

KOCAELİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

PROGRAMLAMA LABORATUVARI I

PROJE II:

SPORCU KART OYUNU

Proje Özeti:

Programlama Laboratuvarı I dersinin 2. Projesinde, sporcu kart oyunu yapılması istendi. Öncelikle belirlenen iki spor dalında sekizer oyuncu seçerek onlara üç özellik atanması istendi. Biz futbol ve basketbol sporlarını tercih ettik. Her sporcuya üç farklı özellik ekledik. Her özellik için farklı farklı puanlar atandı. Oyunun başında kullanıcıya, dört futbolcu, dört basketbolcu olacak şekilde rastgele kartlar verilecek. Oyun bir futbol bir basketbol seçerek ilerleyecek ve kullanıcı ile birlikte bilgisayar da kendisine kalan kartlardan rastgele bir kart atacak. Ardından rastgele bir özellik seçilecek ve bu özellikleri karşılaştırılacak. Kazanana 10 puan verilecek, berabere ise kartlar geri alınacak. Son kartta berabere kalması söz konusuysa otomatik olarak farklı bir özellik karşılaştırılacak. Böylece oyun ortaya çıkacak.

Bunlar için Swing GUI kullanıldı. Bu seçim, Swing GUI'nin kolaylık ve yeterlilik bakımından avantajlı olduğu düşünüldüğü için yapıldı.

Bu projede Java Programlama dili ve Netbeans/Eclipse geliştirme programları kullanılarak yapıldı.

1. GİRİŞ

Bu raporda, rastgele seçilen kartlar için kullandığımız algoritma, rastgele seçilen pozisyon için kullandığımız algoritma, kullanılan arayüzün açıklamaları ve yapılan arayüz için gerekli bilgiler bulunmaktadır.

Proje Java Programlama dilinde ve Netbeans/Eclipse geliştirme ortamında yazılmıştır.

2. YÖNTEM

2.1 Swing GUI

Bu projeyi yaparken kullanılan arayüz tasarımı için öncelikle bir JFrame projesi açılır. Burada bulunan “Design” kısmının içinden proje için gerekli malzemeler sürükleyip bırak yardımı ile arayüze yerleştirilir.

Kartlar buton şeklinde yapıldı. Böylelikle kullanıcı seçtiği kartın üzerine tıkladığı zaman, kartı seçmiş oluyor.

Sol tarafta skorları gösteren veriler, sağ tarafta ise seçilen pozisyon ve turu kazananlar, en üst kısımda en son oyunu kazanan yazıldı. Orta tarafta ise kullanıcının seçtiği kart ve bilgisayarın seçtiği kartlar yer alıyor.

Kullanılan komutlar/metodlar aşağıdaki gibidir.

initComponents() = Arayüzü açar.

ButonVeyaLabelAdi.setIcon() = Adı girilen buton veya labelın içine resim eklemimize olanak sağlar.

ButonAdi.setEnabled(false/true) = “true” girildiği zaman butona erişimi sağlar. “false” girildiği zaman butona erişimimizi durdurur. Yani butona basılamaz.

LabelAdi.setText() = Adı girilen label içine istenenleri yazar.

ButonVeyaLabelAdi.setVisible(false/true) =
“true” girildiği zaman belirtilen nesneyi görünür hale getirir. “false” girildiği zaman ise görünmez hale getirir.

2.2 Rastgele Kart Seçimi

Rastgele kart seçimi için öncelikle futbolcuları ve basketbolcuları belirlemek gerekiyor. Bu tanımlamaların hepsi Test Class’ında yapıldı. Tanımlamalar için dizi tercih edildi. Bu nedenle rastgele kart seçimi için sayıları kullanmak yeterli oldu.

Rastgele seçilen sayılar sporcuların indisini belirtiyor. Ve bu indis, kartKullanildiMi(), kartListesi() metodlarının içindekilerden birisi ile eşleşmeyecek şekilde seçiliyor. Böylelikle bir kez kullanılan kart bir daha kullanılmayacak şekilde ayarlanıyor.

