

Kare Puzzle Oyunu

İsa Oğuz
201307026@kocaeli.edu.tr

Mehmet Yusuf Alpay
201307046@kocaeli.edu.tr

Yunus Emre Dal
201307021@kocaeli.edu.tr

Abstract—This project is carried out for the development of a game called "Square Puzzle". The game is created using C# programming language and Microsoft Visual Studio, with the implementation of linkedlist data structure. Data regarding user's moves and the number of moves required to complete the game have been collected

Keywords—game development, Square Puzzle, linkedlist data structure, C# programming language, Microsoft Visual Studio, user moves, number of moves.

I. GİRİŞ .

Günümüzde bilgisayar oyunları, popüler bir eğlence ve öğrenme aracı haline gelmiştir. Bu çalışma, C# programlama dili kullanılarak geliştirilen bir kare puzzle oyununu tanıtmaktadır. Oyun, oyunculara istedikleri görseli kullanarak kare puzzle bulmacaları sunmaktadır. Oyunun temel amacı, oyuncuların zihinsel becerilerini geliştirmelerine yardımcı olmaktır. Bu çalışmanın amacı, oyuncuların zihinsel becerilerini artırmak için kullanılabilecek bir oyun tasarlamak ve geliştirmektir. Bu amaçla, oyunda kullanılan temel veri yapısı olarak linkedlist kullanılmıştır. Bu çalışmada, linkedlist'in kullanımı, oyunun tasarımı ve geliştirilmesi ayrıntılı bir şekilde açıklanmaktadır. Sonuçlarımız, linkedlist'in kullanımının oyunun performansını artırdığını ve oyuncuların zihinsel becerilerini geliştirmeye yardımcı olduğunu göstermektedir.

II. KULLANILAN PROGRAMLAR

A. C# dili ve Visual Studio Editörü

Bu çalışmada, C# programlama dili ve Microsoft Visual Studio editörü kullanılmıştır. C# dilinin seçilmesi, nesne yönelimli programlama (OOP) konseptlerine sahip bir dil olması ve ADO.NET platformu ile uyumluluğu nedeniyle tercih edilmiştir. C# dilinin ayrıca açık kaynaklı bir topluluğa sahip olması ve gelişmiş araçları kullanarak kod yazmayı kolaylaştırması da tercih edilme sebepleri arasındadır.

Microsoft Visual Studio editörü, C# programlama dili için yaygın olarak kullanılan bir IDE'dir. Bu editör, geliştiricilere kod yazımı, hata ayıklama, derleme ve dağıtım işlemlerinde kullanılabilecekleri bir dizi araç sunar. Bu çalışmada, Microsoft Visual Studio editörü kullanılarak C# dilinde bir kare puzzle oyunu geliştirilmiştir.

B. Bulut Sistemi Azure

Bu çalışmada, bulut tabanlı bir kare puzzle oyunu geliştirirken Microsoft Azure platformundan yararlanılmıştır. Azure, uygulama geliştiricilerine bulut tabanlı uygulama ve hizmetlerin kolayca oluşturulması, dağıtılması ve yönetilmesi için bir dizi araç ve hizmet sunar. Bu nedenle, Azure'un sunduğu hizmetler, oyunumuzun bulut tabanlı bir çözüm olmasını sağlamak için tercih edilmiştir. Azure'un veritabanı

ve depolama hizmetleri gibi araçları, oyunun verilerini bulutta tutmayı ve erişimini sağlamayı kolaylaştırmıştır.

C. Linkedlist

Bağlantılı liste, birçok programlama dilinde sıkça kullanılan bir veri yapısıdır. Özellikle verilerin ekleme, çıkarma ve arama işlemlerinin sık yapıldığı durumlarda kullanımı oldukça avantajlıdır. Bağlantılı liste, verilerin hafızada ardışık olarak tutulmadığı, her verinin kendi bellek adresini sakladığı bir yapıdır. Bu sayede veri ekleme, çıkarma işlemleri oldukça hızlı ve kolay bir şekilde yapılabilir.

