**KARE PUZZLE OYUNU**

160201015 Nilüfer TOKDEMİR , 160201063Özge SOĞUKPINAR

Bilgisayar Mühendisliği Bölümü  
 Kocaeli Üniversitesi

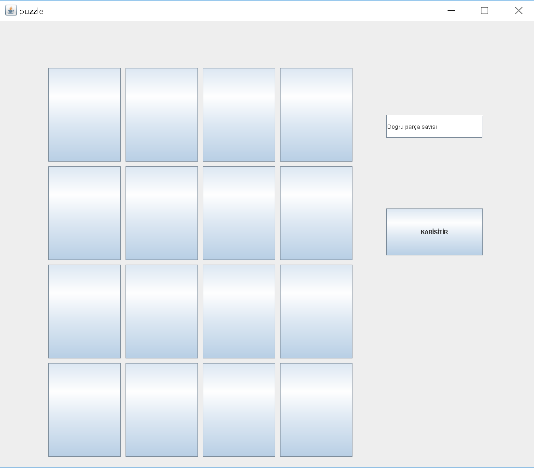
[tokdemirnilufer@gmail.com](mailto:tokdemirnilufer@gmail.com), sogukpnar.ozge@gmail.com

**Özet**

*Proje bizden istenildiği üzere bir puzzle çözme oyunu şeklinde tasarlanmıştır.Okunan herhangi bir resim dosyası 16 eşit parçaya bölünmüş ve yerlerinin karıştırılması işlemi gerçekleştirilmiştir. Sonrasında ise istenilen iki parçanın yerdeğiştirmesi sağlanarak yerleri değiştirilen parçaların doğru yerde olup olmadığı kontrolleri yapılmıştır. Oyunda tasarımcılara ait bir puanlama sistemi vardır ve oyun bittiğinde kullanıcının aldığı puan kendisine gösterilmiştir.*

**1)GİRİŞ**

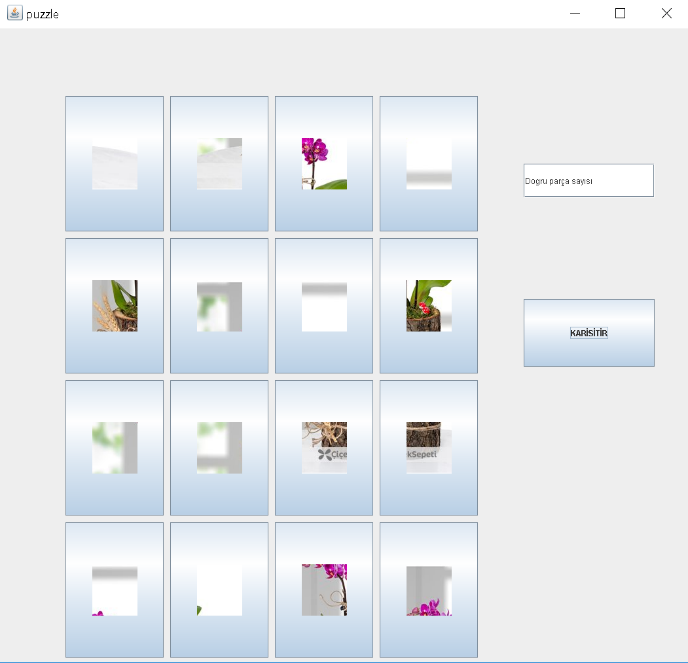
Program çalıştırıldığında istenilen resim dosyası okunur ve işlemler başlar. Kullanıcıyı ilk olarak içerisinde resim olmayan on altı tane boş buton karşılar. Bu on altı tane boş butonun hemen yanında “karistir” adında bir buton daha bulunmaktadır. Karıştır butonuna tıklandığı anda seçilen resim 16 eş parçaya ayrılır ve yerleri karışık olacak şekilde bu boş on altı butona yerleştirilir. Karıştır butonu en az bir resim parçası doğru yere gelene kadar aktif haldedir. En az bir parça doğru yere geldiği takdirde karıştır butonu pasif duruma geçer ve kullanılamaz hale gelir.



**2)TEMEL BİLGİLER**

**2.1 Resmin Parçalanması**

Resmin kaç satır ve kaç sütuna bölüneceği bilgisi alınmıştır. Alınan bu iki bilgi ile resmin kaç parçaya bölüneceği bilgisi elde edilmiş olur. Parçalanmış olan resimlerin boyut belirleme işlemleri, asıl resmin boyutlarının satır ve sütun sayılarına bölünerek elde edilmiş olur. Bazı birkaç method ve elde edilen boyutlar sayesinde resim istenilen sayıda parçalanıp BufferedImage formatında bir dizide saklanır.

