

KARE PUZZLE OYUNU

İsmail Kocatürk, Zeynep Gül Aslan

Bilişim Sistemleri Mühendisliği Teknoloji

Fakültesi/Kocaeli Üniversitesi

201307045@kocaeli.edu.tr, 191307049@kocaeli.edu.tr

Özet—Bu ödev kapsamında, bir puzzle çözme oyunu tasarımı gerçekleştirilmesi beklenmektedir. Proje, kullanıcı arayüzü (GUI) hazırlanarak, masaüstü uygulaması veya web tabanlı olarak geliştirilebilir. Puzzle parça eşleştirmeleri için kullanılacak yöntemin bağlı liste veri yapısı olması zorunludur.

Projenin isteleri arasında, puzzle görselinin 16 parçadan oluşması, Github platformunun kullanımı, herhangi bir dosya veya URL'den resim/fotoğraf dosyasının okunması, parçaların karıştırılması, doğru yere yerleştirildiğinde parçaların kilitlenmesi, en yüksek skor bilgisinin arayüzde yer alması, doğru hamlelerin +5 puan, yanlış hamlelerin -10 puan olarak tasarlanması, puan durumunun arayüz üzerinden görüntülenebilmesi, en yüksek puanların azalan sırada gösterilmesi gibi özellikler yer almaktadır.

Projenin tamamlanması için bir isim istenmekte ve bu isim, skor metin belgesine kaydedilmektedir. Ayrıca, oyun boyunca kullanıcının hamle sayısı da kaydedilmekte ve en yüksek skorlar, her açılışta azalan sırada gösterilmektedir..

Anahtar kelimeler—GUI, Bağlı Liste, Puzzle

Abstract— Abstract—Within this assignment, it is expected to design a puzzle solving game. The project can be developed as a desktop application or web-based by preparing the user interface (GUI). The method to be used for puzzle piece matching must be a linked list data structure.

Among the requirements of the project, the puzzle image consists of 16 pieces, the use of the Github platform, the reading of the image / photo file from any file or URL, the mixing of the pieces, the locking of the pieces when placed in the right place, the highest score information on the interface, the correct moves +5 points, There are features such as designing wrong moves as -10 points, displaying the score status on the interface, displaying the highest scores in descending order.

A name is requested for the completion of the project and this name is recorded in the score text document. In addition, the user's number of moves throughout the game is also recorded, and the highest scores are displayed in descending order at each launch.

Keywords— GUI, Linked List, Puzzle

1. GİRİŞ

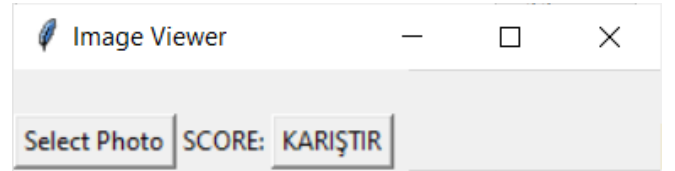
Puzzle çözme oyunları, akıl zekasını geliştirmek ve eğlenceli vakit geçirmek için popüler bir oyun türüdür. Bu ödev kapsamında, bir puzzle çözme oyunu tasarımı gerçekleştirilmiştir. Oyun, kullanıcı arayüzü hazırlanarak, masaüstü uygulamasına çevrilmiştir.

Projenin temel özellikleri arasında, 16 parçadan oluşan bir puzzle görselinin kullanılması, Github platformunun zorunlu olarak kullanılması, resim/fotoğraf dosyasının okunması, parçaların karıştırılması, doğru yerleştirilen parçaların kilitlenmesi, en yüksek skor bilgisinin arayüzde yer alması, doğru hamlelerin +5 puan, yanlış hamlelerin -10 puan olarak tasarlanması, puan durumunun arayüz üzerinden görüntülenebilmesi, en yüksek puanların azalan sırada gösterilmesi yer almaktadır.

Ödevin tamamlanması için, kullanıcının ismi ve hamle sayısı kaydedilmekte ve en yüksek skorlar her açılışta azalan sırada gösterilmektedir. Bu ödev kapsamında, puzzle parça eşleştirmeleri için bağlı liste veri yapısının kullanılması zorunludur.