Bağlantılı liste yapısı, bir başlangıç noktası (head) ve bir sonlandırma noktası (tail) ile oluşturulur. Veriler, birbirlerine bağlı düğümler (node) şeklinde saklanır. Her düğüm, bir veri elemanı ve bir sonraki düğümün bellek adresini içerir. Böylece, herhangi bir düğüme erişildiğinde, bir sonraki düğümün bellek adresi kullanılarak o düğüme erişilebilir.

Bağlantılı liste yapısı, düğüm ekleme, düğüm silme, düğüm arama gibi temel işlemlerin yanı sıra, liste boyutu, liste boş mu dolu mu, veri elemanlarının sıralı olup olmadığı gibi özellikleri de kontrol etmek için kullanılabilir.

Kare Puzzle oyunu geliştirme projesinde de bağlantılı liste veri yapısı kullanılmıştır. Oyun butonları, bağlantılı liste yapısı içinde saklanmış ve kullanıcının hamleleri bu liste üzerinde gerçekleştirilmiştir.

Bağlantılı liste yapısı, diğer veri yapılarına göre bellek kullanımı açısından daha verimli bir yapıdır. Ancak, bağlantılı liste yapısının kullanılması, verilerin sıralı olarak depolanmasını engeller ve bazı durumlarda erişim hızı açısından dezavantajlı olabilir. Bu nedenle, bağlantılı liste yapısının kullanımı, uygulamanın ihtiyacına ve kullanım senaryosuna göre değerlendirilmelidir. Projenin kullanım senaryosunda istendiğinden tercih edilmiştir.

Dağıtma işlemi sırasında, doğru konuma yerleştirilen parçalar "rightPlaceMethod" adlı bir yönteme düşüyor ve burada "isRightPlace" yöntemi kullanılarak parçaların doğru yerleştirildiği kontrol ediliyor. Bu doğruluk kontrolü için bir linkedlist kullanılıyor ve bu linkedlist, tahta parçalarının ve hareketlerinin depolanmasında kullanılıyor. Yapboz parçaları, bağlı liste düğümleri olarak temsil edilebiliyor ve oyuncu yapboz parçalarını hareket ettirdiği zaman, bağlı liste düğümleri eklenip çıkarılıyor.

Bu şekilde yapboz parçalarının konumları hızlı bir şekilde değiştirilebiliyor. Soldan sağa konum kontrolü için ise her parça için boolean bir değer oluşturuluyor ve bu değerler birbirine linkedlist bağlantılarıyla bağlanıyor. Bu şekilde bağlı listenin ilk düğümünden son düğüme kadar gezinerek soldan sağa doğru konum kontrolü yapılabilir.

Resimlerin parçalara ayrılması için ise Bitmap sınıfı

kullanılıyor.

Parçaların setlenme ve dokunulmaz hale getirilmesi "rightPlaceMethod" içinde değil, "isRightPlace" yöntemi içinde olmalıdır. "rightPlaceMethod" sadece parçaların doğru yerlerine yerleştirilmesini sağlar.

Yapboz parçalarının hareket ettirilmesi için linkedlist değil, genellikle dizi kullanılır. Linkedlist kullanımı, tahta üzerindeki parçaların silinip yeniden eklenebileceği durumlarda daha uygundur.

Soldan sağa konum kontrolü için linkedlist kullanılması doğru bir seçim olabilir, ancak yine de diğer seçenekleri de göz önünde bulundurmakta fayda var. Örneğin, bir dizi kullanarak da soldan sağa kontrol sağlanabilir. Bitmap sınıfı, resimler üzerinde işlem yapmayı sağlar ancak resimlerin parçalanması için herhangi bir yöntem sağlamaz. Bu işlemi genellikle, resmin parçalarına bölünmesi için önce den belirlenmiş boyutlarda yeni resimler oluşturarak yapılır.

