



T.C

**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİęİ PROGRAMI**

## **ÖDEV KONUSU**

### **POKEMNON KART OYUNU**

**Hazırlayan**

**Esmanur AYDIN**

**220501005**

**DERS SORUMLUSU**

**Dr. Öğr. Üyesi ERCAN ÖLÇER**

**04/07/2024**

[esm4ydn \(Esmanur AYDIN\) · GitHub](#)

*Yeniköy Mahallesi Ilıca Caddesi No:29, Başiskele,Kocaeli*  
[info@kocaelisaglik.edu.tr](mailto:info@kocaelisaglik.edu.tr) [kocaelisaglik.edu.tr](http://kocaelisaglik.edu.tr)

---

## İÇİNDEKİLER

1.	GİRİŞ .....	3
2.	GEREKSİNİM ANALİZİ .....	3
3.	TASARIM .....	6
4.	UYGULAMA.....	10
5.	TEST VE DOĞRULAMA .....	12
6.	KAYNAKÇA .....	14

## 1. GİRİŞ

### Projenin Amacı:

Bu projede, kullanıcı ve otomatik oyuncu arasında oynanabilen basit bir Pokémon kart oyunu geliştirilmiştir. Oyunda toplam 10 farklı Pokémon kartı bulunmakta ve her iki oyuncuya rastgele 3'er kart dağıtılmaktadır. Oyuncular, sırayla kart koyarak en yüksek hasar puanını elde etmeye çalışırlar. Oyunun hem kullanıcı vs bilgisayar hem de bilgisayar vs bilgisayar modları bulunmaktadır.

## 2. GEREKSİNİM ANALİZİ

### Arayüz Gereksinimleri:

#### Giriş Ekranı:

- Başlık:** "Pokemon Card Game - Player Selection"
- Arka Plan:** Pokemon karakterleri içeren bir arka plan resmi.
- Başlık Metni:** "OYUNCU MOD SEÇİMİ" (Kırmızı renkte, Arial, 30pt, kalın)
- Oyun Modu Seçimi:**
  - Butonlar:**
    - Oyuncu vs Bilgisayar:** Kullanıcı ve bilgisayarın karşı karşıya geleceği oyun modu.
    - Bilgisayar vs Bilgisayar:** İki bilgisayarın karşı karşıya geleceği oyun modu.
  - Buton Yerleşimi:** İki buton ortalanmış olarak yan yana yerleştirilmiştir.
  - Buton Stili:** Butonlar düz, okunabilir metin içerecek şekilde tasarlanmıştır.

#### Kart Görselleri:

- Boyutlandırma:** Kart görselleri 155px genişliğinde olacak şekilde boyutlandırılacak.
- Kalite:** Kartların yüksek çözünürlüklü ve net bir şekilde görünmesini sağlamak için orijinal görüntü kalitesinin korunması sağlanacak.

Ödev No:Mazeret	Tarih 04.07.2024	3/15
-----------------	------------------	------

---

## Oyun Ekranı:

- **Başlık:** "Pokemon Card Game"
- **Arka Plan:** Mavi arka plan üzerinde beyaz yıldız desenleri.
- **Skor Panelleri:**
  - **Oyuncu ve Bilgisayar Skorları:** Ekranın alt kısmında, kullanıcı ve bilgisayarın skorlarını gösterecek iki ayrı panel.
- **Kart Panelleri:**
  - **Oyuncu Paneli:** Sol tarafta oyuncunun kartlarını gösterecek panel.
  - **Bilgisayar Paneli:** Sağ tarafta bilgisayarın kartlarını gösterecek panel.
- **Kart Deste Paneli:**
  - **Orta Destekartı:** Ortada bulunan destedeki kartları gösterecek. Kartın boyutu 250px genişliğinde olacak ve deste tükenince görünmez olacak.
- **Oyun Butonu:**
  - **Tur Oynama Butonu:** "Play Round" butonu ekranın alt kısmında ortalanmış şekilde yer alacak ve kullanıcı her tura geçmek için bu butona basacak.
- **Tur Sonucu Bilgisi:**
  - **Sonuç Etiketi:** Ekranın üst kısmında, tur sonucunu gösterecek (örn. "Player wins the round!").

