

PROGRAMLAMA LABORATUVARI I – PROJE II
Betül CİHAN 1902010147 -- Deniz ATMACA 190201018
KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
betulchn42@gmail.com -- denizatmaca13@gmail.com
Proje Teslim Tarihi: 08.12.2020

Projenin Özeti

Bizden bir kullanıcının bilgisayarla rekabet edebileceği bir kart oyunu tasarlanması istenmektedir. Oyunda, 8'i futbolcu, 8'i basketbolcu olmak üzere toplamda 16 sporcu kartı bulunacaktır. Bilgisayara ve kullanıcıya rastgele 4 basketbolcu, 4 futbolcu kartı dağıtılacaktır. Futbolcuların penaltı, serbest vuruş ve kaleciyle karşı karşıya özellikleri bulunmaktadır. Basketbolcuların üçlük, ikilik ve serbest atış özellikleri bulunmaktadır. Kullanıcı ve bilgisayar kendilerine dağıtılan 8 karttan her hamlede birini seçerek ortaya koyacaktır. Her hamle ardışık sırayla bir futbolcu bir basketbolcu kartlarıyla oynanacak şekilde olmalıdır.

Giriş

Projeye ilk başta IntelliJ IDEA derleyicisi ile başlandı. Oyunun algoritması burada hazırlandı. Ara yüz tasarlanırken daha kullanışlı olan Eclipse derleyicisine geçildi. Gerekli ara yüz kurulumları yapıldı.

Yöntem

Proje 8 adet sınıf kullanılarak oluşturuldu. Bunlar futbolcu, basketbolcu, sporcu, kullanıcı, bilgisayar, oyuncu, test ve kart sınıflarıdır.

Futbolcu sınıfı Sporcu sınıfından miras alır. SporcuIsim, sporcuTakim, img, penaltı, serbestVurus, kaleciKarsiKarsiya parametreleri ile constructor oluşturulur. Sporcu sınıfındaki sporcuPuaniGoster metodu override edilir. Bu metotta futbolcunun özellikleri yazdırılır.

Basketbolcu sınıfı Sporcu sınıfından miras alır. SporcuIsim, sporcuTakim, img, ikilik, ucluk, serbestAtis parametreleri ile constructor oluşturulur. Sporcu sınıfındaki sporcuPuaniGoster metodu override edilir. Bu metotta basketbolcunun özellikleri yazdırılır.

Sporcu sınıfı abstract olarak oluşturulur. SporcuIsim, sporcuTakim, img parametreleri ile constructor oluşturulur. SporcuPuaniGoster metodu abstract olarak yazılır.

Bilgisayar sınıfı Oyuncu sınıfından miras alır. OyuncuID, oyuncuAdi, Skor parametreleri ile constructor oluşturulur. Oyuncu sınıfındaki SkorGoster ve kartSec metodu override edilir. SkorGoster metodunda bilgisayarın skoru konsol ekranına yazdırılır. KartSec metodunda bilgisayara rastgele kart dağıtılır.

Kullanıcı sınıfı Oyuncu sınıfından miras alır. OyuncuID, oyuncuAdi, Skor parametreleri ile constructor oluşturulur. Oyuncu sınıfındaki SkorGoster ve kartSec metodu override edilir. SkorGoster metodunda kullanıcının skoru konsol ekranına yazdırılır. Sporcu sırasının kimde olduğunu belirlemek için fb değişkeni kullanılır. Bu değişkenin iki ile bölümünden kalana göre sıra belirlenir. KartSec metodunda Kart sınıfında yapılan tıklama seçimine göre kullanıcının seçtiği sporcu kartı konsol ekranına yazdırılır.

```
if(Test.fb % 2 == 1) { Basketbolcu
if(Test.fb % 2 == 0) { Futbolcu
```

Oyuncu sınıfı abstract olarak oluşturulur. OyuncuID, oyuncuAdi, Skor parametreleri ile constructor oluşturulur. KartSec ve SkorGoster metotları abstract olarak yazılır. Bilgisayarın seçtiği futbolcu ve basketbolcuları tutacak arraylistler oluşturulur. Kartlar silindiğinde bu verilere tekrar ulaşılabilmesi için yedek arraylistler oluşturulur. KartListesi metodunda 8 futbolcu kartının bulunduğu futbolcular arraylistinin içinden arraylistin boyutunu göz önüne alarak rastgele bir index seçilir. Bu index'te bulunan veriler bilgisayarın futbolcu kartları için oluşturulan ana ve yedek arrayliste aktarılır. Bu veriler futbolcular arraylistinden silinerek futbolcular arraylistinde sadece kullanıcının futbolcu kartları kalması sağlanır. Bu işlemler bir döngü yardımıyla dört kez gerçekleştirilir. Aynı işlemler basketbolcu kartları için de yapılır.