D. Github

Bu çalışmada, Kare Puzzle oyununun kodları GitHub üzerinde depolanmıştır. GitHub, yazılım geliştiricileri için kod depolama, paylaşma, inceleme ve işbirliği yapma imkanı sağlayan bir web tabanlı sürüm kontrol sistemi ve yazılım geliştirme platformudur. Bu nedenle, GitHub'u kullanarak, oyunumuzun kodlarını merkezi bir yerde depolayarak, herhangi bir cihazdan erişilebilir hale getirdik. Ayrıca, GitHub'un hata takip sistemi ve proje tahtası gibi araçları, yazılım geliştirme sürecimizi yönetmemize yardımcı oldu. Bu araçlar, kodların daha iyi yönetilmesini ve geliştirme sürecindeki işbirliğini kolaylaştırdı.

III. KULLANILAN METODOLOJİ

Bu projede, bağlantılı liste veri yapısı kullanılmıştır. Bağlantılı liste, taşların konumlarının depolanmasında kullanılan veri yapısıdır. Bu özellik, bağlantılı listelerin farklı boyutlarda oluşturulabilmesi sayesinde kolayca gerçekleştirilmiştir. Oyunun yapısı, C# programlama dili ve Microsoft Visual Studio aracılığıyla oluşturulmuştur.

A. Proje Yönetimi

Projenin kaynak kodunun C# dilinde yazılması kararı alındıktan sonra, algoritmaların tasarımı için fikir danışmanlığı yapmak üzere bir araştırma yapıldı. Bu araştırmada, özellikle Veri Bilimi ders videoları, dokümanları ve Youtube'da yer alan örnek projeler incelendi.

Linkedlist yapısı, bu incelemeler sonucunda tasarımı yapılan önemli bir veri yapısıdır. Linkedlist, verilerin birbirine bağlı bir dizi şeklinde tutulduğu bir veri yapısıdır. Bu yapı, her elemanın kendinden sonraki elemana bir referans veya işaretçi ile bağlı olduğu bir zincir olarak tasarlanmıştır.

Linkedlist yapısı, özellikle veri manipülasyonunda ve performanslı veri erişiminde kullanılabilir. Örneğin, bir listedeki herhangi bir elemanın eklenmesi veya çıkarılması diğer elemanlara etki etmez, bu da işlemlerin hızlı ve verimli bir şekilde gerçekleştirilmesine olanak tanır.

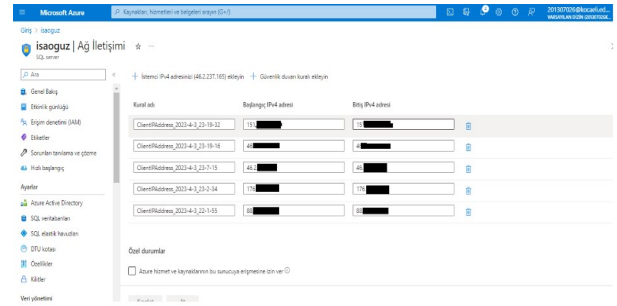
Bu tasarımların sonucunda, proje lideri İsa Oğuz tarafından Linkedlist yapısı için gerekli kodlamalar yapılmıştır. Bu kodlamaların ardından, proje çalışır hale getirilerek test edilmiştir.

B. Proje Detaylı Tanıtımı

1. Azur Bulut Sistemi:

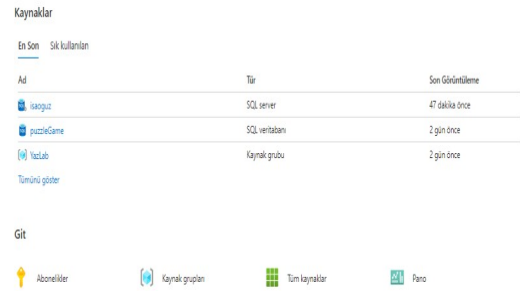
Projede kullanılan Azur bulut sistemi ve database sayesinde, kullanıcı kayıt bilgilerinin saklanması ve kullanıcılara erişim izni verilmesi mümkün olmuştur. Bu sayede, kullanıcıların sisteme erişimini kontrol etmek ve kimlerin hangi verilere erişebileceğini belirlemek mümkün olmuştur. Veritabanının kullanımı, verilerin güvenliği ve erişim kontrolü açısından önemlidir ve projenin başarısı için kritik bir rol oynamaktadır. Bu nedenle, Azur bulut sistemine yüklenen veritabanında kullanıcı adları ve şifreleri saklanmaktadır.

