

Puzzle Oyunu

Masaüstü Puzzle Oyunu

Fatmanur Azbaş
201307038
201307038@kocaeli.edu.tr

Zeynep Serva Yıldız
201307052
201307052@kocaeli.edu.tr

Zehra Betül Taşkın
201307054
201307054@kocaeli.edu.tr

Teknoloji Fakültesi / Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi

Özet

Projemiz C# .Net teknolojisi kullanılarak geliştirilmiş bir puzzle oyunudur. Bu projenin amacı, 4x4'lük bir puzzle oyununu kullanıcıya sunabilmektir. Kullanıcı dışarıdan kendi istediği bir resmi yükler ve oyun bu resmi 4x4'lük buton parçalarına, resim parçalarını endeksleyerek oyunu başlatmış olur. Kullanıcı tüm puzzle parçalarını yerine oturtuktan sonra isterse yeni oyuna başlayabilir isterse de oyunu bitirir. Oyunun bitmesinin ardından kullanıcının isim, puan, hamle sayısı ve süre bilgisi bir .txt belgesine kaydedilir. Kullanıcı daha sonra bu dosyaya erişim sağlayarak bu bilgileri görüntüleyebilir.

I. GİRİŞ

Günümüzde oyunlar her yaş kitlesine hitap etmesiyle ve eğlenceli vakit geçirmeye olanak sunmasıyla oldukça yaygın. Genel olarak çocuk yaş kitlesi için zekâ geliştirici oyunlar ebeveynler tarafından tercih edilmektedir. Çocuk yaş kitlesi için hafıza oyunları, kabiliyet oyunları, mantık ve problem çözme yeteneği geliştiren oyunlar tercih edilmektedir. Bununla birlikte mantık ve problem yeteneği geliştiren oyunlar arasında en başta dağınık parçalardan bir bütün oluşturma amacıyla geliştirilmiş olan puzzle oyunları yer almaktadır. Puzzle oyunları birçok farklı çeşidi olan bir oyun türüdür. Bunlardan biri de yine her yaş kitlesinin vazgeçemediği resimli puzzle oyunlarıdır. Bu oyunlar hem eğlenceli hem de zekâ geliştirici oyunlar olması sebebiyle zararsız ve zevkli kabul edilen oyunlar arasındadır. Bizde böyle bir puzzle oyunu geliştirdik. Kullanıcının sıkılmadan oynayabileceği bir oyun tasarladık.

1. Puzzle Oyunu

Oyunumuz kullanıcı tarafından yüklenen bir resimle başlamaktadır. Ardından kullanıcıya, ne kadar sürede bir puzzle oyununu bitirdiği, tıklama sayısı ve bununla birlikte güncel puanı tutularak, bu bilgileri görüntüleme şansı verdik. Bununla birlikte oyun boyunca puan ve zaman bilgilerini eş zamanlı listeleterek kullanıcının oyunda daha zaman

geçirmesine ve önceki skorunu geçmesi için hırslanmasına olanak tanıdık. Yani kullanıcı biten her bir oyunun ardından tekrar tekrar oyunu oynayabilmektedir.

2. Puzzle Oyunlarında Çok Tercih Edilen Teknolojiler

Literatür araştırması sırasında edindiğimiz bilgilere göre puzzle oyunları en çok aşağıdaki programlama dilleri kullanılarak yazılmıştır. (Aşağıdaki sıralama rastgeledir. Bir istatistik belirtmemektedir.)

- JavaScript: Genel olarak web bazlı puzzle oyunlarının Javascript kullanılarak yazıldığını gözlemledik.
- Python: Yazımı diğer dillere göre daha az iş yükü veren bir dil olduğu için kullanımı puzzle oyunlarında oldukça yaygındır.
- C# ve Unity: Araştırmalarımız sırasında fark ettiğimiz bir diğer programlama dili C#' tır. Genel olarak bu teknolojiler kullanılarak yapılan oyunların kullanıcı arayüzü Unity'de tasarlanmakta ve ardından C#'ta arka yüz kodu yazılmış olduğunu gözlemledik.
- C# .Net: Son olarak bizimde kullandığımız C# .Net Teknolojisi var. Bu tercihimizin sebebi aslında Windows Form aracılığıyla aynı uygulama içerisinde hem arayüzü tasarlayıp hem de arka yüz kodunu rahatlıkla yazabilmemizi sağlanmasıdır.