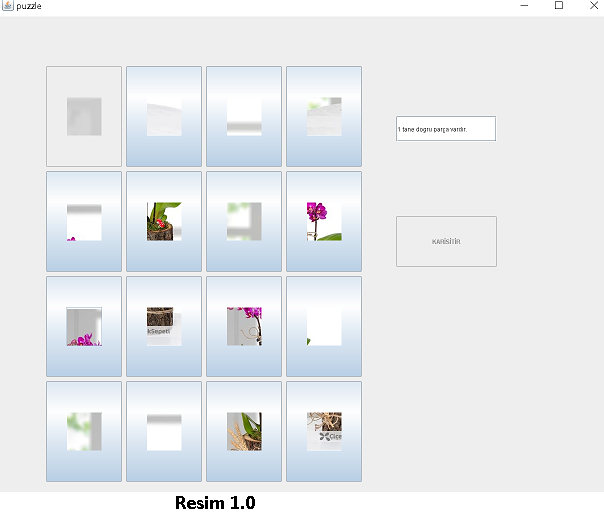
****

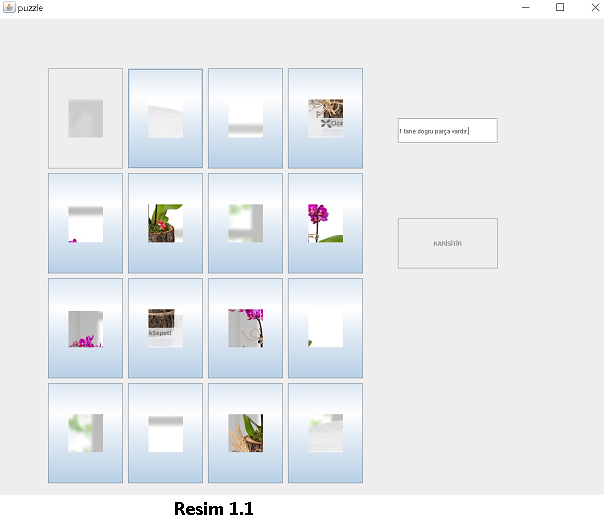
**2.2 Resmin Karıştırılması**

Sıfır ile on altı arasında birbirinden farklı ve random olarak atılan sayılar ile bir dizi oluşturulmuştur. Bu dizi sayesinde BufferedImage fomatındaki dizinin elemanları çekilip yeni bir BufferedImage diziye atılmıştır. Yeni oluşturulan bu dizinin elemanlarını oluşturan resimler ise on altı tane butona sırayla yerleştirilmiştir.

**2.3 Resimlerin Yer Değiştirilmesi**

Butonlar ActionListener methodu ile dinlenir ve ilk basılan butondaki resim ve bu buton bir değişkende tutulur. İkinci kez basılan butonda da resim ve hangi buton olduğu bir değişkende tutulur. Yine burada daha önce basılan butondaki resim ikinci basılan butona,ikinci basılan butonda bulunan resim de ilk basılan butona yüklenir. Böylece sırayla basılan iki butondaki resimlerin yerleri değiştirilmiş olur.





Resim 1.0 ve resim 1.1 de görüldüğü üzere sağ üst ve sağ alttaki resimlerin yerleri değişmiştir.

**3)KONTROL BÖLÜMÜ**

**3.1 Referans Klasörün Oluşturulması**

Yer değiştirme ve karıştırma işlemlerinden sonra doğru parçaların yerini belirlemek amacı ile resimlerin doğru sırada bulunduğu bir referans diziye ihtiyaç olmuştur. Ele alınan resmin ilk olarak on altı parçaya bölünme işleminden hemen sonra resmin parçaları sıralı olarak bir klasör içinde saklanmıştır. Sonrasında gerçekleşecek olan karıştırma ve yer değiştirme işlemlerinin her birinden sonra karışık olan resim parçaları teker teker referans olarak ele aldığımız sıralı resim klasörü ile karşılaştırılmış ve böylece hangi parçanın doğru yerde olduğu bilgisine ulaşılmıştır.

**3.2 İki Resmin Karşılaştırılması**

Referans olarak alınan klasördeki resimler ile karışık halde bulunan resimlerin karşılaştırılması gerekmektedir. Farklı birçok yöntem ile karşılaştırma işlemi yapılabilmektedir. Geliştirdiğimiz bu projede resimlerin aynı olup olmadığının kotrolünü yaparken ilk olarak boyutlarını kontrol ettik.Eğer boyutlar aynı ise ikinci bir kontrol için her iki görüntünün RGB değerleri alındı ve iki resmin tüm pikselleri karşılaştırılıp fark olup olmadığı bilgisi elde edilmeye çalışıldı. Eğer pikseller arası fark sıfıra eşitse karşılaştırılan iki resim birbirinin aynısı demektir. Eğer herhangi bir pikselde farklılık varsa bu iki resim farklı resimler demektir.

**4)PUANLAMA**

Karıştır butonunu kullanarak kullanıcı en az bir doğru parça bulduktan sonra ilk karıştır butonuna basışta 16 da 16 dogru parçayı tutturmuşsa eğer kullanıcı100 puan alır.Kullanıcı karıştır butonuna 2 veya daha fazla basışta tüm butonlara doğru parçayı getirebilirse 100 puandan basış sayısının 10 ile çarpımı kadar bir puan sahibi olur.İdeal hamle sayısını ise karıştır butonuna basışta gelen doğru sayısı tüm parça sayısından çıkarıldıktan sonra kalan sayının iki ile bölümünden geriye kalan sayı olarak buluruz.Bu sayıyı geçmesi durumunda üç değişiklikte üç ve daha fazla değişiklikte ise beş puan kaybeder.Hatalı işlem yapılırsa beş puan daha kesilir.Eğer kullanıcı ideal işlem sayısını tutturur ise on puanlık bir bonus kazanır.Karıştır butonuna her basış ise on puan kaybına denk gelir.

**5) SONUÇ**

Proje genel anlamda istenilen sonuçları verdi.Büyük bir oranda başarılı oldu.Kullanıcının istediği resim on altı eşit parçaya bölünerek bir puzzle oyunu oluşturuldu. Gerekli yer ve hamle kontrolleri sayesinde puanlama işlemi gerçekleştirildi.

**5) KAYNAKÇA**

Bu projeyi geliştirirken iki resmin kıyaslanması konusunda bilgi edinmek için[1] internetteki siteden yardım alınmıştır.

[1]: https://www.geeksforgeeks.org/image-processing-java-set-14-comparison-two-images/

