NESNEYE YÖNELİK

TIC TAC TOE

160201079 İlknur GÖK,

160201012 Elanur OCAK

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[ilknurgok.1@hotmail.com](mailto:ilknurgok.1@hotmail.com) , [elaocak45@gmail.com](mailto:elaocak45@gmail.com)

**Özet**

Kullanıcıdan boyut, kimin başlayacağı ve harf bilgileri alındı. Alınan bilgilere göre bir oyun tahtası oluşturuldu. Sırayla bilgisayar ve kullanıcıdan harf değeri istendi. Alınan değerler her seferinde belli kontrollerden geçti ve kimin kazandığı belirlendi. Oyunun sonunda kayıt edilip edilmeyeceği bilgisi alındı. Diğer oyunun başında isteğe göre yarım kalan oyundan başlatıldı.

**1.Giriş**

Projenin konusu nesneye yönelik programlama mantığını kullanarak proje geliştirmeyi sağlamaktır.

Projenin amacı ise nesneye yönelik programlama mantığını kullanarak Tic Tac Toe oyununu geliştirmektir.

**2.Temel Bilgiler**

Proje Java dilinde yazıldı ve NetBeans’te hazırlandı.

* **NetBeans**, Oracle tarafından geliştirilen bir [Java](http://www.wikizero.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvSmF2YV8oeWF6JUM0JUIxbCVDNCVCMW1fb3J0YW0lQzQlQjEp) geliştirme ortamıdır (IDE) ve ücretsiz olarak dağıtılmaktadır. Özellikle kullanıcı arayüzü tasarımında sağladığı kolaylıklardan dolayı tercih edilmektedir.

**3.Diğer Bölümler**

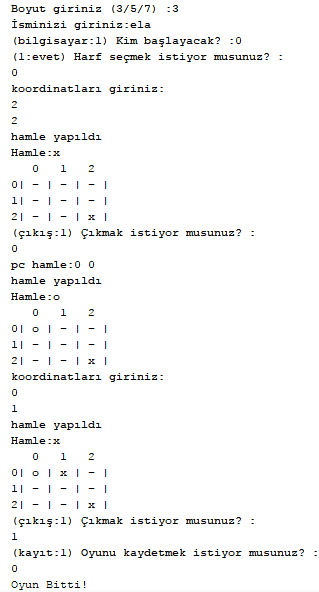
Hazırladığımız projede ilk olarak oynanılan oyunu dosyaya kaydetme gibi bir işlem yapıldığından, kullanıcı oyuna başlar başlamaz dosyaya kayıtlı olan oyundan oyuna başlamak isteyip istemediğini sorguladık. Verilen cevaba göre oyun ya dosyaya önceden kaydedilen oyundan ya da yeni bir oyundan başlanmasını sağladık.

*3.1.Yeni Oyun Oynanması*

Kullanıcıdan alınan cevaba göre oyununun oynanmasını sağlamak için kullanıcıdan ilk olarak oyun oynanacak tahtanın boyutunu kullanıcıdan istedik. Sonrasında kullanıcıdan isim bilgisini aldık. Daha sonra oyuna kullanıcının mı yoksa bilgisayarın mı başlanmasını istenmesini sorguladık.