3. Kullanılan Teknolojiler, Diller ve Uygulamalar

A. C# .Net Framework

i. C#

C#; Microsoft tarafından .NET Teknolojisi için geliştirilen modern bir programlama dilidir. Sözdizimi C-like (C benzeri) bir deneyim sunar.

Microsoft tarafından geliştirilmiş olsa da ECMA ve ISO standartları altına alınmıştır.

C programlama dilinde bir tam sayı değişkeni 1 artırmak için değişkenin sonra "++" eki kullanılır. C++ dilinin adı, C diliyle Nesne Yönelimli Programlama yapabilme olanağı (C with Classes) için eklentiler sağladığı için "C++" şeklindedir. Benzer şekilde C++ diline yeni eklentiler yapılarak ((C++)++) bir adım daha da ileriye götürülmüş ve tamamen nesneye yönelik tasarlanmış C# dilinin isimlendirilmesinde, + karakterlerinin birbirlerine yaklaşmış hali ve bir melodi anahtarı olan C# Major kullanılmıştır.

Bu dilin tasarlanmasına Pascal, Delphi derleyicileri ve J++ programlama dilinin tasarımlarıyla bilinen Anders Hejlsberg liderlik etmiştir.[4]

ii. .Net Teknolojisi

C#, .NET orta seviyeli programlama dillerindendir. Yani hem makine diline hem de insan algısına eşit seviyededir. Buradaki orta ifadesi dilin gücünü değil makine dili ile günlük konuşma diline olan mesafesini göstermektedir. Örneğin; Visual Basic .NET (VB.NET) yüksek seviyeli bir dildir dersek bu, dilin insanların günlük yaşantılarında konuşma biçimine yakın şekilde yazıldığını ifade etmektedir. Dolayısıyla VB.NET, C#.NET'ten daha güçlü bir dildir diyemeyiz.[4]

C#'ta kodlar MSIL (Microsoft Intermediate Language) yapısına çevrilir. Bu kod işlemci tarafından tanınan bir kod yapısı değildir ve direk olarak çalıştırılmaz. Kodu çalıştırılabilir hale getiren CLR(Common Language Runtime) yapısıdır. Yazılan kodlar .NET Framework olan bütün bilgisayarlarda çalışır durumdadır ayrıca .NET ortamında bir projeyi iki veya daha fazla dil ile ortak yazabilmek mümkündür.[5]

B. Visual Studio

Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır.[6]

C. Azure Virtual Machine

Sanallaştırma olarak en stabil ve yüksek kullanıcı oranlarına sahip Azure firmasını tercih ettik. Azure Sanal Makinesi, hem ücretsiz kullanım hakkı hem de çeşitli donanım avantajlarıyla bize oldukça kolaylık sağladı. Bu projede Azure için ücretsiz hesap oluşturma yöntemi ile değil, Öğrenci hesabı ile kredisiz ücretsiz sanal makine kurulumu gerçekleştirdik.

