



**T.C**  
**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR/ MÜHENDİSLİĞİ**

**PROJE ADI: POKEMON KART OYUNU**

**ÖĞRENCİ ADI: Anıl Erdoğan-Akın Turan**

**ÖĞRENCİ NUMARASI: 220501006-220501013**

**DERS SORUMLUSU: DR. ÖĞR. ÜYESİ Ercan Ölçer**

**TARİH:4.07.2024**

# 1 GİRİŞ

## 1.1 Projenin amacı

Yapılan projenin temel amacı bir pokemon kart oyunu tasarlamak. Bunun için de python dili kullanılarak ve pygame kütüphanesi aracılığıyla bir uygulama tasarlandı ve istenen gereksinimler uygulamaya eklendi.

Projede gerçekleştirilmesi istenilen özellikler ise şu şekilde:

- Kullanıcı için sade ve anlaşılabilirliği yüksek bir arayüz
- Basit bir skor tablosu
- Oluşturulan tasarımların istenilen şekilde doğru ve kurallara uygun biçimde çalışması.

## 2 GEREKSİNİM ANALİZİ

### 2.1 Arayüz gereksinimleri

Projede istenilen arayüz gereksinimleri şu şekilde

- Uygulamadaki bileşenler için fonksiyonlar bölümünün oluşturulması ve uygulama için sade ve anlaşılır bir tarz kullanılması.
- Her bir fonksiyonun kendine has özelliği olması
- Bileşenlerin uygun konumlarda gözükmesi ve yerleşmesi
- Her işlem sonrası gerekli işlemlerin doğru bir biçimde gerçekleşmesi

