

SPORCU KARTI OYUNU PROJESİ

Aslı Özen, Beyza Hatip

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

asliiozen@hotmail.com, beyzahatip17@gmail.com

Özet

Bu projede 8 voleybol, 8 artistik buz pateni sporuna ait kartlar, 2 oyuncuya dağıtıldı, swing paketini kullanılarak oluşturulan ara yüz ile bir kart oyunu tasarlandı. Her kart için 3 pozisyon bilgisi tanımlandı. Bir voleybolcu bir buz patenci sırasıyla oynanan bu kart oyununda rastgele seçilen pozisyonda yüksek değere sahip olan oyuncu 10 puan aldı. Eldeki kartlar bitene kadar oyun devam etti. 2 sporcu ve 2 oyuncu sınıfı için birer üst sınıf oluşturuldu. Bunlardan kalıtım alan alt sınıf özelliklerine test sınıfında atama yapılarak, oyun test sınıfında takip edildi.

Giriş

Çalışmada amaç ;bizlerin nesneye yönelik programlama yapısını anlaması ve çözüm sağlayabilmesidir. Bizden sporcu ana classından kalıtım alan 2 adet sporcu alt sınıf ve bir oyuncu ana sınıfından kalıtım alan 2 adet oyuncu alt sınıfı oluşturmamız; buradaki özelliklere atama yapmak için super() fonksiyonu kullanmamız ve metotları override edebilmemiz, oluşturduğumuz test sınıfında sporculara özellikleri atamamız, oluşturulan kartları oyunculara dağıtmamız, sporcular için belirlenen 3 pozisyondan rastgele seçilenler arasında karşılaştırma yapıp yüksek değerli olan kartın oyuncusuna puan eklememiz, seçilen pozisyonlarda beraberlik durumunda kartı desteye geri koymamız ve bu kart karşılaştırmalarını sırayla her seferinde bir diğer spor branşını seçerek yapmamız istendi. Bunun için bir “Athlete” ana sınıfını oluşturduk. Burada oluşturduğumuz özelliklerin ataması için bir parametrelili constructor ve set get methodları kullandık. Alt sınıflarda override edilmek üzere pozisyon değerlerini yazdıran bir metot tanımladık.”volleyPlayer” ve “figureSkater” alt klaslarında “Athlete” sınıfından kalıtım aldığımız

özelliklere atama yapmak için super() fonksiyonu kullandık. Böylece sporcu adı, ülkesi ve pozisyon bilgilerine atama yapmış olduk. Aynı zamanda pozisyon bilgileri için set, get methodları da kullandık. Test sınıfında birden fazla işi aynı anda yapabilmek için “thread” sınıfını extend ediyoruz. Burada kartların ara yüzde kullanılması ve başla butonu için JButtonları, ekranda çıkacak yazılar için JLabel, JTextFieldları ve ara yüzü oluşturmak için JFrame özelliklerini oluşturuyoruz. Girdi-çıkı işlemlerinde bir hata olduğunda daha iyi yönetebilmemiz için eklediğimiz IOException sınıfının altında her iki oyuncunun kartları ve ortaya konulacak kartlar için butonları nesnesi, tüm kartları tutmak için bir ArrayList oluşturduk. Kartların arka yüzünün fotoğrafı için dosya yolunu belirterek bir atama yaptık. Computer ve Player sınıfları için nesne oluşturarak 3 özelliğe atama yaptık. Ekrandaki yazılar için JTextField ve JLabel nesnelerini oluşturduk. Oyuna başlamak için konulan butona “Start Game” yazısı atadık”. createCardButton() metodunda parametre olarak gelen x ve y konumlarını kullanarak bir buton yarattık ve daha önce dosya yolunu eklediğimiz kartların arka yüzünün fotoğrafını butonlara ekledik. createScreen() metodu ile oyunda kullanacağımız yazıların tiplerini, renklerini ve konumlarını belirledik ve bunları frame’e ekledik. Başlat butonunu da aynı şekilde özelleştirdikten ve butona tıklandığında oyunun başlaması için startgame() metodunu da çağırdıktan sonra frame’e ekliyoruz. Daha sonra tüm kart butonlarının ekrandaki konumlarını belirledikten sonra açılacak oyun penceresinin boyutlarını belirleyip görünürlüğü “true” yapıyoruz. startgame() metodu ile oyun başladıktan sonra görünmesini istemediğimiz buton ve yazıların görünürlüğünü “false” edip kartları tanımlamak ve kartları dağıtmak için kullandığımız defineCards() ve dealCards() fonksiyonlarını çağırdık. this.start() ile

oyunu başlattık. Thread sınıfından kalıtım aldığımız için run metodunu override ederek gerekli komutları yazmaya başladık. Her iki spor dalı için kart butonlarını tutan “dnmf” ve “dnmv”;0’ dan 7’ye kadar indisleri tutan “dnm1f” ve “dnm1v” ArrayListleri oluşturduk. Bütün kart butonlarını görünür yaptık. Oyun 8 kere döneceği için bir while döngüsü açıyoruz ve 8 kere döndürüyoruz.p1(Computer),ve p2(Player) kart seçene kadar orta kartların görünürlüğü (false) ediyoruz. Gamer sınıfında yazdığımız showScore() fonksiyonunu p1 ve p2 için kullanarak oyuncuların skorlarını gösteriyoruz ve p2’nin seçtiği kart indeksini bulmak için Gamer sınıfından override ettiğimiz pickCard() metoduna p2 için tanımladığımız kart butonlarını ve dnmf, dnmv,test ArrayList ve değişkeni gönderiyoruz. p2 için bunlar kullanılmayacak sebebi daha sonra anlatılacaktır.”Gamer” sınıfından kalıtım alan “Player” sınıfında butonlara tıklamak için oluşturduğumuz ActionListenerlar ve Test classına return etmek için bir “chosenCard” değişkeni tanımladık. Parametrelili constructor ile “Gamer” sınıfından kalıtım aldığımız özelliklerin atanması için super() fonksiyonu kullandık.”Gamer” sınıfından override ettiğimiz “pickCard” metodu “test” sınıfından gelen butonlara ActionListener ekleyip actionPerformed() metodu ile tıklamamızı ve seçilen karta uygun indisi Test’e return etmemizi sağlar. Burada Test sınıfında tanımladığımız “test” değişkeni ile oyuncunun sırayla bir voleybolcu bir figürücü seçmesini ve en başta her ikisini seçmekte de özgür olmasını kontrol ediyoruz. Yöntem kısmında daha ayrıntılı bahsedilecektir. ”test” sınıfına gelen indis ile birlikte bir switch-case yapısı kurduk ve seçilen indise ait kart butonunun görünürlüğü (false) ederken orta kart butonuna fotoğrafını koyup görünür yaptık. ”test==0” if koşulu ilk durumda voleybolcu seçilmesi durumunda sıranın buz patenciye geçmesi için konulmuştur. Aynı şekilde “test” değişkeninin döngüde her seferinde artırılması bu sıranın düzgün işlemesi için yapılmıştır.”p1” oyuncusu bilgisayar olduğundan onun için bir tıklama işlemi yapılmamıştır. Aynı şekilde “Gamer” sınıfından kalıtım alan “Computer” sınıfında da “pickCard()” metodu override edildi ve Test sınıfına bu sefer p1’in kart butonları ile