Kurulum olarak hesabın öğrenci olduğu “Kocaeli Üniversitesi Mail Sistemi” ile onaylandıktan sonra “sanal makine oluştur” seçeneğinden kredi limiti olmadan çeşitli seçenekler ile sanal makine oluşturulabilmektedir. Burada öğrenci hesabı olması sebebi ile bazı donanım sınırlamaları göz ardı edilmemelidir. Kurulum gerçekleştikten sonra makinenin IP adresi, kullanıcı adı ve şifre bilgilerini,

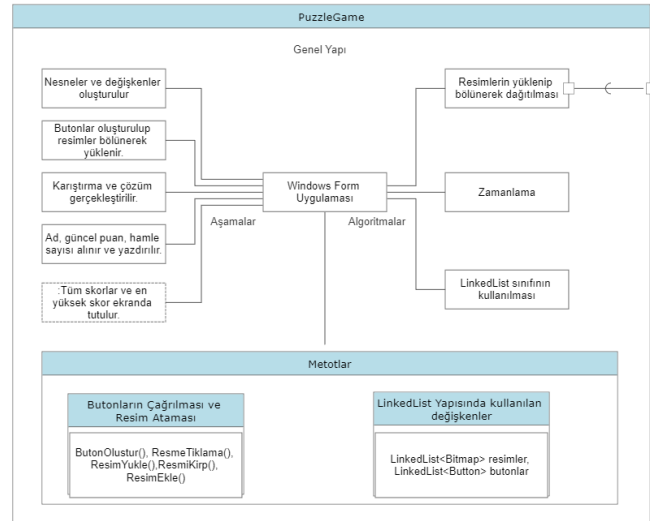
bilgisayarda halihazırda bulunan “Uzak Masaüstü Bağlantısı” programına girdikten sonra sanal makineye ulaşım kolaylıkla sağlanmıştır. Proje gösterimi ise burada kurulan “Visual Studio 2022” programı üzerinden gerçekleştirilecektir.

D. GitHub

Geliştirici toplulukları arasında en büyük platformlardan biri olan GitHub’ı, projemizi konuşlandırmada kullandık. Web tabanlı bulut depolama imkanı sağladığı için projeyi tanımlama ve yayınlamaya imkan sağladı. Bu proje için PuzzleGame repository’si oluşturarak GitHub profilimize aktararak, projenin ne olduğu, ne işe yaradığı, nasıl kullanıldığını bildiren README dosyasını ekledik. Projenin tüm class yapılarının aynı işletim sistemindeki dosya yapısı hassasiyetinde olması, projenin okunabilirliğini artırmıştır. Projemizi bu şekilde yayınlayarak hem diğer tüm kişilerin erişim sağlamasını hem de indirilip yüklenerek herkes tarafından kullanılabilmesini sağladık.

II. GELİŞME

İlk olarak puzzle oyunu için yazacağımız algoritmanın taslağını oluşturduk. Daha sonra hem LinkedList yapısının kullanım kolaylığı hem de çok çeşitli geliştirme seçenekleri sebebiyle C# dilinde yazmaya karar verdik. Bu nokta da oyunun yapısı gereği web uygulamasına göre masaüstü bir oyun geliştirmenin daha etkili olacağına karar verdik. Ardından “windows form app” yapısında bir uygulama geliştirip, Azure Virtual Machine ile de sanallaştırmış olduk.



(UML Diyagramı)

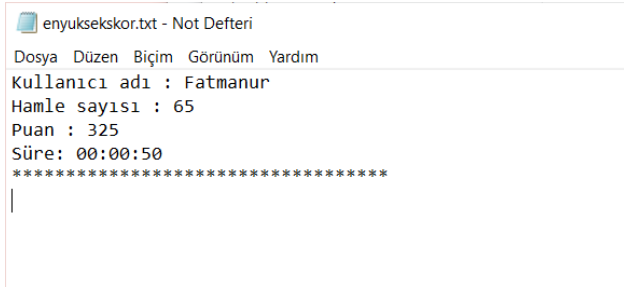
Yazılım Mimarisi

Yazılım tarafında C# dili tercih edilmiştir. Bu sebep ile .NET ortamını tercih ettik. İlk başlarda .NET Framework ile geliştirmeye başladığımız projeyi daha sonra .NET 6 yapısının mevcut zamanda geliştirmeye daha uygun

olduğuna karar verip .NET 6 ortamına projemizi taşıdık. Bu sayede bazı fonksiyonları daha dinamik kullanabilmiş olduk.

