

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU:**

**Pokemon**

**ÖĞRENCİ ADI: Müslüm Batın KILIÇ**

**ÖĞRENCİ NUMARASI:220501023**

**DERS SORUMLUSU:**

**PROF. DR./DR. ÖĞR. ÜYESİ Ercan ÖLÇER**

**TARİH:**

**29.06.2003**

1. **GİRİŞ**
   1. **Projenin amacı:**

Bu projenin amacı, bir oyuncunun bilgisayara karşı veya bilgisayarın bilgisayara karşı oynayabileceği basit bir kart oyunu geliştirmektir. Oyunda, her iki tarafın da ellerinde bulunan kartlarla savaşarak en yüksek skoru elde etmeye çalışması amaçlanmıştır. Bu proje, hem yazılım geliştirme becerilerini pekiştirmek hem de oyun programlamaya yönelik temel konseptleri anlamak amacıyla tasarlanmıştır.

* 1. **Projede Gerçekleştirilecekler:**

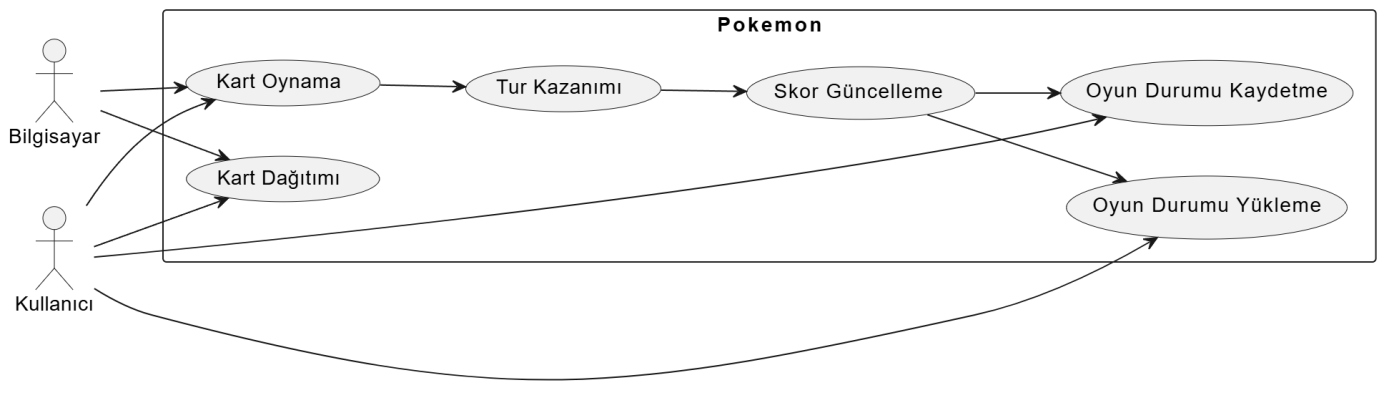
1. Pokemon Kartlarının Oluşturulması: Toplamda 10 adet Pokemon kartı oluşturulacaktır. Her bir kart, bir isim, hasar puanı ve resim dosyası içerecektir.
2. Kartların Dağıtılması: Her iki oyuncuya (kullanıcı ve bilgisayar) başlangıçta 3'er kart rastgele dağıtılacaktır.
3. Orta Kartların Belirlenmesi: Ortada 4 kapalı kart olacaktır.
4. Kart Oynama Mekanizması: Oyuncular sırayla ellerindeki kartlardan birini ortaya koyacaktır.
5. Tur Kazanımı ve Skor: Daha yüksek hasar puanına sahip kartı oynayan oyuncu turu kazanacak ve ortadaki kartlardan birini alacaktır. Kazanan oyuncu 5 puan kazanacaktır.
6. Oyunun Sonlandırılması: Oyun sonunda en yüksek skora sahip oyuncu kazanacaktır.
7. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

· **Kullanıcı Arayüzü**:

* Kullanıcının ve bilgisayarın elindeki kartlar gösterilmelidir.
* Ortadaki kapalı kartlar arayüzde gösterilmelidir.
* Oyuncunun oynayacağı kartı seçebilmesi için butonlar bulunmalıdır.
* Skorlar ekranda gösterilmelidir.
* Oyun durumu ve sonuçları kullanıcıya bildirilecek şekilde bir mesaj kutusu ile gösterilmelidir.