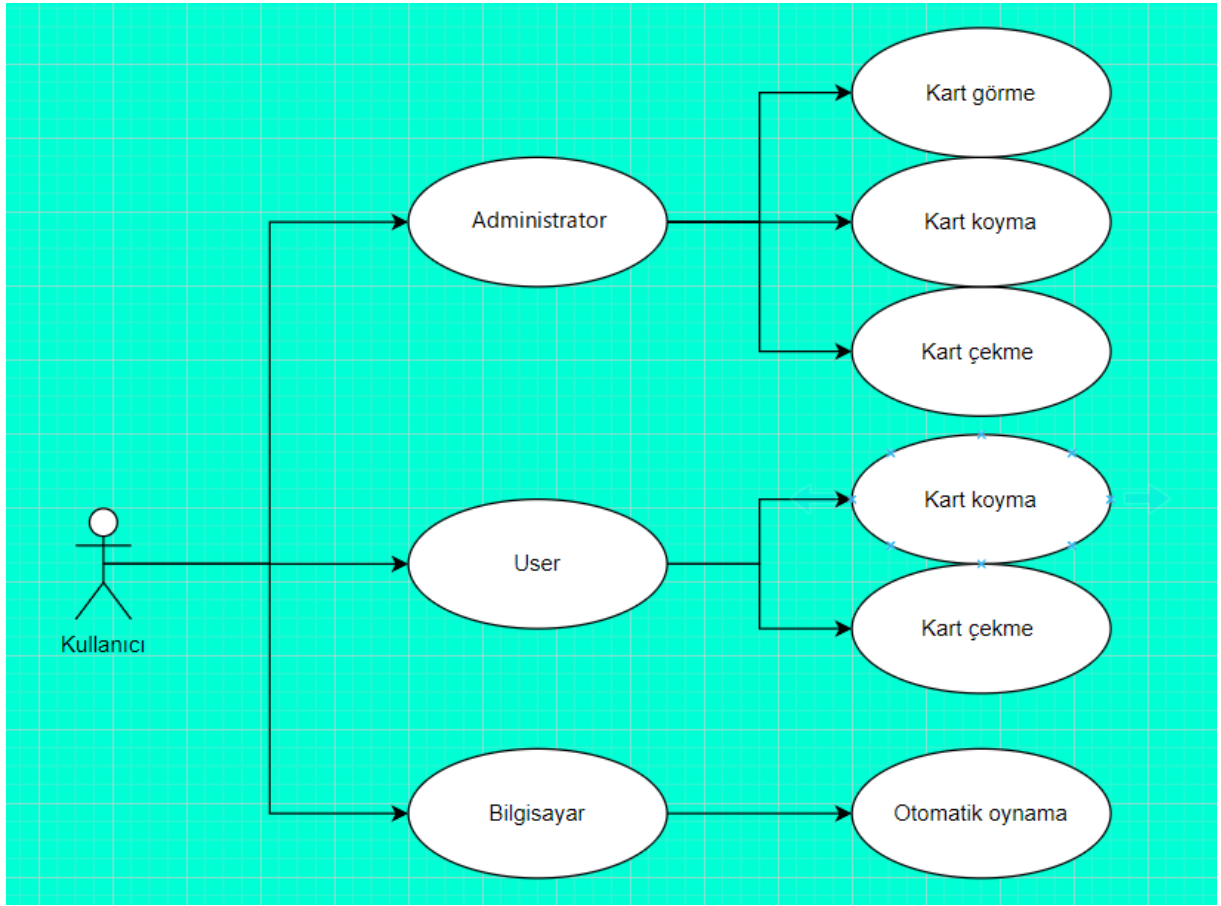
### 2.2 Fonksiyonel gereksinimler

Projede istenilen fonksiyonel gereksinimler şu şekilde:

- Kullanıcıların oyun başında eksik olmayan öğelere sahip olması ve oyunun düzgün başlatılması.
- Skor tablosunun doğru skorları göstermesi
- Kart bittikçe kartların ortak kart bölümünden otomatik seçilmesi.
- Administrator ve userlerin doğru yetkilere sahip olması.
- Oyundan çıkma, oyunu başlatma ve seçim yapabilme görevlerinin doğru çalışması.

### 2.3 Use-Case diyagramı

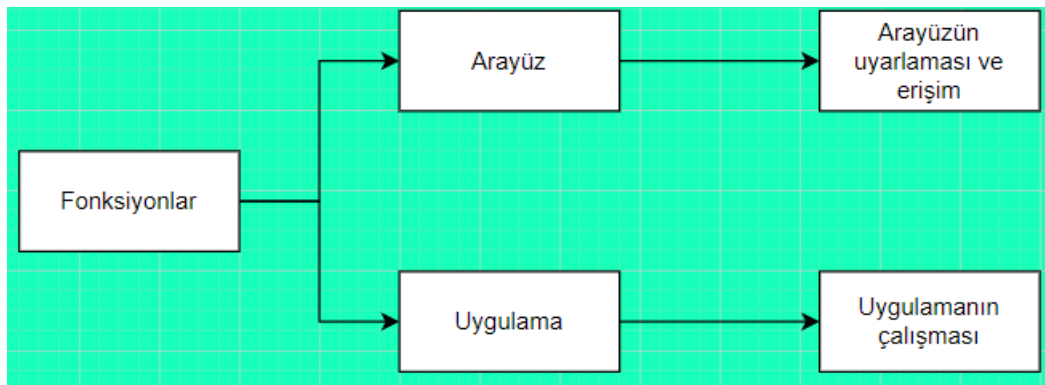
Uygulama için tasarlanan use-case diyagramı şu şekilde:



### 3 TASARIM

#### 3.1 Mimari tasarım

- İlk olarak arayüz ve arayüz ile ilgili fonksiyonlar ve modüller tasarlandı.
- İkinci olarak kodun ana mantığını çalıştıracak modüller ve bileşenler arayüzün üstüne entegre edildi.
- Son olarak ise uygulamanın tasarım kısmı tamamlandı. Kartlar doğru yerlere ve ek kartlar da pencerenin sağ orta bölümüne yerleştirildi.
- Modül Diyagramı:



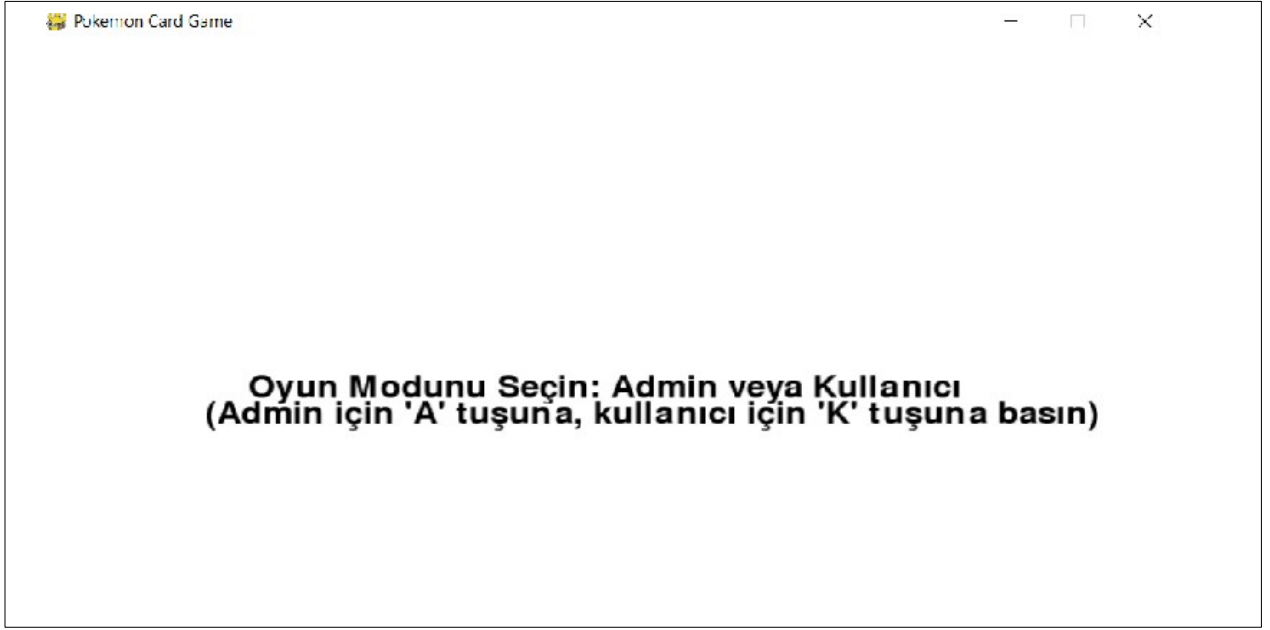
### 3.2 Kullanılacak teknolojiler

- Yapılan simülasyon için Python dili kullanıldı. Pygame kütüphanesi aracılığıyla uygulama tasarlandı ve random kütüphanesi ile desteklendi ve gösterilen program oluşturuldu. Ekstra bir teknoloji kullanılmadı.

### 3.3 Veri tabanı tasarımı

- Uygulamada herhangi bir veri tabanı tasarımı yapılmadı veya hazır olarak kullanılmadı

### 3.4 Kullanıcı arayüzü tasarımı



- Kullanıcı arayüzü tasarlanırken basit ve sade bir tasarım tercih edildi oluşturulan arka plan üzerine gerekli araçlar eklendi ve simetrik bir görüntü elde edildi.
- Ardından çeşitli fonksiyonlar tanımlandı ve arayüz tasarımına devam edildi.
- Son olarak kod test edildi.
- Uygulamanın çalışması ise oldukça basit kod çalıştırıldıktan sonra oyunu harekete geçirecek hamlelerin gerçekleşmesi.

## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

- **PokemonCard:** Bu sınıf her bir pokemon kartını name, damage gibi ara fonksiyonlarla temsil eder ve çeşitli metodlarla yönetir.
- **Player:** Bu sınıf her bir oyuncuyu name, el, skor gibi özelliklerle tanımlar ve çeşitli metodlarla yönetir.
- **AdminPlayer:** Bu sınıf player'ın yani oyuncunun yönetici olduğu durumda devreye girer ve bazı özellikleri devreye sokar. `view_opponent_cards` gibi bazı fonksiyonlarla destekler.
- **HumanPlayer:** Bu sınıf her bir insan oyuncuyu temsil eder.
- **ComputerPlayer:** Bu sınıf bilgisayar (yapay zeka) yı temsil eder.
- **ComputerVsComputerPlayer:** Bu sınıf 2 bilgisayarın arasındaki oyunu kontrol eder.
- **Game:** Oyunun resmen var olması için gerekli fonksiyonları yöneten bir sınıftır.
- `create_deck()`: Desteyi oluşturur
- `deal_cards()`: Kartları dağıtır
- `set_player_mode()`: Oyuncu modunu ayarlar
- `play_round()`: Puanları günceller ve turları idare eder
- `is_game_over()`: Oyunu sonlandırır
- `determine_winner()`: Kazananı belirler
- `view_opponent_cards()`: Rakip kartlar admin için görünür olur

### 4.2 Görev dağılımı

- Kodun ana kullanıcı ara yüzü tasarımını Akın yaparken Anıl ise arka planda çalışmasını ve test kodunu tasarladı. Ayrıca takıldığımız yerlerde birbirimize danıştık. Sosyal iletişim uygulamaları bu süreçte ödev dağılımını destekledi.
- Raporda ise görsel iletişim kurarak beraber yazdık

### 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Metodları eklemekte ve arayüzü tasarlamakta zorluk çektik.
- Bilgisayarın kendi kendine oynaması işleminin düzenli ve işlenebilir bir hale gelmesi.
- Bazı textlerin uygun anlarda uygun yerlere yerleştirilmesinde zorlanıldı.