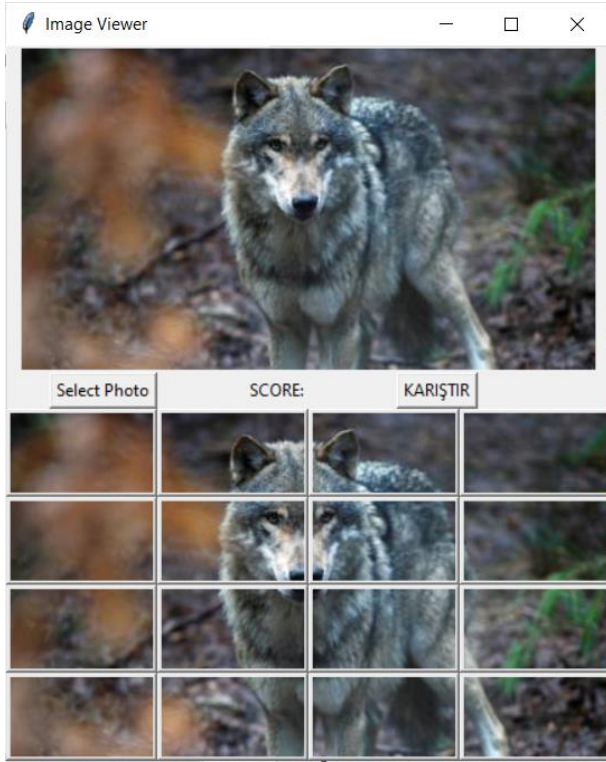
1.1 Kullanıcıya Fotoğraf Seçtirme ve Fotoğrafi Butonlar üstünde görüntüleme

Bu uygulama, kullanıcının bir fotoğraf seçmesine izin verir. Seçilen fotoğraf daha sonra 16 parçaya bölünür. Her bir parça, bir butonla temsil edilir ve kullanıcı butonları tıklayarak fotoğrafın farklı bölümlerini görüntüleyebilir.



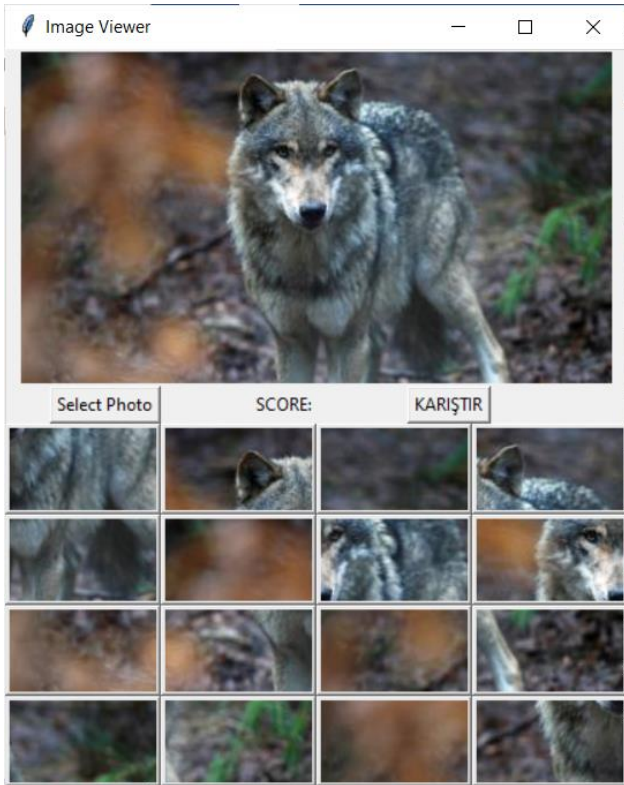
Şekil-1 Kullanıcının fotoğraf seçtiği yer.

Uygulamada kullanılan 'ImageViewer' sınıfı, uygulamanın ana yapısını oluşturur. Bu sınıf, '__init__', 'select_photo' ve 'display_photo' adlı üç yöntem içerir. '__init__' yöntemi, uygulamanın GUI yapısını oluşturur. 'select_photo' yöntemi, kullanıcının fotoğraf seçmesine olanak tanır ve 'display_photo' yöntemi, seçilen fotoğrafı ekranda gösterir.



Şekil-2 Fotoğrafın butonlar ile görüntülediği yer.

Ayrıca, seçilen fotoğrafın 16 parçaya bölünmesi için 'display_photo' yöntemi, 'PIL' kütüphanesi içindeki 'Image' sınıfını kullanır. Bu sınıf, seçilen fotoğrafın boyutlarını ve her bir parçanın boyutlarını hesaplar. Daha sonra, her bir parça, 'crop' yöntemi ile kesilir ve her bir butona yerleştirilir.

