

KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ

BÖLÜMÜ PROGRAMLAMA LAB. I

PROJE 2

190201017 Fatma Gül YILDIRIM

190201076 Kerem KARATAŞ

Özet:

Projede bizden istenen bilgisayarın bir kullanıcı ile karşılıklı oynayabileceği bir oyun tasarlamızdır.

Tasarladığımız oyunda 16 adet oyun kartı olacak ve bunların 8 i futbol kartı 8 i ise basketbol kartıdır. Kullanıcıya rastgele 4 futbol 4 basketbol kartı verilecek ve kalan 8 kart bilgisayarın olacaktır. Hamle sırasında kullanıcı istediği kartı seçerken bilgisayarın kartı random olarak seçilir. Ortaya konan bu iki kartın 3 özelliği vardır ve bu özelliklerden biri (random olarak seçiyoruz 3 özellikten birini) karşılaştırılır. Kartın seçilen özellik puanları aynıysa kartlar geri alınıp yeniden sıradaki sporcu için seçim yapılmalıdır. Eğer aynı değilse kazanan tarafın skoru 10 puan artar. Aynı zamanda kazanılan bir hamle içerisinde oyuna sürülen kartlar tekrar kullanılamayacaktır. Bu işleyişe göre oyunumuz elimizdeki kartlar bitene kadar devam eder. Sıra son karta geldiğinde karşılaştırılan özellik puanları aynı ise diğer özelliklerine bakılır. Eğer 3 özellik puanı da aynı ise oyun sonlandırılır. Oyun kurallarına ek olarak oyunu takip edebileceğimiz kartların görünür olduğu bir arayüz tasarlamamız beklenmektedir.

Giriş:

Projeyi yaparken ide olarak Netbeans idesini kullandık. Arayüz tasarımı için ise swing kütüphanesinden yararlandık.

Java Swing, zengin bir widget kümesi içeren hafif bir Grafik Kullanıcı Arabirimi (GUI) araç takımıdır. Paket, Java uygulamalarımız için GUI bileşenleri yapmamızı sağlar ve platformdan bağımsızdır.

Oluşturulması Gereken Sınıflar:

- ❖ Sporcu Sınıfı
sporculsim ve sporcuTakim özelliklerine sahiptir. Kart puanlarını göstermek için kartPuanınıGoster metodunu içerir.
- ❖ Futbolcu Sınıfı

Sporcu sınıfından kalıtım alır.

penalti, serbestVurus, kaleciyleKarşıKarşıya ve kartKullanıldıMi özelliklerine sahiptir.

❖ Basketbolcu Sınıfı

Sporcu sınıfından kalıtım alır.

İkilik, ucluk, serbestAtis ve kartKullanıldıMi özelliklerine sahiptir.

❖ Oyuncu Sınıfı

oyuncuID, oyuncuAdi ve skor özelliklerine sahiptir. BasketbolKartListesi ve FutbolKartListesi ile oyun kartlarımızı bir arraylistte tutuyoruz. FKartSec ve BKartSec metotları ile kart seçimi yapıyoruz. skorGoster metodu ile kart skorlarını ekrana bastırıyoruz.

❖ Kullanıcı Sınıfı

Oyuncu sınıfından kalıtım alır. .FKartSec ve BKartSec metotları override edilir.

❖ Bilgisayar Sınıfı

Oyuncu sınıfından kalıtım alır. .FKartSec ve BKartSec metotları override edilir.

❖ Test Sınıfı

Futbolcu ve basketbolcuların isimleri ve özellikleri burada tanımlanmalıdır. Pozisyon bilgisi için pozisyonBilgisi metodunu içerir.

Yöntem:

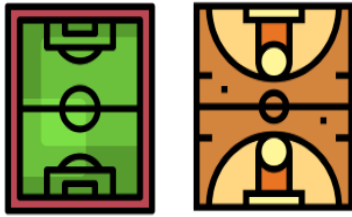
- ❖ Projede ilk olarak bizden istenilen sınıfları oluşturduk. Ardından test sınıfında futbol ve basketbol kartlarımızı oluşturup bir arraylistte tuttuk. Daha sonra bu kartlardan 8ini random olarak kullanıcıya kalan 8ini ise bilgisayara verdik. Bu kullanıcı ve bilgisayarın kartlarını da bir arraylistte tuttuk. Bu bizim işimizi kolaylaştırdı.
- ❖ Daha sonra sıra arayüzü yapmaya geldi. Arayüzde ilk olarak çizimleri swing kütüphanesinden yararlanarak kendimiz çezecektik ve üç boyutlu bir oyun tasarlayacaktık. Ancak böyle bir durumda

sürükle bırak özelliğini kullanamıyorduk ve bu bizim işimizi zorlaştırıyordu.O yüzden her kart için bir buton oluşturduk ve kullanıcının butonlarına tıklama özelliği verdik.Bu sayede kullanıcı istediği kartı seçebilecekken bilgisayarın kartları karşısına random olarak gelecekti.