## Oyun Kuralları:

- **Tur Başlangıcı:** Her oyuncuya başlangıçta 3 kart dağıtılacak.
- **Puanlama:** Turu kazanan oyuncu 5 puan kazanacak. En yüksek hasarı veren kart sahibi turu kazanacak.
- **Oyun Sonu:** Kartlar bitene kadar oyun devam edecek ve en yüksek puanı alan oyuncu oyunu kazanacak.
- **Kart Çekme:** Desteden kart çekme işlemi, ortadaki desteye tıklayarak gerçekleşecek ve bilgisayar otomatik olarak kart çekecek.

## Donanım Gereksinimleri:

- **İşlemci (CPU):** En az Intel i5 veya eşdeğeri bir AMD işlemci.
- **Bellek (RAM):** En az 8 GB RAM.
- **Depolama:** En az 500 MB boş disk alanı.

Ödev No:Mazeret	Tarih 04.07.2024	4/15
-----------------	------------------	------

- 
- **Grafik Kartı (GPU):** En az 2 GB VRAM'e sahip bir grafik kartı.
  - **Ekran:** En az 1024x768 çözünürlüğü destekleyen bir monitör.
  - **Giriş Aygıtları:** Klavye ve fare.
  - **Ses:** Hoparlör veya kulaklık.
  - **İşletim Sistemi:** Windows 10, macOS 10.14 veya üstü, Linux (Ubuntu 18.04 veya üstü).

#### Fonksiyonel Gereksinimler:

##### ☐ **Kullanıcı Arayüzü (UI):**

- Başlangıç ekranında kullanıcıya "Oyuncu vs Bilgisayar" ve "Bilgisayar vs Bilgisayar" seçeneklerini sunan bir arayüz bulunur.
- Oyuncuların ve bilgisayarın ellerindeki kartları görüntüleyebileceği bir oyun ekranı vardır.
- Oyun ekranında her iki oyuncunun da skorlarını gösteren paneller yer alır.
- Ortada, kart destesini gösteren bir alan vardır.
- "Play Round" butonu ile turu başlatma ve ilerletme işlevi sağlar.

##### ☐ **Oyun Kuralları:**

- Her oyuncuya başlangıçta 3 kart dağıtılır.
- Her turda, oyuncular ellerindeki kartlardan birini oynar ve en yüksek hasar puanına sahip olan kartın sahibi turu kazanır.
- Turu kazanan oyuncu 5 puan alır.
- Tüm kartlar oynandıktan sonra oyun biter ve en yüksek puana sahip oyuncu kazanır.

##### ☐ **Kart Yönetimi:**

- Kartlar rastgele dağıtılır.
- Kartlar oyuncuların elinden çıkarıldıktan sonra ortadaki desteye geri döner.
- Her oyuncunun elindeki kart sayısı her zaman doğru bir şekilde güncellenir ve görüntülenir.

##### ☐ **Skor Takibi:**

Ödev No:Mazeret	Tarih 04.07.2024	5/15
-----------------	------------------	------

- 
- Oyuncuların ve bilgisayarın skorları her tur sonunda güncellenmeli ve ekranda görüntülenebilir.
  - Oyun sonunda kazanan oyuncu ve final skorları ekranda gösterilir.

□ **Oyun Modları:**

- "Oyuncu vs Bilgisayar" modunda bir oyuncu ve bir bilgisayar arasında oyun oynanır.
- "Bilgisayar vs Bilgisayar" modunda iki bilgisayar arasında oyun oynar ve oyuncu yalnızca turların ilerlemesini izler.

