Pokemon Kart Oyunu Projesi

|  |
| --- |
| Taha Batuhan TÜRK - İbrahim PAMUK  180202007 - 180202023  *Bilgisayar Mühendisliği Bölümü*  *Kocaeli Üniversitesi*  180202007@kocaeli.edu.tr - 180202023@kocaeli.edu.tr |

***Özet*–Nesneye yönelik programlama yapısını kullanarak görsel arayüze sahip, takip edilebilecek bir algoritmaya sahip kart oyunu geliştirdik.**

***Anahtar Kelimeler – nesneye yönelik, grafik arayüz***

I. GİRİŞ

Programımız bir oyuncunun(bilgisayar ya da insan) otomatik oyuncuyla (bilgisayar) savaşabileceği basit bir kart oyunu. Oyunda karşılıklı iki oyuncu, önceden karıştırılmış desteden elinde 3 kart bulunana kadar sırayla kartlar çekecek. Kartlar dağıtıldıktan sonra her raund kullanıcılar sırayla ortaya kapalı olarak birer kart atacak ve hasar puanı yüksek karta sahip oyuncu raundu kazanacak. Kazanılan her raund başına kullaıcılara 5(beş) puan eklenir.

Bu durum destede ve kullanıcıların elinde kart kalmayana kadar devam eder.

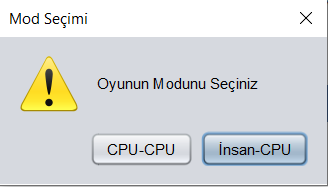
II. YÖNTEM

Programımızı Java programlama dili ve grafik arayüzünü swing API’si ile geliştirdik. Kodumuzda nesneye yönelik programlama( encapsulation, inheritance, polymorphism, abstraction ) ve temel programlama fonksiyonlarını kullandık.

III. DENEYSEL SONUÇLAR

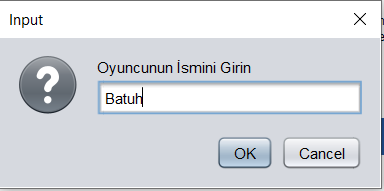
Programımız on altı adet sınıftan oluşuyor. Bunlardan bir tanesi arayüzümüzü yazdığımız arayuz.java, diğeri sık kullandığımız metodları içeren transactions.java sınıfı. Diğerleri de projede kullandığımız diğer sınıflar.

İlk başta kullanıcının karşısına oyun modunu seçmesi için Şekil.1 deki gibi bir menü çıkmaktadır.



