

# POKEMON KART OYUNU

Taha İsmet Başar  
Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi  
basartaha@gmail.com

Şevki Karagöl  
Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi  
sevkikaragol@outlook.com

**Özet-** Bu projede bir oyuncunun (kendimizin) otomatik oyuncuyla (bilgisayar) aynı zamanda otomatik oyuncunun, otomatik oyuncuyla savaşılabileceği bir kart oyunu tasarladık. Tasarladığımız oyunda toplamda 10 pokemon kartı bulunuyor. Her bir kullanıcıya 3'er pokemon kartı dağıtılıyor ve ortada 4 tane pokemon kartı kalmış oluyor. Bu işlemlerin hepsi random olarak yapılıyor. Kullanıcı ve otomatik oyuncu kendilerine dağıtılan 3 pokemon kartından birini seçerek ortaya koyuyor. İki taraf kartları ortaya açık bir şekilde koyuyor. (Normalde kapalı olması gerekiyor fakat kontrolün sağlanması ve ödevin değerlendirilebilmesi için kartlar açık olarak tasarlanmıştır.) Ortaya konulan kartların hasar puanı kontrol ediliyor ve daha yüksek hasar puanına sahip olan kart, kartı ortaya atan kullanıcıya 5 puan kazandırıyor. Daha sonra kullanıcı ve bilgisayar ortada kalan kartlardan birer tane alıyor. Ortadaki ve kullanıcıların elindeki kartlar bitene kadar oyun devam ediyor. En yüksek puana sahip oyuncu oyunu kazanıyor.

**Anahtar Kelimeler-**  
**Oyuncu, bilgisayar, pokemon ve kart**

## I. GİRİŞ

Proje klasörünün içinde bulunan daha önceden hazırladığımız kartlar gerekli kodlamaların

yapılması ile proje kapsamında istenen işlemleri gerçekleştirir. Kartların kullanılması ile oyun oynanır ve kazanan belirlenir. Bu projede görsel tasarım ve nesneye yönelik programlamanın bir arada kullanıldığı bir çalışma gerçekleştirilmektedir. Aynı zamanda öğrencinin projede tanıtımında sunulan isterler doğrultusunda, kafalarında oluşturdukları tasarımı bir ide aracılığıyla bilgisayar ortamına aktarılması amaçlanmıştır. Proje için tercih edilen programlama dili "Java" olmuştur ve grafik tasarımı için GUI ortamı kullanılmıştır.

## II. TEMEL BİLGİLER

Bu proje Java programlama dilinde geliştirilmiş olup, geliştirme ortamı olarak "NetBeans IDE 8.2" kullanılmıştır. İlk etapta proje için bir yol haritası çıkarılarak ön hazırlık sürecine girilmiştir. Bu aşamada projenin isterlerine yönelik araştırmalar gerçekleştirilmesi adına grup içerisinde bir iş bölümü yapılmış olup elde edilen veriler doğrultusunda projenin ana hatları ortaya çıkarılmış ve büyük ölçüde karşılaşılabilecek problemler saptanıp çözümlendirildikten sonra ide ortamında projenin ilk adımları atılmıştır. Proje yaklaşık 1 hafta gibi süre içerisinde tamamlanmıştır.

### III.YÖNTEM

Öncelikler projede kullanılması için pokemon kartlarının görselleri ve oyun ekranında kullanılacak görseller indirildi ve “Photoscape” uygulaması ile tasarlayacağımız oyuna uygun hale getirildi ardından bu görseller proje klasörüne aktarıldı. Daha sonra, proje tanıtım belgesinde bizlerden yazılması istenen sınıflar ve bunlardan kalıtım alan diğer sınıflar yazıldı. Ardından özelleştirilmesi gereken sınıflar “Override” metodu kullanılarak özelleştirildi. Oyunda kullanılacak pokemonlar, oyuncular ve kontrolleri sağlayacak diğer öğeler tanımlandı. Ardından projenin görsel tasarım kısmına geçildi. Gui ortamı kullanılarak menü ve savaş animasyonları tasarlandı ve isterlerin gerçekleştirilmesine yönelik kodlama işlemleri yapıldı. Projede Encapsulation, Inheritance, Polymorphism, Abstraction yapıları kullanıldı.