“dnmf”, “dnmv”,test”, “p2ChoosenCardIndex” gönderildi. ”Computer” class’ı da super() ile atamalarını gerçekleştirdikten sonra “pickCard()” metodu ile kendisine dağıtılan kartlardan buz patencilerin odluğu ArrayList(dnmf) ve voleybolcuların olduğu ArrayList(dnmv)’nin dizi boyutlarına göre random bir sayı seçip “test” classına gönderiyor. Kendisine parametre olarak gelen ”p2ChoosenCardIndex” küçük ise 4’ten figürücü,4’ten büyük ise voleybolcu listelerinin boyutlarına göre random sayı gönderiyor. Test classında 3 pozisyon değerinin karşılaştırılması için bir “probSelection” değişkeni tanımladık. Her seferinde random olarak seçilen bu değişken için if koşulları açtık. Bu koşullar seçilen pozisyon için oyuncu kartlarını karşılaştırıyor ve yüksek değerli pozisyon sahip oyuncuya puan ekleyip bunu ekrana yazdırıyor. Eşitlik durumunda kartı desteye geri ekliyor. Son durumda(ctrl==7) iken oyunun bitmemesi için “ctrl” 1 azaltılıyor. Döngüye girmeden önce tanımladığımız ve 0’ a eşitlediğimiz“tmp1”, “tmp2”, “tmp3” değişkenleri son elde tüm pozisyonların eşit olması durumunda bütün pozisyonları karşılaştırıp oyunu bitirmesi için kullanılmıştır. Bu pozisyon karşılaştırmaları ”p2ChoosenCardIndex” in 4’ten küçük(buz patenci) veya 4’e eşit-büyük olması (voleybolcu) durumları için birer kez yaptırılıyor. p1 için “pickCard()” metodu ile gelen değere ait kartın o pozisyonundaki değerini karşılaştırıyor. Bu arada p1 için gelen değer hangi spora ait ise o değer ve o değere ait buton listelerden bir daha seçilmemesi için siliniyor. Oyun bitimden showscore() metodundan yararlanarak kazanma ya da berabere bitme durumu ekrana yazdırılıyor. Run() bittikten sonra addPicsToCards() metodu ile Player’ın kart fotoğraflarını ekliyoruz, Computer’ın her zaman arkası dönük olacağı için burada bir ekleme yapmıyoruz. defineCards() metodu ile bütün kartları tanımlıyoruz, özelliklere atama yapıyoruz ve her kartın fotoğrafları için dosya yollarını ekliyoruz. Bu kartları daha önce tanımladığımız “cards” listesine ekliyoruz. dealCards() metodu ile Player’a random kart ataması yapıyoruz geri kalan kartlar ise Computer’a atanıyor. Atanan kartı listeden siliyoruz. main metodu altında frame nesnesini oluşturuyoruz. Oyunun arka planının dosya yolunu ekliyoruz. Pencerenin köşesindeki kapatma tuşuna tıkladığımızda programımızın

sonlanması için JFrame'in setDefaultCloseOperation metodunu, yine JFrame'de tanımlı EXIT_ON_CLOSE sabitiyle çağırıyoruz.

Yöntem

Athlete classında öncelikle javax.swing.ImageIcon kütüphanesini import ediyoruz. Daha sonra abstract tipinde Athlete classında athleteName, athleteTeam, prob1, prob2, prob3, pic özelliklerini tanımlıyoruz. Athlete classının parametrelili constructureında parametre olarak pic özelliği hariç bütün özellikleri göndererek ve this keywordü ile atama yapılmasını sağlıyoruz. Daha sonra parametresiz bir Athlete constructureı daha tanımlıyoruz. getAthleteName(), getAthleteTeam() ve getPic() getter metodlarıyla athleteName, athleteTeam ve pic özelliklerini return ederken setAthleteName() ve setAthleteTeam() setter metodları ile bu özelliklere atama yapılmasını sağlıyoruz. Daha sonra figureSkater ve volleyPlayer subclasslarında özelleştirmek amacıyla showAthleteScore() metodunu tanımlıyoruz.

figureSkater classını oluştururken öncelikle extends keywordünü kullanarak figureSkater classını Athlete classının bir subclassı haline getiriyoruz daha sonra fName, fTeam, quadAxel, deathSpiral ve spin özelliklerini tanımlıyoruz.figureSkater adlı parametrelili constructureımıza parametre olarak yukarıda tanımladığımız bütün özellikleri gönderiyoruz ve atama işleminin yapılması için super() metodunu kullanarak bu parametreleri Athlete classımızdaki parametrelili constructureımıza gönderiyoruz. getSpin(), getDeathSpiral(), getQuadAxel() getter metodlarını kullanarak spin,quadAxel ve deathSpiral özelliklerini return ederken setSpin(), setDeathSpiral(), setQuadAxel() setter metodlarını kullanarak bu özelliklere atama yapılmasını sağlıyoruz. Athlete classında oluşturduğumuz showAthleteScore() metodunu burada özelleştiriyoruz ve spin,quadAxel ve deathSpiral özelliklerinin konsola yazdırılmasını sağlıyoruz.isCardUsed

