

KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA LAB. I, 2019-2020 Güz Dönemi
PROJE 2

PROJE TESLİM TARİHİ: 29.11.2019

POKEMON KART OYUNU



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.

Amaç: Proje gerekleřtirmesi ile beraber ğrencilerin nesneye ynelik programlama yapısını anlamasını ve czm saėlayabilmesini amalanmaktadır.

Programlama Dili: Proje C++ veya Java dili kullanılarak gerekleřtirilecektir. Grafik tasarım iin C++ 'da Graphic.h ya da Allegro.h kullanılabilir. Java iin ise GUI ortamından yararlanabilirsiniz.

1. İřterler

Oyunu ve oyuncuların ellerinde bulunan kartlar grnebilecek ve takip edilebilecek bir arayz tasarlanamaz beklenmektedir.

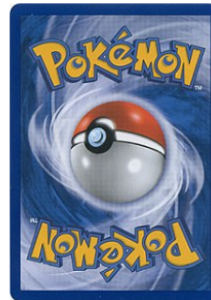
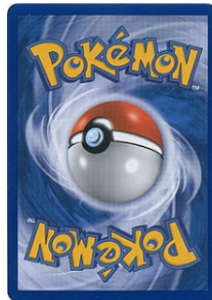
- Destede 10 adet Pokemon kartı bulunmalıdır.



- Oyuncu desteden  rastgele kart alır.
- Bilgisayar desteden  rastgele kart alır.
- Oyuncu ve bilgisayar elinde bulunan  karttan birini seerek kapalı bir řekilde ortaya koyar. Bilgisayar seimi random olarak gerekleřtirir. Burada kullanıcı bilgisayarın hangi kartı setiėini bilmeyecek (Fakat sunum sırasında, oyunun doėru alıřtıėının kontrol yapılabilmesi iin kartlar gsterilmeli).

Oyuncu 1

Bilgisayar



- Daha sonra kartlar evrilir. (Grseller rnek olarak verilmiřtir.)

Oyuncu 1

Bilgisayar



- Yukarıda bulunan şekilde gösterildiği üzere, hasar puanı daha büyük olan Charmander kazanacaktır.

Oyuncu 1

Bilgisayar

Skor : 0

Skor : 5



- Oyuna sürülen kartlar bir daha kullanılamayacak. Her hamleden sonra desteden kullanıcı ve bilgisayar tarafından birer adet kağıt 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.**

İstenen Sınıflar

Pokemon Sınıfı

Bir Pokemon sınıfı oluşturun. Sınıf tanımı şunları içermelidir:

- Yapıcı (constuctor) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Parametreler pokemonAdi ve pokemonTip olmalı.
- Pokemonun hasar puanını göstermek için hasarPuanıGoster() metodu yazılmalıdır.

Pokemon Tülerine Ait Sınıflar (10 Adet):

Tablo 1’de verilen pokemonların herbiri birer sınıf olarak tanımlanacaktır. Sınıflar şu tanımlamaları içermeli:

- **Pokemon sınıfından kalıtım alacaktır.**
- Yapıcı (constuctor) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Pokemon sınıfında bulunan pokemonAdi ve pokemonTip özelliklerine atama yapmak için super() kullanılacaktır.
- hasarPuani özelliği bu sınıflarda tutulacaktır ve Pokemon sınıfında bulunan hasarPuaniGoster() metodu override edilerek her bir pokemon için özelleştirilecektir.
- Boolean veri tipinde kartKullanildiMi bilgisi tutulmalıdır. **Kullanılan kartların oyunda bir daha kullanılması engellemek için bu veri tipinden yararlanılacaktır.**

Tablo 1. Pokemon hasar puanları

Pokemon Kart Adı	Hasar Puanı	Tip
Pikachu	40	Elektrik
Bulbasaur	50	Çim
Charmander	60	Ateş
Squirtle	30	Su
Zubat	50	Hava
Psyduck	20	Su
Snorlax	30	Normal
Butterfree	10	Hava
Jigglypuff	70	Ses
Meowth	40	Normal

Oyuncu Sınıfı:

Bilgisayar ya da kullanıcı olmak üzere oyunu oynayan iki oyuncu olacaktır. Bu iki oyuncunun farklı ve aynı özellikleri bulunmaktadır. Aynı özellikleri temsil etmek için Oyuncu temel sınıfı oluşturulacaktır. Oyuncu temel sınıfı **Abstract** sınıf olacaktır. Burada **Abstract** olması gereken fonksiyonu bulmanız beklenmektedir.