*Projede kullanılan sınıflar:*

#### **1)Pokemon**

Özellikler:

pokemonID, pokemonAdi ve pokemonTip

İçinde bulunan fonksiyonlar:

Kurucu fonksiyon(constuctor), Get, Set metotları

#### **Pokemonlar**

Her bir pokemon için ayrı ayrı tasarlanmışlardır ve “Pokemon” sınıfından kalıtım almışlardır. Kurucu fonksiyonlar, Get, Set metotları, “hasarPuani” metodu ve gerekli kontrollerin sağlanması için “kullanıldıMi” metodunu içermektedirler.

Başlıkta pokemonlar olarak tanımlanan 10 adet classın isimleri şu şekildedir:

2-Pikachu

3-Bulbasaur

4-Charmander

5-Squirtle

6-Zubat

7-Psyduck

8-Snorlax

9-Butterfree

10-Jigglypuff

11-Meowth

Bu sınıflarda, kalıtım aldıkları Pokemon sınıfına ek olarak, hasarPuani tanımlaması yapılmış ve proje belgesinde bizlere verilen, kartların tasarımında kullandığımız hasar puanı değerleri bu tanımlamaya atanmıştır.

#### **12)Oyuncu**

Bilgisayar ya da kullanıcı olmak üzere oyunu oynayan iki oyuncu vardır. Bu iki oyuncunun farklı ve aynı özellikleri bulunmaktadır. Aynı özellikleri temsil etmek için **Oyuncu** temel sınıfı oluşturulmuştur. Oyuncu temel sınıfı **Abstract** sınıf olarak tasarlanmıştır.

Bu sınıfta bulunan özellikler ve fonksiyonlar:

\* oyuncuID, oyuncuAdi ve Skor özellikleri

\*Yapıcı (constuctor) metotları

\*kartListesi özelliği

\* SkorGoster() fonksiyonu

\*kartSec() fonksiyonu

#### **13)Bilgisayar**

Bu sınıfta bulunan özellikler ve fonksiyonlar:

\* Oyuncu sınıfından kalıtım almaktadır.

\*Yapıcı (constuctor) metotları

\* kartSec() metodu

#### **14)Kullanıcı**

Bu sınıfta bulunan özellikler ve fonksiyonlar:

\* Oyuncu sınıfından kalıtım almaktadır.

\*Yapıcı (constuctor) metotları içermektedir.

\* kartSec() metodu

*Projede kullanılan görsel tasarım elemanları:*

- 1)Button
- 2)Frame
- 3)Label
- 4)OptionPane

#### **IV.SÖZDE(PSEUDO) KOD**

*Verilen proje için hazırladığımız programın pseudo kodları şu şekildedir:*

- 1-Program çalıştı.
- 2-Kullanıcının menüyü görmesini sağlayan komut çalıştı.
- 3-Kullanıcıya menü aracılığıyla,gerçekleştirmek istediği işlem soruldu. *\*ek 1*
- 4-Kullanıcı “Oyuncu x Bilgisayar” butonuna tıkladı.
- 5-Kullanıcı x Bilgisayar için tasarlanan ekran açıldı. *\*ek2.1*
- 6-Kullanıcı kartları dağıt butonuna tıkladı.
- 7-Kartlar dağıtıldı.  
*\*ek 2.2*
- 8-Kullanıcı ortaya atmak istediği kartın üzerine tıkladı.
- 9-Kullanıcının atmak istediği kart ve bilgisayar kullanısından bir kart ortaya atıldı ve oynanan kartlar oyundan silindi. *\*ek 2.3*
- 10-Atılan kartlar yerine,ortada bulunan kartlardan birer tane çekildi. *\*ek 2.4*

11-Kullanıcı atmak istediği kartın üzerine tıkladı.

