



# Kare Puzzle Oyunu

Deniz Beltan  
Kocaeli Üniversitesi Teknoloji  
Fakültesi  
Bilişim Sistemleri Mühendisliği  
İstanbul, Türkiye  
201307016@kocaeli.edu.tr

Oğuzhan Öztürk  
Kocaeli Üniversitesi Teknoloji  
Fakültesi  
Bilişim Sistemleri Mühendisliği  
Kocaeli, Türkiye  
201307007@kocaeli.edu.tr

Kaan Yıldırım Eser  
Kocaeli Üniversitesi Teknoloji  
Fakültesi  
Bilişim Sistemleri Mühendisliği  
Kocaeli, Türkiye  
201307083@kocaeli.edu.tr

**Özet—** Kare puzzle oyununu geliştirmeden önce oyunun mantığına ve istelere uygun bir algoritma tasarlanması gerekmektedir. Bu algoritma, kullanıcının oyun alanında hareket ettirdiği görsel parçalarının konumlarını takip etmeli ve doğru şekilde yerleştirilip yerleştirilmediğini puanlama durumu ile yönlendirmelidir.

Kare puzzle oyunu kodları geliştirirken, kullanıcıların oyunu kolayca oynaması ve ara yüzün temiz ve anlaşılabilir olması için uygun düzenlemelerin yapılması gerekmektedir.

**Anahtar Kelimeler—**Kare puzzle, 16 parça puzzle, görsel, bağlı liste, oyun, bulmaca

## I. GİRİŞ

Kare puzzle oyunu kullanıcıların kendi yüklediği veya seçtiği fotoğraflar ile oynadığı eğlenceli bir bulmaca oyunudur. Oyun, akıl ve strateji gerektiren bir bulmacadır. Kullanıcının amacı, kare şeklinde yer alan 16 parçalık görselleri doğru karelere taşıyarak tam bir görsel oluşturmaktır. Kare puzzle oyunu, sadece eğlenceli bir oyun değil, aynı zamanda zihin egzersizi de yapmak için de iyi bir yoldur. Oyuncu, problem çözme becerilerini ve dikkatini artırabilir.

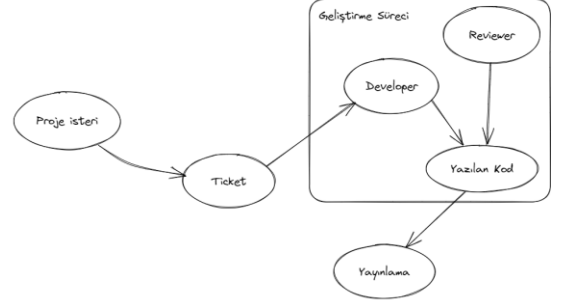
Projemiz arka yüz kısmında NodeJS, ön yüz kısmında EJS, CSS ve Javascript kullanarak oluşturulmuştur. Oyunumuz web tabanlıdır. Bulut bilişim teknolojilerinden Render/Back4app servisleri kullanılmıştır.

## II. AMAÇ VE HEDEFLER

Proje amacı; kullanıcının kendi yüklediği veya seçtiği görselden kare şeklinde yer alan 16 parçayı doğru karelere taşıyarak tam bir görsel oluşturmaktır. Yüklenen görsellerin karıştırılmış halinden kullanıcının hamleler yaparak bu görseli oluşturmaya beklenmekte ve her hamle sonunda kullanıcının yaptığı hamleye göre görsel parçasının doğru konumda olup olmama durumuna göre puanlaması sistem tarafından otomatik olarak yapılmalıdır. Bulmaca tamamlandığında görsel de tamamlanmalı, kullanıcının puanı “enyuksekskor.txt” adında bir dosyada hamle sayıları, kullanıcının adı ve puanı bilgileriyle saklanmalıdır.