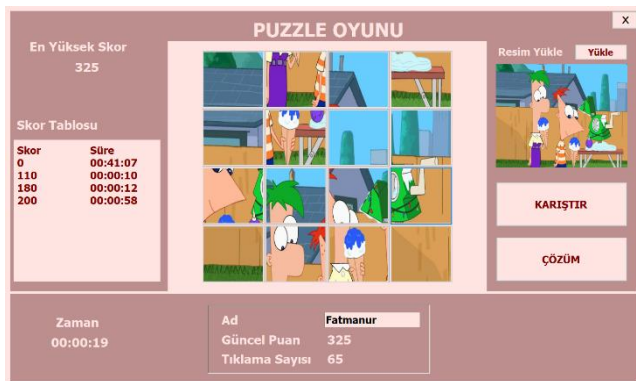
Mimarisinde ise projenin genel olarak LinkedList yapısına uygun olmasına özen gösterdik. Bu bağlamda Butonlar için bir List ve Fotoğraflar için LinkedList yapısını kullandık.

Puzzle oyununun genel yapısı kullanıcı tarafından yüklenecek resmin 16 parçaya ayrılması, bunların yazılım tarafında oluşturulan butonlara atanması ve butonların uygulama tarafından karıştırılarak kullanıcıdan doğru yerlerine yerleştirilmesi istenmesi şeklindedir. Burada kullanıcı bütün parçaları doğru yerleştirdiğinde uygulama oyunun bittiğine dair bir uyarı vermekte ve kullanıcıya tekrar oynama durumunu sorup cevaba göre skor durumunu güncellemektedir. Kullanıcı oyunu her bitirdiğinde Tasarım tarafında oluşturulan ListBox aracına skor ve süre bilgisi gelmektedir. Aynı zamanda kullanıcının isim bilgisi ile birlikte proje dosyasının içerisinde bulunan “enyuksekskor.txt” skor durumunu yazdırmaktadır.



(enyuksekskor.txt)

Uygulama aynı zamanda puan durumunu da ölçmektedir. Her doğru hamlede -5 puan her yanlış hamlede ise +10 puan yazılmaktadır. “Karıştır” butonu ile kullanıcı istediği anda resimleri tekrar karıştırabilir, “Çözüm” butonu ile de istediği zaman puzzle’ın orijinal haline gelebilmektedir. Ayrıca programda tanımlanan timer ile oyun başladığı andan itibaren bir sayaç tutulup, kullanıcı oyunu bitirdiğinde geçen süre belirtilmektedir.



(Kullanıcı Arayüzü)

Uygulamadaki ana fonksiyonlara gelinirse, Yükle butonu ile Program öncelikle kullanıcıdan “OpenFileDialog()”

metoduyla seçilen resmi alır. Alınan resmin bir Bitmap haritası oluşturulur. Daha sonra “ButonOlustur()” fonksiyonu, tanımlanan liste ile 16 adet buton oluşturur. “ResimKırp()” fonksiyonu Bitmap haritasına yerleştirilen resmi istenen boyutta parçalara ayırıp bunu <Image> olarak tanımlı bağlı listeye atar. “ResimEkle()” fonksiyonu eşit parçalara ayrılıp bağlı listeye atılan resimleri oluşturulan butonlara resim olarak (BackgroundImage) atanmasını sağlar. “ButonlarıYerlester()” fonksiyonu 16 butonun tamamının tasarım tarafında nerede duracağını belirler. Ayrıca buradan butonların resme göre doğru konumları alınır ve oyun bitişi için sorgulamada kullanılır. “Kontrol()” fonksiyonu var olan konumları butonlara atar ve bunu kullanılan “Control” metodu ile kontrol edilmesini sağlar. Burada mevcut_konum doğru konumlar ile sorgulanır ve her parça doğru yerde ise ekrana “Tebrikler” şeklinde yazı yazdırır. “Yazdır()” fonksiyonu StreamWriter() metodu ile yukarıda da belirtilen “enyuksekskor.txt” ‘nin yazdırılmasını sağlar.

