

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU:**

**POKEMON KART OYUNU**

**DOĞUŞ HAN ALAN 220502014**

**EMRE ÇOLAK 220501011**

**DERS SORUMLUSU:**

**PROF. DR./DR. ÖĞR. ÜYESİ**

**ERCAN ÖLÇER**

**TARİH:04/07/2024**

1. **GİRİŞ**
   1. **Projenin amacı**

Bu projenin amacı, kullanıcıların ve bilgisayarın birbirleriyle veya bilgisayarın diğer bilgisayarlarla savaşabileceği basit bir kart oyunu geliştirmektir. Bu oyun, kullanıcıya keyifli bir deneyim sunmayı ve aynı zamanda nesneye yönelik programlama (OOP) prensiplerini uygulayarak yazılım geliştirme becerilerini pekiştirmeyi hedefler.

#### Projede Gerçekleştirilmesi Beklenenler

* Kullanıcı ve bilgisayarın rastgele dağıtılan 3'er Pokémon kartı ile oyuna başlaması.
* Ortada 4 Pokémon kartının kalması.
* Kullanıcı ve bilgisayarın kart seçerek kapalı bir şekilde ortaya koyması ve kartların açılması ile en yüksek hasar puanına sahip kartın kazananın belirlenmesi.
* Her el sonunda yeni kartların desteden çekilmesi ve oyun sonunda en yüksek skora sahip oyuncunun belirlenmesi.
* Kullanıcı arayüzü tasarımı ile kartların ve oyunun durumunun takip edilebilmesi.
* Oyun modları: Kullanıcı vs. Bilgisayar ve Bilgisayar vs. Bilgisayar.

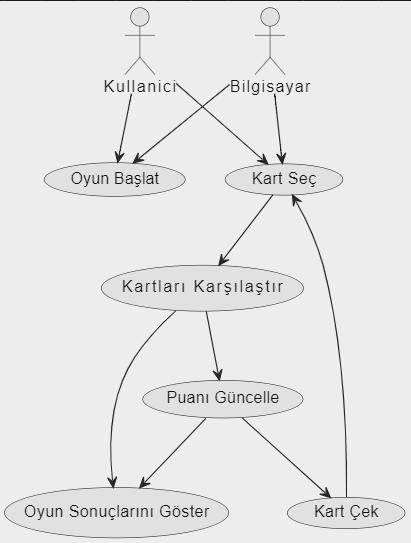
1. **GEREKSİNİM ANALİZİ**
   1. **Arayüz gereksinimleri**

* **Kullanıcı Arayüzü Gereksinimleri:**
  + Kartların ve oyuncuların puanlarının görüntülenmesi.
  + Kullanıcının kart seçimini yapabileceği bir alan.
  + Oyunun durumunu ve sonuçlarını gösteren bir alan.
  + Bilgisayarın seçtiği kartların kullanıcı tarafından görülmemesi ancak kontrol amacıyla sunum sırasında görüntülenebilmesi.
* **Donanım Arayüzü Gereksinimleri:**
  + Ekstra donanım gereksinimi yoktur; standart bir bilgisayar yeterlidir.

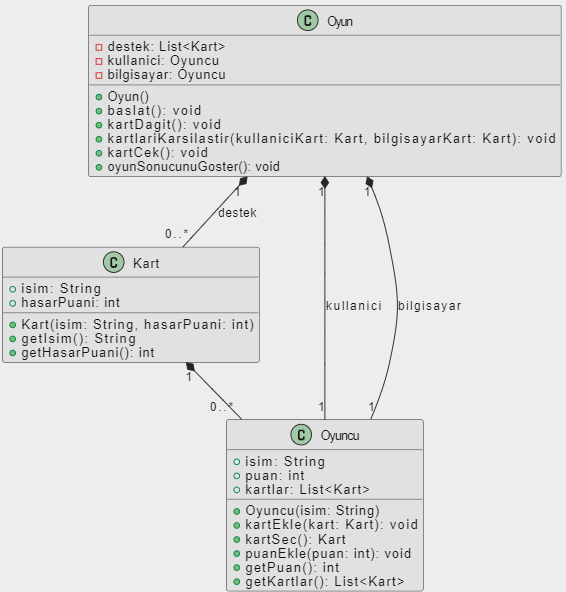
#### 2.2 Fonksiyonel Gereksinimler

* Kullanıcı ve bilgisayarın rastgele kart seçimi yapabilmesi.
* Seçilen kartların kapalı olarak ortaya koyulması ve aynı anda açılması.
* Kartların hasar puanlarının karşılaştırılması ve kazananın belirlenmesi.
* Yeni kartların desteden çekilmesi.
* Oyun sonunda skora göre kazananın belirlenmesi.