Test sınıfı kart oyununun gidişatının izlendiği ana sınıftır. Başlangıçta bütün kartların tutulduğu daha sonrasında sadece kullanıcı kartlarının tutulacağı futbolcular ve basketbolcular arraylistleri burada oluşturulur. Kullanıcı ve bilgisayar sınıflarından nesne oluşturulur ve istenen bilgiler girilir. Oyuncu sınıfı abstract bir sınıf olduğundan Kullanıcı sınıfı yardımıyla bir nesne oluşturulur. Futbolcular ve basketbolcular arraylistlerine sporcuların bilgileri ve sporcuların resimleri girilir. Resimler için ImageIcon yapısı kullanılır. Futbolcu ve basketbolcu kartlarının sırayla ilerlemesi için kullanılan fb değişkeni burada oluşturulur. Bu yapıların static olarak oluşturulmasının nedeni diğer sınıflardan nesne oluşturulmadan çağırılmasını sağlamaktır. Oyuncu sınıfının içindeki kartListesi metodu çağırılarak kartların rastgele dağıtılması sağlanır. Bilgisayarın ve kullanıcının kartları konsol ekranına yazdırılır. Fb değişkeni sıfıra eşitlenerek sıranın futbolculardan başlaması sağlanır. Oynanış ve sonuç metotları çağırılır.

Pozisyon bilgisi adında bir metod oluşturulur. Static olarak tanımlanan pozisyon değişkenine 1 ile 3 arasında rastgele bir sayı atanır. Fb'nin 2 ile bölümünden kalan sayının durumuna göre switch-case yapısı oluşturulur. Rastgele belirlenen pozisyon değişkenine göre o turda karşılaştırılacak özellik bilgisi elde edilir. Her karşılaştırılacak özellik bilgisi için kullanıcının kartının özellik puanının bilgisayarındakinden büyük olup olmama ya da eşit olma durumu incelenir. Eğer iki özellik puanı eşit değilse kim kazandıysa onun skoru setSkor ile 10 artırılır ve bilgisayarın kartı arraylistinden silinir. Eğer eşitlik varsa kart silinmez. Böylece kartların geri dağıtılması mekanizması uygulanmış olur. Static ve boolean olarak tanımlanan beraberemi değişkeni eşitlik sağlandığında true'ya çevrilir. Bu değişken daha sonra ara yüz için kullanılacaktır.

Oynanış metodu oluşturulur. Bu metot oyunun devamlılığını sağlar. While döngüsü içine bilgisayarın futbolcu veya basketbolcu kartlarının sıfırdan farklı olma şartı yazılır. Üst üste eşitlikler gelme durumunda bir sporcu tarafı bitebileceği, diğerinde hala kart kalabileceği için bir şart oluşturulması gerekir. Örneğin futbolcu kartı bitip basketbolcu kartları kaldığında algoritmanın tekrar futbolcu kartını sormaması gerekmektedir. Bunun için while içinde iki if bloğu oluşturulur. Fb, futbolcular kartı biterse 1'e, basketbolcular kartı biterse 0'a eşitlenerek sıranın diğer tarafa geçmesi sağlanır. Bilgisayar ve kullanıcı için kartSec metotları çağrılarak kart seçtirilir. Fb bir artırılarak sıra diğer sporcu takımına geçirilir. Pozisyon bilgisi metodu çağrılarak pozisyon ve skor belirlenir. SkorGoster metotları çağrılarak konsol ekranına oyuncuların skorları yazdırılır.

Sonuç metodu oluşturulur ve içine oyun bittiğinde kullanıcı ve bilgisayarın skorlarını karşılaştıran if blokları yazılır.

Kart sınıfında ara yüz işlemleri yapılır. Bu sınıf JFrame'den miras alır. Öncelikle design kısmından butonlar, labellar oluşturulur. Oluşturulan her şeyin kodu otomatik olarak IDE tarafından yazılır. Butonlardan 4 tanesi futbolcular için, 4 tanesi basketbolcular için, 2 tanesi bilgisayar ve kullanıcı kartları için oluşturulur. Sporcuların butonları için arraylistler oluşturulur.