Bu sınıfta bulunması gereken özellikler ve fonksiyonlar:

- oyuncuID, oyuncuAdi ve Skor özellikleri olmalı.
- Yapıcı (constuctor) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Parametreler oyuncuID, oyuncuAdi ve Skor olmalı.
- kartListesi özelliği ile oyuncuların elinde bulunan kartlar listede tutulacaktır.
- SkorGoster() fonksiyonu ile oyuncuların skorları gösterilecektir.
- kartSec() fonksiyonu yazılmalı, fakat bu fonksiyonun bilgisayar ve kullanıcı için farklı durumlarda çalışacağı unutulmamalıdır.

Bilgisayar Sınıfı

- **Oyuncu sınıfından kalıtım alacaktır.**
- Yapıcı (constuctor) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Oyuncu sınıfında bulunan oyuncuID, oyuncuAdi ve Skor özelliklerine atama yapmak için super() kullanılacaktır.
- Oyuncu sınıfında bulunan kartSec() metodu override edilecektir. Bilgisayar random olarak aldığı 3 adet kart arasından yine random kart seçerek ortaya koyacaktır.

Kullanıcı Sınıfı

- **Oyuncu sınıfından kalıtım alacaktır.**
- Yapıcı (constuctor) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak. Oyuncu sınıfında bulunan oyuncuID, oyuncuAdi ve Skor özelliklerine atama yapmak için super() kullanılacaktır.
- Oyuncu sınıfında bulunan kartSec() metodu override edilecektir. Kullanıcı random olarak aldığı kartlar arasından kendi istediği kartı seçerek ortaya koyacaktır.

Tablo 2. Çalışmada bulunması gereken sınıflar

Sınıf	Özellikler	Kalıtım Aldığı Sınıf	Fonksiyonlar
Pokemon	pokemonID, pokemonAdi, pokemonTip	Yok	Kurucu fonksiyon (constuctor) Get, Set metotları
Pokemonlar#10 adet Herbir pokemon için ayrı ayrı 10 adet sınıf tanımlanacak	hasarPuanı	Pokemon	Kurucu fonksiyon (constuctor) Get, Set metotları
Oyuncu	oyuncuID, oyuncuAdi, Skor	Yok	Kurucu fonksiyon (constuctor) Get, Set metotları KartSecim
BilgisayarOyuncusu	kartListesi	Oyuncu	Kurucu fonksiyon (constuctor) Get, Set metotları, KartSecim (override)
InsanOyuncusu	kartListesi	Oyuncu	Kurucu fonksiyon (constuctor) Get, Set metotları, KartSecim (override)

Her sınıf için ortak olan özellikler:

- **Projede Encapsulation, Inheritance, Polymorphism, Abstraction yapılarının (hepsinin) kullanılması gerekmektedir.**
- **Yapıcı (constuctor) metotları (parametrelili ve parametresiz olarak en az iki) yazılacak.**
- **Tüm özellikler için get, set metotları tanımlanacak.**

2. Ödev Teslimi

- Proje sunum gününde rapor (hard copy) teslim edilmesi gerekmektedir.
- Rapor IEEE formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır. Raporda UML SINIF DIAGRAMI nın çizilmesi beklenmektedir.
- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki forum üzerinden Arş. Gör. Fulya Akdeniz veya Arş. Gör. Seda Kul'a sorulabilir.
- Demo tarihleri daha sonra duyurulacaktır.

- Demo sırasında algoritma, geliřtirdiđiniz kodun eřitli kısımlarının ne amaçla yazıldıđı ve geliřtirme ortamı hakkında sorular sorulabilir.
- Kullandıđınız herhangi bir satır kodu açıklamanız istenebilir.
-

İNTİHAL: İNTERNETTEN ALINAN KOD PARACIKLARI MUTLAKA KOD İERİSİNDE BELİRTİLECEK VE AIKLAMA SATIRI İLE KAYNAK GÖSTERİLECEKTİR. AKSİ DURUMDA KOPYA OLARAK DEĞERLENDİRİLECEKTİR. KOPYA ÇEKTİĐİ YA DA KOPYA VERDİĐİ TESPİT EDİLEN ÖĐRENCİLER SUNUMA ALINMAYACAKTIR.