*3.1.a. Kullanıcının Oyuna Başlaması*

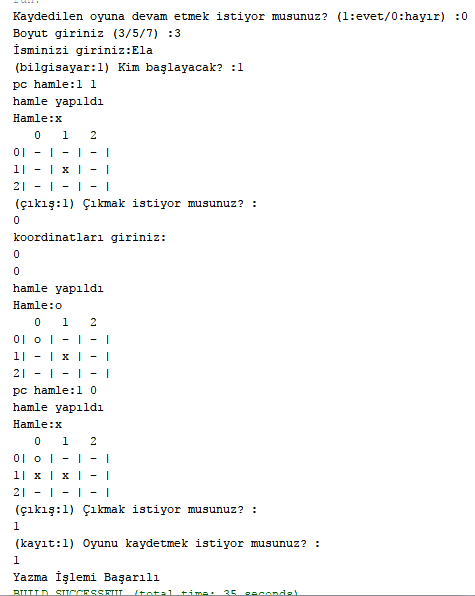
Oyuna ilk olarak kullanıcının başlamak istemesi üzerine ilk olarak kullanıcıya ait olarak kullanıcı tipini belirlememizde yardımcı olmasında kullanmak üzere boolean first değerini true olarak atadık. (false olduğunda kullanıcı tipini bilgisayar olarak kabul ettik.) Sonrasında kullanıcıya oyunda kullanmak üzere harf seçip seçmek  isteyip istemediğini sorduk. Eğer kullanıcı harf seçmek istemez ise, ilk olarak Oyuncu class'ı için nesne oluşturduk. Bunu yaparken Oyuncu class'ında oluşturduğumuz default constructor yapısı ile kullanıcının oyunda kullanacağı harfi 'x' olarak belirlemiş olduk. Sonrasında OyunTahtasi nesnesini oluşturduk. Bunu yaparken OyunTahtasi class'ında oluşturduğumuz default constructor yapısı ile oynanılacak oyun tahtasının boş olarak oluşturulmasını sağladık. Daha sonrasında kullanıcıdan aldığımız boyutu OyunTahtasi içinde tanımladığımız oyunTahtasiniAl() method'una gönderdik ve bu method ile (boyut) x (boyut) 'luk oyun tahtasını oluşturduk. Sonrasında oyun tahtasına hamle yapmak için kullandığımız döngüye girdik. Döngüde ilk olarak kullanıcıdan hamle yapmak istediği herhangi bir koordinat değerini Oyuncu method'u olan oyuncununHamlesiniAl() ile istedik. Alınan koordinat değerlerinin geçerli olup olmadığını OyunTahtasi method'u olan hamleyiYaz() ile kontrol ettirdik.Koordinatlar doğru olmadığında (girilen koordinat değeri dolu olduğunda veya girilen boyut değerinden fazla olduğunda) kullanıcıdan tekrar koordinat girmesini istedik. Kullanıcıdan hamle yapılabilecek koordinat alındığında alınan koordinat değerlerine kullanıcının hamlesini atadık. Sonrasında hamle yapılmış olan oyun tahtasını ekrana yazdırdık. Kullanıcı hamle yaptığı için kazanıp kazanmadığını OyunTahtasi method'u olan kazanan() ile kontrol ettirdik. Kazanma durumu olduğunda ise kullanıcının kazandığını ekrana yazdırıp programı kapattık. Eğer kazanma olmaz ise sonrasında beraberlik olup olmadığını OyunTahtasi method'u olan beraberlikKontrol() ile kontrol ettirdik. Beraberlik olduğu durumunda oyunun berabere bittiğini ekrana yazdırıp programı kapattık. Eğer beraberlik de olmaz ise kullanıcıya oyundan çıkış yapmak isteyip istemediği soruldu. Çıkış yapmak istendiğinde ise oynanılan oyunun dosyaya kaydedilmesinin isteyip istenmediği soruldu. Kayıt yapılması istendiği zaman kullanıcıya ait bilgiler (isim, kullandığı harf) ve oyun tahtasını dosyaya kaydettik. Ve sonrasında programın kapanmasını sağladık. Eğer kullanıcı çıkış yapmak istemez ise kullanıcımıza ait oyunun en başında belirlemiş olduğumuz boolean first verisini bilgisayar hamle yapacağı için değiştirdik. Bilgisayar hamle yaptığında kullanacağı harfi ise Oyuncu method'u olan karakteriAl() ile değiştirdik. Ve tekrar while döngüsüne bilgisayarın hamle yapmasını sağlamak için girdik. Bilgisayar için döngüde rastgele oluşturduğu koordinat değerlerine hamle yapılmasını sağladıktan sonra ise yine kazanan olup olmadığını, olmaz ise beraberlik durumunu kontrol ettirdik. Sonrasında döngüye girmeden önce yine first ve harf değerlerini değiştirdik. Oluşturduğumuz bu döngünün herhangi bir kazanma veya beraberlik durumundan sonra bitmesini ve programı kapatmasını sağladık. Eğer kullanıcı harf seçmek ister ise, ilk olarak kullanıcıdan istediği harfi girmesini istedik. Sonrasında Oyuncu class'ı için nesne oluşturduk. Bunu yaparken Oyuncu class'ında oluşturduğumuz constructor yapısı ile kullanıcının oyunda kullanacağı harfi, girmiş olduğu harf olarak belirlemiş olduk. Sonrasında OyunTahtasi nesnesini oluşturduk. Bunu yaparken OyunTahtasi class'ında oluşturduğumuz default constructor yapısı ile oynanılacak oyun tahtasının boş olarak oluşturulmasını sağladık. Daha sonrasında kullanıcıdan aldığımız boyutu OyunTahtasi içinde tanımladığımız oyunTahtasiniAl() method'una gönderdik ve bu method ile (boyut) x (boyut) 'luk oyun tahtasını oluşturduk. Sonrasında oyun tahtasına hamle yapmak için kullandığımız döngüye girdik. Döngüde ilk olarak kullanıcıdan hamle yapmak istediği herhangi bir koordinat değerini Oyuncu method'u olan oyuncununHamlesiniAl() ile istedik. Alınan koordinat değerlerinin geçerli olup olmadığını OyunTahtasi method'u olan hamleyiYaz() ile kontrol ettirdik. Koordinatlar doğru olmadığında(girilen koordinat değeri dolu olduğunda veya girilen boyut değerinden fazla olduğunda) kullanıcıdan tekrar koordinat girmesini istedik. Kullanıcıdan hamle yapılabilecek koordinat alındığında alınan koordinat değerlerine kullanıcının hamlesini atadık. Sonrasında hamle yapılmış olan oyun tahtasını ekrana yazdırdık. Kullanıcı hamle yaptığı için kazanıp kazanmadığını OyunTahtasi method'u olan kazanan() ile kontrol ettirdik. Kazanma durumu olduğunda ise kullanıcının kazandığını ekrana yazdırıp programı kapattık. Eğer kazanma olmaz ise sonrasında beraberlik olup olmadığını OyunTahtasi method'u olan beraberlikKontrol() ile kontrol ettirdik. Beraberlik olduğu durumunda oyunun berabere bittiğini ekrana yazdırıp programı kapattık. Eğer beraberlik de olmaz ise kullanıcıya oyundan çıkış yapmak isteyip istemediği soruldu. Çıkış yapmak istendiğinde ise oynanılan oyunun dosyaya kaydedilmesinin isteyip istenmediği soruldu. Kayıt yapılması istendiği zaman kullanıcıya ait bilgiler (isim, kullandığı harf) ve oyun tahtasini dosyaya kaydettik. Ve sonrasında programın kapanmasını sağladık. Eğer kullanıcı çıkış yapmak istemez ise kullanıcımıza ait oyunun en başında belirlemiş olduğumuz boolean first verisini bilgisayar hamle yapacağı için değiştirdik. Bilgisayar hamle yaptığında kullanacağı harfi ise Oyuncu method'u olan karakteriAl() ile değiştirdik. Ve tekrar while döngüsüne bilgisayarın hamle yapmasını sağlamak için girdik. Bilgisayar için döngüde rastgele oluşturduğu koordinat değerlerine hamle yapılmasını sağladıktan sonra ise yine kazanan olup olmadığını, olmaz ise beraberlik durumunu kontrol ettirdik. Sonrasında döngüye girmeden önce yine first ve harf değerlerini değiştirdik. Oluşturduğumuz bu döngünün herhangi bir kazanma veya beraberlik durumundan sonra bitmesini ve programı kapatmasını sağladık.