### 3. TASARIM

#### *Mimari Tasarım*

Projenin mimarisi, Model-View-Controller (MVC) tasarım desenini takip edecektir. Bu desen, kullanıcı arayüzü ile iş mantığı ve veri arasında açık bir ayırım sağlar, böylece kodun bakımını ve genişletilmesini kolaylaştırır.

**1. Model (Veri Katmanı):**

- Kartlar, Desteler ve Oyuncu el modelleri gibi oyun verilerini ve mantığını temsil eder.
- Bu katman, oyun kurallarını ve veritabanı işlemlerini içerir.

**2. View (Görünüm Katmanı):**

- Kullanıcı arayüzünü (UI) ve kullanıcı ile etkileşimleri yönetir.
- Bu katman, Java Swing kullanarak oyun ekranlarını ve bileşenlerini oluşturur.

**3. Controller (Kontrol Katmanı):**

- Kullanıcı girişlerini alır ve uygun model güncellemelerini ve görünüm değişikliklerini yönetir.
- Oyunun işleyişini ve akışını kontrol eder.

Ödev No:Mazeret	Tarih 04.07.2024	6/15
-----------------	------------------	------

---

## *Kullanılacak Teknolojiler*

### **1. Java:**

- Oyun mantığı, veri modelleri ve genel işleyiş için temel programlama dili.
- Nesne yönelimli programlama (OOP) prensipleri kullanılarak sınıflar ve nesneler oluşturur.

### **2. Java Swing:**

- Kullanıcı arayüzü (UI) tasarımı ve bileşenleri için kullanılır.
- Oyuncu ve bilgisayar panelleri, kartlar ve skor gösterimi gibi grafiksel öğeler bu kütüphane ile oluşturulur.

### **3. Java Collections Framework:**

- Kart destesi ve oyuncu elleri gibi veri yapıları için kullanılır.
- ArrayList ve diğer koleksiyon sınıfları veri yönetiminde kullanılır.

### **4. ImageIcon:**

- Kart görsellerini ve diğer grafiksel öğeleri yüklemek ve görüntülemek için kullanılır.
- Yüksek kaliteli ve yeniden boyutlandırılabilir görüntü desteği sağlar.

### **5. Event Handling:**

- Kullanıcı etkileşimleri (tıklamalar, seçimler vb.) için olay işleyicileri.
- ActionListener, MouseListener gibi arayüzler ile kullanıcı girişleri yönetir.