## III. YÖNTEM

Projenin geliştirme sürecine başlamadan önce bir iş akış diyagramı oluşturulmuş ve bu iş akış diyagramına uygun şekilde geliştirme ortamları sağlanmıştır. Bu diyagramdan hareketle proje altta belirtilen başlıklar üzerinde değerlendirilebilmektedir:

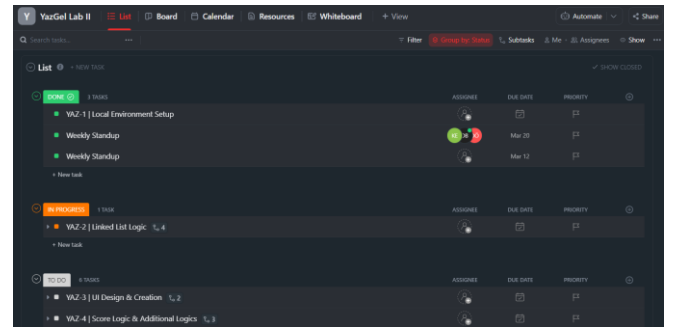


Görsel 1. İş Akış Diyagramı

### A. Ticket ve Proje İsterleri

Projede gerçekleştirilmesi beklenen isterler parçalara bölünmüş ve her biri belirli numaralarda “Ticket” olarak oluşturulmuştur. Oluşturulan “Ticket” numaralarının her biri yazılımcılara atanmıştır ve belirtilen kısmı, yine belirtilen sürelerde tamamlamaları hedeflenmiştir. Bu süreç “ClickUp” iş ve ekip yönetim platformunda kontrol edilmiştir.

ClickUp, kısaca birbirinden işlevlerdeki araçların bir arada olduğu bir üretkenlik platformudur. ClickUp, ekipler ve çalışanlar arasında proje yönetimi, hedef izleme, sorunsuz iş birliği ve görev yönetimini kolaylaştıran bulut tabanlı bir araçtır. Proje takımları, yaptıkları ya da yapacakları görevleri listeye ekleyip takım arkadaşlarına ne durumda olduklarını gösterebilmektedirler. Bu bağlamda ekip olarak edinilen görevler için haftalık toplantılar ClickUp üzerinde tanımlanarak düzenlenmiş ve işlerin durumları kontrol altında tutulmuştur. ClickUp üzerinde yapılan iş dağılımları neticesinde Deniz Beltan *Scrum Master*, Oğuzhan Öztürk *Project Quality Manager* ve Kaan Yıldırım Eser *Project Manager* unvanlarını edinmiştir.

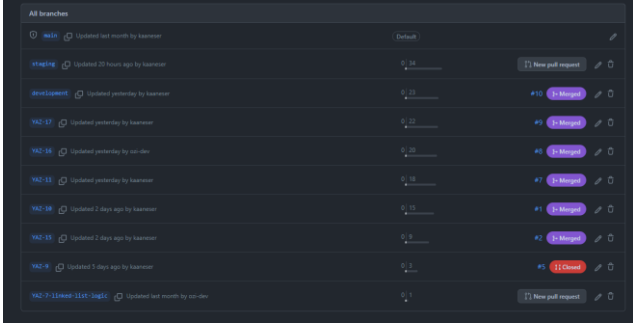


Görsel 2. ClickUp Platformu

### B. Geliştirme Süreci

Görevler, yazılımcılar tarafından ClickUp üzerinde belirtilen süre doğrultusunda görevi kod ortamına

implemente edilmiş, daha sonra oluşturulmuş GitHub ortamında kendi numaralarıyla oluşturulan dallara *commit* işlemi sağlanarak gönderilmiştir. Daha sonra “Pull Request (PR)” ile atanan inceleyci tarafından kod incelenmiş, stil ve mantık hataları olup olmadığı kontrol edildikten sonra ana dal olan “*geliştirme (development)*” dalına gönderilmiştir. Bu süreçlerin hepsi ClickUp üzerinde “*In Progress*” ve “*Code Review*” durumlarıyla kontrol edilmiştir. İnceleme sürecinden geçmiş olan görev, “*Done*” durumuna atanarak tamamlanır. [11]

