

# Puzzle Oyunu

Atayağz USLU  
Bilişim Sistemleri Mühendisliği  
191307062

Ömer Faruk SEVİNÇ  
Bilişim Sistemleri Mühendisliği  
191307007

Hüseyin NAS  
Bilişim Sistemleri Mühendisliği  
191307043

**ÖZET — Proje kapsamında yüklenen görseli 4 satır ve 4 sütun olmak üzere 16 parçaya bölen bir puzzle oyunu tasarlanmıştır.**

**Anahtar Kelimeler — Puzzle, Jigsaw, Linked List**

## I. GİRİŞ

Yapbozlar, bir resmin tam resmi oluşturmak için birleştirilmesi gereken daha küçük parçalara bölüldüğü popüler bir eğlence biçimidir. Bir yapbozda, her parçanın, tüm resmi oluşturmak için diğer parçalarla birbirine uyan benzersiz bir şekli vardır. Benzer şekilde, bağlantılı bir listede her öğenin benzersiz bir değeri ve bir sonraki öğeye işaretçisi vardır.

Bağlı liste kullanarak geliştirdiğimiz puzzle konsol uygulaması, kullanıcının belirli bir puzzle'ı tamamlamasına yardımcı olan bir programdır. Bu uygulama, puzzle parçalarını depolamak ve yönetmek için bağlı liste veri yapısını kullanır. Bağlı liste, puzzle'ın farklı parçalarının depolanması ve yönetimi için ideal bir veri yapısıdır. Bu veri yapısı, puzzle'ın parçalarının eklenmesini, silinmesini ve yeniden düzenlenmesini kolaylaştırır. Uygulamamızda, puzzle'ın her parçası bir düğüm olarak temsil edilir ve bağlı liste yapısı ile birbirine bağlanır.

## II. KULLANILAN TEKNOLOJILER

### A. Unity

Unity, oyun ve diğer interaktif 3D uygulamaların oluşturulması için kullanılan bir oyun motorudur. Oyun motoru, birçok farklı özelliği bir araya getirerek, oyun geliştiricilerinin oyunlarını oluşturmalarına ve bunları farklı platformlarda yayınlamalarına olanak sağlar.

Unity'nin en önemli özelliklerinden biri, oyun ve uygulamaların farklı platformlarda çalıştırılabilmesine olanak sağlamasıdır. Bu nedenle, oyunlar ve uygulamalar Windows, Mac, Linux, iOS, Android, Xbox, PlayStation, Nintendo Switch ve daha birçok platformda çalıştırılabilir.

Unity ayrıca, oyun geliştiricilerine kolaylık sağlamak için hazır şablonlar, önceden oluşturulmuş bileşenler ve hazır materyaller sunar. Bu, oyun geliştiricilerinin zaman kazanmasına ve hızlı prototipleme yapabilmelerine olanak sağlar.

Unity kullanmanın bize avantajları:

**Çoklu Platformlarla Uyumlu:** Unity oyun geliştirme motoru, iOS, Android, Mac ve Steam gibi neredeyse tüm işletim sistemleriyle uyumludur.

**Kullanıcı Dostu:** Unity'nin kullanımı diğer oyun geliştirme motorlarına kıyasla daha kolaydır. Unity'de komut dosyası oluşturmak için C# veya JavaScript kullanılır. Bunlar da öğrenmesi ve kullanması kolay olan dilleridir.

**Hata Ayıklama:** Unity ile oyun geliştirme sırasında karşılaşılan hatalar kolayca tespit edilir. Bu da hata oranını azaltır, hata ayıklamayı daha kolay hale gelir.

**Hızlı Geliştirme:** Unity, oyun geliştiricilerine gelişmiş araçlar sağlar ve bu araçlar sayesinde hızlı bir şekilde oyunlar oluşturulabilir. Karakter animasyonları, fizik simülasyonları, ses efektleri, yapay zeka gibi özellikler bulunur ve bu özellikler, oyun geliştiricilerinin daha kapsamlı ve ilgi çekici oyunlar oluşturmalarını sağlar.

### B. GitHub

GitHub, dünyanın farklı yerlerinden geliştirici, programcıların bir araya gelerek yazılım, framework ve kodlarını paylaşarak ortak çalışma projeleri oluşturabildiği, ya da kendilerine ait projelerin GitHub bulut sisteminde kaydedilerek çeşitli araçlarla kolayca tekrar ulaşılmasının sağlandığı, milyonlarca kullanıcısı olan web tabanlı bir platformdur.

GitHub, aynı zamanda birçok farklı programlama diliyle uyumlu olduğu için, geliştiricilerin birçok farklı proje için kullanabilmesine olanak tanır. Ayrıca, GitHub kullanıcıları arasında işbirliği yapmak için birçok özellik sunar, örneğin kod incelemeleri, sorun takip sistemi ve tartışma forumları gibi.