### **6. JOptionPane:**

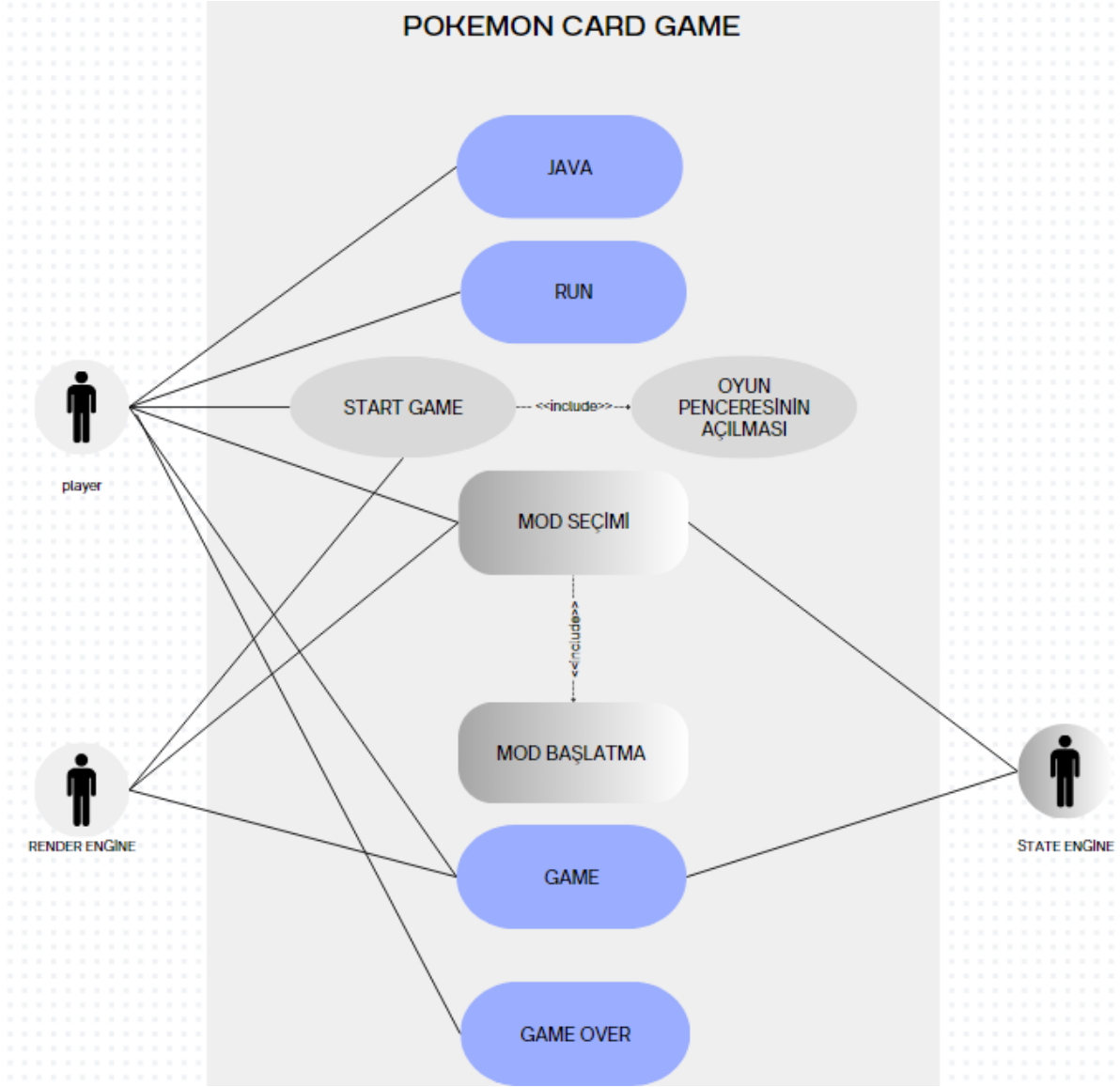
- Bilgilendirme ve sonuç mesajları için kullanılır.
- Oyun sonunda kazananı ve skorları göstermek için diyalog pencereleri oluşturur.

