

Projenin Amacı ve Genel Tanımı

Bu proje, oyuncunun kontrolünde bir renkli karo birleştirme ve patlatma oyununu geliştirmeyi hedeflemektedir. Oyunun temel amacı, aynı renkteki en az 3 nesneyi yatay veya dikey olarak yan yana getirerek patlatmak ve puan kazanmaktır. Oyunu daha eğlenceli hale getirmek için joker nesneler ve dinamik oyun kontrolleri eklenmiştir. Proje nesneye yönelik programlama (OOP) ilkelerine uygun olarak geliştirilmiştir.

Proje Gereksinimlerinin Karşılanması

Aşağıda, ödevde belirtilen gereksinimlerin nasıl karşılandığı detaylı olarak açıklanmıştır:

Menü Ekranı

Menü ekranında oyuncunun ismini yazması ve oyuna giriş yapması sağlanmıştır. Menü ekranı özellikleri şunlardır:

- Oyunu başlatmak için bir buton bulunmaktadır (button1_Click metodu).
- Oyuncu ismini yazmak için bir TextBox bulunmaktadır.
- Oyuncu ismi kontrolü yapılmıştır. Eğer isim girilmezse, uyarı mesajı gösterilir.
- "En İyi Skorlar" linkine tıklanarak en yüksek puanlar ayrı bir formda görüntülenebilir (linkLabel1_LinkClicked metodu).

Ekranı ayrıca oyun bilgileri ve yönlendirmeler sağlanmıştır:

- “Nesneleri hareket ettirmek için Mouse ile seçiniz ve/veya yön tuşlarını kullanınız.”
- “Oyunu durdurmak için P tuşuna basınız.”

Oyun Ekranı

Oyun ekranı dinamik ve kullanıcı dostu bir yapıda tasarlanmıştır:

Oyun Alanı

- 8x8 boyutunda bir oyun alanı oluşturulmuştur.
- Oyun başlangıcında renkli nesneler ve jokerler rastgele yerleştirilmiştir (InitializeGameBoard metodu).
- İlk yerleşimde eşleşme olmaması sağlanmıştır (WouldCauseMatch metodu).

Oyun Nesneleri

- 4 farklı renkli nesne ve 4 farklı joker (Roket, Helikopter, Bomba, Gökkuşağı) bulunmaktadır. Jokerlerin özellikleri:
 - **Roket:** Yatay veya dikeydeki tüm nesneleri patlatır (RocketTile sınıfı).
 - **Helikopter:** Rastgele bir nesneyi patlatır (CopterTile sınıfı).
 - **Bomba:** Çevresindeki 8 komşusunu patlatır (BombTile sınıfı).
 - **Gökkuşağı:** Rastgele bir renk seçer ve bu renkteki tüm nesneleri patlatır (RainbowTile sınıfı).
- Kutuların hareket ettirilmesi sağlanmıştır. Hareketler yalnızca yatay ve dikey olabilir (AreAdjacent metodu).
- Jokerlerin özel yetenekleri başarılı bir şekilde uygulanmıştır.

Patlatma Mekanizması

- Aynı renkte en az 3 nesne yatay veya dikey olarak yan yana geldiğinde nesneler patlatılmaktadır (CheckAndHandleMatches metodu).
- Patlatılan boşluklara yeni nesneler dinamik olarak eklenmektedir (DropTiles ve FillEmptySpaces metotları).
- Dinamik kontrol sağlanmış ve patlatmalar devam ettiği sürece oyun alanı kontrol edilmektedir.

Skor ve Zamanlama

- Nesneler patlatıldığında oyuncuya puan verilmektedir (UpdateScore metodu). Her eşleşme için 10 puan verilmiştir.
- Oyunun başlangıcında bir zamanlayıcı başlatılmıştır (GameTimer_Tick metodu). Zaman sıfırlandığında oyun sona ermektedir (EndGame metodu).
- Toplam süre ve puan, oyun ekranında gerçek zamanlı olarak gösterilmektedir (lblScore, lblTime).

Kontroller

- Oyuncu fare ile kutucukları seçerek ve klavye yön tuşlarını kullanarak hareket edebilir (Tile_Click ve HandleKeyboardInput metotları).
- Oyun sırasında “P” tuşuna basıldığında oyun duraklatılmakta ve devam ettirilebilmektedir (PauseGame metodu).

En İyi Skorların Kaydedilmesi ve Gösterimi

- Oyunun sonunda oyuncunun skoru kontrol edilmekte ve en yüksek 5 skor arasında ise kaydedilmektedir (CheckHighScore metodu).
- Skorlar bir metin dosyasına kaydedilmekte ve en yüksek 5 skor her seferinde güncellenmektedir (SaveHighScores ve LoadHighScores metotları).
- Menü ekranındaki “En İyi Skorlar” linkine tıklanarak en yüksek puanlar ayrı bir ekranda görüntülenebilmektedir (ShowHighScores metodu).

Nesneye Yönelik Programlama (OOP) Kullanımı

Proje, nesneye yönelik programlama prensiplerine uygun olarak geliştirilmiştir. Aşağıda OOP ilkelerinin projede nasıl uygulandığı açıklanmıştır:

- **Kalıtım (Inheritance):** Oyun nesneleri için bir Tile sınıfı tanımlanmış ve jokerler bu sınıftan türetilmiştir (NormalTile, RocketTile, CopterTile, BombTile, RainbowTile). Tile sınıfı, ortak özellikleri ve işlevleri içerirken, türetilmiş sınıflar jokerlerin özel davranışlarını içermektedir.
- **Polimorfizm (Polymorphism):** Tile sınıfındaki sanal metotlar (OnMatched, Destroy) türetilmiş sınıflar tarafından ezilmiş ve jokerlerin özel işlevleri uygulanmıştır.
- **Kapsülleme (Encapsulation):** Sınıflardaki özellikler ve metotlar uygun erişim belirleyicilerle (private, protected, public) korunmuştur.
- **Soyutlama (Abstraction):** Tile sınıfı soyut bir yapı olarak tasarlanmış ve jokerler bu sınıfı genişletmiştir. Böylece kod tekrarından kaçınılmıştır.
- **Statik Özellikler:** Menü ekranında oyuncu adını tutmak için PlayerName adlı statik bir özellik tanımlanmıştır. Bu özellik Form2 sınıfında kullanılmaktadır.

Sonuç ve Değerlendirme

Bu proje, belirtilen gereksinimleri başarıyla karşılamış ve eğlenceli bir oyun deneyimi sunmuştur. Proje kapsamında:

- Nesneye yönelik programlama ilkeleri uygulanmış,
- Dinamik bir oyun alanı oluşturulmuş,
- Joker nesneler ile oyun daha ilgi çekici hale getirilmiştir.