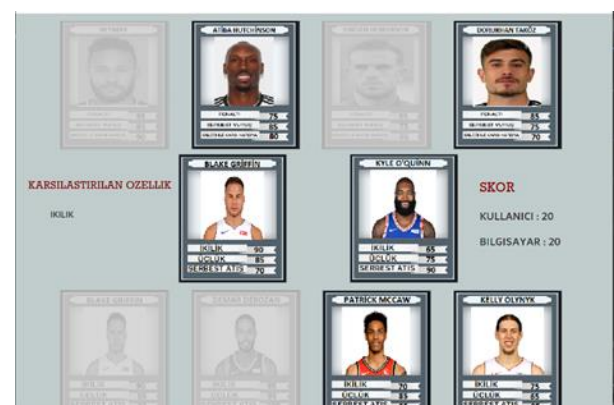
Kartlar isminde bir metod yazılır. Burada kart sınıfından oluşturulan frame nesnesinin görünürlüğü true yapılır. Geri kalan bütün işlemler constructor’da yapılır. Static ve boolean kartKullanildiMi değişkeni oluşturulur. Bu değişken başlangıçta false yapılır. Pencerenin sınırları belirlenir. Arka plan ve açıkta kalan vs1- vs2 butonları için renklendirme yapılır. Burada ek olarak yazı tipleri ve yazı rengi de ayarlanır. 10 adet buton oluşturulur ve 4’er tanesi sırayla arraylistlere eklenir. Karşılaştırılacak özellik ve skorlar için labellar oluşturulur. For döngüleri yardımıyla butonlara resim eklenir. Her bir buton için ActionListener oluşturulur. Fb değişkeni kontrol edilerek sıra hangi sporcudaysa diğerine tıklanılması engellenir. Kullanıcının tıkladığı butonun index’i seçimine eşitlenir ve seçtiği sporcunun

resmi onun için ayrılan butona eklenir. KartKullanildiMi true'ya çevrilir. Seçilen butonun tıklanabilirliğinin kapanması için `setEnabled(false)` ifadesi kullanılır. Bu ifade butonun rengini soluklaştırır. Bilgisayarın rastgele seçtiği kartın resmi bilgisayar için ayrılan butona eklenir.

Pozisyon bilgisi ve skorlar kart seçimlerinden daha sonra ekranda gözükmesi için timer yapısı kullanılır. Run metodu içerisinde beraberlik durumu için if bloğu oluşturulur. 4 farklı durum incelenir ve bu durumlarda kartların geri dağıtılması işleminin ara yüzde uygulanabilmesi için eşitlik gelen kart `setEnabled(true)` yapılır. Bu şartlardan biri gerçekleşirse berabereMi tekrar false yapılır. Fb değişkenin durumuna göre switch-case yapısı kullanılarak pozisyon bilgisi için ayrılan yere rastgele gelen pozisyon yazılır. Kullanıcı ve bilgisayarın skoru ilgili label'a yazılır. Oyunun sonunda kimin kazandığını belirlemek için bir koşul daha yapılmalıdır. If bloğunun içine bilgisayarın futbolcu ve basketbolcu kartlarının bitme şartı yazılır. Kullanıcı ve bilgisayarın skorlarını karşılaştırarak kim kazandıysa `OptionPane.showMessageDialog(null, "");` ifadesi ile ekranda bir kutucuk oluşturulur. Eğer skorlar eşitse farklı bir değerlendirme yapılmalıdır. Bu durumda oyuncuların son kartları tekrar oynanmalıdır. Son kartlar futbolcu ise iki kartın penaltı özellikleri karşılaştırılır. Onlar da eşitse serbest vuruş özelliğine bakılır. Bu pozisyonlar ve yenilenen skorlar tekrar ekrana yazdırılır. Kazanan durumu tekrar bir kutucuk içinde belirtilir. Son kartlar basketbolcu ise iki kartın ikilik özellikleri karşılaştırılır. Onlar da eşitse serbest atış özelliğine bakılır. Diğer işlemler aynen yapılır. Timer'ın oyun bittikten sonra devamlı çalışmaması için `timer.cancel()` ifadesi yazılır. Kazanan durumundan sonra kart penceresinin de kapanması için `setVisible(false)` ifadesi kullanılır.