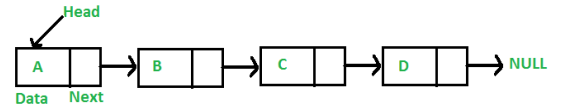


Şekil-3 Kariştir butonu.

1.2 Puzzle Parçalarını Eşleştirmek İçin Bağlı Liste Kullanımı

Puzzle parçaları, bağlı listelerdeki düğümler olarak temsil edilebilir. Her düğüm, parça hakkındaki bilgileri, örneğin şekli, rengi ve konumu gibi, içeren verileri içerir. Bir düğümden diğerine olan bağlantı, parçaların eşleştirilmesini temsil eder.

Bağlı liste kullanarak puzzle parçalarını eşleştirmek için, her bir puzzle için bir bağlı liste oluşturarak başlayabiliriz. Bağlı listenin her düğümü, parçayı temsil eder ve veri alanları, parça hakkındaki bilgileri içerir. Daha sonra, bağlı listeleri dolaşarak, her düğümün veri alanlarını karşılaştırarak eşleşmeleri bulabiliriz.

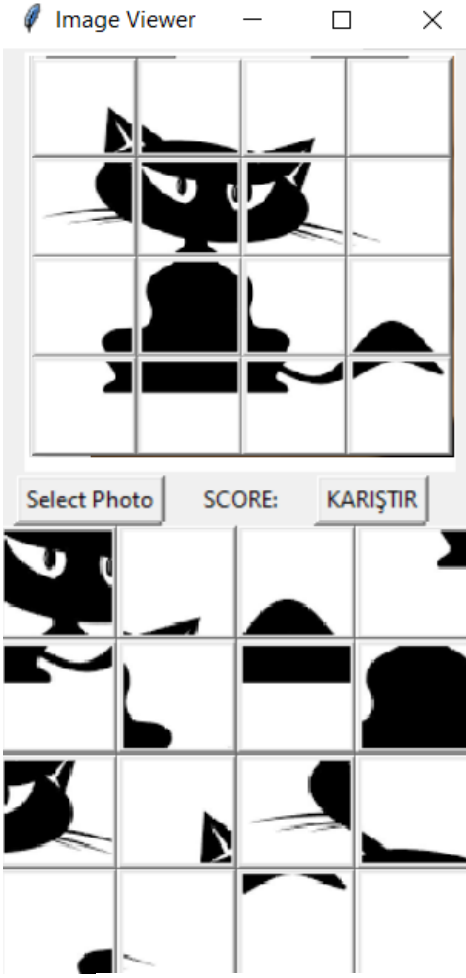


Şekil-4 Bağlı Liste.

Bir eşleşme bulunduğunda, düğümler arasındaki bağlantıları güncelleyerek parçaların eşleştirilmesini temsil edebiliriz. Örneğin, A düğümü B düğümüyle eşleşirse, A düğümündeki bağlantıyı B düğümüne, B düğümündeki bağlantıyı da A düğümüne yönlendirebiliriz. Bu, puzzle parçalarının eşleştirilmesini temsil eden bağlantılı düğümler zinciri oluşturur.

2. GUITASARIMI

GUI (Graphical User Interface), Grafiksel Kullanıcı Arayüzü, elektronik cihazların simgeler, ikonlar ve diğer görsel grafikler yardımıyla kullanmasına yardımcı olması amacıyla geliştirilmiş tasarımlardır. GUI öncesi komut bazlı işletim sistemi kullanılan eski nesil bilgisayarlar ve elektronik cihazlarda herhangi bir işlemi gerçekleştirmek için komut satırı kullanılıyordu. Kullanıcılar bilgisayar üzerindeki her işlemi klavye üzerinden ve komutları kullanarak gerçekleştiriyordu.



Şekil-13 GUI Tasarımı.

3. BENZER ÇALIŞMALAR VE LİTARATÜR TARAMASI

Puzzle çözme oyunları, yazılım geliştirme alanında sıklıkla kullanılan ve popüler olan bir oyun türüdür. Oyunların temel amacı, oyuncuların verilen parçaları doğru bir şekilde birleştirerek tam bir resim oluşturmalarını sağlamaktır.