özelliğini tanımlıyoruz ve getIsCardUsed() metodu ile return edilmesini sağlarken setIsCardUsed() metodu ile atama yapılmasını sağlıyoruz.

volleyPlayer classını oluştururken öncelikle extends keywordünü kullanarak volleyPlayer classını Athlete classının bir subclassı haline getiriyoruz daha sonra vPName, vPTeam, dunk, serve, leadPass ve isCardUsed özelliklerini tanımlıyoruz.volleyPlayer adlı parametrelili constructureımıza parametre olarak yukarıda tanımladığımız isCardUsed dışındaki bütün özellikleri gönderiyoruz ve atama işleminin yapılması için super() metodunu kullanarak bu parametreleri Athlete classımızdaki parametrelili constructureımıza gönderiyoruz. getDunk (), getServe(), getLeadPass() getter metodlarını kullanarak dunk,serve ve leadPass özelliklerini return ederken setDunk (), setServe(), setLeadPass() setter metodlarını kullanarak bu özelliklere atama yapılmasını sağlıyoruz. Athlete classında oluşturduğumuz showAthleteScore() metodunu burada özelleştiriyoruz ve dunk,serve ve leadPass özelliklerinin sout kullanılarak konsola yazdırılmasını sağlıyoruz. isCardUsed özelliğinin getIsCardUsed() metodu ile return edilmesini sağlarken setIsCardUsed() metodu ile atama yapılmasını sağlıyoruz.

abstract tipli Gamer classımızı yazmadan önce javax.swing.*, java.util.List, java.util.ArrayList kütüphanelerini import ediyoruz. Daha sonra classımızın içinde gamerName, score ve cards özelliklerini tanımlıyoruz. Parametrelili Gamer constructureımıza parametre olarak gamerName ve score özelliklerini gönderip this keywordünü kullanarak bu özelliklere atama yapılmasını sağlıyoruz. Gamer adlı parametresiz constructureımızda score özelliğine 0 değerini atıyoruz. addPoint() isimli metot ile score özelliğinin değerini 10 artırdık. showScore() metodu ile score özelliğini return ettirdik.

8 adet kart butonumuzu, 2 adet listemizi ve 1 adet int tanımlı p2si değerini göndereceğimiz abstract pickCard() metodunu tanımladık.

Computer classımızı yazmadan önce javax.swing.*, java.util.Random, java.util.ArrayList kütüphanelerini import ediyoruz. Computer classını oluştururken öncelikle extends keywordünü kullanarak Computer classını Gamer classının bir subclassı haline getiriyoruz daha sonra cScore, ComputerName özelliklerini tanımlıyoruz. Computer adlı parametrelili constructureımıza parametre olarak yukarıda tanımladığımız özellikleri gönderiyoruz ve atama işleminin yapılması için super() metodunu kullanarak bu parametreleri Gamer classımızdaki parametrelili constructureımıza gönderiyoruz. Parametresiz bir Computer constructureı tanımlıyoruz. Gamer classında tanımladığımız pickCard() metodunu burada override ediyoruz. pickCard() metodu kendisine “Test” classından gönderilen ArrayList dnmf, ArrayList dnmv, int p2ChoosenCardIndex parametrelerini kullanacak. Gelen diğer parametreler bu metot override edildiği için Player sınıfı kart butonlarını kullanabilmesi sebebiyle yazılmıştır.”Test” sınıfında tanımladığımız “dnmf” listesi buz patencilerin kartlarını tutarken;”dnmv” listesi voleybolcuların kartlarını tutuyor.”dnm1f” buz patencilerin 4’ten küçük olan indislerini tutarken;”dnm1v”, 4’eşit-büyük olan indisleri tutuyor.Player tarafından seçilen indisli kart 4’e eşit-büyükse voleybolcu;küçükse buz patenci göndermesi için bu classta gelen indise göre seçilen sporun kartlarının olduğu listenin boyutunu parametre olarak alan random classından alınan değer“Test”e return ediliyor.”Test”e gelen değerle birlikte pozisyon karşılaştırmaları için random olarak belirlediğimiz probSelection için 3 adet if açıyoruz.Herhangi seçilen pozisyonda Playerın kartlarında tanımladığımız seçilen karttaki o pozisyon değeri ile Computer sınıfından “Test”e return edilen değeri “dnm1f” veya “dnm1v” sayı tutan listelerin indisi şeklinde yazıp bu değerın Computerın kartlarında denk gelen pozisyonu karşılaştırıyoruz. setText ile yüksek değerli oyuncun adını yazdırıyoruz ve Gamer sınıfında yazdığımız addPoint() metodu ile o oyuncuya puan ekliyoruz. Eşitlik

durumunda döngünün dönmesini kontrol ettiğimiz ctrl değişkenini bir azaltıyor. ArrayListlere ise seçilen indis değerindeki elemanlarını geri ekliyoruz. Böylece bu el oynanmamış gibi devam ediyor ama sıra hangi sporcudaysa el yine ona geçiyor. Son elde(ctrl==7) pozisyonların eşitliğini kontrol etmek ve hepsi eşitse oyunu bitirmek için while döngüsünden önce tanımladığımız ve 0’a eşitlediğimiz “tmp1”,”tmp2”,”tmp3” değişkenlerini kullanıyoruz. Random olarak seçilen pozisyon 0 ise tmp1’i;1 ise, tmp2’yi;3 ise tmp3’ü if(ctrl==7) koşulunun altında 1 artırıyoruz. if(tmp1!=0 && tmp2!=0 && tmp3!=0) koşulu sadece bu değerler 0’dan farklıysa Playerın kartını ve Computerın kartını görünmez yapıp oyunu bitiriyor.Bu durumda son elde bütün pozisyonların eşiliği durumunu kontrol ettikten sonra oyunu bitirebiliyoruz. Bu karşılaştırmalar Playerın seçtiği kartın voleybolcu veya buz patenci olması durumu için iki ayrı if koşulu atında yaptırılıyor. Eğer p2ChoosenCardIndex >= 4 (Player bir voleybolcu seçmiştir) ise Computer dnm1v ve dnmv ArrayListlerini kullanırken ;tam tersi ise dnm1f ve dnmf ArrayListlerini kullanıyor. Yüksek değerli olan kartın sahibinin Seçilen kartını ortaya koymak için setVisible() ile görünürlüğü false edip orta kart butonlarına dnm1f veya dnm1v de kullanarak Computer kartlarına atama yaptığımız foroğrafı getPic() ile koyuyoruz. Computer için yine aynı kart seçilmesin diye kullandığımız ArrayListlerden p1ChoosenCardIndex değerini siliyoruz.