### **7. Design Patterns:**

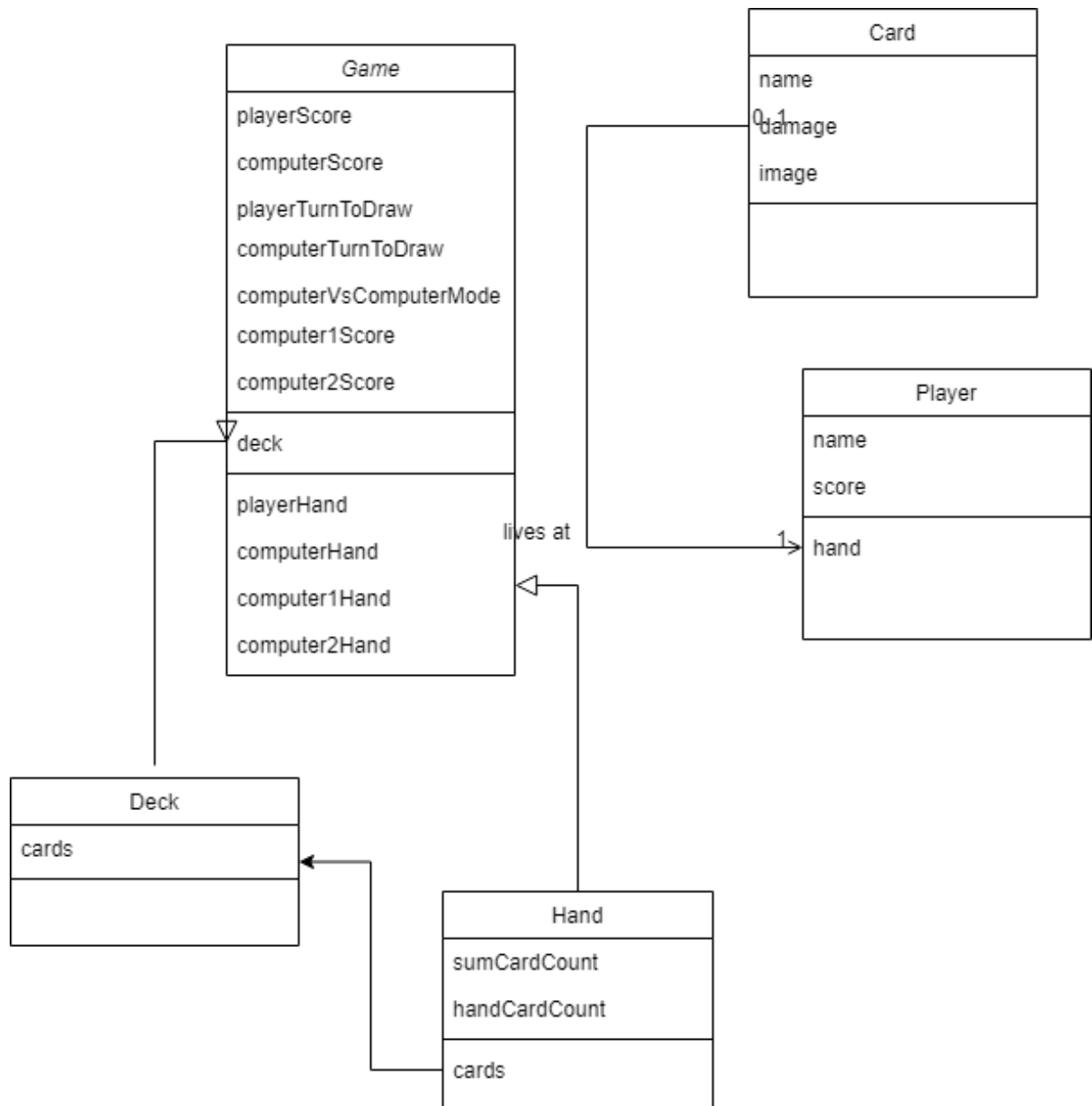
- MVC (Model-View-Controller) tasarım deseni, kodun düzenli ve sürdürülebilir olmasını sağlamak için kullanılır.
- Bu desen, bileşenlerin birbirinden bağımsız olarak geliştirilmesini ve test edilmesini kolaylaştırır.

Bu tasarım ve teknolojiler ile oyun, kullanıcı dostu bir arayüze sahiptir ve kodun bakımını kolaylaştırır şekilde yapılandırılmıştır. Ayrıca, yüksek kaliteli grafikler ve akıcı bir oyun deneyimi sunulacaktır.

Ödev No:Mazeret	Tarih 04.07.2024	7/15
-----------------	------------------	------







---

## 4. UYGULAMA

### 4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

#### 1. Zorluk: Kart Dağıtımı ve Görüntülenmesi

- **Açıklama:** Oyunculara ve bilgisayara başlangıçta 3 kart dağıtmak ve bu kartların doğru bir şekilde görüntülenmesini sağlamak.
- **Çözüm:** dealCards() metodunda kart dağıtımı yapılırken her iki oyuncuya da kesinlikle 3'er kart verildiğini kontrol eden bir döngü eklendi. Ayrıca, kartların UI'da doğru görüntülenmesi için displayCards() metodunda kartların sayısını dinamik olarak ayarlayan GridLayout kullanıldı.

