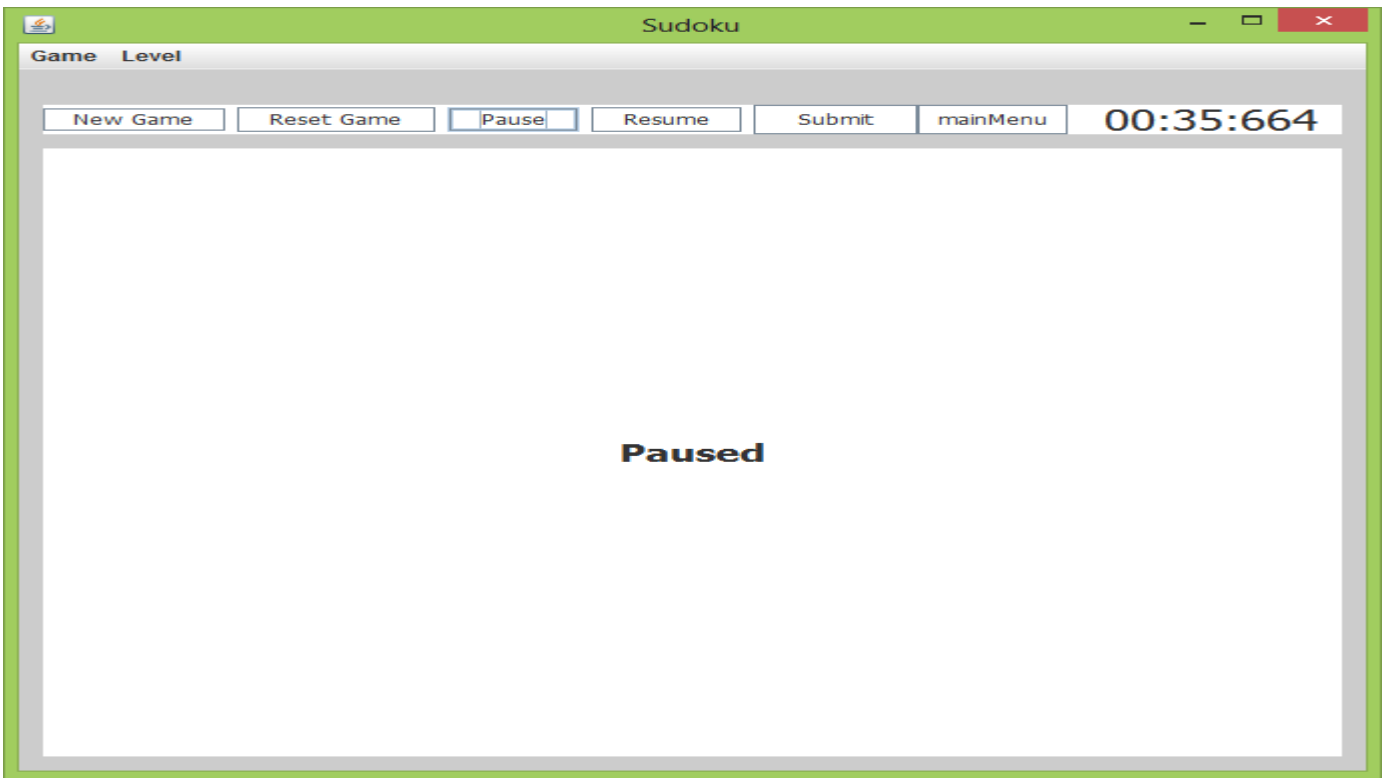
- Kullanıcı arka plan resmini ayarlaya bilmektedir.



- Kullanıcı oyuna başladığın da karşısına gelen ekran görüntüsü
- Burada kullanıcı yeni bir oyuna başlaya bilmektedir.
- Oyunu resetleye bilmektedir.
- Oyunu durdurabilmektedir.
- Oyunu başlatabilmektedir.
- Oyunun doğru olup olmadığına bakabilmektedir.
- Çıkış yapabilmektedir.
- Ana menüye geri dönebilmektedir.
- Oyunun zorluk seviyesini ayarlayabilmektedir.



- Kullanıcı oyunun bitip bitmediğini kontrol ettiğinde ve yanlış sudoku girdiğinde karşısına çıkan ekran



- Kullanıcı oyunu durduğunda karşısına çıkan ekran