Bu sistemde Ip adresinin bulut sistemi sahibi tarafından girilmelidir. Aksi halde uygulama kullanılamaz. Ip adreslerinin yüklendiği sayfanın görseli Resim-1 de gösterilmiştir.



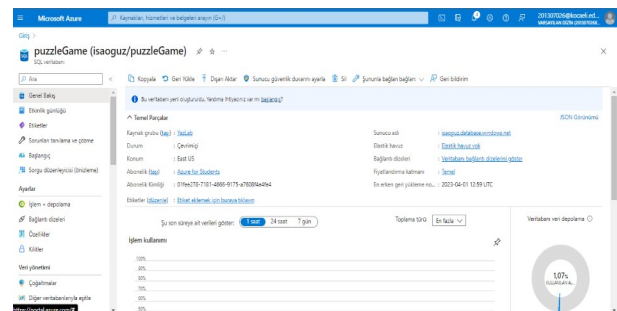
Resim I: Azur Ip Adreslerini Ekleme

Azur server ve veritabanı bağlantısının Azur Web sitesinde yapıldığı arayüz Resim-2de gösterilmiştir.



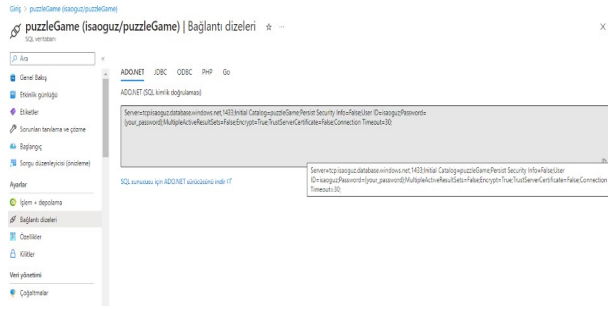
Resim II: Azur Veritabanı Bilgi Ekranı

Veritabanı-Azur ilişki ayarlamalarının ve veritabanı detaylarının bulunduğu arayüz Resim-3 de gösterilmiştir



Resim III: Azur Veritabanı Detayları

Azure bağlantı dizeleri, Azure servisleriyle çalışırken kullanılan, veritabanı ve diğer Azure hizmetleriyle bağlantı kurmak için gerekli olan özel bir yapıdır. Bu dizeler, veritabanı sunucusunun adresi, veritabanı adı, kullanıcı adı, parola ve diğer bağlantı seçenekleri gibi bilgileri içerir Resim-4 de gösterilmiştir.



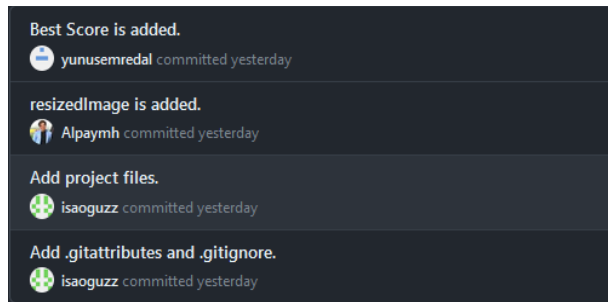
Resim IV:Azur Bağlantı Dizeleri

2. Github :

Proje lideri İsa Oğuz tarafından, Yunus Emre Dal ve Mehmet Yusuf Alpay ile paylaşılan bir gizli repo, Github platformu üzerinde açıldı. Proje planı kapsamında, Visual Studio kullanılarak yapılacak görevler belirlendi ve bunlar için al, gönder ve yorumlar yapıldı.