#### 2. Zorluk: Bilgisayar vs Bilgisayar Modu

- **Açıklama:** Bilgisayar vs Bilgisayar modunda her iki bilgisayar oyuncusunun otomatik olarak tur oynamasını sağlamak.
- **Çözüm:** Bu mod için özel bir Game nesnesi ve ilgili UI bileşenleri oluşturuldu. Bilgisayarların otomatik olarak tur oynamasını sağlamak için playRound() ve drawCardForComputer() metodları otomatik olarak çağırıldı.

#### 3. Zorluk: Grafik Kalitesi ve Görsellerin Boyutlandırılması

- **Açıklama:** Kart görsellerinin yüksek kalitede ve uygun boyutta görüntülenmesini sağlamak.
- **Çözüm:** ImageIcon sınıfı kullanılarak kart görselleri yüklendi ve getResizedImage() metodu ile yeniden boyutlandırıldı. Bu sayede, kartların kalitesi korunarak ekranda doğru boyutlarda görüntülendi.

#### 4. Zorluk: Skor Takibi ve Güncellenmesi

- **Açıklama:** Oyuncuların ve bilgisayarın skorlarının doğru bir şekilde takip edilmesi ve ekranda güncellenmesi.
- **Çözüm:** Skorları güncellemek için updateScores() metodu eklendi. Bu metod, her tur sonunda oyuncuların ve bilgisayarın skorlarını günceller ve UI'da gösterir.

#### 5. Zorluk: MVC Yapısının Uygulanması

Ödev No:Mazeret	Tarih 04.07.2024	10/15
-----------------	------------------	-------

- 
- **Açıklama:** Model-View-Controller (MVC) yapısının projeye uygun şekilde uygulanması ve bileşenlerin birbirinden bağımsız olarak çalışmasının sağlanması.
  - **Çözüm:** Model katmanı, oyun verilerini ve mantığını temsil ederken, View katmanı kullanıcı arayüzünü ve Controller katmanı kullanıcı girişlerini ve oyun akışını yönetti. Bu sayede, kodun daha okunabilir, sürdürülebilir ve genişletilebilir olması sağlandı.

#### 6. Zorluk: Oyun Sonu ve Kazananın Belirlenmesi

- **Açıklama:** Oyunun sonunda kazanan oyuncunun doğru bir şekilde belirlenmesi ve kullanıcıya bildirilmesi.
- **Çözüm:** `getGameResult()` metodu eklenerek oyun sonunda oyuncuların puanları karşılaştırıldı ve kazanan belirlenerek `JOptionPane` ile kullanıcıya gösterildi.