Şekil 1: Örnek Çıktı 1

*3.1.b. Bilgisayarın Oyuna Başlaması*

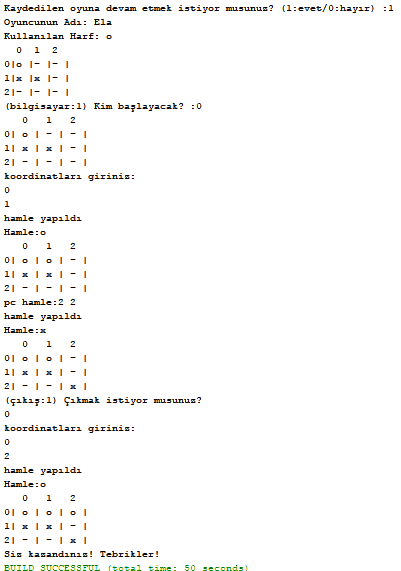
Oyuna ilk olarak kullanıcının başlamak istememesi üzerine ilk olarak bilgisayara ait olarak kullanıcı tipini belirlememiz de kullanmak üzere boolean first değerini false olarak atadık. (true olduğunda kullanıcı tipini kullanıcı olarak kabul ettik.) Sonrasında Oyuncu class'ı için nesne oluşturduk. Bunu yaparken Oyuncu class'ında oluşturduğumuz constructor yapısı ile bilgisayarın oyunda kullanacağı harfi 'o' olarak belirlemiş olduk. Sonrasında OyunTahtasi nesnesini oluşturduk. Bunu yaparken OyunTahtasi class'ında oluşturduğumuz default constructor yapısı ile oynanılacak oyun tahtasının boş olarak oluşturulmasını sağladık. Daha sonrasında kullanıcıdan aldığımız boyutu OyunTahtasi içinde tanımladığımız oyunTahtasiniAl() method'una gönderdik ve bu method ile (boyut) x (boyut) 'luk oyun tahtasını oluşturduk. Sonrasında oyun tahtasına hamle yapmak için kullandığımız döngüye girdik. Bilgisayar için döngüde rastgele oluşturduğu koordinat değerlerine hamle yapılmasını sağladıktan sonra ise yine kazanan olup olmadığını, olmaz ise beraberlik durumunu kontrol ettirdik. Sonrasında döngüye girmeden önce yine first ve harf değerlerini değiştirdik. Daha sonra kullanıcı için döngünün çalışmasını sağladık. Oluşturduğumuz bu döngünün herhangi bir kazanma veya beraberlik durumundan sonra bitmesini ve programı kapatmasını sağladık.