12-Kullanıcının seçtiği kart ve bilgisayar oyuncusundan bir kart ortaya atıldı.Atılan kartlar oyundan çıkarıldı.

13-Atılan kartlar yerine ortadan yeni kartlar çekildi ve ortada bulunan kartlar tükendi.

14-Kullanıcı elinde bulunan kartlar bitene kadar seçme işlemine devam etti.Kullanılan kartlar oyundan çıkarıldı.

15-Oynanacak kartlar tükendiğinde,kullanıcı kazananı göster butonuna tıkladı.

16-Ortaya atılan kartların hasar puanlarının karşılaştırılması ile belirlenen kazanan ,ayrı bir pencerede kullanıcıya bildirildi. *\*ek 2.5*

17-Kullanıcı “Bilgisayar x Bilgisayar” butonuna tıkladı.

18- Bilgisayar x Bilgisayar için tasarlanan ekran açıldı.

19-Kullanıcı “kartları dağıt” butonuna tıkladı.

20-Kartlar dağıtıldı. *\*ek 3.1*

21-Kullanıcı “Kart At!” butonun a tıkladı.

22-İki taraftan da birer kart ortaya atıldı ve kullanılan kartlar oyundan çıkarıldı.

23-Kullanıcı “Kart Çek” butonuna tıkladı ve atılan kartların yerine ortadan birer kart çekildi.

24-Kullanıcı “Kart At!” butonuna tıkladı.

25-İki taraftan da birer kart ortaya atıldı ve kullanılan kartlar oyundan çıkarıldı.

26-Kullanıcı “Kart Çek” butonuna tıkladı ve atılan kartların yerine ortadan birer kart çekildi ve ortadaki kartlar tükendi.

24-Kullanıcı “Kart At!” butonuna tıkladı ve bu işlem kartlar tükenene kadar devam ettirildi.

25-Oyuncuların elindeki kartlar tükendi ve kazananı göster butonu aktif hale geldi. \*ek3.2

26-Kullanıcı “Kazananı Göster!” butonuna tıkladı.

27- Ortaya atılan kartların hasar puanlarının karşılaştırılması ile belirlenen kazanan ,ayrı bir pencerede kullanıcıya bildirildi.

28-Program sonlandırıldı.