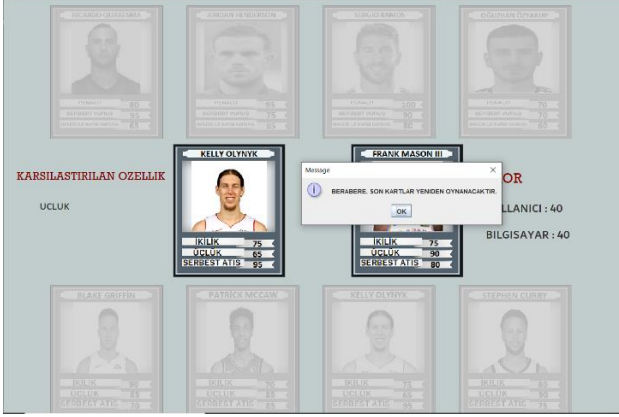
Deneysel Sonuçlar

Projeye öncelikle algoritmayı yazarak başladık. Daha sonrasında ara yüze uyarladık. Ancak bu kısım projenin en zorlu kısmı oldu. Yazdığımız yerlerde değişiklikler ve eklemeler yapmak zorunda kaldık. İki kartı karşılaştırmak için bir buton daha ekledik. Ancak bu buton sıralamaları iyice karıştırdı. Örneğin pozisyon bilgilerini kartları seçmeden önce görüyorduk. Çözümü timer yapısını kullanmakta bulduk.



Sonuç

Sonuç olarak 16 sporcu kartını rastgele dağıtan, karşılaştırılacak özellikleri rastgele seçen, bir sporcu tarafının kartı tükendiğinde diğer tarafın kartları bitene kadar soran, eşitlik durumunda kartları geri iade eden, skor eşitliği durumunda son kartları tekrar oynayan bir kart oyunu elde ettik.



Kaynakça

https://www.youtube.com/playlist?list=PLEcJSEQK_cD5KHgg9sXumeg659hAr2j4W

<https://medium.com/@tugrulbayrak/uml-class-diagramlari-4c3bb7e9cc4c>

<https://web.cs.hacettepe.edu.tr/~burcucan/BBS514-Algorithms.pdf>

Yalancı Kod

Futbolcu veri tipinde ArrayList oluştur.
Basketbolcu veri tipinde ArrayList oluştur.
Oyuncu sınıfından oyuncu nesnesi oluştur.
Ekle: futbolcular ArrayListine (SporcuAdi, SporcuTakim, img, penalti, serbestVurus, kaleciKarsiKarsiya)
Ekle: basketbolcular ArrayListine (SporcuAdi, SporcuTakim, img, ikilik, ucluk, serbestAtis)
Yazdır: "Bilgisayarın Futbolcu Kartları"
for(i 0'dan 4'e kadar)
 random index al
 Yazdır: "Isim, Takim, Penalti, Serbest Vurus, Kaleci ile Karsi Karsiya"
 Ekle: pcSecilenFutbolcular ArrayListine futbolcular ArrayListinin random indexini
 Ekle: pcSecilenFutbolcularYedek ArrayListine futbolcular ArrayListinin random indexini
 Sil: futbolcular ArrayListinden random indexi
Yazdır: "Bilgisayarın Basketbolcu Kartları"
for(i 0'dan 4'e kadar)
 random index al
 Yazdır: "Isim, Takim, Ikilik, Ucluk, Serbest Atis"
 Ekle: pcSecilenBasketbolcular ArrayListine basketbolcular ArrayListinin random indexini
 Ekle: pcSecilenBasketbolcularYedek ArrayListine basketbolcular ArrayListinin random indexini
 Sil: basketbolcular ArrayListinden random indexi

Yazdır: "Kullanıcının Futbolcu Kartları"

for(i 0'dan futbolcular'ın boyutuna kadar)

Yazdır: "Isim, Takim, Penalti, Serbest Vurus, Kaleci ile Karsi Karsiya"

Yazdır: "Kullanıcının Basketbolcu Kartları"

for(i 0'dan basketbolcular'ın boyutuna kadar)

Yazdır: "Isim, Takim, Ikilik, Ucluk, Serbest Atis"

fb'ye 0 ata.

while(pcSecilenFutbolcular'ın ya da

pcSecilenBasketbolcular'ın boyutu 0'dan farklı)

if(pcSecilenFutbolcular'ın boyutu 0'a eşit)

fb'ye 0 ata.

if(pcSecilenBasketbolcular'ın boyutu 0'a eşit)

fb'ye 1 ata.

if (fb'nin 2 ile bölümünden kalan 1'e eşit)

bpc'ye random sayı ata.

Yazdır: "Bilgisayarın seçtiği basketbolcu kartı -> Isim: , Takim: , Ikilik: , Ucluk: , Serbest Atis: "

if (fb'nin 2 ile bölümünden kalan 0'a eşit)

fpc'ye random sayı ata.