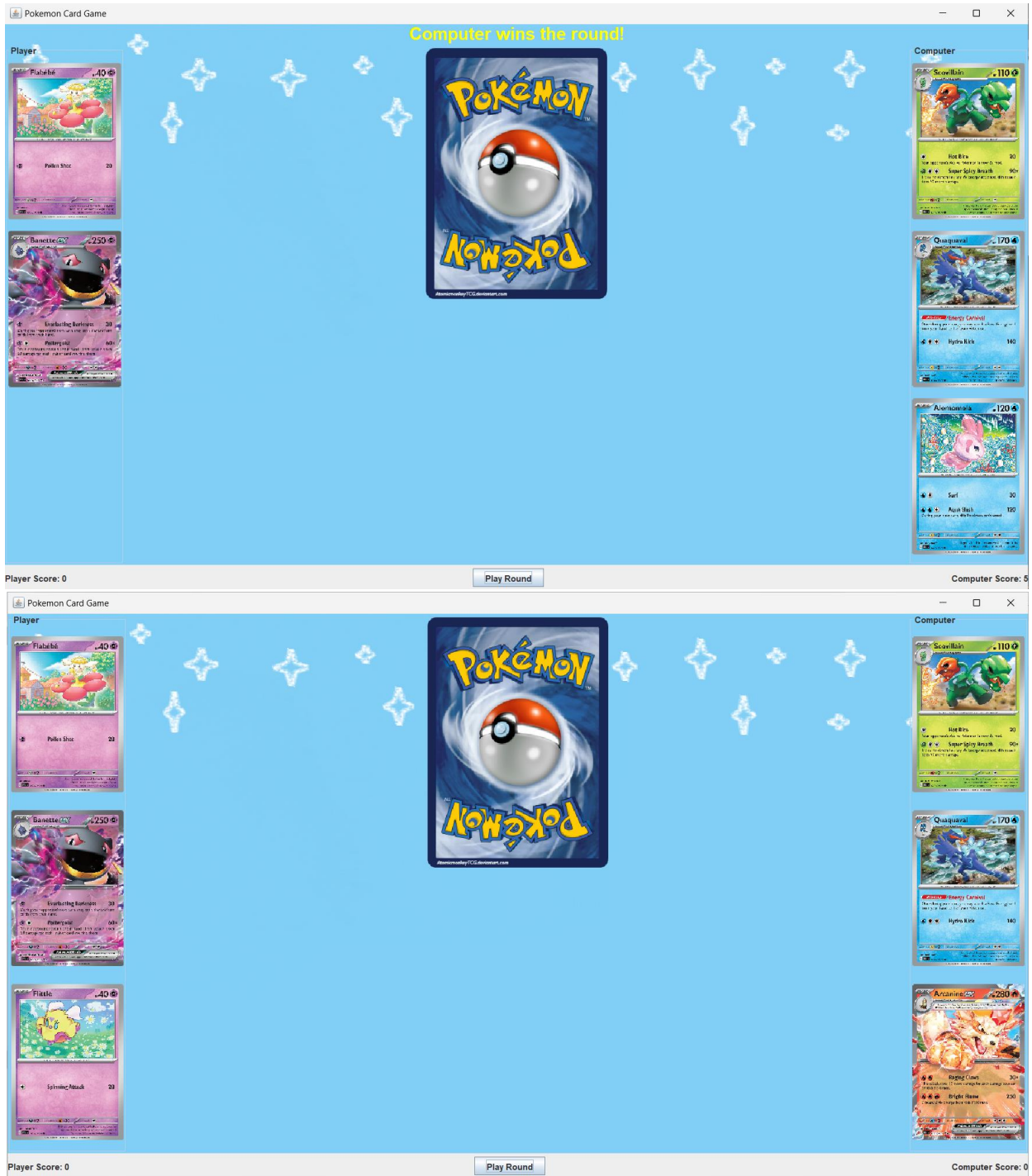
Bu zorluklar, dikkatli bir planlama ve uygun çözüm yöntemleri ile aşılmış ve projenin başarılı bir şekilde tamamlanması sağlanmıştır.