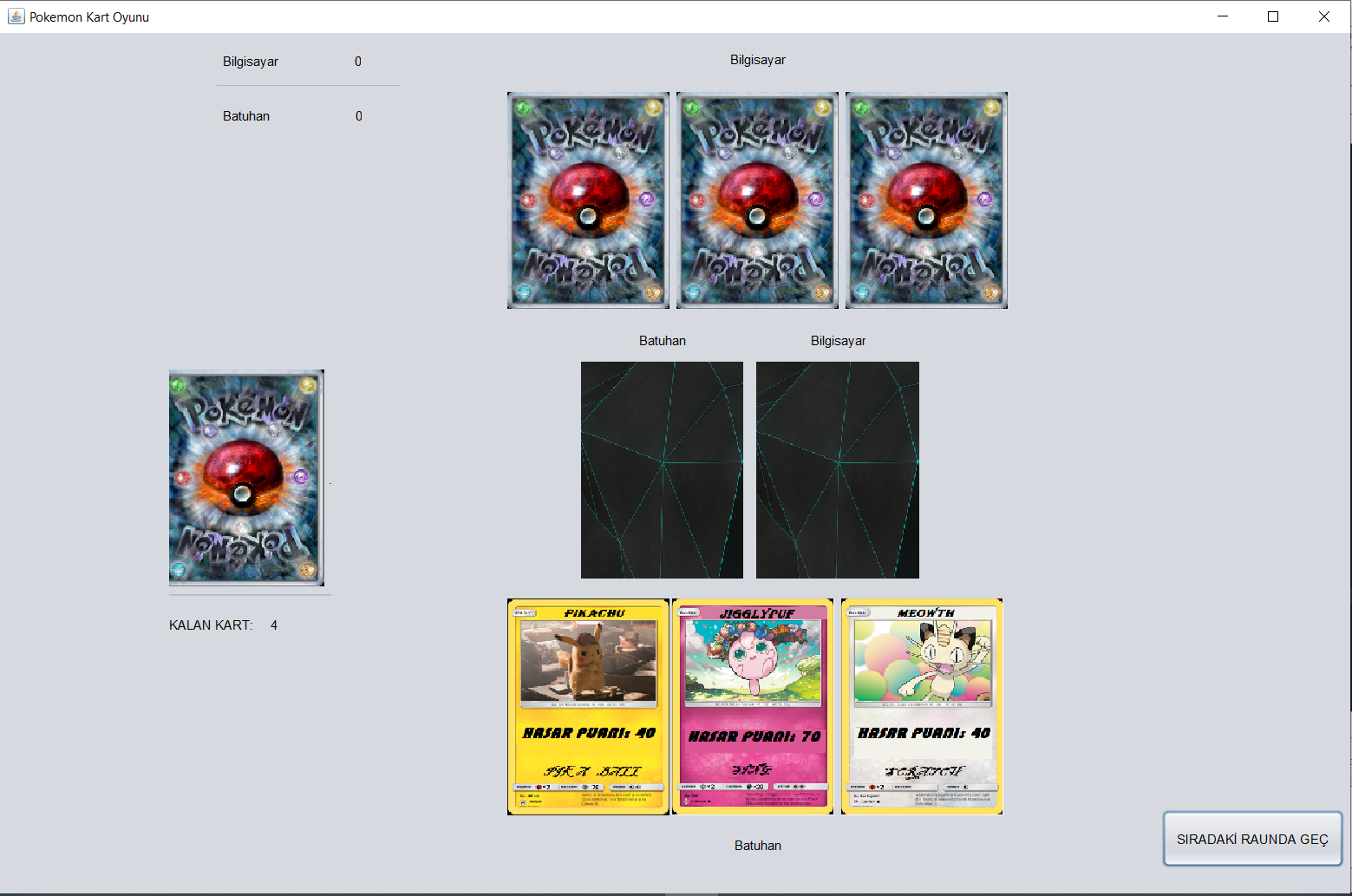
Şekil.1 Kullanıcının karşısına çıkan menü.

Kullanıcı seçim yapmadan Şekil.1 deki pencereyi kapatırsa oyun başlamadan sonlandırılır. Eğer kullanıcı “İnsan-CPU” seçeneğini seçerse, karşısına oyuncunun ismini girmesi için şekil.2 deki isim girme penceresi gelir. Diğer seçenek(CPU-CPU) seçilirse direk sonraki aşamaya geçilir.



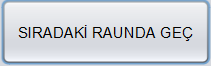
Şekil.2 İsim girme penceresi

Kullanıcı seçimini ve girişini tamamladıktan sonra artık oyunun gerçekleşeceği arayüz olan şekil.3 deki ekranla karşılaşır.

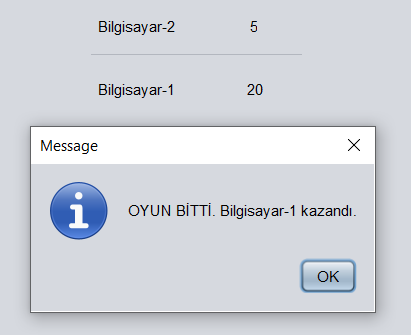


Şekil.3 Oyun Ekranı

Oyun ekranı geldikten sonra kullanıcı-1 insan ise bilgisayar(kullanıcı-2) kartını oynamadan önce insan kullanıcısının mutlaka elindeki kartlardan birtanesine tıklayarak ortaya atması gerekmektedir. Kullanıcı-1, kartını seçtikten sonra şekil.4’ deki “Sıradaki raunda geç” butonuna tıklamalı. Bu tuşa basıldıktan sonra savaştırılan kartların kazanma durumuna göre oyun ekranında bulunan skor tabelasında(şekil.5) anlık puan düzeltmesi gerçekleştirilir.