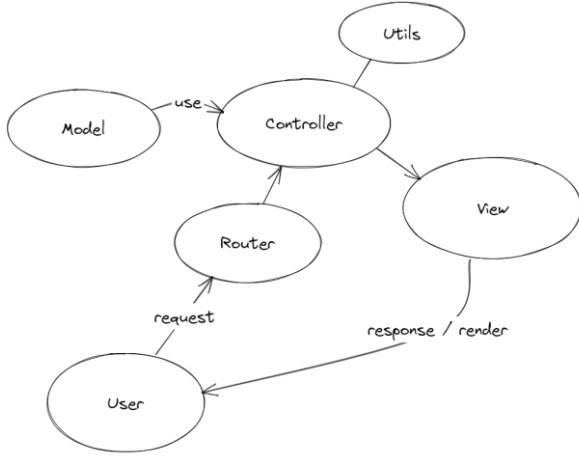


Görsel 3. Oluşturulan dalların GitHub'da gösterimi

#### IV. ARKA YÜZ TASARIMI

##### A. MVC Yapısı

Arka yüz tasarımını ve işlevini parçalara ayırarak daha modüler ve esnek bir yapıya getirmek için *MVC (Model, View, Controller)* tasarım deseni kullanılmıştır. Bu yapıda, kod tekrarları engellenerek kod optimizasyonu ve kodun yeniden kullanılabilirliğinin artırılması hedeflenmiştir.

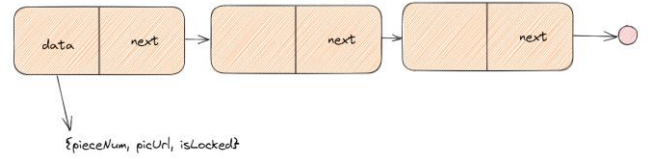


Görsel 4. MVC Diyagramı

Diyagramda (Görsel 3) belirtilen yapıda kullanıcı, yapmak istediği isteği (*request*) “Router” adındaki katmana gönderir, ardından Router, gerekli rotada çalışacak olan iş kodlarını ve eğer kullanacaksa küçük fonksiyon parçalarını yani “Util” fonksiyonlarını kullanarak “Controller” adındaki katmanda, “Model” adındaki katmanı işleme sokar. Bu işlemler sonucunda halihazırda oluşturulan veya kullanılan Model’e belirli veriler yazılır ya da okunur. Daha sonra tamamlanan işlerden geriye bir cevap (*response*) döner ve “View” adı verilen katmanda hazırlanarak kullanıcıya mesaj ya da bir sayfa olarak geri döndürülür. Bu yapı sayesinde kullanıcılar HTTP isteklerinde bulunabilir ve bu isteklerine yazılım tarafından cevap alabilirler.

##### B. Bağlı Liste (Linked List)

Bağlı listeler, kısaca soyut veri tipi olarak tanımlanmakta olup, öğelerin birbirine bağlı olduğu bir dizi öğeden oluşmaktadır. Bağlı listede olan her öğe, değeri ve bağlı olduğu bir sonraki öğeye işaret eden bir referans içerir. Böylece bu liste dinamik olarak değiştirilebilir, bu yüzden bulmacaya istendiği takdirde daha az parçalı ya da çok parçalı bir sistem kurulmak istendiğinde gerçekleştirilmesi kolaylaşmaktadır. Bu yüzden bağlı liste yapısı, uygulamada “Model” katmanının bir öğesi olarak tanımlanmıştır. Tanımlanan bu bağlı listede ekleme, arama ve karşılaştırma işlemleri için hızlı bir yöntem sağlanmış ve bellek yönetimi konusunda da diğer klasik veri tiplerinden daha etkili bir yöntem oluşturulmuştur. Proje içerisinde kullanılan bağlı listede, her parçanın kendine has numarası ve görsel adresleri “Puzzle” isimli Modelden oluşturularak tutulmuş, aynı zamanda bir sonraki bağlı liste düğümünün referansı da tutulmuştur.



Görsel 5. Bağlı liste yapısı

##### C. Fonksiyonlar

Projedeki işlemlerden sorumlu olması için birtakım fonksiyonlar ve modüller tanımlanmıştır. Tanımlanan modüller başlıklarıyla şu şekildedir:

