A logo with a flower design

Description automatically generated

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU: PROGRAMLAMA LAB 2 MAZERET PROJESİ**

**ÖĞRENCİ ADI: UTKU EMRE ERMİŞ**

**ÖĞRENCİ NUMARASI: 220501034**

**DERS SORUMLUSU:**

**DR. ÖĞR. ÜYESİ ERAY DURSUN**

**TARİH: 04/07/2024**

1. **GİRİŞ**

**1.1 Projenin amacı:**

**Bir oyuncunun (kendiniz olacak) otomatik oyuncuyla (bilgisayar) savaşabileceği basit bir kart oyunu**

**yaratacaksınız. Aynı zamanda bilgisayar, bilgisayar ile de oynayabilecek. Tasarlayacağınız oyunda,**

**toplamda 10 pokemon kartı olacaktır ve her bir kullanıcıya ilk başta random olarak 3 er pokemon kartı**

**dağıtılacaktır. Dağıtımdan sonra ortada 4 tane pokemon kartı kalacaktır. Kullanıcı ve bilgisayar kendilerine**

**dağıtılan 3 pokemon karttan birini seçerek ortaya koyacaktır. İki taraf kartları ortaya kapalı bir şekilde**

**koyacak ve kartlar aynı anda çevrilerek yüksek hasar puanına sahip olan pokemon kartına sahip olan kişi ya**

**da bilgisayar 5 puan kazanacaktır. Daha sonra kullanıcı ve bilgisayar ortada kalan kartlardan birer tane**

**(kartların ne olduğunu bilmeden) alacaklardır. Ortadaki ve eldeki kartlar bitene kadar oyun devam edecektir.**

**En yüksek puana sahip oyuncu, oyunu kazanacaktır. Bu oyunu nesneye yönelik programlama yöntemini**

**kullanarak yapmanız beklenmektedir.**

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

**Oyunu ve oyuncuların ellerinde bulunan kartlar görünebilecek ve takip edilebilecek bir arayüz tasarlamanız**

**beklenmektedir.**

**• Destede 10 adet Pokemon kartı bulunmalıdır.**

**• Oyuncu desteden üç rastgele kart alır.**

**• Bilgisayar desteden üç rastgele kart alır.**

**• Oyuncu ve bilgisayar elinde bulunan üç karttan birini seçerek kapalı bir şekilde ortaya koyar. Bilgisayar seçimi**

**random olarak gerçekleştirir. Burada kullanıcı bilgisayarın hangi kartı şeçtiğini bilmeyecek (Fakat sunum**

**sırasında, oyunun doğru çalıştığının kontrolü yapılabilmesi için kartlar gösterilmeli).**

**• Daha sonra kartlar çevrilir.**

**• Oyuna sürülen kartlar bir daha kullanılamayacak. Her hamleden sonra desteden kullanıcı ve bilgisayar**

**tarafından birer adet kâğıt alınır ve yukarıdaki işlemler ortada ve elde, pokemon kart kalmayıncaya kadar devam eder.**

**• Skoru yüksek olan oyuncu, oyunu kazanır.**

**• Oyunu kullanıcı bilgisayara karşı oynayabildiği gibi bilgisayar bilgisayara karşı da oynayabilmelidir.**

**Her sınıf için ortak olan özellikler:**

**• Projede Encapsulation, Inheritance, Polymorphism, Abstraction yapılarının (hepsinin) kullanılması gerekmektedir.**

**• Yapıcı (constuctor) metotları (parametreli ve parametresiz olarak en az iki) yazılacak.**

**• Tüm özellikler için get, set metotları tanımlanacak.**

**• Kullanılan yapıların her biri nerede kullanıldı, neden kullanıldı sunumda açıklanacak.**

* 1. **UML diyagramı:**

**//www.plantuml.com/plantuml/png/ZLNRZjCm47tlL-Gn1Of47rS2hK12Me402PPNoqb6MIk-LFOOA9T\_nx77tJfkkrrfa-QSuvEvZEDRXzoYLtBLI-vSyo4yDNzMJVZRC2QSKCrDuz1cv84xRWCaD4Re-GWOPUsQX1bDqWWVn2wBvqkKhpEHzV4Qbn2Lf3jhTc1ndp0IbI1ZmO-nGo3\_LfJJ5pE1CdfEJOJSI2oqGCOU8tdH7B\_wiFZptklUNoe9o04cmQPWBsDjf72OoPtd-hfobJQHJqOaY5RNbhFsAn-JTrPdy\_doLEbBdoMR74fibl8O3eD9GZi\_tyH1MrUMo7Muoq7OIXQzeqCLv-dEK\_CUMpU0Gv2mR9697NuLqDJh4jzMk3B3idVPkUvQTeyC-IouRSMvOqlwKRSiEzCzMnIAv4\_tXOACgnwItsbukuHf40GR-2YmNH-BOqkPWxuD\_qAbbZMfgKdN2JaAn\_VSicoa1HkDmwgOwAqF6McU4qxkGW\_mw\_Kdes6Mezz9U0o1TG9HC2MYLsUCp47\_vfhhqmDpVvT7nXg3i-dqyjkKaDEpaYUbHWbAmALaqslUXWY1w-\_SfXLSSwP0UzQ3xU47E95qxorNa2IAQm-ITHwD9XEFuJV5W4Pn55FD5\_ai\_J\_o3cHyIIc3O0LufAaIksHKAizuKMKkP1ndYGqGZ2RGnDzfJ5KZYvJmNE9Yjy\_paoXV5yC5j5yQVqKhz4ZUfA7OZtCyss7gR\_7ITm\_GJrtTXlWLliByhAuyskRnfUcbRVbJ32EWX12MaTwXjoqnUTBsdWBE1CrNsatSIOwPoRDraWkGPtEiahVA37GurUjeIo-fmsOlmwZnW8t61XyrUusXZ20bjF40ByxzpURlc\_JXYOV2iys6Dcke7vN\_BBEDpFtTkLN4FRsEcDKjwC4hkVeF**