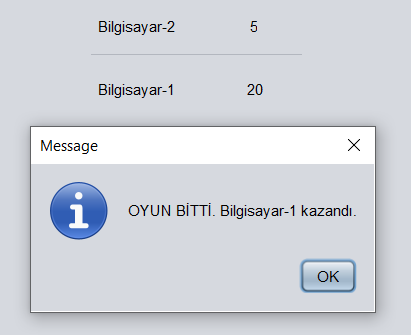
Şekil.4 Sıradaki Raunda Geç Butonu



Şekil.5 Skor Tablosu

Eğer kullanıcı-1 bilgisayar ise mause ile kart seçmesine gerek kalmaz bu bilgisayar tarafından otomatik olarak gerçekleştirilir. Kullanıcının tek yapması gereken her raund için şekil.4’ deki “Sıradaki raunda geç” butonuna tıklamak olur.

Bu işlemler (destedeki kart sayısı/2) kadar gerçekleşir ve oyunun sonucunda elde edilen skorlara göre şekil.6 deki sonuç ekranı gelir.



Şekil.6 Skora göre gelen sonuç ekranı

IV. YALANCI KOD

BAŞLA

1. Pokemon sınıfı Arraylisti oluştur.
2. ImageIcon Arraylisti oluştur
3. Pokemonları Arrayliste ekle
4. İkonları ImageIcon arraylistine ekle.
5. Kullanıcıdan oyun modunu al.

6- İf

{oyun mod == 0

Oyuncu1= BilgisayarOyuncusu

Oyucu2= BilgisayarOyuncusu

}

Else if

{ oyun mod == 1

Oyuncu1= InsanOyuncusu

Oyucu2= BilgisayarOyuncusu

}

1. Oyuncu1 başlangıç kart ata.

Oyuncu2 başlangıç kart ata.

8- Kart ikonları ata.

9- İf

{oyuncu1== InsanOyuncusu

Kullanıcıdan kart girişi iste.

OrtaKart1 ikon ata.

} else if

{ oyuncu1== BilgisayarOyuncusu

bilgisayar kart seç.

OrtaKart1 ikon ata.

}

10- oyuncu2 kart seç

OrtaKart2 ikon ata.

11- İf

{ OrtaKart1 hasar puanı< OrtaKart2 hasar puanı

Oyuncu2 skor +=5

}

else if

{ OrtaKart1 hasar puanı> OrtaKart2 hasar puanı

Oyuncu1 skor +=5

}

Else

{

Oyuncu1 skor +=0

Oyuncu2 skor +=5

}

12- İf

{raund < (ToplamKartSayısı/2)

8. aşamaya dön.

Else

{ İf

{oyuncu1 skor < oyuncu2 skor

Print “OYUN BİTTİ. Oyuncu1 kazandı."

}

Else if

{oyuncu1 skor < oyuncu2 skor

Print “OYUN BİTTİ. Oyuncu2 kazandı."

}

Else

{

Print “OYUN BERABERE BİTTİ”

}

}

BİTİŞ

V. SONUÇ

Bu projeyi geliştirirken nesneye yönelik programlamanın; encapsulation, inheritance, polymorphism, abstraction, method overloading ve overriding gibi temel kavramlarını aktif bir biçimde kullanıp ve bu bağlamda nesneye yönelik program geliştirmeyi öğrendik. Ayrıca javanın sahip olduğu swing API’ sinin olanaklarını kullanıp projemize oynanış açısından görsellik kattık. Projede en zorlandığımız kısım arayüz ile oyunun temel işlemlerinin gerçekleştiği kodları birleştirmek oldu.

VI. KAYNAKÇA

1. Bilgisayar Kavramları. (n.d.). Retrieved from http://bilgisayarkavramlari.sadievrenseker.com
2. Theano, Flutter, KNime, Mean.js, Weka, Solidity, Org.Json, AWS QuickSight, JSON.Simple, Jackson Annotations, Passay, Boon, MuleSoft, Nagios, Matplotlib, Java NIO, PyTorch, SLF4J, Parallax Scrolling, Java Cryptography. (n.d.). Retrieved from https://www.tutorialspoint.com
3. Where Developers Learn, Share, & Build Careers. (n.d.). Retrieved from https://stackoverflow.com