· **Donanım Gereksinimleri**:

* Kullanıcının oyunu oynayabilmesi için bir bilgisayar gereklidir.
* Grafiksel kullanıcı arayüzünü destekleyen bir işletim sistemi (Windows, macOS, Linux).
  1. **Fonksiyonel gereksinimler**
* **Kartların Rastgele Dağıtılması**: Oyunun başlangıcında kartlar rastgele dağıtılmalıdır.
* **Kart Oynama**: Oyuncu ve bilgisayar sırayla kart oynamalıdır.
* **Tur Sonunda Kazananın Belirlenmesi**: Tur sonunda kazanan oyuncuya ortadaki kartlardan biri verilmelidir.
* **Oyun Sonunun Belirlenmesi**: Oyun sonunda skorlar kontrol edilmeli ve kazanan belirlenmelidir.
* **Oyunun Kaydedilmesi ve Yüklenmesi**: Oyun durumu kaydedilebilmeli ve yeniden yüklenebilmelidir.
  1. **Use-Case diyagramı**



1. **TASARIM**
   1. **Mimari tasarım**

· **Pokemon Sınıfı**: Pokemon kartlarının isim, hasar puanı ve resim dosyası gibi özelliklerini tutar.

* **Özellikler**: isim, hasar\_puani, resim\_dosyasi
* **Metodlar**: isim\_al(), hasar\_puani\_al(), resim\_al()

· **Deste Sınıfı**: Kartları karıştırır ve dağıtır, kart çekme işlevini sağlar.

* **Özellikler**: kartlar (Pokemon kartlarından oluşan liste)
* **Metodlar**: kart\_cek()

· **Oyuncu Sınıfı**: Oyuncunun elindeki kartları yönetir, kart çekme ve kart oynama işlevlerini sağlar.

* **Özellikler**: isim, el (eldeki kartlar)
* **Metodlar**: kart\_cek(), kart\_oyna(), el\_al(), isim\_al()

· **Oyun Sınıfı**: Oyun akışını ve kurallarını yönetir, turları oynatır, skorları hesaplar ve oyunu bitirir.

* **Özellikler**: deste, kullanici, bilgisayar, ortadaki\_kartlar, kullanici\_skor, bilgisayar\_skor, oyuncu\_turu
* **Metodlar**: tur\_oyna(), oyun\_bitti\_mi(), skorlar\_al(), oyun\_durumunu\_kaydet(), oyun\_durumunu\_yukle()

· **OyunArayuzu Sınıfı**: Tkinter kullanarak arayüzü oluşturur ve yönetir, kullanıcı etkileşimlerini işler.

* **Özellikler**: root, oyun, kullanici\_etiketi, kullanici\_kartlari\_frame, bilgisayar\_etiketi, bilgisayar\_kartlari\_frame, ortadaki\_kartlar\_etiketi, ortadaki\_kartlar\_frame, skor\_etiketi, skor\_frame, kaydet\_butonu, yukle\_butonu
* **Metodlar**: kartlari\_guncelle(), kapali\_kart\_goster(), karti\_goster(), skorları\_guncelle(), kart\_oyna(), oyunu\_kaydet(), oyunu\_yukle()
  1. **Kullanıcı arayüzü tasarımı**
* **Kartların Gösterimi**: Kullanıcı ve bilgisayar kartları arayüzde gösterilir. Kartlar resim dosyaları ile temsil edilir.
* **Kapalı Kartlar**: Ortada kapalı kartlar gösterilir.
* **Kart Seçim Butonları**: Kullanıcı, elindeki kartları butonlar aracılığıyla seçer.
* **Skor Gösterimi**: Kullanıcı ve bilgisayar skorları ekranda gösterilir.
* **Oyun Durumu ve Sonuçları**: Oyun durumu ve sonuçları mesaj kutuları ile gösterilir.

1. **UYGULAMA**
   1. **Kodlanan bileşenlerin açıklamaları**

**Pokemon Sınıfı**: Bu sınıf, her bir Pokemon kartının isim, hasar puanı ve resim dosyasını tutar. Bu sınıf, kartların temel özelliklerini saklamak ve bu özelliklere erişim sağlamak amacıyla kullanılır.

* + **Neden Kullanıldı**: Her Pokemon kartının farklı özellikleri vardır ve bu özelliklerin saklanması gereklidir.
  + **Özellikler ve Metodlar**:
    - **isim**: Pokemon kartının adı.
    - **hasar\_puani**: Pokemon kartının hasar puanı.
    - **resim\_dosyasi**: Pokemon kartının resim dosyası.
    - **isim\_al()**: Pokemon kartının adını döndürür.
    - **hasar\_puani\_al()**: Pokemon kartının hasar puanını döndürür.
    - **resim\_al()**: Pokemon kartının resim dosyasını döndürür.