1. **TASARIM**
   1. **Kullanılacak Teknolojiler:**

**• Python kullanılarak yapılacak.**

**• Arayüz tasarımı için “tkinter” kütüphanesi kullanılacak.**

* 1. **Kullanıcı arayüzü tasarımı:**

**A screenshot of a computer

Description automatically generated**

**• Oynanışın manuel mi (oyuncu vs bilgisayar) yoksa auto mu (bilgisayar vs bilgisayar) olduğu seçilir.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**•** **Kartlar rastgele dağıtılır.**

**• Alttaki seçeneklerden elindeki kartlardan birini kullanırsın. Kullandığın kartı sonraki tur kullanamazsın.**

**• İstediğin zaman “Bilgisayarın Kartlarını Göster” kısmından bilgisayarın kartlarını görebilirsin.**

**A screenshot of a computer

Description automatically generated**

**• En yüksek skoru alan kazanır.**

**4 UYGULAMA**

**4.1 Kodlanan bileşenlerin açıklamaları:**

**“Kart” Sınıfı**

**A screenshot of a computer program

Description automatically generated**

Yapıcı Metot (Constructor):

Sınıfın yapıcı metodu, yeni bir “Kart” nesnesi oluştururken çalıştırılır ve kartın adını ve hasar değerini ayarlar.

“isim”: Kartın adı.

“hasar”: Kartın hasar değeri.

Getter ve Setter Metotları:

Bu metotlar, özel (private) olarak tanımlanan “\_\_isim” ve ” \_\_hasar” özelliklerine erişim sağlar ve bu özelliklerin güvenli bir şekilde değiştirilmesini sağlar.

“get\_isim” ve “set\_isim”:

Bu metotlar, isim özelliğine erişim sağlar ve bu özelliğin değerini değiştirir.

“get\_hasar” ve “set\_hasar”:

Bu metotlar, “hasar” özelliğine erişim sağlar ve bu özelliğin değerini değiştirir.

Özellikler (Properties):

Python'da özellikler (properties), sınıf değişkenlerinin kontrollü bir şekilde okunmasını ve yazılmasını sağlar. “property” fonksiyonu, getter ve setter metotlarını kullanarak özellikler tanımlar.

“isim”: “get\_isim” ve “set\_isim” metotlarını kullanarak “\_\_isim” özelliğine erişim sağlar.

“hasar”: “get\_hasar” ve “set\_hasar” metotlarını kullanarak “\_\_hasar” özelliğine erişim sağlar.

“\_\_str\_\_” Metodu

“\_\_str\_\_” metodu, “print” fonksiyonu kullanıldığında veya kartın bir metin olarak temsil edilmesi gerektiğinde çağrılır. Bu metodun amacı, kartın adını ve hasarını içeren okunabilir bir metin döndürmektir.

“PokemonKarti” Sınıfı:

A screen shot of a computer code

Description automatically generated

“PokemonKarti” sınıfı, pokemon kart oyunları için özel olarak tasarlanmış kartların temel özelliklerini ve işlevlerini tanımlar. Her pokemon kartının adı (isim) ve hasar değeri vardır, bu özellikler “Kart” sınıfından miras alınmıştır.

Miras Alınan Sınıf

“PokemonKarti” sınıfı, “Kart” sınıfından miras alır ve bu sayede “Kart” sınıfının tüm özelliklerine ve işlevlerine erişebilir.

“isim” ve “hasar” özellikleri, “Kart” sınıfından miras alınan yapıcı metot aracılığıyla atanır.

Özel Metotlar

“PokemonKarti” sınıfı, “Kart” sınıfından miras alınan “\_\_init\_\_” metodu dışında özel bir metot içermez. Ancak, “Kart” sınıfının “\_\_str\_\_” metodu otomatik olarak miras alınır ve pokemon kartının ismi ve hasar değerini içeren bir metin olarak döndürülmesini sağlar.

“oyuncu” Sınıfı:

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

Github hesabı: https://github.com/UtkuEmreErmis