Player classımızı yazmadan önce javax.swing.*, java.awt.event.ActionEvent, java.util.ArrayList ve java.awt.event.ActionListener kütüphanelerini import ediyoruz. Player classını oluştururken öncelikle extends keywordünü kullanarak Player classını Gamer classının bir subclassı haline getiriyoruz daha sonra pScore, PlayerName, ActionListener tipinde choosenCard,choosenCardB1, choosenCardB2, choosenCardB3, choosenCardB4, choosenCardB5, choosenCardB6, choosenCardB7, choosenCardB8 özelliklerini tanımlıyoruz. Player adlı parametrelili

constructureımıza parametre olarak pScore ve playerName özellikleri gönderiyoruz ve atama işleminin yapılması için super() metodunu kullanarak bu parametreleri Gamer classımızdaki parametrelili constructureımıza gönderiyoruz. Parametresiz bir Player constructureı tanımlıyoruz. Gamer classında tanımladığımız pickCard() metodunu burada override ediyoruz. pickCard() metodu "Test" classı tarafından gönderilen Playera ait 8 adet kart butonunu ve test adlı değişkeni kullanır. Diğer parametreler Computer classı için yazılmıştır. test değişkeni "Test" classında while döngüsünden önce tanımlanmış ve 0 eşitlenmiştir. Döngü her döndüğünde "Test" değişkeni bir artırılmıştır. Bu metotta gelen "Test" değerine göre hangi kartı seçebileceğimiz if koşullarıyla belirlenmiş ve seçilen karta ait değer "Test" classına return edilmiştir. Bu durumda ilk elde test==0 iken bütün kart butonları için yukarıda yarattığımız ActionListenerlara actionPerformed metodunu uyguluyoruz ve tıklanılan karta ait indis numarasını "chosenCard" değişkenine atıyoruz. Bu mantıkla döngü her döndüğünde test 1 artacağı için test'in çift olduğu durumlarda buz patenci kartlarına; testin tek olduğu durumlarda voleybolcu kartlarına tıklanabiliyor ve "Test" classına o kartın indis numarası return ediliyor. Yalnız test ilk başta 0 iken çift olduğundan sadece buz patenci seçmemesi için buz patencileri seçmek için kurulan if'e 0 olmaması şartını da koyduk. Seçilen kartlara tıklanabilmesi için de parametre olarak gönderdiğimiz buton kartlarına ActionListener ekledik. "Test" classındaki if içinde ilk elde seçilen kartın voleybolcu olması durumunda tekrar voleybolcu seçmemesi için test değişkenini 1 arttırdık. switch case yapısı ile seçilen kartın görünürlüğünü false edip return ettirilen değere ait Player kartının fotoğrafını orta karta getPic() ile koyduk ve o kartın görünürlüğünü true yaptık.

Test classımızda öncelikle
javax.imageio.ImageIO, javax.swing.*,
java.awt.*, java.awt.event.ActionEvent,
java.io.File, java.io.IOException, java.util.List,
java.util.Random, java.util.ArrayList

kütüphanelerini import ediyoruz. Daha sonra extends keywordünü kullanarak Test classımızı swing GUI için gerekli olan Thread classının subclassı haline getiriyoruz. Öncelikle private final olarak Computer tipinde p1, Player tipinde p2, Athlete tipinde list olan cards, ImageIcon tipinde backcard, JButton tipinde pcard1, pcard2, pcard3, pcard4, pcard5, pcard6, pcard7, pcard8, p2card1, p2card2, p2card3, p2card4, p2card5, p2card6, p2card7, p2card8, ocard1, ocard2, start, JLabel tipinde texts, texts1, p1texts, p2texts, endofgame, JTextField tipinde gameName, crtrName, JFrame tipinde frame özelliklerini tanımladık.

Parametrelili Test(JFrame aframe) constructureı tanımladık ve hataları bulabilmemiz için throws keywordünü kullanarak IOException yazdık. Bu constructureın içinde yukarıda tanımladığımız özelliklere atama yapıyoruz.

Kart butonlarımızı daha rahat oluşturmak için JButton tipinde createCardButton adlı bir metot oluşturuyoruz. Parametre olarak butonu ve bulunmasını istediğimiz x ve y değerlerini aldığımız bu metotda butonlarımıza setIcon() metodunu kullanarak backcard adlı fotoğrafımızı icon olarak ekliyor, setBounds() metodunu kullanarak butonlarımızın bulunması gereken x ve y değerlerini; olması gereken yükseklik ve genişliği belirliyor ve setVisible() metodunu kullanarak butonlarımızı görünmez kılıyoruz. Son olarak da butonumuzu return ediyoruz.

Ekranımızı oluşturmak için yazdığımız createScreen() adlı metotda texts, texts1, p1texts, p2texts, endofgame, gameName, crtrName özelliklerinin setBounds() ile bulunmaları gereken x ve y değerlerini; olmaları gereken yükseklik ve genişliği belirliyor, setHorizontalAlignment() ile hizalarını ayarlıyor, setFont ile yazı stilini ve boyutunu ayarlıyor, setForeground ile yazı rengini ve setBackground ile textfieldların arka plan rengini ayarlıyor aynı zamanda setEditable() metodu ile düzenleme yapılmasını engelliyoruz. Daha sonra bu özellikleri add() metodu ile frame'e ekliyoruz. Start butonun yazı rengini, fontunu, arkaplan

rengini, bulunması gereken yeri ve büyüklüğü ayarladıktan sonra addActionListener() metodu ile butona basıldığında startgame() metodunun çağırılarak oyunun başlatılmasını sağlıyoruz. start butonu ile beraber kart butonlarımızın hepsini add() metodu ile frame'e ekliyoruz. setsize() metodu ile frame boyutumuzu belirleyip, setLayout() metodu ile düzenimizi null yapıyoruz ve son olarak setVisible() metodu ile frame'imizi görünür kılıyoruz.

startgame() metodunda setVisible() metodu ile start butonunu, gameName ve crtrName text fieldlarını görünmez kılıyor defineCards() ve dealCards() metodlarını çağırıyoruz. Swing GUI'a ait start() metodu ile oyunumuz başlamış oluyor.