2.3 Rastgele Pozisyon Seçimi

Basketbolcular ve futbolcular için farklı farklı seçimler uygulanıyor. Eğer futbolcuysa Test Class’ındaki metod “f” parametresini alıyor. Basketbolcuysa “b” parametresini alıyor ve 0’dan 2’ye kadar rastgele bir sayı üretiliyor

Futbolcuysa ve sayı bir ise Kaleci ile Karşı Karşıya, iki ise Serbest Atış, üç ise Penaltı pozisyonunu seçiyor.

Basketbolcuysa ve sayı bir ise İkilik, iki ise Üçlük, üç ise Serbest Atış pozisyonunu seçiyor.

Böylece rastgele bir pozisyon seçilmiş oluyor. Eğer son kartta seçilen pozisyonların değerleri aynı ise bu işlemi birkaç kez tekrarlıyor ki son kart da kullanılmış olsun.

3.DENEYSEL SONUÇLAR

Bu projede kullanılan java dilinin nesneye yönelik programlama yapısı kullanılmıştır.

Bu projede ;

Inheritance kullanılmıştır. Inheritance miras alma işlemidir. Bir sınıfın kendisine ait özellikleri ve işlevleri bir başka sınıfa aynen aktarması ya da bazı özellik ve işlevlerini diğer sınıfların kullanmasına izin vermesidir. Inheritance, daha önce yazılmış sınıf içindeki kod parçacıklarının tekrar tekrar yazılmadan başka sınıflar için kullanılmasını sağladığından iş yükünü hafifleten ve zaman kazandıran bir yöntem olduğu görülmüştür.

Metotlar override edilerek kullanılmıştır. Override etme işlemi, bir classtaki metodu başka bir classta kullanmak için süper classtaki metodun diğer classın nesnesine uygun olarak çalışmasını sağlamaktır. Override etmek aynı metodun diğer classta tekrar yazılmamasını sağladığından override işleminin iş yükünü hafifleten ve programlamacıya zaman kazandıran bir yöntem olduğu görülmüştür.

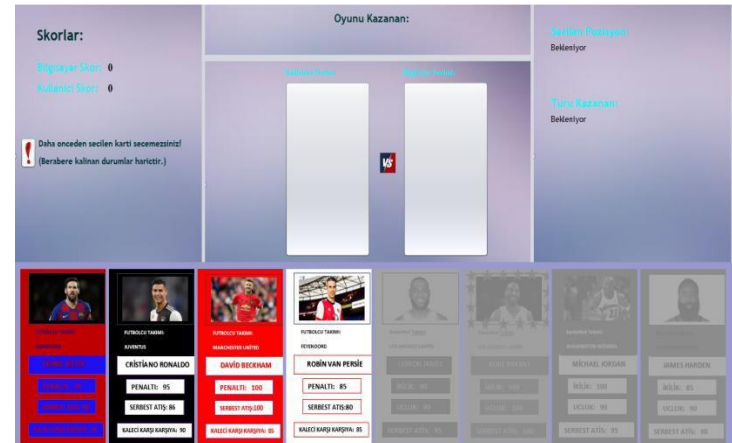
Swing kütüphanesi kullanılmıştır. Swing ile arayüz oluşturulmuştur. Swing kütüphanesinde bulunan JButton ile seçme işleminin yapılacağı butonlar(projede bu butonlar kart olarak kullanılmıştır) oluşturulmuştur. JPanel ile paneller açılmış, JLabel ile arka plan resimlendirilmiş ve oyun takibini sağlayan skor, kullanıcıSecimi, bilgisayarSecimi vb. yazılmıştır. Bu işlemlerle swing kütüphanesi kullanılarak design tasarlanmış, Actionperformed ile oyun işleyişi tasarlanmıştır. Swing kütüphanesi ve java programlama dili kullanımıyla oyun yapımı öğrenilmiştir.