- ❖ Oyun giriş ekranında kullanıcıdan ad bilgisini alıp oyunu başlatıyoruz.

Adınız:

- ❖ Daha sonra başta oluşturduğumuz bu 16 butona futbol ve basketbol kartları için ayrı ayrı resim ekledik.



- ❖ Kartların başta kullanılma durumu false idi.Oyunda bir futbol bir basketbol kartı olmak üzere sıra bozulmadan oynanacak şekilde bir düzen kurduk.Sıra futbolcudayken basketbol kartı seçmeyi engelledik.
- ❖ Oyun kullanıcının herhangi bir futbol kartını tıklamasıyla başlıyor ve tıkladığı anda hem oyuncunun hem bilgisayarın oyun kartları açılıyor.



- ❖ Açılan bu kartların random olarak seçilen özellik puanları karşılaştırılıyor ve kartların özellikleri ekranda gözüktüyor.Daha yüksek özellik kartına sahip olan oyuncuya 10 puan veriliyor.KartKullanildiMi bilgisi true yapılıyor. Ve kazanma durumu gerçekleştiği için bu 2 kart oyundan siliniyor.Bir daha kullanılamıyor.
- ❖ Eğer karşılaştırılan özellik aynıysa KartKullanildiMi bilgisi false olarak kalıyor ve kartlar tekrardan kullanılabilir.
- ❖ Her bir hamle sırasında skor tablosunda puanlarımız görüntüleniyor.

SKOR TABLOSU

Deniz
skor
20
Bilgisayar
skor
10

Deniz	Kaleciyle Karşı Karşıya:10 Serbest Vuruş:79 Penaltı:77
Bilgisayar	Kaleciyle Karşı Karşıya:45 Serbest Vuruş:62 Penaltı:50

- ❖ Elimizdeki kartlar bitene kadar bu şekilde ilerliyor oyun.
- ❖ Sıra son kartlara geldiğinde karşılaştırılan özellik puanları aynı ise diğer bir özelliğe geçiliyor.Eğer tüm özellikler aynıysa oyun bitiriliyor.
- ❖ Oyunun sonunda oyun bitti mesajı ve kazanan kişi ... şeklinde bir uyarı yazısı geliyor.

Yalancı Kod:

- 1-)Pencere boyutunu ayarla ve panellerini ekle.
- 2-)8 buton insan için 8buton bilgisayar için ekle.
- 3-)Giriş ekranı ,skor tablosu ve kart özellikleri için panel,label,text vs. ekle.
- 4-)Sadece insanın butonlarına tıklama event i ver.
- 5-)int x=4;

```
int y;  
int fsay=0;  
int bsay=0;  
int fsayi=0;  
int bsayi=0;  
int pozisyon;  
int cik=0;  
int a;
```

- 6-)Kartlarını oluştur ve bir arraylistte tut. Fkartseç ve Bkart seç metodunda kartları random olarak dağıt.
- 7-)Giriş dışındaki panellerin görünürlüğüünü kapat.
- 8-)Giriş dışındaki panellerin görünürlüğüünü aç. Tüm kartlara arka yüz ekle resimEkle(arkaYüz);
- 9-)if(fsay==bsay||bsayi==4)
Futbol kartına tıklayabilirsin.

- fsay++;
- yaz(sporcuPuanı);
- if(futbol kartı==false)
resimEkle(futbolcu);
random sayı al
resimEkle(futbolcuRandomBilgisayar);

```

yaz(sporcuPuanıRandomBilgisayar);
• pozisyon bilgisini al
• pozisyon = Test.pozisyonBilgisi();
• if(cik==7)
son kart için karşılaştırma yapıyoruz.
İf(insan1.özellik>pc1.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan1.özellik<pc1.özellik)
pc.skor+=10;
Kart=true;
else İf(insan2.özellik>pc2.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan2.özellik<pc2.özellik)
pc.skor+=10;
Kart=true;
else İf(insan3.özellik>pc3.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan3.özellik<pc3.özellik)
pc.skor+=10;
Kart=true;
else
kart=true;
• if(cik<7)
random pozisyon bilgisi al.
• İf(pozisyon==0)
İf(insan1.özellik>pc1.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan1.özellik<pc1.özellik)
pc.skor+=10;
Kart=true;
• İf(pozisyon==1)
İf(insan2.özellik>pc2.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan2.özellik<pc2.özellik)
pc.skor+=10;
Kart=true;
• Yaz(İnsan.skor);
Yaz(pc.skor);
• İf(kart==true)
3 saniye bekle
Sil(kartlar)
Cik++;
Fsayi++;
Sil(randomSayiindisi);
c--;
• İf(cik==8)
Mesaj(oyun bitti);
Mesaj(kazanan ....);
• İf(kart==false)
3 saniye bekle.
Resim(arkaYüz)
(kart tekrar kullanılabiliyor bu sayede)
10-)if(fsay-1=bsay||fsayi==4)