Swinge GUI'a ait projenin çalışmasını sağlayan run() metodunda öncelikle addPicsToCards() metodunu çağırıyoruz. Daha sonra rand, ctrl, tmp1,tmp2, tmp3, test, dnmf, dnmv, dnm1v, dnm1f özelliklerini tanımlıyoruz. Bütün butonlarımızı görünür hale getiriyoruz. dnmf listesine pcard1,pcard2,pcard3,pcard4; dnmv listesine pcard5,pcard6,pcard7,pcard8 butonlarını atıyoruz.dnm1f listesine 0,1, 2, 3 değerlerini dnm1v listesine ise 4, 5, 6, 7 değerlerini atıyoruz. ctrl<8 durumunda dönecek olan bir while döngüsü tanımlıyor ve daha sonra yukarıda anlattığımız oyunun ilerleyiş algoritmasını kodluyoruz. Oyun bitiminde texts ve texts1 textfieldlarımızı görünmez kılıyor ve puanı yüksek olan oyuncunun adının yanına wins! kelimesini ekleyip bunu ekrana yazdırarak oyunu bitiriyoruz.

addPicsToCards() metodunda setIcon() metodu ile Player'ın sahip olduğu kartlara(butonlara) fotoğraf eklemesi yapıyoruz.

defineCards() metodunda bütün figureSkater ve volleyPlayer nesnelerinin özelliklerine atama yapıyoruz ve bu nesneleri cards isimli listemize atıyoruz.

dealCards() metodunda rand, randomNum, randomNum2, min, max, min2, max2 card isimli değişkenleri tanımlıyoruz. for döngüsü açıyor ve

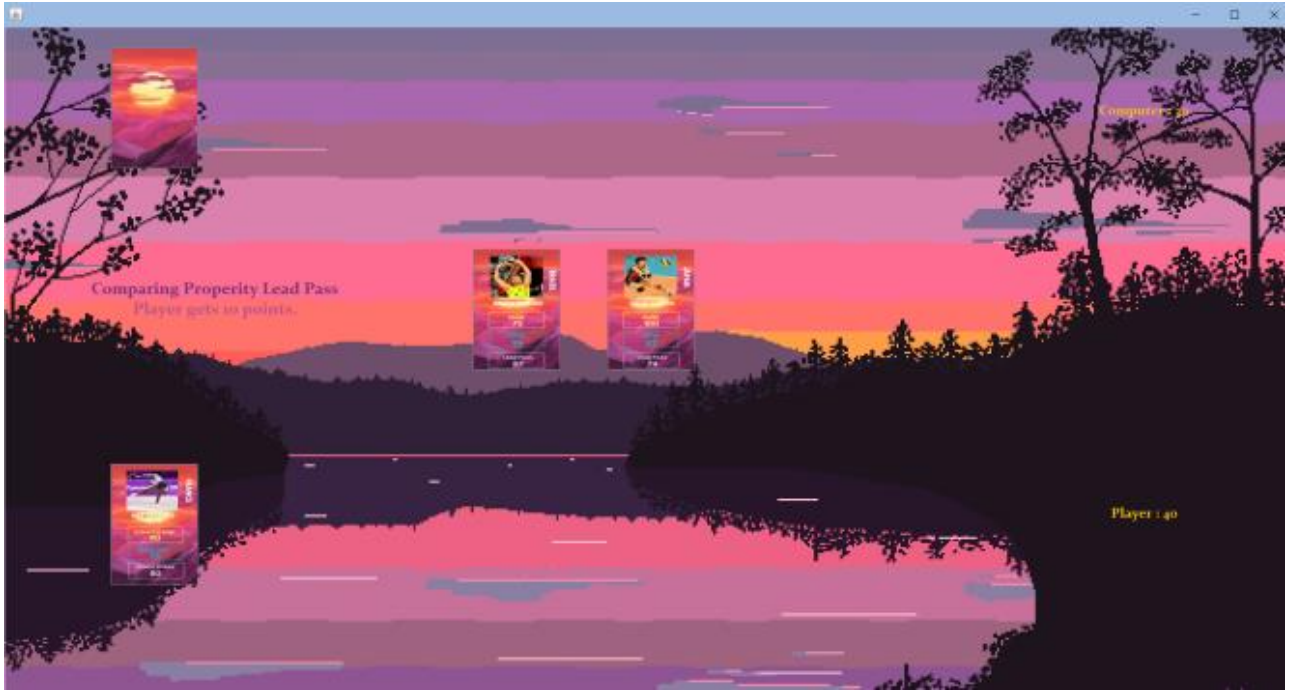
bu döngüyü 4 defa döndürerek Player oyuncusunun kart listesine rastgele 4 adet figureSkater nesnesi atıyoruz. Bir tane daha for döngüsü açıyor ve bu döngüyü de 4 defa döndürerek Player oyuncusunun kart listesine rastgele 4 adet volleyPlayer nesnesi atıyoruz. Son olarak 8 defa dönecek bir for döngüsü açıyor ve kalan figureSkater ve volleyPlayer nesnelerini Computer oyuncusunun kart listesine atıyoruz.