## 4.4 Proje isterlerine göre eksik yönler

- Projede eksik bir ister bulunmamaktadır.

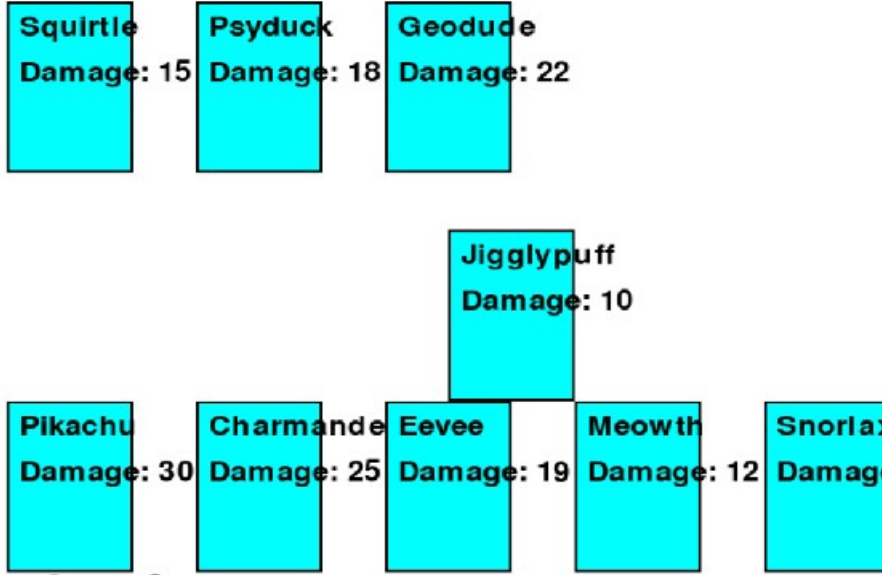
# 5 TEST VE DOĞRULAMA

## 5.1 Yazılımın test süreci

- Test bileşeni olarak arayüzün çalışması için gerekli konum adresleri test edilerek bulundu ve nesneler buraya yerleştirildi
- Kod başlarken ki oyunu başlatma ayarları belli test aşamalarından geçti.
- Test sürecinde bazı gözden kaçan ufak hatalarda tespit edilip düzeltildi.

## 5.2 Test fotoğrafları

Computer Score: 0



Player Score: 0

**Bilgisayar Puanı: 0**

Squirtle  
Hasar: 15

Bulbasaur  
Hasar: 20

Pikachu  
Hasar: 30

Snorlax  
Hasar: 28

Charmander  
Hasar: 25

Jigglypuff  
Hasar: 10

Psyduck  
Hasar: 18

Meowth  
Hasar: 12

**Kullanıcı Puanı: 10**

**Computer Score: 15**

Card

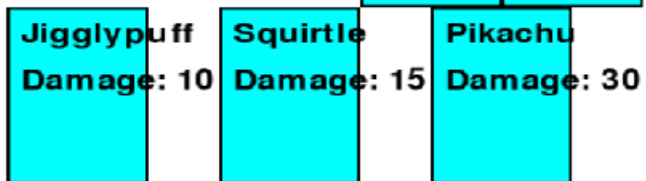
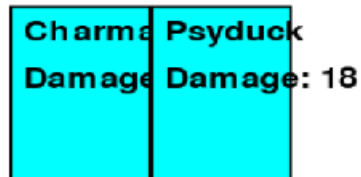
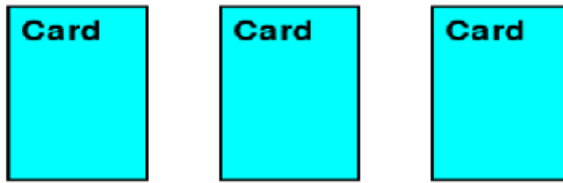
Card

Card

Jigglypuff Geodude  
Damage: 22

**Player Score: 10**

**Computer Score: 5**



**Player Score: 5**