### C. Visual Studio

Visual Studio, Microsoft tarafından geliştirilen bir entegre geliştirme ortamıdır (IDE). Windows, macOS ve Linux işletim sistemleri için mevcuttur.

Visual Studio, birçok farklı programlama diliyle uyumlu olup, uygulama geliştirme sürecinde gerekli olan birçok aracı tek bir yerde sunar. C++, C#, VB.NET, Python, JavaScript, TypeScript, F# ve diğer birçok programlama dili için kullanılabilir.

Visual Studio, kod editörü, hata ayıklama araçları, sürüm kontrolü, birim testleri ve diğer birçok geliştirme aracını içerir. Geliştiriciler, projelerini oluşturabilir, kod yazabilir, derleyebilir ve hata ayıklama yapabilirler.

Sonuç olarak, Visual Studio, geliştiricilerin birçok farklı programlama diliyle uyumlu olarak tek bir yerde uygulama geliştirme sürecini yönetmelerine olanak sağlayan bir IDE'dir.

### D. C#

C# yani diğer bir adıyla C Sharp, Microsoft tarafından geliştirilen sunucu ve gömülü sistemleri çalıştırmak için tasarlanmış programlama dilidir. .NET Framework ortamında kullanılmak üzere geliştirilmiş nesne tabanlı programlama dillerinden birisidir.

C# programlama dili C ve C++ programlama dillerinden türemiştir. C, Java veya C++ dillerinden biri ile ilgilenmiş yazılım geliştiriciler C#'ın kurallarına kolayca uyum sağlayabilirler. Debugger özelliği yani gelişmiş hata giderme özelliği ile yazılımcıların işlerini kolaylaştırır. Bu özellik ile “syntax” hatalarını kolayca tespit eder. Dizinlerdeki hataları yazılımcıya ayrı bir ekranda göstererek yazılımcının bu hataları kolayca tespit etmesini sağlar. Bu özellikleri sonucunda Backend Developer’lar tarafından yaygın kullanılan bir programlama dili haline gelmiştir.

### III. PROJEYİ NASIL OLUŞTURDUK

Projeyi yapabilmek için her birimiz öncelikle konu hakkında ayrıntılı bir araştırma yaptık. Daha sonra adım adım aşağıdaki gibi ilerledik.

1. Önce proje hedefleri ve gereksinimleri belirledik: Projenin ne yapması gerektiğini ve hangi adımların takip edilmesi gerektiğini belirledik. Bu, projede ne yapılması gerektiğine ve hangi görevlerin tamamlanması gerektiğine karar vermenize yardımcı olacaktır.
2. Grup üyeleri arasında görevleri paylaştık: Proje için yapılması gereken görevleri belirledik ve her görevi bir grup üyesine atadık. Görevleri, herkesin yeteneklerine, deneyimine ve ilgisine göre paylaştık.
3. Görevlerin süresini ve önceliğini belirledik: Her görev için bir tamamlanma süresi ve öncelik belirledik. Bu, her görevin zamanında tamamlanmasını sağladı ve projenin ilerlemesini takip etmemize yardımcı oldu.

Daha sonra tekrar bir araya gelerek yaptığımız işleri gözden geçirdik. Read.me ve raporu tamamlayarak projemizi sonlandırmış olduk.

### IV. BAĞLI LİSTE

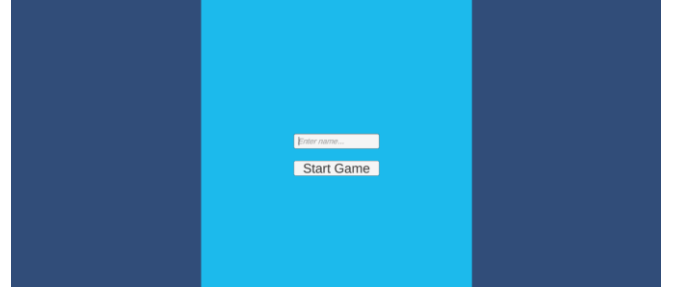
Bağlı liste, bir veri yapısı türüdür ve verileri düğümler şeklinde depolar. Her düğüm, veriyi ve verinin bulunduğu konumu tutar. Her düğüm aynı zamanda, bağlı listenin bir sonraki düğümünün yerini de içerir. Bu nedenle, bağlı liste birbirine bağlı düğümlerden oluşur.

Bağlı listeler, elemanları dinamik olarak eklenebilen veya çıkarılabilen bir veri yapısıdır. Bu nedenle, bağlı listeler, özellikle dizi gibi sabit boyutlu veri yapılarından daha esnek ve verimli bir seçenektir. Bağlı listeler, veriye erişmek için dizilerdeki gibi doğrudan indeks kullanımı yerine, listenin başından itibaren elemanlara doğru gezinme yaparak erişim sağlar.