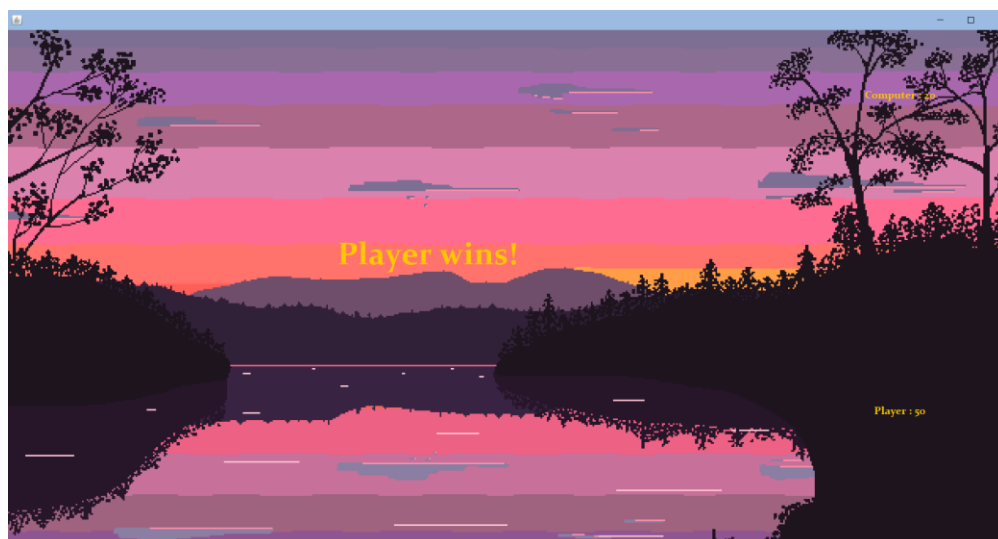
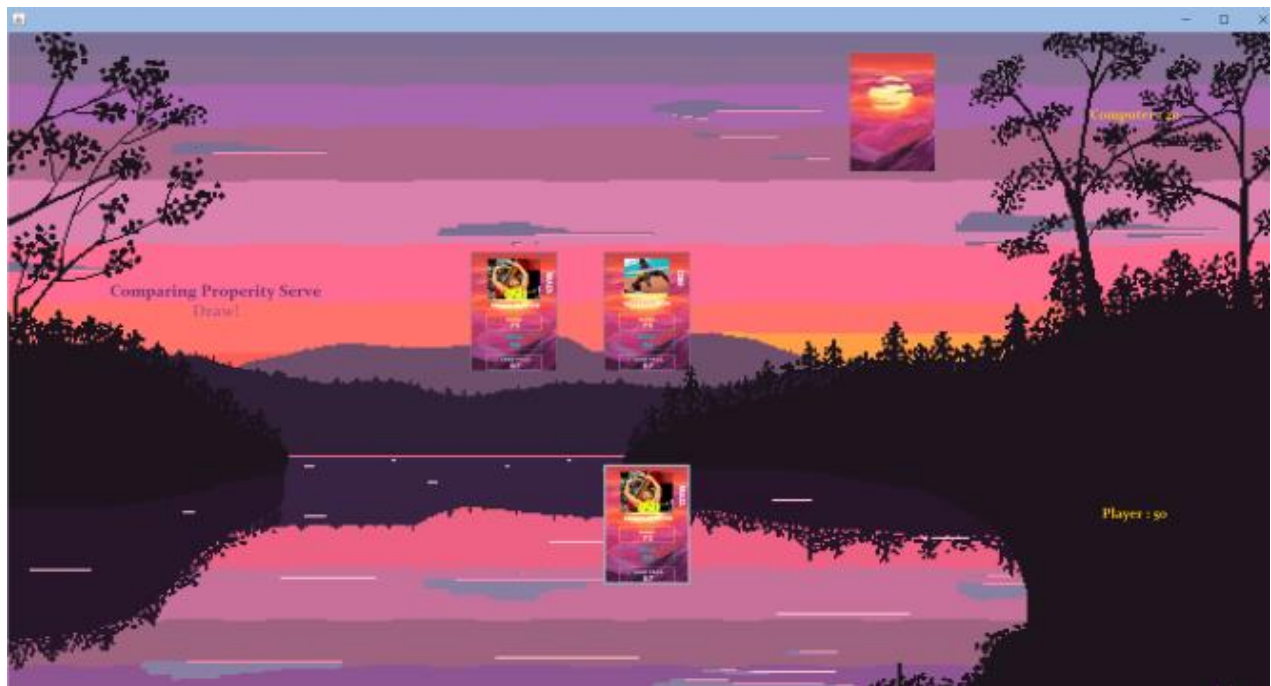
main() metodunu yazarken yine hataları bulması için throws keywordünü kullanarak java.io.IOException yazıyoruz. Bu metodun içinde frame adlı bir JFrame oluşturuyoruz. setDefaultCloseOperation() metodu ile ekranın sağ üstündeki çarpı işaretine basıldığında frame'in kapanmasını sağlıyoruz. setResizable() metodu ile yeniden boyutlandırılabilmesini sağlıyoruz. arkaplan adlı bir JLabel tanımlıyoruz ve oyunumuzun arkaplanını bu JLabel'a atıyoruz. setContentPane() metodunu kullanarak bu arka planı frame'in arka planı olarak ayarlıyoruz. Game adlı bir Test(frame) tanımlıyor ve son olarak game nesnesinde createScreen() metodunu çağırarak kodlamayı bitiriyoruz.

Sonuçlar

Kart oyunu eksiksiz bir şekilde çalışmaktadır fakat son hamlede kalan kartların tüm özellikleri eşit ise bütün özellikler kontrol edilene kadar kartlar oynamalıydı ve kontrol edilen özellikler bir daha kontrol edilmemeliydi. Bu projede son hamlede farklı bir özellik gelmesi durumunu yapamamış bulunmaktayız. Özellikler kontrol edilirken bir özellik birden fazla kontrol edilmektedir fakat tüm özellikler kontrol edildiğinde istenildiği gibi iki oyuncu da puan almadan oyun bitirilmekte ve kazanan tarafın adı ekranda yazmaktadır.