- **Görsel Yükleme:** Görselin yüklenmesi için Node.js temel kütüphaneleri ve URL üzerinden görsel yüklenmesi için *image-downloader* adındaki 3. parti paket kullanılmıştır.
- **Görselin 16 Parçaya Bölünmesi:** Görselin 16 parçaya bölünüp kaydedilmesi için *image-size* ve *image-to-slices* 3. parti paketleri kullanılmış, daha sonra *fs* modülü de kullanılarak görsellerin uygun formatta kaydedilmeleri sağlanmıştır. Görsel parçalanırken 4x4 şeklinde eşit bölünmesi gerektiğinden dolayı görselin yükseklik ve genişlik değerleri koordinat düzleminde bölünmüştür.
- **Görsel Karıştırma:** Bağlı liste metodlarından biri olan görsel karıştırma fonksiyonunda bağlı liste elemanları *Fisher-Yates* algoritmasına göre karıştırılmıştır.
- **Puan Hesaplama ve Kaydetme:** Kullanıcının yaptıkları hamlelerin doğru ve yanlışlığını kontrol etmektedir. Yapılan hamleler doğrultusunda kaydedilen skorların verileri “Model” olarak tutulmaktadır. Hamlelerin hesaplanması için matematiksel fonksiyon kullanılmış, kullanıcının her doğru hamlede 5 puan, her yanlış hamlede 10 puan kazanması/kaybetmesi sağlanmıştır. Kullanıcı oyunu tamamladığında skor modeli içerisinde kaydolun bu veriler *fs* modülü ile “*enyuksekskor.txt*” dosyasına kaydedilmektedir. En yüksek puanların azalan sırada yazdırılması için de yine aynı modül kullanılarak aynı dosyadan bu kez okuma işlemi sağlanmıştır.

## V. ÖN YÜZ TASARIMI

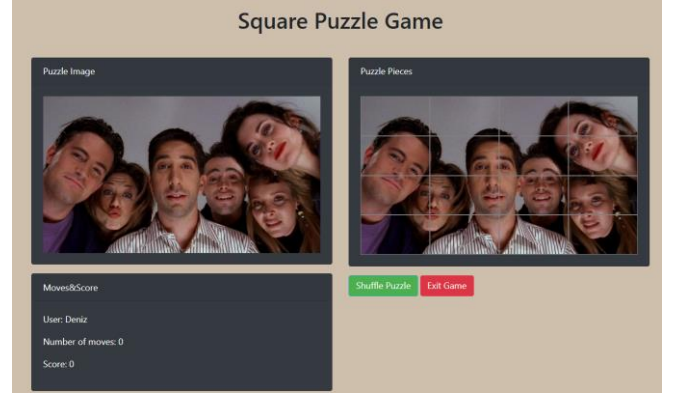
Projenin ön yüz tasarımı yapılırken ara yazılım katmanı olarak “ejs” adı verilen *view engine* kullanılarak “Controller” katmanından gelen verilerin başarılı bir şekilde HTML sayfalarına gönderilmesi sağlanmıştır. Bu bağlamda sistemin ön yüz tasarımı statik olmak yerine dinamik bir yapıya sahip olmuştur. Gönderilen veriler API (*Application Programming Interface*) tarafından kaynak erişimi sağlamakta ve arka yüz bağlantısını API bitiş noktaları (*endpoints*) ile sağlamaktadır. Bu teknolojiler kullanılarak ön yüz tasarımı iki ana ekran üzerinde değerlendirilebilmektedir:

### A. Start Ekranı

Start, yani oyunun başlama ekranında kullanıcı istediği görseli masaüstünden yükleyebileceği veya URL olarak yükleyebileceği bir yazı alanı görür. Ayrıca kullanıcının kendi ismini girebileceği bir yazı alanı da bulunur. Kullanıcı, görseli yükleme aşamasını tamamlamadan adını girdiği alanı kullanamamakta ve oyunu başlatamamaktadır. Kullanıcı görsel yüklerken de iki seçenekten birini tercih etmek zorundadır, yani sadece bir adet görsel yüklenebilmektedir. Kullanıcı, giriş ekranının altında bulunan bölümde oyunu oynayıp en yüksek skora sahip olan ilk üç kişinin hamle sayısı, aldığı puan ve isim bilgilerini görüntüleyebilir.