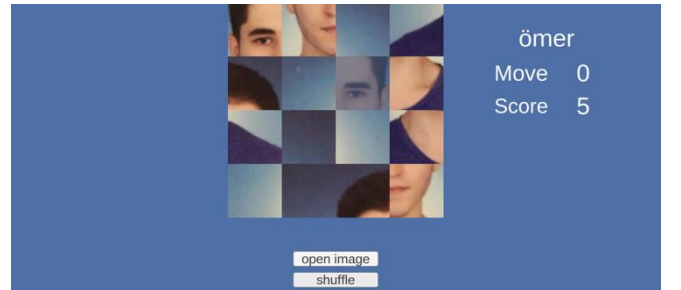
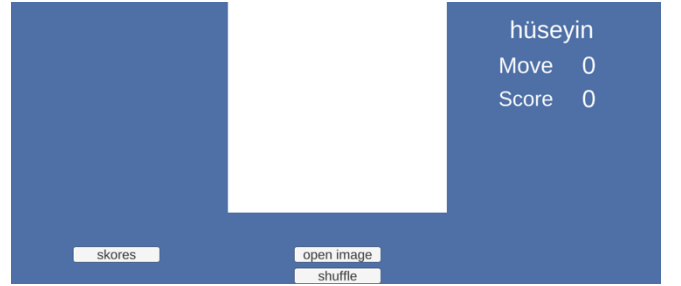
Bağlı listeler, genellikle verilerin sıralı olarak depolanması gerektiği zamanlarda kullanılır. Örneğin, bir oyun sıralaması veya bir müzik çalma listesi gibi durumlarda bağlı liste yapısı kullanılabilir.

### V. PROJEMİZDEN EKRAN GÖRÜNTÜLERİ

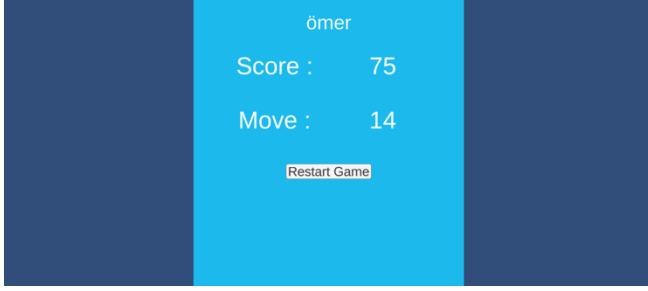
1. Giriş ekranı: Proje açıldığında bir isim girmeniz beklenmektedir. Daha sonra “start game” diyerek başlatılır.



2. Oyun ekranı: Start game dedikten sonra açılan ekranda bir görsel yüklemek gerekiyor. Daha sonra “shuffle” diyerek karıştırıyoruz. Sağ tarafta girilen isim, hamle sayısı ve skor puanımız geliyor. Altta görselde sol alt kısımda geçmiş skor puanları görüntülenir.

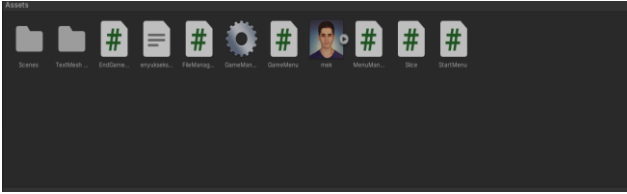


3. Puzzle tamamlandıktan sonra karşınıza aşağıdaki ekran gelir. Burada isim, skor puanı ve hamle sayısı görülmektedir. İstenirse “Restart Game” diyerek tekrar oynanabilir.



- [1] <https://unity.com/community>  
[2] <https://stackoverflow.com/>  
[3] <https://www.udemy.com/>  
[4] <https://github.com/>  
[5] <https://learn.microsoft.com>

4. Proje Unity ekranında da aşağıdaki şekilde görülmektedir.



## VI. SONUÇ

Projenin sonucunda, ekibimiz başarıyla işbirliği içinde çalışarak, projenin amacına uygun bir şekilde tasarım ve uygulama aşamalarını gerçekleştirdi. Projenin gereksinimleri tamamen karşılandı ve sonuç olarak başarılı bir yazılım projesi ortaya çıktı.

Proje sırasında bazı zorluklar yaşandı ve bu zorluklar, ekibimizin takım çalışması ve iletişim becerilerini geliştirmemize yardımcı oldu. Ayrıca, proje yönetimi sürecinde belirlenen plana sadık kalarak, proje zamanında tamamlandı.

Projenin sonuçları, amacına uygun bir şekilde gerçekleştirilmiş olmasıyla birlikte, bazı geliştirme alanları da olduğu görülmektedir. Bu nedenle, gelecekte benzer projelerde, ekibimizin daha iyi bir tasarım ve uygulama süreci için daha fazla çaba sarf etmesi gerektiği sonucuna varılmıştır.

Sonuç olarak, bu proje ekibi, takım çalışması, iletişim ve problem çözme becerileri konusunda büyük bir ilerleme kaydetti. Ayrıca, projenin başarıyla tamamlanması, ekibimizin birlikte çalışarak yazılım projelerinde başarılı sonuçlar elde edebileceğini gösterdi.