```

Basketbol kartına tıklayabilirsin.

```

• bsay++;
• yaz(sporcuPuanı);
• if(basketbol kartı==false)
resimEkle(basketbolcu);
random sayı al
resimEkle(basketbolcuRandomBilgisayar);
yaz(sporcuPuanıRandomBilgisayar);
• pozisyon bilgisini al
• pozisyon = Test.pozisyonBilgisi();
• if(cik==7)
son kart için karşılaştırma yapıyoruz.
İf(insan1.özellik>pc1.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan1.özellik<pc1.özellik)
pc.skor+=10;
Kart=true;
else İf(insan2.özellik>pc2.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan2.özellik<pc2.özellik)
pc.skor+=10;
Kart=true;
else İf(insan3.özellik>pc3.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan3.özellik<pc3.özellik)
pc.skor+=10;
Kart=true;
else
kart=true;
• if(cik<7)
random pozisyon bilgisi al.
• İf(pozisyon==0)
İf(insan1.özellik>pc1.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan1.özellik<pc1.özellik)
pc.skor+=10;
Kart=true;
• İf(pozisyon==1)
İf(insan2.özellik>pc2.özellik)
İnsan.skor+=10;
Kart=true;
Else İf(insan2.özellik<pc2.özellik)
pc.skor+=10;
Kart=true;
• Yaz(İnsan.skor);
Yaz(pc.skor);
• İf(kart==true)
3 saniye bekle
Sil(kartlar)
Cik++;
Bsayi++;
Sil(randomSayiindisi);
b--;
• İf(cik==8)

```

- Mesaj(oyun bitti);
Mesaj(kazanan);
- If(kart==false)
3 saniye bekle.
Resim(arkaYüz)
(kart tekrar kullanılabilir bu sayede)

11-)Oyun bu algoritmik sırada ilerliyor.Bir futbolcu bir basketbolcu şeklinde.

Deneysel Sonuçlar:

1

Karşılaştırılan kartların kaçınıcı özelliğine bakıldığını yazdırdık.

İkilik:105 Serbest Atış:85 Üçlük:67

İkilik:90 Serbest Atış:85 Üçlük:69

Kartların özelliklerini yazdırdık

Kazanan Bilgisayar

OK

Puanlar

Merve:30

Bilgisayar:50

OK

Kazanan yok.Oyun berabere bitti

OK

Oyun bitimizde gelebilecek mesajlar

Kazanımlar ve Sonuç:

Bizden istenen

1. Soyutlama (Abstraction) Soyutlama denilince, nesneyi bazı karakteristikleri olan ve bazı eylemleri gerçekleştirebilen bir veri tipi olarak genelleştirmek anlaşılmalıdır. ...
2. Sarmalama / Paketleme (Encapsulation) ...
3. 3. Miras Alma (Inheritance) ...
4. Çok Biçimlilik (Polymorphism)

bu 4 ilkeyi öğrenmemizdi.

Proje sayesinde classlar arasındaki hiyerarşiye daha bir hakim olduk.Projede inheritance(kalıtım) kullandık.Bir yerde abstract class kullanmamız gerekiyordu ancak abstract classın nesnesi oluşturulmadığı için biraz sıkıntı çektik ve classı abstract yapmaktan vazgeçtik.İçindeki metodları extend edildiği class da override ettik.

- Projede bizden istenen tüm özellikleri classları ve metodları oluşturduk,kullandık.
- Projede istenen özel durum şartlarını da doğru bir şekilde ekledik.

Kaynakça:

<https://www.tutorialspoint.com/java/index.htm>

<https://www.javatpoint.com/java-swing>

<https://www.yusufsezer.com.tr/java-thread/>

<https://youtu.be/ZOg1i27i04s>

SKOR TABLOSU

Toprak

skor

40

Bilgisayar

skor

40