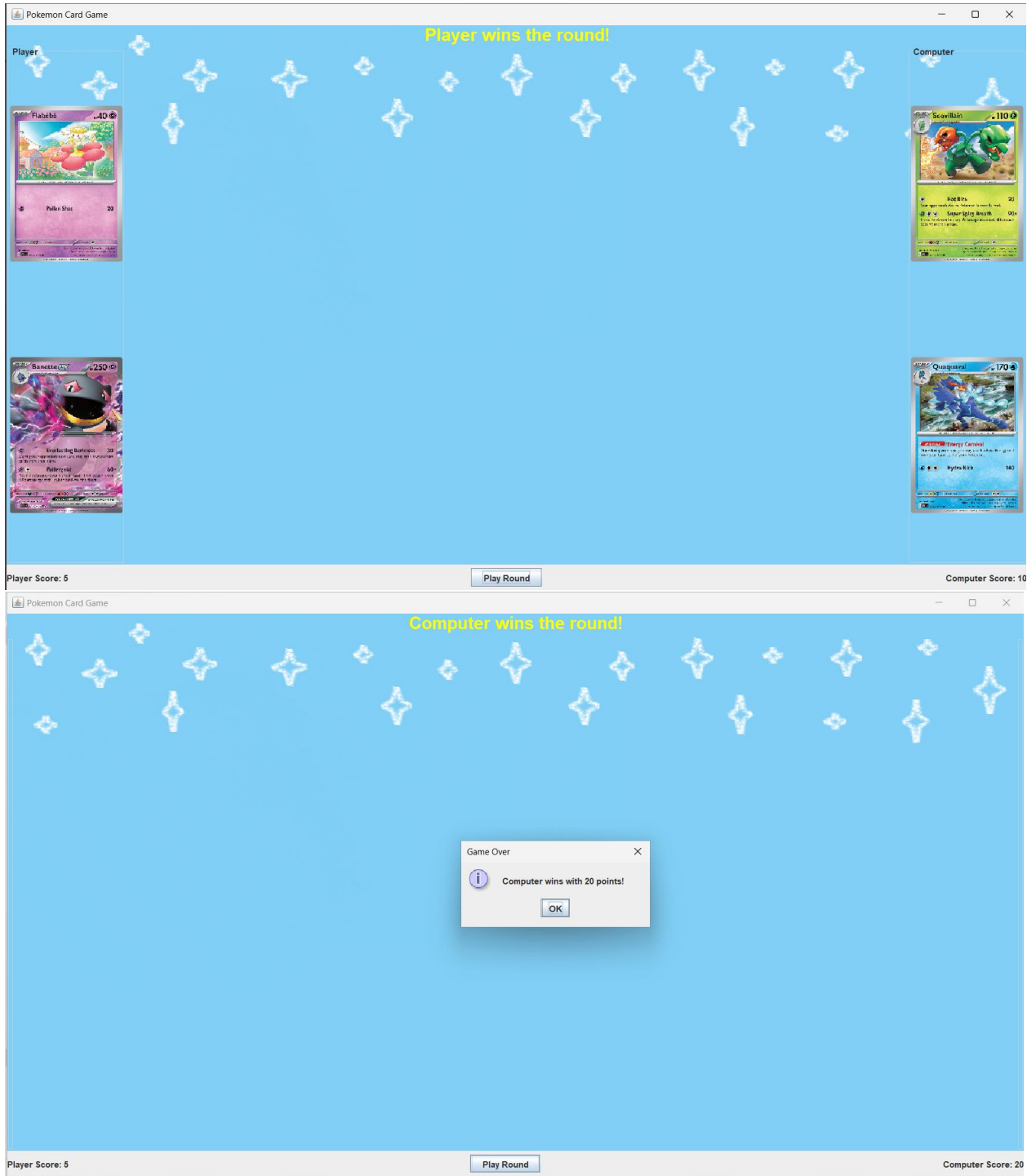
Ödev No:Mazeret	Tarih 04.07.2024	11/15
-----------------	------------------	-------

## 5. TEST VE DOĞRULAMA



Ödev No:Mazeret	Tarih 04.07.2024	12/15
-----------------	------------------	-------





Ödev No:Mazeret	Tarih 04.07.2024	14/15
-----------------	------------------	-------

---

## 6. KAYNAKÇA

- † <https://chat.openai.com>
- † <https://app.diagrams.net/>
- † <https://www.tasarimkodlama.com/java-programlama/java-swing-ile-gui-olusturma-ornekleri/>
- † <https://muhammedtalhacevik.medium.com/java-swing-ve-gui-programlama-kullan%C4%B1c%C4%B1-aray%C3%BCzleri-olu%C5%9Fturman%C4%B1n-temel-rehberi-by-muhammed-talha-7ec9fd9992cb>
- † <https://www.yusufsezer.com.tr/java-awt-swing-ve-javafx/>
- † <https://kodlamavakti.com/java/encapsulation/>
- † <https://www.canva.com/design/DAGAcdmB1v4/NUeL-2TUUU3wfhq9rygEpA/edit>

Ödev No:Mazeret	Tarih 04.07.2024	15/15
-----------------	------------------	-------