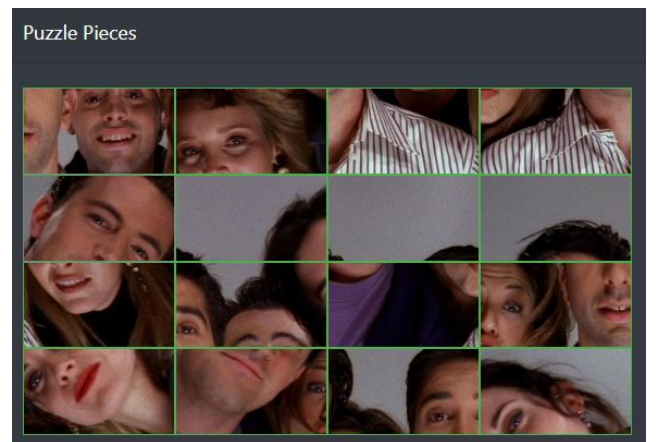
Görsel 6. Start Ekranı

### B. Oyun Ekranı

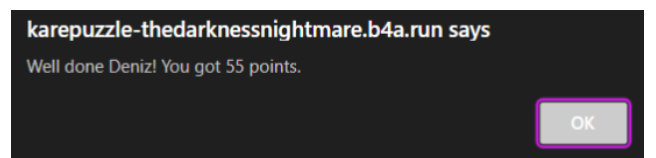


Görsel 7. Oyun Ekranı

Kullanıcı, oyun sayfasına ilk kez geldiğinde sol tarafta, yüklediği veya URL’ini girdiği fotoğrafın orjinal halini; sağ tarafta ise 16 parçaya bölünmüş halini görüntülemektedir. Kullanıcı, oyuna başlamak için “Shuffle” adındaki butona basmalı ve görselin 16 parçaya bölünmüş halini karıştırmalıdır. Butona basıldığı andan sonra ekran yenilenir ve sağ tarafta bulunan parçalardan en az biri kilitli olacak şekilde karıştırılmış şekilde görüntülenir, aynı zamanda “Shuffle” butonu da oyun süresince kullanılamayacak şekilde kilitlenir. Kullanıcı değiştirmek istediği parçalara sırasıyla tıklar ve görsel parçaları yer değiştirir. Parçalar doğru pozisyonlarına geldiklerinde hali hazırda olan yeşil renkli çerçeveleri, gri renkli bir çerçeveye dönüşerek kilitlenir. Oyuncu istediği vakitte “Exit Game” butonuna basarak oyundan çıkış yapabilir, ancak herhangi bir skor kaydedilmez. Sol alt tarafta bulunan tabloda kullanıcı yaptığı hamle sayısını, adını ve aktif puanını görüntüleyebilir. Yapılan her hamle sonrasında aktif puanı dinamik olarak görüntülenmektedir. Kullanıcı oyunu tamamladığında, ekrana oyunun bittiğine dair bir bildirim, açılır pencere şeklinde görüntülenir. Bu bildirim içerisinde yine kullanıcının son puanı görüntülenmektedir. Kullanıcı bildirimi kapattığında “Start” ekranına geri dönmektedir.



Görsel 8. Karıştırılmış Görsel Parçaları



Görsel 9. Açılır Pencere

## VI. BULUT BİLİŞİM

Bulut Bilişim; kullanıcılara internet kullanarak bilgiye erişme, bilgi paylaşma ve depolama yeteneği vermek için bilgi işlem teknolojisini dağıtmanın yeni bir yolunu temsil eder. Bu bağlamda projemizi herhangi bir bulut bilişim teknolojisinde depolamak; erişilebilirlik, veri güvenliği, kolay paylaşım, verimlilik ve daha fazla depolama alanı sağlar ve aynı zamanda sağladığı ölçeklenebilirlik ile geliştirme alanlarına da uygunluk sağlar.

### A. Versiyon Kontrol

Versiyon kontrolü, bir proje üzerinde yapılan değişiklikleri takip etmek, yönetmek ve kontrol etmek için kullanılan bir yazılım aracıdır. Projede versiyon kontrol araçlarından GitHub kullanılmıştır. Proje yönetiminde proje değişikliklerini takip etmek, hatayı düzeltmek, işbirliği yapmak ve projenin yedeklenmesi konularında yardımcı olmasının yanı sıra, bulut bilişim platformlarında ölçeklenebilir bir ortam sağlamaktadır. [11]

### B. Back4App ve Render

Back4App ve Render, uygulamalar için barındırma hizmeti sağlayan platformlardır.