## V. IMPORT İŞLEMLERİ VE EKLER

**\*import java.util.ArrayList;**  
Pokemonları saklamak için kullanıldı.

**\*import java.util.Collections;**  
Kartları rastgele atamak için kullanıldı.

**\*import java.util.Random;**  
Rastgele sayı üretmek için kullanıldı.

**\*import javax.swing.ImageIcon;**  
Tasarlanan kartları oyuna ekleyebilmek için kullanıldı.

**\*import javax.swing.JButton;**  
Buton oluşturmak için kullanıldı.

**\*import javax.swing.JOptionPane;**  
Kazananı bildirdiğimiz panelleri yaratmak için kullanıldı.

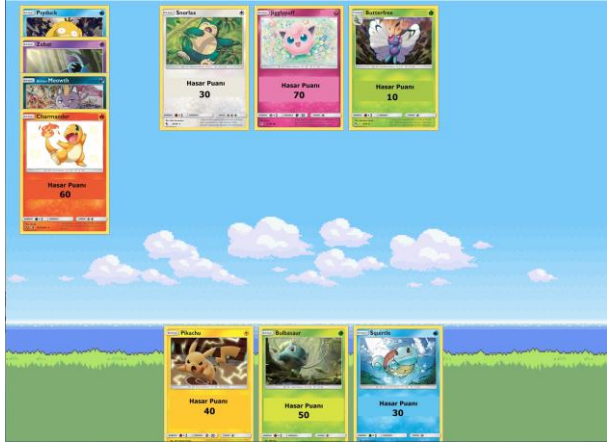
**\*ek1**



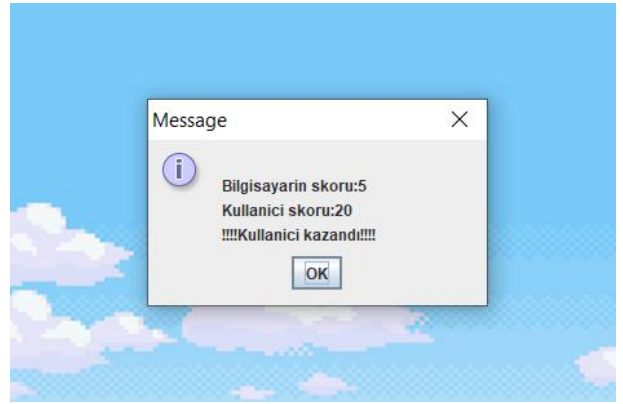
**\*ek 2.1**



\*ek 2.2



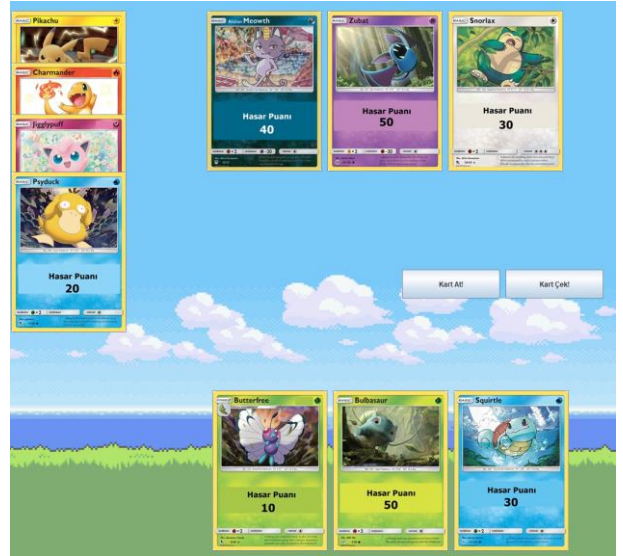
\*ek 2.5



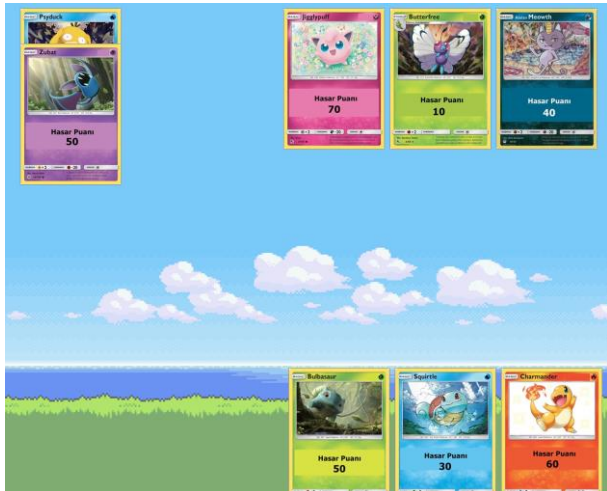
\*ek 2.3



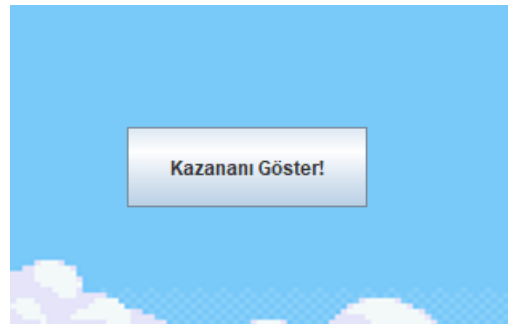
\*ek 3.1



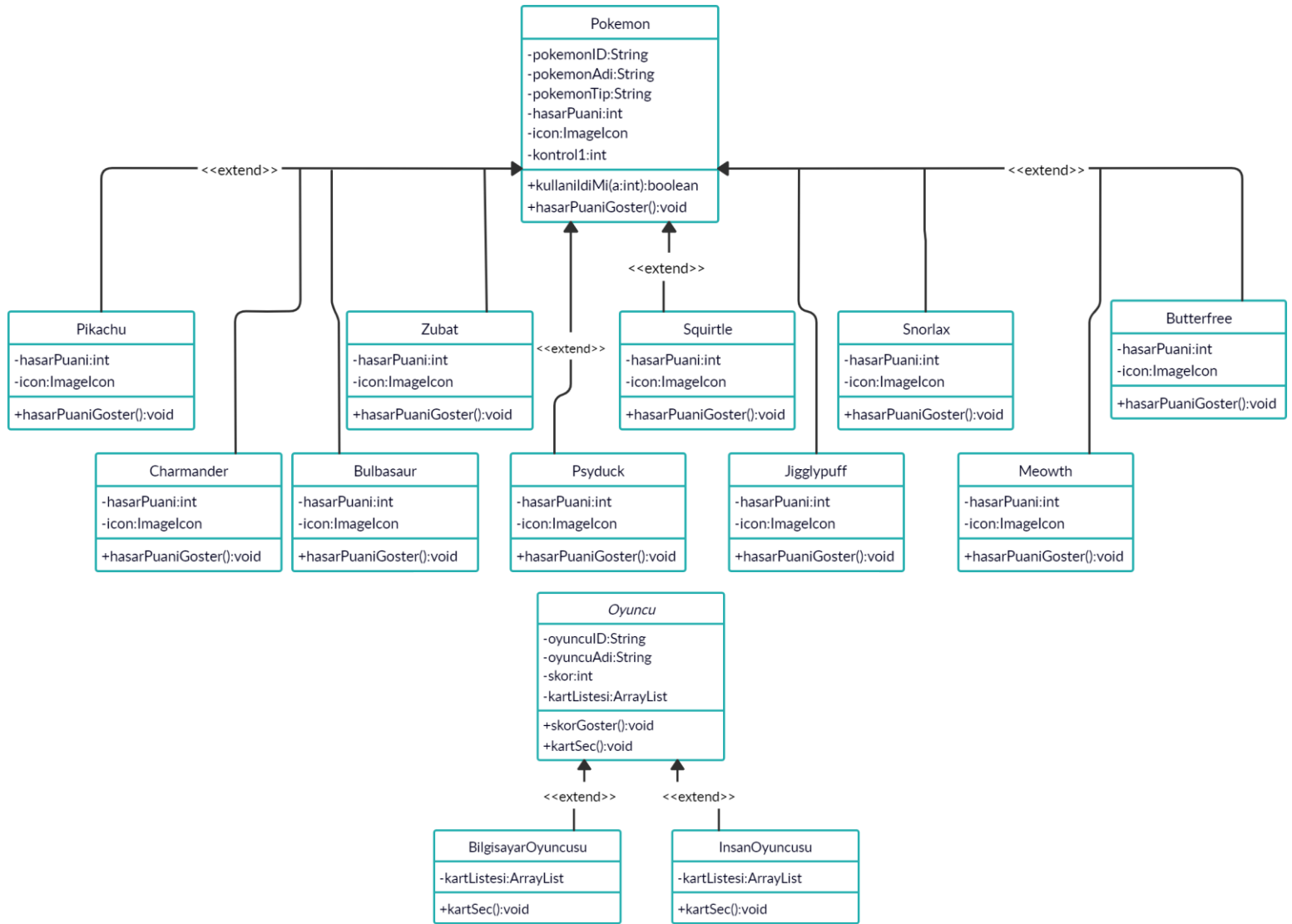
\*ek 2.4



\*ek 3.2



## VI.UML SINIF DIAGRAMI



## VII.KAYNAKÇA

- [1]medium.com
- [2]youtube.com
- [3]creatly.com
- [4]edestek.kocaeli.edu.tr
- [5]it-swarm.net
- [6]pkmncard.com
- [7]wallpaperplay.com