Puzzle çözme oyunlarının, yazılım geliştirme sürecinde problem çözme ve mantıksal düşünme becerilerinin geliştirilmesinde önemli bir araç olduğu bilinmektedir. Bu oyunlar, yazılım geliştiricilerin kodlama problemlerine alternatif çözümler düşünmelerine ve daha etkili bir şekilde kod yazmalarına yardımcı olabilir. Bu ödevde de olduğu gibi bu tür oyunlar, yazılım geliştiricilerin algoritmaların ve veri yapılarının nasıl çalıştığını daha iyi anlamalarını sağlayabilir.

4. PROJEYİ GELİŞTİRİRKEN KULLANDIĞIMIZ YAZILIMSAL MİMARİ YÖNTEM VE TEKNİKLER

Projeyi geliştirirken Geleneksel Süreç Modellerinden Arttırımsal (Augmentative) Model'i kullandık.

Arttırımsal Modelde üretilen ve uygulamaya alınan her ürün sürümü birbirini içeren ve giderek artan sayıda işlev içerecek biçimde geliştirilmektedir. Öncelikle ürüne ilişkin çekirdek bir kısım geliştirilerek uygulamaya alınmakta, ardından yeni işlevsellikler eklenerek yeni sürümler elde edilmektedir. Tıpkı bu ödevi yaptığımız hafta boyunca her gün gelişimini izlememiz gibi.

Uygulama, kullanıcının bir fotoğraf seçmesine, fotoğrafı 16 parçaya bölmeye, bu parçaları karıştırmasına ve parçaları doğru sıraya yerleştirmesine olanak tanır. Ayrıca uygulama, kullanıcının puanını tutar ve doğru veya yanlış hamleler için kullanıcıya geri bildirim sağlar. Kodların ayrıntılı açıklaması şöyledir:

tkinter: Python'da grafiksel kullanıcı arayüzü (GUI) oluşturmak için kullanılan bir modüldür.

PIL: Python Imaging Library, Python ile görüntü işleme işlemleri yapmak için kullanılan bir modüldür.

random: Rastgele sayılar oluşturmak için kullanılan bir modüldür.

ImageViewer: Ana sınıf, kullanıcı arayüzünü oluşturur ve fotoğrafı görüntüler.

init: Yapılandırıcı metot, sınıf özelliklerini ve başlangıç durumunu tanımlar.

self.master: Tkinter penceresini tutar.

self.score: Başlangıç puanıdır.

self.image_label: Fotoğrafın görüntülediği etiket.

self.select_button: Kullanıcının fotoğraf seçmesine olanak tanıyan düğme.

self.score_label: Kullanıcının puanını görüntüleyen etiket.

self.mix_button: Parçaları karıştırmak için düğme.

self.grid_layout: 16 parçanın görüntülediği çerçeve.

self.piece_list: Parça indekslerini tutar.

select_photo: Fotoğraf seçme işlemini gerçekleştirir.

display_photo: Seçilen fotoğrafı görüntüler ve parçalara ayırır.

mix_pieces: Parçaları karıştırır ve yeni konumlara yerleştirir.

move_piece: Kullanıcının parçaları sürükleyip bırakarak hareket ettirmesine olanak tanır ve doğru veya yanlış hamleler için geri bildirim sağlar.

is_adjacent: İki parçanın bitişik olup olmadığını kontrol eder.

name == 'main': Kodların doğrudan çalıştırılıp çalıştırılmadığını kontrol eder ve uygulamayı başlatır.

5. KARŞILAŞTIĞIMIZ ZORLUKLAR

- Yapboz parçalarının yerini değiştirme özelliği olan `def move_piece(self, index)` fonksiyonunu yazmak.
- `Def move_piece` fonsiyonun da doğru cevap olduğunda parça kitleniyor fakat karıştır butonu ile sabit kalmıyor bu problemi de çözmeye çalışıyoruz.

6. KAYNAKÇA

- [1] [https://wmaraci.com/nedir/gui#:~:text=GUI%20\(Graphical%20User%20Interface\)%2C,yard%C4%B1mc%C4%B1%20olmas%C4%B1%20amac%C4%B1yla%20geli%C5%9Ftirilmi%C5%9F%20tasar%C4%B1mlard%C4%B1r.](https://wmaraci.com/nedir/gui#:~:text=GUI%20(Graphical%20User%20Interface)%2C,yard%C4%B1mc%C4%B1%20olmas%C4%B1%20amac%C4%B1yla%20geli%C5%9Ftirilmi%C5%9F%20tasar%C4%B1mlard%C4%B1r.)
- [2] www.notion.so
- [3] Her Yönüyle Python Fırat Özgül