**Deste Sınıfı**: Bu sınıf, oyunun başında kartları karıştırır ve rastgele dağıtır. Ayrıca oyun sırasında kart çekme işlemlerini gerçekleştirir. Kartların rastgele dağıtılması ve oyuncuların kart çekebilmesi için gereklidir.

* + **Neden Kullanıldı**: Kartların rastgele dağıtılması ve oyun sırasında kart çekme işlemlerinin yapılması için gereklidir.
  + **Özellikler ve Metodlar**:
    - **kartlar**: Pokemon kartlarından oluşan liste.
    - **kart\_cek()**: Destenin en üstündeki kartı çeker.

**Oyuncu Sınıfı**: Bu sınıf, her bir oyuncunun elindeki kartları yönetir. Oyuncuların kart çekme ve kart oynama işlevlerini sağlar. Oyuncuların ellerindeki kartları yönetmesi ve oyun sırasında kart oynayabilmesi için gereklidir.

* + **Neden Kullanıldı**: Oyuncuların ellerindeki kartları yönetmesi ve oyun sırasında kart oynayabilmesi için gereklidir.
  + **Özellikler ve Metodlar**:
    - **isim**: Oyuncunun adı.
    - **el**: Oyuncunun elindeki kartlar.
    - **kart\_cek()**: Desten kart çeker ve oyuncunun eline ekler.
    - **kart\_oyna()**: Oyuncunun elinden bir kart oynar ve kartı elden çıkarır.
    - **el\_al()**: Oyuncunun elindeki kartları döndürür.
    - **isim\_al()**: Oyuncunun adını döndürür.

**Oyun Sınıfı**: Bu sınıf, oyun akışını ve kurallarını yönetir. Turları oynatır, skorları hesaplar ve oyunu bitirir. Oyunun kurallarını ve akışını yönetmek için gereklidir. Ayrıca, oyuncuların kartlarıyla etkileşimde bulunmasını ve oyunun sona ermesini sağlar.

* + **Neden Kullanıldı**: Oyunun kurallarını ve akışını yönetmek için gereklidir. Ayrıca, oyuncuların kartlarıyla etkileşimde bulunmasını ve oyunun sona ermesini sağlar.
  + **Özellikler ve Metodlar**:
    - **deste**: Oyun için kullanılan deste.
    - **kullanici**: Oyundaki kullanıcı oyuncu.
    - **bilgisayar**: Oyundaki bilgisayar oyuncu.
    - **ortadaki\_kartlar**: Ortada duran kapalı kartlar.
    - **kullanici\_skor**: Kullanıcının skoru.
    - **bilgisayar\_skor**: Bilgisayarın skoru.
    - **oyuncu\_turu**: Hangi oyuncunun sırası olduğu.
    - **tur\_oyna()**: Bir turu oynar, kartları karşılaştırır ve kazananı belirler.
    - **oyun\_bitti\_mi()**: Oyunun bitip bitmediğini kontrol eder.
    - **skorlar\_al()**: Kullanıcı ve bilgisayar skorlarını döndürür.
    - **oyun\_durumunu\_kaydet()**: Oyun durumunu kaydeder.
    - **oyun\_durumunu\_yukle()**: Oyun durumunu yükler.

**OyunArayuzu Sınıfı**: Bu sınıf, Tkinter kullanarak arayüzü oluşturur ve yönetir. Kullanıcı etkileşimlerini işler. Kullanıcıya görsel bir arayüz sunmak ve etkileşimli bir oyun deneyimi sağlamak için gereklidir.