4. SONUÇ

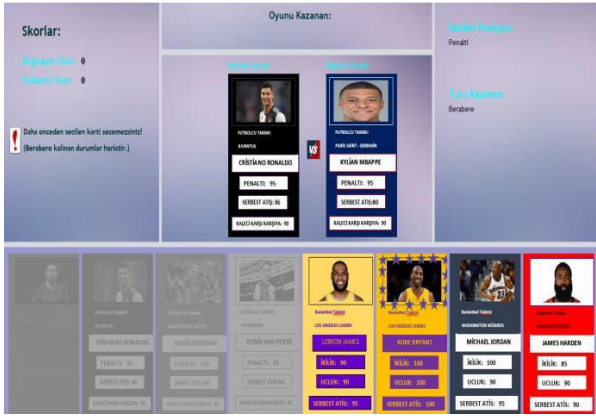
Bu projeye java programlama dilinde classlarla ilgilendik ve kendimizi inheritance, metod override gibi konularda geliştirdik. Ara yüz kullanarak algoritma kurma yeteneğimizi geliştirdik. Nesneye yönelik programlamanın mantığını daha iyi kavradık ve nesneye yönelik programlamada kendimizi geliştirdik. Ara yüz aracılığıyla java dilini kullanarak 2D oyun tasarlamayı öğrendik. Daha verimli ve işlevsel bir kod nasıl yazabiliriz gibi sorularla ilgilendik. UML diyagramı çizmeyi öğrendik. Kullanıcının etkileşimde olduğu bir kod yazdık. Bu şekilde nesneye yönelik programlamanın mantığını anlamış olduk.

5.GÖRSELLER

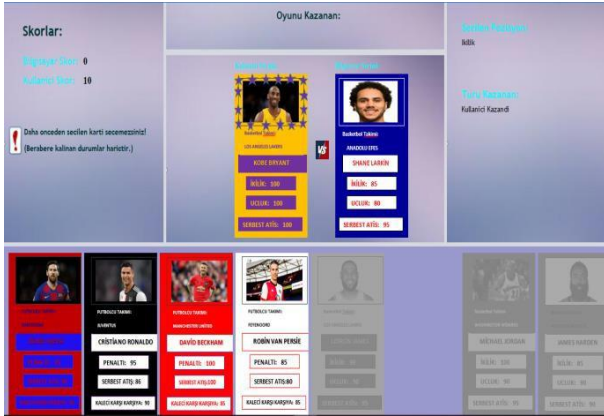
Oyun başlangıcı:



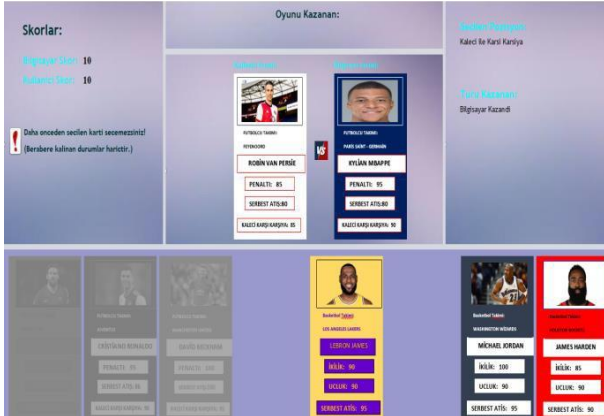
Berabere durumunda:



Kullanıcının kazandığı durumda:



Bilgisayarın kazandığı durumda:



6. EKSİKLİKLER

Bu projede istenenler yapıldı. Fakat onun dışında arayüz olarak daha fazla gelişmeler gösterilebilirdi. Farklı uyarılar ve farklı davranışlar içerebilirdi. Görselleştirmeler biraz daha ileri seviye olabilirdi.

Kod olarak ise bazı override edilen fonksiyonların yazımında gereksizlik vardır. Ve bazı kodlarda tekrar olduğu için tam olarak sade bir kod olmadı.

7. YAKLAŞIM

Nesne yönelimli programlama, nesneyi merkezine alan bir programlama yaklaşımıdır. Nesne yönelimli yaklaşımda uygulama, nesneler ve onların ilişkileri çerçevesinde belirli bir iş yapmak için geliştirilebilmelidir. Her nesnenin bir sınıfı olmalıdır ve sınıflar nesnelerin ortak davranışlarını ifade etmelidir. Nesneler birbirleriyle iletişime geçebilmelidir. Nesne yönelimli yaklaşımda dört temel özellik vardır:

1. Abstraction(Soyutlama)

Soyutlama, nesneyi bazı karakteristik özellikleri olan ve bazı fonksiyonları gerçekleştirebilen bir veri tipi olarak genelleştirmektir. Soyutlama işlemi class yapısı ile gerçekleştirilir. Class yapısı içinde o nesneye ait özellikler ve metodlar tanımlanır. Soyut sınıflardan nesne oluşturulamaz. Soyut yöntemlere sahip bir sınıfın kendisi de otomatik olarak soyut olur ve öyle tanımlanır. Static,final,private olarak tanımlı yöntemler soyut olarak tanımlanamazlar.