Çıktılar





Yalancı Kod

TEST CLASSI

- p1 Computer nesnesi oluştur.
- p2 Player nesnesi oluştur.
- backcard ImageIcon nesnesi oluştur.
- start,pcard1,pcard2,pcard3,pcard4,pcard5,pcard6,pcard7,pcard8,p2card1,p2card2,p2card3,p2card4,p2card5,p2card6,p2card7,p2card8,ocard1,ocard2 JButton nesnesi oluştur.
- gameName,crtrName JTextField nesnesi oluştur.
- texts,texts1,p1texts,p2texts,oyunSonu JLabel nesnesi oluştur.
- Gelen butonun iconunu backCard olarak ayarla.
- Gelen butonun bulunması istenilen x ve y değerlerine; olması istenen en ve boy oranlarına göre butonu ayarla.
- Butonu görünülmez kıl.
- Butonu geri döndür.
- texts nesnesinin fontunu ayarla.
- texts nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- texts nesnesinin rengini ayarla.
- texts1 nesnesinin fontunu ayarla.
- texts1 nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- texts1 nesnesinin rengini ayarla.
- p1texts nesnesinin fontunu ayarla.
- p1texts nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- p1texts nesnesinin rengini ayarla.
- p2texts nesnesinin fontunu ayarla.
- p2texts nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- p2texts nesnesinin rengini ayarla.
- gameName nesnesinin fontunu ayarla.
- gameName nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- gameName nesnesinin rengini ayarla.
- crtrName nesnesinin fontunu ayarla
- crtrName nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- crtrName nesnesinin rengini ayarla.
- Yukarıdaki nesneleri frame'e ekle.
- start nesnesinin fontunu ayarla
- start nesnesinin bulunması istenilen x ve y değerlerini; olması istenen en ve boy oranlarını ayarla.
- start nesnesinin yazı rengini ayarla.
- start nesnesinin arka plan rengini ayarla.
- start butonuna action listener ekle.
- Action listener'a gir ise kartlarigör O'a eşitle ve startgame metodu çağır.
- start nesnesi frame'e ekle.
- pcard1,pcard2,pcard3,pcard4,pcard5,pcard6,pcard7,pcard8,p2card1,p2card2,p2card3,p2card4,p2card5,p2card6,p2card7,p2card8,ocard1,ocard2 nesneleri createCardButton metoduna gönder,daha sonra frame'e ekle.
- Frame büyüklüğü ayarlanır,layout null olarak belirt ve frame görünür kıl.
- start,gameName,crtrGame nesneleri görünmez kıl.
- defineCards,dealCards ve start metodları çağır.
- addPicsToCards metodu çağır.
- rand,dnmv,dnmf,dnm1v,dnm1f nesneleri tanımla.
- ctrl,tmp1,tmp2,tmp3,test değerleri tanımlanır ve O'a eşitle.
- pcard1,pcard2,pcard3,pcard4,pcard5,pcard6,pcard7,pcard8,p2card1,p2card2,p2

card3,p2card4,p2card5,p2card6,p2card7,p2card8 nesneleri görünür kıl.

- pcard1,pcard2,pcard3,pcard4 nesneleri dnmf listesine ekle.
- pcard5,pcard6,pcard7,pcard8 nesneleri dnmv listesine ekle.
- 0,1,2,3 değerleri dnm1f; 4,5,6,7 değerleri dnm1v listelerine ekle.
- ctrl değişkeninin değeri 8'den küçükken döngü devam eder.
 - Ortadaki kart butonları görünmez kıl.
 - Oyuncuların adları ve skorları yazar.
 - ctrl 7'ye eşit ise test 0'a eşitle.
 - Player oyuncusu kart seçimi yapar.
 - Choosen cards are opening yazdır.
 - Eğer test 0'a eşit ise ve eğer Player oyuncusunun seçtiği kart indisi büyük eşit 4 ise test'i 1 artır.
 - Player oyuncusunun seçtiği kart indisinin değerine göre butonlardan biri görünmez yap ve ocard1 butonuna Icon ata ve görünür kıl.
 - test 1 artır.
 - Computer is choosing card yazdır.
 - secim değişkeni tanımlanır ve 0 ile 3 arasında rastgele bir değer alır.
 - Computer oyuncusu kart seçimi yapar
 - Eğer Player oyuncusunun seçtiği kart 4'dan küçük ise
 - Computer oyuncusunun seçtiği kart butonu görünmez olsun.

- ocard2 butonuna Icon ata ve nesne görünür kıl.

- Secim 0 ise

- Comparing Property Scratch Spin yazdırılır.
- Eğer p1 Spin büyüktür p2 Spin ise Computer gets 10 points yazdır ve Computer'a 10 puan ekle.
- Eğer p1 Spin küçüktür p2 Spin ise Player gets 10 points yazdırılır ve Player'a 10 puan ekle.
- Eğer p1 Spin eşittir p2 Spin ise seçilen kart butonları görünür kıl.
 - Ctrl 1 eksilt.
 - Draw! Yaz.

- Secim 1 ise

- Comparing Property Quad Axel yazdır.
- Eğer p1 Quad Axel büyüktür p2 Quad Axel ise Computer gets 10 points yazdır ve

- Computer'a 10 puan ekle.
- Eğer p1 Quad Axel küçüktür p2 Quad Axel ise Player gets 10 points yazdır ve Player'a 10 puan ekle.
- Eğer p1 Quad Axel eşittir p2 Quad Axel ise seçilen kart butonları görünür kıl.
 - Ctrl 1 eksilt.
 - Draw! Yaz.
- Secim 2 ise
 - Comparing Properity Death Spiral yaz.
 - Eğer p1 Death Spiral büyüktür p2 Death Spiral ise Computer gets 10 points yazdır ve Computer'a 10 puan ekle.
 - Eğer p1 Death Spiral küçüktür p2 Death Spiral ise Player gets 10 points yazdır ve Player'a 10 puan ekle.
 - Eğer p1 Death Spiral eşittir p2 Death Spiral ise seçilen kart butonları görünür kıl.
 - Ctrl 1 eksilt.
 - Draw! Yaz.
- butonları görünür kıl.
 - Ctrl 1 eksilt.
 - Draw! Yaz.
- Eğer Player oyuncusunun seçtiği kart 4'dan büyük ise
 - Computer oyuncusunun seçtiği kart butonu görünmez olsun.
 - ocad2 butonuna Icon atanır ve nesne görünür kıl.
 - Secim 0 ise
 - Comparing Properity Dunk yazdır.
 - Eğer p1 Dunk büyüktür p2 Dunk ise Computer gets 10 points yazdırılır ve Computer'a 10 puan ekle.
 - Eğer p1 Dunk küçüktür p2 Dunk ise Player gets 10 points yazdırılır ve Player'a 10 puan ekle.
 - Eğer p1 Dunk eşittir p2 Dunk ise seçilen kart butonları görünür kılınır.
 - Ctrl 1 eksilt.