* **Neden Kullanıldı**: Kullanıcıya görsel bir arayüz sunmak ve etkileşimli bir oyun deneyimi sağlamak için gereklidir.
* **Özellikler ve Metodlar**:
  + - **root**: Tkinter ana penceresi.
    - **oyun**: Oyun nesnesi.
    - **kullanici\_etiketi**: Kullanıcı kartlarını gösteren etiket.
    - **kullanici\_kartlari\_frame**: Kullanıcı kartlarını içeren çerçeve.
    - **bilgisayar\_etiketi**: Bilgisayar kartlarını gösteren etiket.
    - **bilgisayar\_kartlari\_frame**: Bilgisayar kartlarını içeren çerçeve.
    - **ortadaki\_kartlar\_etiketi**: Ortadaki kartları gösteren etiket.
    - **ortadaki\_kartlar\_frame**: Ortadaki kartları içeren çerçeve.
    - **skor\_etiketi**: Skorları gösteren etiket.
    - **skor\_frame**: Skorları içeren çerçeve.
    - **kaydet\_butonu**: Oyunu kaydetmek için buton.
    - **yukle\_butonu**: Oyunu yüklemek için buton.
    - **kartlari\_guncelle()**: Kartları günceller ve arayüzde gösterir.
    - **kapali\_kart\_goster()**: Kapalı kartları arayüzde gösterir.
    - **karti\_goster()**: Bir kartı arayüzde gösterir.
    - **skorları\_guncelle()**: Skorları günceller ve arayüzde gösterir.
    - **kart\_oyna()**: Bir kartın oynanmasını sağlar.
    - **oyunu\_kaydet()**: Oyunu kaydeder.
    - **oyunu\_yukle()**: Oyunu yükler.
  1. **Karşılaşılan zorluklar ve çözüm yöntemleri**
     + **kaydet\_butonu**: Oyunu kaydetmek için buton.
     + **yukle\_butonu**: Oyunu yüklemek için buton.
     + **kartlari\_guncelle()**: Kartları günceller ve arayüzde gösterir.
     + **kapali\_kart\_goster()**: Kapalı kartları arayüzde gösterir.
     + **karti\_goster()**: Bir kartı arayüzde gösterir.
     + **skorları\_guncelle()**: Skorları günceller ve arayüzde gösterir.
     + **kart\_oyna()**: Bir kartın oynanmasını sağlar.
     + **oyunu\_kaydet()**: Oyunu kaydeder.
     + **oyunu\_yukle()**: Oyunu yükler.

## UYGULAMA

### Görev Dağılımı

* **Tasarım ve Geliştirme**:
  + [Öğrenci adı]: Tüm kodun tasarımı ve geliştirilmesi.
* **Raporun Hazırlanması**:
  + [Öğrenci adı]: Raporun hazırlanması.

### Karşılaşılan Zorluklar ve Çözüm Yöntemleri

**Tkinter ile Dinamik Kart Güncelleme**:

* + **Zorluk**: Kartların dinamik olarak güncellenmesi ve gösterilmesi zorluğu.
  + **Çözüm**: Tkinter widget'larının doğru şekilde yönetilmesi ve güncellenmesi sağlandı.
    - **Detaylı Çözüm**: Tkinter widget'larının güncellenmesi için mevcut widget'lar yok edilip yeniden oluşturuldu. Bu sayede kartlar her oynandığında veya güncellendiğinde doğru şekilde arayüzde gösterildi.
    - **Kod Parçası**: Kullanıcı ve bilgisayar kartlarını güncellemek için kartlari\_guncelle() metodu kullanıldı.

**Rastgele Kart Dağıtımı ve Oyun Sonlandırma**:

* + **Zorluk**: Rastgele kart dağıtımı ve oyunun doğru şekilde sonlandırılması.
  + **Çözüm**: Rastgele dağıtım algoritması ve oyun bitirme koşulları dikkatlice kontrol edilerek çözüldü.
    - **Detaylı Çözüm**: random.shuffle() fonksiyonu kullanılarak kartlar karıştırıldı ve kart\_cek() metodu ile rastgele kart çekildi. Oyun bitirme koşulları oyun\_bitti\_mi() metodunda kontrol edildi.
    - **Kod Parçası**: Deste sınıfında kartların karıştırılması ve çekilmesi işlemleri yapıldı. Oyun sınıfında oyun bitirme koşulları belirlendi.
  1. **Proje isterlerine göre eksik yönler**
* Görselleştirme daha iyi olabilirdi.
* Bilgisayara karşı bilgisayar oynamıyor.

1. **TEST VE DOĞRULAMA**
   1. **Yazılımın test süreci**

* **Manuel Testler**: Yazılım için manuel testler uygulanmıştır.
* **Kart Dağıtımı**: Kart dağıtımının rastgele olup olmadığı kontrol edilmiştir.
* **Kart Oynama**: Kart oynama işlemlerinin doğru çalışıp çalışmadığı kontrol edilmiştir.
* **Skor Hesaplama**: Skor hesaplama işlemlerinin doğruluğu kontrol edilmiştir.
  1. **Yazılımın doğrulanması**
* **Doğrulama Sonuçları**: Yazılımın test edilmesi sonucunda tüm bileşenlerin doğru çalıştığı doğrulanmıştır.
* **Hatalı Çalışan Bileşenler**: Hatalı veya eksik çalışan bir bileşen bulunmamaktadır.
* **Doğru Çalışan Bileşenler**: Tüm bileşenler (kart dağıtımı, kart oynama, skor hesaplama) doğru çalışmaktadır.