Aldığımız Hatalar ve Çözümleri

Genel olarak aldığımız hatalar mantıksal hatalardır. Çözmesi fazla zor olmayan hatalar aldık. Bunlardan birkaçı aşağıda sıralanmıştır.

- Windows Form içerisine eklediğimiz arayüz kontrol araçlarının Click olaylarının gerçekleştirilmesi konusunda hatalar aldık. Bunları koyduğumuz isimlerin Form.Designer.cs dosyasındaki isimlerle aynı olup olmadığını kontrol etmemiz sırasında birtakım isim çakışmalarını tespit edip düzelterek çözdük.
- Arka yüz kod kısmında oluşturduğumuz bazı değişkenlere veri gelmemesi sebebiyle System.NullReferenceException hatası aldık. Bu hatayı değer veya veri atamamız gereken değişkenleri kontrol edip gerekli atamaları yaparak çözdük.
- Butonlara gelen bilginin tek bir butona iletilmesi sebebiyle kalan diğer butonları hareket ettiremediğimizi tespit ettik. Ardından bu hatayı tüm butonların tek tek sorgulandığı bir döngü kodu yazarak çözdük.
- Puzzle oyunun çözümü için yazdığımız kodda bir hataya rastladık. Hatanın sebebi normal çözüm konumlarının ve anlık konumların birbiriyle eşleşmemesinden kaynaklanıyordu. Bu hatayı if sorgusuna alarak anlık konum ile çözüm konumun eşleşip eşleşmediğini kontrol ederek çözdük. [1][2]

III. SONUÇ

Bahsedilen tüm teknolojiler ile kullanıcının rahat bir şekilde kullanmasını istediğimiz bir arayüz, bu arayüzdeki tüm bileşenleri işlevlendiren ve algoritmalarımızı kapsayan fonksiyonları bulunduran bir class yapısı sağladık. Kullandığımız linkedlist yapısı, dışarıdan alınan resmi bölen ve eş parçalara ayıran resim bölme fonksiyonları ve

nihayetinde butonların hareket etmesini sağlayan fonksiyon, üzerinde en çok durduğumuz algoritmalar oldu. C# .NET Framework'ü çerçevesinde LinkedList ve çeşitli algoritmalar ile geliştirdiğimiz bu masaüstü projesinde istelere uygun bir şablon hazırlamış ve kullanıma sunmuş olduk. Windows Form uygulamasının sağladığı bileşenlerin bulunduğu designer arayüzünde komplike bir tasarım sağladık. Bu şekilde projenin çalışması durumunda hem kullanıcının kolay kullanıma erişebileceği bir önyüz, hem de geliştiricilerin rahatlıkla anlayabileceği fonksiyonları içeren class yapısı sağlanmış oldu.

IV. KAYNAKLAR

- [1] [Hataların Çözümlerini Araştırma-I](#)
- [2] [Hataların Çözümlerini Araştırma-II](#)
- [3] [Benzer Projelerin Araştırılması](#)
- [4] [C# Nedir?](#)
- [5] [.Net Framework](#)
- [6] [Microsoft Visual Studio](#)
- [7] [Word Search Puzzle using Multi-Linked Lists](#)
- [8] [LinkedList<T> Sınıf \(System.Collections.Generic\) | Microsoft Learn](#)
- [9] [Bulut Bilişim Hizmetleri | Microsoft Azure](#)
- [10] [Control Sınıf \(System.Windows.Forms\) | Microsoft Learn](#)
- [11] [Download .NET 6.0 \(Linux, macOS, and Windows\) \(microsoft.com\)](#)
- [12] [Windows Form Araç Kutusu Bileşenleri](#)
- [13] [C# Karıştırma Metodu](#)
- [14] [UML Diyagram](#)
- [15] [LinkedList Mantığı](#)