2. Encapsulation(Paketleme)

Kapsülleme, bir nesnenin iç yapısını, özelliklerini bozulmalara ve hatalara karşı dış dünyadan korumaktır. Bir nesnenin iç yapısının korunması demek başka sınıflar tarafından korunan özelliklere erişilmemesi demektir. Bu koruma işlemi programlama dillerinde erişim belirleyiciler sayesinde olmaktadır. Bir sınıfın özelliklerini dışardan erişime kapatabilmek için sınıfı private olarak tanımlamamız gerekir. Tanımladığımız sınıfın özelliklerine doğrudan erişmemiz mümkün değildir. Dolaylı yoldan erişebilmek için getter ve setter metodlarını kullanırız.

3. Inheritance(Kalıtım alma)

Bir sınıfın kendisine ait özelliklerini ve metodlarını bir başka sınıfa aynen aktarması ya da bazı özellik ve metodların diğer sınıflarda kullanılmasına olanak sağlamasıdır. Üst sınıf özelliklerini alt sınıflarda kullanmak için extends kullanılır. Bir alt sınıf üst sınıfa erişmek istediğinde süper() kullanmalıdır. Süper() in iki kullanımı vardır. Üst classtaki öğelere erişmek için ve üst sınıfa ait nesne yaratmak için kullanılır.

4. Polymorphism(Cok biçimlilik)

Farklı şekilde çalışan nesnelere aynı şekilde erişmek şeklinde tanımlayabiliriz. Diğer bir tanım olarak bir nesnenin farklı bir nesne gibi davranmasıdır.

Projede nesneye yönelik yaklaşımın 4 temel özelliğinden 2si kullanılmıştır. Kullanılan yöntemler: Inheritance ve Encapsulation yöntemleridir.

Inheritance projede;

Basketbolcu() ve Futbolcu() classlarının Sporcu() classından özellik ve metod alması için kullanılmıştır.

Bilgisayar() ve Kullanici() classlarının Oyuncu() classından özellik ve metod alması için kullanılmıştır.

Encapsulation projede;

Private değişken tanımlarken kullanılmıştır. Oyuncu(),Sporcu() classlarında değişken tanımlarken kullanılmış daha sonrasında getter() ve setter() metodları ile projenin gerekli yerlerinde private değişkene erişim sağlanmıştır.

8. KAZANIMLAR

Nesneye yönelik programla ile ilgili temel kavramlar hakkında ve uygulamada yetkinlik kazanıldı.

Java dilinde class yapısı ve nesnelerin birbiriyle nasıl etkileşim sağlayabileceği öğrenildi.

Inheritance kavramı öğrenildi ve uygulamada inheritance ın nasıl kullanılacağı ve faydaları öğrenildi.

Encapsulation kavramı öğrenildi, private kavramının üzerinde duruldu. Kullanılma sebebi ve programcıya sağlayacağı kolaylıklar öğrenildi.

Metodların nasıl override edileceği ve bunun getireceği kolaylıklar anlaşıldı.Static kavramı öğrenildi ve uygulamada kullanıldı.

Swing kütüphanesi öğrenildi, arayüz tasarımının nasıl yapılacağı anlaşıldı.

Java dilinde algoritma geliştirme öğrenildi.

Nesne tabanlı programlamanın mantığı öğrenildi ve uygulamaya nasıl döküleceği görüldü.

9. KAYNAKÇA

<https://www.youtube.com/watch?v=50hiy0TTaU4&list=PLzIWkToFwqHRjHdDsrJhQSWGaoJkAHIU4&index=4>

<https://www.youtube.com/watch?v=Z0Pq0oysC8I&list=PLzIWkToFwqHRjHdDsrJhQSWGaoJkAHIU4&index=5>