- Back4App, bulut tabanlı BaaS (*Backend as a Service*) platformudur. Back4App, uygulama geliştirme sürecini hızlandırmak için kullanıcı yetkilendirmesi, veritabanı yönetimi, oturum yönetimi, dosya depolama gibi işlemlere sahiptir. Back4App projelerini dağıtmak için *Docker* adı verilen ve uygulamaları dağıtmak, çalıştırmak paketlemek için kullanılan açık kaynaklı konteyner platformunu kullanmaktadır.
- Render, web uygulamaları ve API'ler için bir bulut bilişim hizmetidir. Render, uygulamaların hızlı, kolay bir şekilde ölçeklendirmesini sağlar. Aynı zamanda Render; SSL sertifikaları, ölçeklenebilir veritabanları, otomatik yedekleme, otomatik ölçeklendirme gibi özellikleri de sunmaktadır.

Projeyi dağıtmak için Back4App ve Render platformları kullanılmıştır. Bunun sebebi her iki platformun da diğer bulut platformlara kıyasla ücretsiz olması ve projede versiyon kontrol sistemi olarak kullanılan GitHub'a entegrasyonlarının bulunmasıdır. Proje "*staging*" dalında dağıtılmış ve kullanıma sunulmuştur. [8, 9]

## VII. SONUÇ

Kare Puzzle oyunu geliştirme sürecinde, oyun mantığına uygun bir algoritma tasarlanması gerektiği ortaya konulmuştur. Bu algoritmanın kullanıcının hareket ettirdiği görsel parçalarının konumlarını takip etmesi ve doğru şekilde yerleştirilerek puanlama durumu ile yönlendirilmesi gerekmektedir. Ayrıca, oyunun kolayca oynanabilmesi için uygun düzenlemeler yapılmış ve arayüzün anlaşılabilir olması hedeflenmiştir. Arka yüz tasarımı için MVC yapısı kullanılmış ve bağlı listeyi bir model olarak kullanarak ilgili fonksiyonlar başarılı bir şekilde gerçekleştirilmiştir. Ön yüz tasarımında iki ayrı ekran tasarlanmış ve oyun, kullanıcı tarafından oynanabilir hale getirilmiştir. Geliştirme sürecinde versiyon kontrolü için GitHub kullanılmış ve son olarak Back4App ve Render ile bulut bilişim platformlarına entegre edilmiştir.

Projenin gelecek sürümleri için eklenebilecek birkaç özellik listelenmiştir:

- Zorluk Seviyeleri: Oyun, birden fazla zorluk seviyesi eklenerek oyuncuların deneyimlerine ve becerilerine göre ayarlanabilir. Böylece oyunu daha da keyifli hale getirerek oyuncuların daha uzun süre ilgilenmeleri sağlanabilir.
- Ses Efektleri ve Müzikler: Kullanıcıların oyunu oynarken müzik ve ses efektlerini içtirmeleri kullanıcının deneyimini olumlu yönde etkileyecektir.
- Süre: Kullanıcıların kısıtlı bir süre içerisinde bulmacaları tamamlamaya çalışmaları rekabeti arttıracaktır.

## KAYNAKÇA

- [1] <https://scrapingant.com/blog/download-image-javascript>
- [2] [https://edestek3.kocaeli.edu.tr/pluginfile.php/35913/mod\\_resource/content/1/git-cheat-sheet-education.pdf](https://edestek3.kocaeli.edu.tr/pluginfile.php/35913/mod_resource/content/1/git-cheat-sheet-education.pdf)
- [3] <https://www.digitalocean.com/community/tutorials/how-to-process-images-in-node-js-with-sharp>
- [4] <https://www.npmjs.com/package/image-downloader>
- [5] <https://www.npmjs.com/package/image-to-slices>
- [6] <https://www.npmjs.com/package/express>
- [7] <https://www.npmjs.com/package/image-size>
- [8] <https://karepuzzle-thedarknessnightmare.b4a.run/start>
- [9] <https://kare-puzzle.onrender.com/start>
- [10] <https://excalidraw.com>
- [11] <https://github.com/kaaneser/kare-puzzle>
- [12] Crown Stephen W., "Improving Visualization Skills of Engineering Graphics Students Using Simple JavaScript Web Based Games", Journal of Engineering Education, Vol 90 (2001).