Github platformu üzerinde yapılan değişikliklerin beyanları ve yorumları için commitler atıldı. Bu commitler, Resim-5’da gösterildi.

Proje takımının, yapılacak işler hakkındaki görüşlerini ve yaptıkları değişiklikleri takip etmek için Github platformu kullanması oldukça faydalı bir yöntemdir.



Resim V:Github Raporları

3. ADO.NET:

ADO.NET (ActiveX Data Objects .NET), Microsoft tarafından .NET Framework’ün bir parçası olarak sunulan bir veritabanı bağlantı teknolojisidir. ADO.NET, veritabanlarına bağlanmak, verileri sorgulamak, güncellemek ve yönetmek için bir dizi sınıf ve yöntemler içerir.

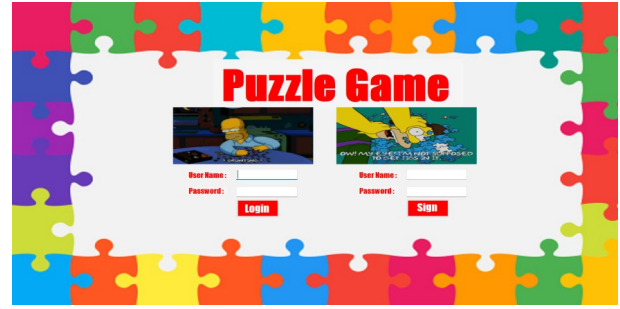
ADO.NET, .NET Framework sınıf kütüphanesinde System.Data adlı bir ad alanı altında yer almaktadır. ADO.NET, bir veritabanı sunucusu ile bir .NET uygulaması arasındaki etkileşimi kolaylaştıran bir dizi nesne ve sınıf içermektedir. Bu nesneler ve sınıflar, veritabanı bağlantıları, sorgular, güncelleme işlemleri, veri okuma ve yazma işlemleri, transaksyon yönetimi ve diğer veritabanı işlemlerinin yapılmasını sağlar.

ADO.NET, birçok farklı veritabanı yönetim sistemi ile çalışabilen bir teknolojidir ve birçok veritabanı sunucusu için

özel sürücüler ve veri sağlayıcıları mevcuttur. Bu da, .NET uygulamalarının farklı veritabanlarına kolaylıkla bağlanabilmesini sağlar.

C. Arayüz Tasarımı ve Uygulama Kullanımı :

Giriş Ekranı : Uygulamaya erişmek için kullanıcının kimlik doğrulama işlemini gerçekleştirdiği yerdir. Bu ekran, kullanıcı adı ve şifre gibi kimlik bilgilerini alarak, bu bilgilerin doğru olup olmadığını kontrol eder ve kullanıcının uygulamaya erişim izni verir ya da vermemeye kararı verir. Ayrıca kullanıcıların uygulamaya kaydolmalarını ve hesaplarını oluşturmalarını sağlar. Bu ekran, kullanıcılardan belirli bilgileri girmelerini isteyerek, bu bilgileri uygulamanın veritabanına kaydeder aynı zamanda giriş için veritabanına kayıtlı bilgileri kontrol eder . Resim-6 de giriş ekranı gösterilmiştir.



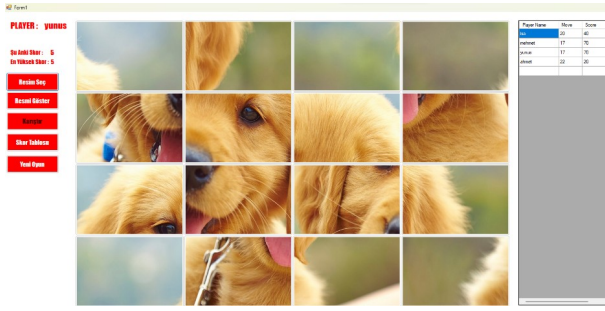
Resim VI:Giriş Ekranı

Oyun Ekranı : Oyun ekranında sol panelde çeşitli öğeler yer almaktadır. Bu öğeler arasında:

- Oyuncu ismi: Kullanıcının oyunda kullandığı isim bilgisini içermektedir.
- Şuan ki skor: Kullanıcının oyun sırasında topladığı puanları ve o anki skorunu göstermektedir.
- En yüksek skor: Kullanıcının daha önce oynadığı oyunlarda elde ettiği en yüksek skor bilgisini içermektedir.
- Resim seç: Kullanıcının oyun için kullanacağı resmi seçebileceği butondur.
- Resim göster: Kullanıcının seçtiği resmi harici bir sayfa olarak açıp göstermesini sağlar
- Karıştır: Kullanıcının seçtiği resmi karıştırarak oyunu başlatmasını sağlar.
- Skor tablosu: Kullanıcının daha önce oynadığı oyunların skor tablosunu gösterir.
- Yeni oyun: Kullanıcının yeni bir oyun başlatması için butondur.

Sol panelde yer alan bu öğeler, kullanıcının oyunu daha verimli ve kolay bir şekilde oynamasına yardımcı olmaktadır. Ayrıca, kullanıcının oyun sırasında performansını takip edebilmesi ve önceki oyunlardan elde ettiği en yüksek skor bilgisine erişebilmesi için skor tablosu ve en yüksek skor öğeleri de önemli bir rol oynamaktadır.

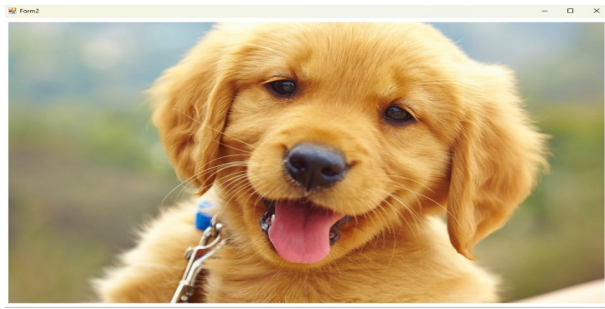
Projenizdeki arayüzün orta panelinde puzzle oyunu için 16 parça bulunmaktadır. Bu parçalar oyunda doğru yerlerine yerleştirildiğinde tamamlanacak olan bir görüntüyü oluştururlar. Sağ tarafta bulunan panelde oyuncu ismi, hamle sayısı ve skor bilgilerini içeren bir tablo bulunmaktadır. Bu tabloda tüm oyuncuların isimleri ve oyun içinde yaptıkları hamle sayısı ve skorları yer almaktadır Resim-7 de gösterilmiştir.



Resim VII: Oyun Ekranı

Resmi göster" butonuna tıklandığında, resim ayrı bir ekran üzerinde açılır ve kullanıcının daha büyük bir boyutta incelemesine olanak sağlanır. Bu özellik Resim-8'de gösterilmiştir

Skor tablosu butonuna basıldığında, Giriş Ekranının



Resim VII: Görsel Ekranı

sağında gözükten tabloyu daha büyük bir şekilde yeni bir ekranda açılır. Bu sayede kullanıcılar skorları daha net bir şekilde görebilme imkanı elde ederler. Bu özellik Resim-9'da gösterilmiştir.

Form2			
Player Name	Move	Score	
İsa	20	40	
mehmet	17	70	
yunus	17	70	
ahmet	22	20	

Resim VIII:Skor Ekranı

Ayrıca, skor tablosu ekranı, kullanıcılara yüksek skor sahibi olan diğer oyuncuların kullanıcı adlarını ve elde ettikleri puanları gösterir. Bu sayede kullanıcılar, kendi performanslarını diğer oyuncularınkiyle karşılaştırabilir ve daha iyi bir skor elde etmek için çalışabilirler.