**2.3 Use-Case diyagramı**



**2.4 UML Sınıf Diyagramı**

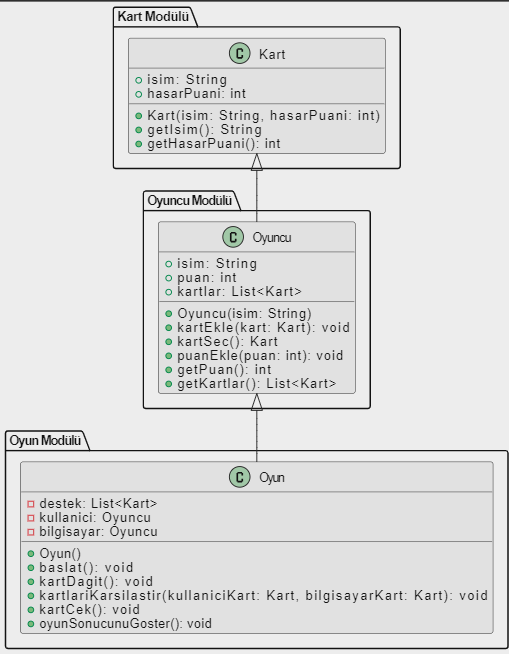


**3.TASARIM**

**3.1Mimari tasarım**

* **Kart Sınıfı:** Pokémon kartlarının özelliklerini (isim, hasar puanı) saklar.
* **Oyuncu Sınıfı:** Kullanıcı ve bilgisayar oyuncuları temsil eder. Kartları seçer ve puanları takip eder.
* **Oyun Sınıfı:** Oyunun ana mantığını içerir. Kart dağıtımı, kart seçimi ve sonuçların hesaplanması gibi işlemleri yönetir.
* **Arayüz Sınıfı:** Kullanıcı arayüzünü yönetir ve kullanıcı etkileşimlerini işler.

**3.2 Modül Diyagramı**



**3.3 Kullanılacak teknolojiler**

* **Yazılım Dili:** Python
* **Kütüphaneler:** Tkinter (arayüz için), random (rastgele seçimler için)
* **Diğer Teknolojiler:** Proje için herhangi bir veri tabanı gereksinimi bulunmamaktadır.

**3.4 Kullanıcı arayüzü tasarımı**

* Kullanıcı arayüzü, Tkinter kütüphanesi kullanılarak geliştirilecektir.
* Ekran çıktıları, kullanıcının kartlarını seçmesini, puanları görüntülemesini ve oyun durumunu takip etmesini sağlayacaktır.

**4.UYGULAMA**

#### 4.1 Kodlanan Bileşenlerin Açıklamaları

* **Kart Sınıfı:** Pokémon kartlarının oluşturulmasını sağlar.
* **Oyuncu Sınıfı:** Oyuncuların kartlarını ve puanlarını yönetir.
* **Oyun Sınıfı:** Oyun akışını ve mantığını yönetir.
* **Arayüz Sınıfı:** Kullanıcı etkileşimlerini ve görsel bileşenleri yönetir.

**4.2 Görev dağılımı**

**Kodun %60 lık kısmı ile Emre Çolak**

**Raporun %70 lik kısmı ile Doğuş Han Alan**

**İlgilenmiştir.**

### 5. KARŞILAŞILAN ZORLUKLAR VE ÇÖZÜM YÖNTEMLERİ

* **Zorluk:** Kullanıcı arayüzü tasarımında kartların doğru şekilde gösterilmesi ve güncellenmesi.
* **Çözüm:** Tkinter kütüphanesinin sağladığı widget'ların ve güncelleme mekanizmalarının kullanılması.

**6. Proje isterlerine göre eksik yönler**

* Projenin arayüz kısmı tam anlamıyla istendiği olmadığından bulunmamaktadır.

**7. TEST VE DOĞRULAMA**

#### 7.1 Yazılımın Test Süreci

* Test uygulaması, her bileşeni test edecek şekilde tasarlanacaktır.
* Testler, kart dağıtımı, kart seçimi ve sonuç hesaplamalarını kontrol edecektir.

**7.2 Yazılımın Doğrulanması**

* Testler sonucunda yazılımın doğru çalıştığı doğrulanacak ve eksik ya da hatalı çalışan bileşenler düzeltilecektir.

8. GİTHUB LİNK

[**https://github.com/Emre-Clk**](https://github.com/Emre-Clk)

[**https://github.com/dogussalann**](https://github.com/dogussalann)