- Draw!
Yaz.
 - Secim 1 ise
 - Comparing
Properity Serve
yaz.
 - Eğer p1 Serve
büyüktür p2
Serve ise
Computer gets
10 points
yazdırılır ve
Computer'a 10
puan ekle.
 - Eğer p1 Serve
küçüktür p2
Serve ise Player
gets 10 points
yazdırılır ve
Player'a 10
puan ekle.
 - Eğer p1 Serve
eşittir p2 Serve
ise seçilen kart
butonları
görünür kıl.
 - Ctrl 1
eksilt.
 - Draw!
Yaz.
 - Secim 2 ise
 - Comparing
Properity Lead
Pass yazdırılır.
 - Eğer p1 Lead
Pass büyüktür
p2 Lead Pass ise
Computer gets
10 points
yazdırılır ve
Computer'a 10
puan ekle.
 - Eğer p1 Lead
Pass küçüktür
p2 Lead Pass ise
Player gets 10
points yazdırılır
ve Player'a 10
puan ekle.
 - Eğer p1 Lead
Pass eşittir p2
Lead Pass ise
seçilen kart
butonları
görünür kılınır.
 - Ctrl 1
eksilt.
 - Draw! Yaz.
 - texts ve texts1 görünmez kıl.
 - Eğer Computer'ın skoru büyüktür
Player'ın skoru ise orta kartları
görünmez kıl ve Computer Wins! yaz.
 - Eğer Computer'ın skoru küçüktür
Player'ın skoru ise orta kartları
görünmez kıl ve Player Wins! yaz.
 - Eğer Computer'ın skoru eşittir Player'ın
skoru ise orta kartları görünmez kıl ve
Draw! yazdır.
 - addPicsToCards() metodu ile Player'ın
kartlarına resim ata.
 - defineCards() metodu ile kart
nesnelerinin özelliklerini ata ve kartları
listeye ekle.
 - dealCards() metodu ile Player'ın ve
Computer'ın kart listesine kart ata.
 - psvm() 'de frame'i oluştur ve arka plan
resmi koy.
 - Oyunu başlat.
- ATHLETE CLASSI
- Takım adı, sporcu adı,
prob1,prob2,prob3 özelliklerine atama
yapınması sağlanır.

- `getAthleteName()`, `getAthleteTeam()` metodları ile `athleteName` ve `athleteTeam` return edilir.
- `setAthleteName()`, `setAthleteTeam()` metodları ile `athleteName`'e ve `athleteTeam`'e atama yapılır.
- `getpic()` methodu ile `pic` özelliği return edilir.

FIGURESKATER CLASSI

- `super()` ile figure skater takımı, figure skater adı, `spin`, `deathSpiral`, `quadAxel` özelliklerine atama yapınması sağlanır.
- `getSpin()`, `getQuadAxel()`, `getDeathSpiral()` metodları ile `spin`, `deathSpiral`, `quadAxel` return edilir.
- `setSpin()`, `setQuadAxel()`, `setDeathSpiral()` metodları ile `spin`'e, `deathSpiral`'a, `quadAxel`'e atama yapılır.
- `showAthleteScore()` methodu `spin`, `deathSpiral`, `quadAxel` yazdırılır.
- `getIsCardUsed()` methodu ile `isCardUsed` return edilir.
- `setIsCardUsed()` methodu ile `isCardUsed`'a atama yapılır.

VOLLEYPLAYER CLASSI

- `super()` ile volleyPlayer takımı, volleyPlayer adı, `dunk`, `serve`, `leadpass` özelliklerine atama yapınması sağlanır.
- `getDunk()`, `getServe()`, `getLeadPass()` metodları ile `dunk`, `serve`, `leadpass` return edilir.
- `setDunk()`, `setServe()`, `setLeadPass()` metodları ile `dunk`'a, `serve`'e, `leadpass`'e atama yapılır.
- `showAthleteScore()` methodu `dunk`, `serve`, `leadpass` yazdırılır.
- `getIsCardUsed()` methodu ile `isCardUsed` return edilir.

- `setIsCardUsed()` methodu ile `isCardUsed`'a atama yapılır.

GAMER CLASSI

- `gamerName`, `score` özelliklerine atama yapınması sağlanır.
- `Gamer()` methodu ile skor 0'a eşitlenir.
- `addPoint()` methodu ile skor 10 artırılır.
- `showScore()` methodu ile `score` özelliği return edilir.

COMPUTER CLASSI

- `super()` ile `ComputerName`, `cScore` özelliklerine atama yapınması sağlanır.
- `pickCard()` methodu ile 0 ile 4 arasında rastgele bir değer seçip gönderilir.

PLAYER CLASSI

- `super()` ile `PlayerName`, `pScore` özelliklerine atama yapınması sağlanır.
- `pickCard()` methodu ile 0 ile 8 arasında rastgele bir değer seçip gönderilir.

Kaynakça:

<https://www.youtube.com/watch?v=QW3iL3zFJSU>

https://www.youtube.com/watch?v=M14D7q8UoCE&list=PLu_zq6omCvuQ_ZoKnE8-CE2nF113p9pxd

<https://elifyonel.wordpress.com/2019/06/14/java-egitimi-40-java-swing/>

<https://medium.com/gokhanyavas/javada-i%CC%87stisnalar-f7ec3838a623>

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

<https://e-bergi.com/y/swing-ile-arayuz-tasarimi/>

<https://www.tutorialspoint.com/java-program-to-set-horizontal-alignment-of-content-in-a-jtextfield>

<https://netbeans.org/kb/docs/java/gui-image-display.html>

<https://www.tutorialspoint.com/java-program-to-set-horizontal-alignment-of-content-in-a-jtextfield>