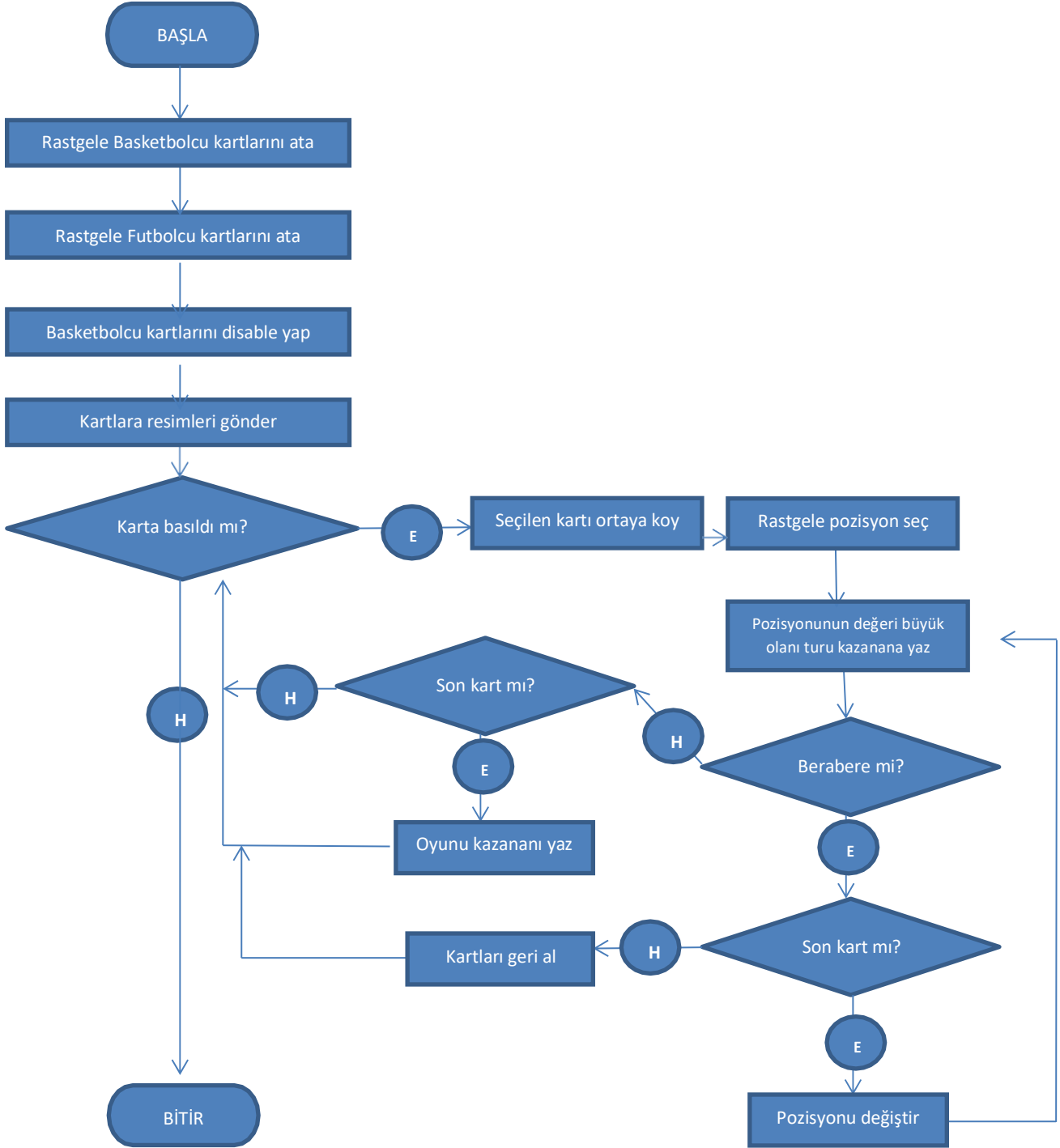
<https://www.youtube.com/watch?v=T1eGtpGuJNU&list=PLzIWkToFwqHRjHdDsrJhQSWGaoJkAHIU4&index=22>

<https://www.youtube.com/watch?v=ZOg1i27j04s>

https://www.youtube.com/watch?v=hbI6pOpCdjc&list=PLop1ia2yhdnnAYjJa-0izua1CP83RkUJ_&index=6

<https://www.udemy.com/course/sifirdan-ileri-seviyeye-komple-java-gelistirici-kursu/learn/lecture/8605716?start=75>

10. AKIŞ DİYAGRAMI



11. UML DİYAGRAMI