Uygulama Kullanımı :Uygulamada kullanıcının yeni bir oyun başlatmak için "Yeni Oyun" butonuna tıklaması durumunda, oyun ekranı sıfırlanarak kullanıcının giriş ekranına yönlendirildiği görülmektedir. Bu işlevsellik, kullanıcının herhangi bir zamanda yeni bir oyun başlatma ihtiyacı olduğunda kolayca erişebilmesini ve oyunun baştan başlamasını sağlamaktadır.

Oyun ekranında yer alan "Karıştır" butonuna basıldığında, puzzle parçaları rastgele bir şekilde karıştırılır ve kullanıcının amacı, parçaları doğru konumlarına yerleştirerek tamamlamaktır.

Kocaeli University Information System Engineering ©2023 IEEE

Kullanıcının bir parçayı doğru yere yerleştirdiği zaman, puanı artar ve artık o parça yerinden çıkarılamaz. Bu, kullanıcının oyunu daha zorlu hale getirerek daha fazla puan kazanmasını sağlar. Bu şekilde, kullanıcının her bir parçayı doğru yere yerleştirerek puan kazanması ve her bir parça doğru yere yerleşince bir daha değiştirilmesine izin verilmez. "Karıştır" butonu ile doğru yere yerleşen parçalar olursa direkt puanı verilir.

D. Yapılabilecek geliştirmeler

- Oyuna farklı zorluk seviyeleri eklenerek, oyuncuların seviyelerine göre farklı zorluklarda oynamaları sağlanabilir.
- Oyunun daha fazla parça sayısına sahip versiyonu da eklenebilir, böylece oyuncuların zorluk seviyeleri artırılabilir.
- Oyun süresi eklenebilir, böylece oyunculara belirli bir süre içinde tamamlamaları gereken bir hedef verilebilir.
- Projenin exe versiyonu oluşturularak Setup şeklinde bir hale gelmesi imkanı sağlanabilir.

IV. SONUÇ

Projenin sonucunda, C# dilinde başarılı bir şekilde bir Kare Puzzle oyunu geliştirildi. Bu oyun, kullanıcılara keyifli bir oyun deneyimi sunarken aynı zamanda programlama becerilerini geliştirmelerine yardımcı olacak bir özellik taşıyor. Oyun, Visual Studio editörü kullanılarak geliştirildi ve Azure bulut platformunda barındırıldı. Azure SQL veritabanı kullanılarak kullanıcı kayıt bilgileri ve oyun skorları saklandı. Ayrıca, Github kullanılarak proje takip edildi.

Oyunun işlevselliği, oyun ekranındaki düğmelerin kullanıcı dostu tasarımı ve puanlama sistemi başarılı bir şekilde uygulandı. Oyun ekranında yer alan sol paneldeki bilgiler, oyuncuların skorlarını takip etmelerine ve en iyi performanslarını izlemelerine olanak tanırken, sağ paneldeki tablo ile de diğer oyuncularla karşılaştırılabilirler.

Sonuç olarak, bu proje sayesinde C# dili, Visual Studio editörü, Azure bulut platformu ve Github kullanımı gibi konularda bilgi sahibi olunmuştur. Kare Puzzle oyununun başarılı bir şekilde geliştirilmesi, bu teknolojilerin başarılı bir şekilde uygulanmasının göstergesidir.

REFERENCES

- <https://www.youtube.com/watch?v=3fEGgSWibE0>
- <https://www.ktu.edu.tr/bilgiislem/ipv4adresiogrenme>
- https://www.youtube.com/watch?v=njTh_OwMljA
- <https://www.gokhan-gokalp.com/c-linked-list-singly-linked-list-nedir/>
- <https://learn.microsoft.com/tr-tr/dotnet/api/system.collections.generic.linkedlist-1?view=net-7.0>
- https://www.youtube.com/watch?v=bgSC-Wgmm_s
- https://www.youtube.com/watch?v=cEsicmlaWDk&list=PL3iOx6lykrwr1VijTcZI3l8Hz_9W3Pb9F