Yazdır: "Bilgisayarın seçtiği futbolcu kartı -> Isim: , Takim: , Penalti: , Serbest Vurus: , Kaleci ile Karsi Karsiya:"

while(kartKullanildiMi false' a eşit)

100 ms uyu.

kartKullanildiMi'ya false ata.

if (fb'nin 2 ile bölümünden kalan 1'e eşit)

Yazdır: "Hangi basketbolcu kartını seçeceksiniz?"

Yazdır: "Sectiginiz basketbolcu kartı->Bilgisayarın seçtiği basketbolcu kartı -> Isim: , Takim: , Ikilik: , Ucluk: , Serbest Atis: "

if (fb'nin 2 ile bölümünden kalan 0'a eşit)

Yazdır: "Hangi futbolcu kartını seçeceksiniz?"

Yazdır: "Sectiginiz futbolcu kartı->Isim: , Takim: , Penalti: , Serbest Vurus: , Kaleci ile Karsi Karsiya: "

Ekle: fb'ye 1 ekle.

pozisyon'a random sayı ata.

switch(pozisyon)

case :

Yazdır: "Karsilastirilacak Ozellik: pozisyon"

if(kullanıcının ArrayListinin pozisyonu bilgisayarın ArrayListinden büyük)

Yazdır: "Kullanici bu eli kazandi."

Kullanıcının skor'una 10 ekle.

Sil: bilgisayarın ArrayListinden pc'yi.

else if(kullanıcının ArrayListinin pozisyonu bilgisayarın ArrayListinden küçük)

Yazdır: Bilgisayar bu eli kazandi."

Bilgisayarın skor'una 10 ekle.

Sil: bilgisayarın ArrayListinden pc'yi.

else

Yazdır: "Pozisyon ozellikleri esittir. Kartlar geri dagitildi."

berabereMi'ye true ata.

Buton oluştur.

Ekle: Butonları ArrayListlere ekle.

Label oluştur.

for(i 0'dan 4' e kadar)

Ekle: butonlara resim ekle.

```

if(fb'nin 2 ile bölümünden kalan)
    secim'e kalanı ata.
Ekle: vs1 butonuna resim ekle.
kartKullanildiMi'ya true ata.
Ekle: vs2 butonuna resim ekle.
if(berbabereMi)
    if(fb'nin 2 ile bölümünden kalan 0 ve
pcSecilenBasketbolcular'ın boyutu 0 değil)
        bButon'un secim indexinin tıklanabilirliğini aç.
    if(fb'nin 2 ile bölümünden kalan 0 ve
pcSecilenBasketbolcular'ın boyutu 0'a eşit)
        fButon'un secim indexinin tıklanabilirliğini aç.
    if(fb'nin 2 ile bölümünden kalan 1 ve
pcSecilenFutbolcular'ın boyutu 0 değil)
        fButon'un secim indexinin tıklanabilirliğini aç.
    if(fb'nin 2 ile bölümünden kalan 1 ve
pcSecilenFutbolcular'ın boyutu 0'a eşit)
        bButon'un secim indexinin tıklanabilirliğini aç.
berabereMi'ye false ata.
if(fb'nin 2 ile bölümünden kalan)
    switch(pozisyon)
    case:
        Label'a yazdır: "pozisyon"

```

```

Label'a yazdır: "KULLANICI: "
Label'a yazdır: "BILGISAYAR: "
if(pcSecilenFutbolcular'ın ve pcSecilenBasketbolcular'ın
boyutu 0'a eşit)
    if(kullanıcının skoru bilgisayarın skorundan büyük)
        Mesaj yazdır: "KAZANDINIZ"
    else if(kullanıcının skoru bilgisayarın skorundan küçük)
        Mesaj yazdır: "KAYBETTINIZ"
    else
        Mesaj yazdır: "Berabere. Son kartlar tekrar
oyunacaktır."
    if(fb'nin 2 ile bölümünden kalan)
        if(kullanıcının ArrayListinin pozisyonu bilgisayarın
ArrayListinden büyük)
            Label'a yazdır: "pozisyon"
            Ekle: kullanıcının skor'una 10 ekle.
            Mesaj yazdır: "KAZANDINIZ"
        else if(kullanıcının ArrayListinin pozisyonu bilgisayarın
ArrayListinden küçük)
            Label'a yazdır: "pozisyon"
            Ekle: bilgisayarın skor'una 10 ekle.
            Mesaj yazdır: "KAYBETTINIZ"
        else
            Mesaj yazdır: "BERABERE"

```

UML SINIF DİYAGRAMI