Şekil 2: Örnek Çıktı 2

*3.2.Kayıtlı Oyunun Oynanması*

Kullanıcıdan alınan cevaba göre önceden dosyaya kaydettiğimiz oyunun oynanması istendiği için öncelikle dosyadan okuma yapılmasını sağladık. Dosyayı okuyarak önceki oyundaki oyun tahtasını geri almış olduk. Sonrasında ekrana bu matrisi yazdırdık. Daha sonra oyuna kullanıcının mı yoksa bilgisayarın mı başlanmasını istenmesini sorguladık.  Oyuna ilk olarak kullanıcının başlamak istemesi üzerine ilk olarak kullanıcıya ait olarak kullanıcı tipini belirlememiz de kullanmak üzere boolean first değerini true olarak atadık.  Bilgisayarın başlanması istendiğinde ise first değerini false olarak atadık. Sonrasında ise ilk olarak Oyuncu class'ı için nesne oluşturduk. Bunu yaptıktan sonra dosyadan çektiğimiz isim ve  harfi kullanıcının özellikleri olarak atadık. Sonrasında OyunTahtasi nesnesini oluşturduk. Bunu yaparken OyunTahtasi class'ında oluşturduğumuz constructor yapısına dosyadan çektiğimiz oyun tahtasını göndererek oynanılacak oyun tahtası olarak oluşturulmasını sağladık. Sonrasında ise oyun tahtasına hamle yapmak için kullandığımız döngüye girdik. Döngüde ilk olarak kullanıcıdan veya bilgisayardan hamle yapmak istediği herhangi bir koordinat değerini Oyuncu method'u olan oyuncununHamlesiniAl() ile oluşturduk. Alınan koordinat değerlerinin geçerli olup olmadığını OyunTahtasi method'u olan hamleyiYaz() ile kontrol ettirdik. Koordinatlar doğru olmadığında (girilen koordinat değeri dolu olduğunda veya girilen boyut değerinden fazla olduğunda) kullanıcının tekrar koordinat girmesini veya bilgisayar için ise yeniden rastgele koordinat değerleri oluşturmasını sağladık. Kullanıcıdan veya bilgisayardan hamle yapılabilecek koordinat alındığında alınan koordinat değerlerine kullanıcı veya bilgisayar hamlesini atadık. Sonrasında hamle yapılmış olan oyun tahtasını ekrana yazdırdık. Kullanıcı hamle yaptığı için kazanıp kazanmadığını OyunTahtasi method'u olan kazanan() ile kontrol ettirdik. Kazanma durumu olduğunda ise hangi kullanıcı tipinin kazandığını ekrana yazdırıp programı kapattık. Eğer kazanma olmaz ise sonrasında beraberlik olup olmadığını OyunTahtasi method'u olan beraberlikKontrol() ile kontrol ettirdik. Beraberlik olduğu durumunda oyunun berabere bittiğini ekrana yazdırıp programı kapattık. Eğer beraberlik de olmaz ise kullanıcı için oyundan çıkış yapmak isteyip istemediğini sorgulattık. Çıkış yapmak istendiğinde ise oynanılan oyunun dosyaya kaydedilmesinin isteyip istenmediğini sorgulattık. Kayıt yapılması istendiği zaman kullanıcıya ait bilgiler (isim, kullandığı harf) ve oyun tahtasini dosyaya kaydettik. Ve sonrasında programın kapanmasını sağladık. Eğer kullanıcı çıkış yapmak istemez ise kullanıcımıza ait oyunun en başında belirlemiş olduğumuz boolean first verisini bilgisayar hamle yapacağı için değiştirdik. Oluşturduğumuz bu döngünün herhangi bir kazanma veya beraberlik durumundan sonra bitmesini ve programı kapatmasını sağladık.



Şekil 3: Örnek Çıktı 3

**4. Sonuçlar**

Nesneye yönelik programlama mantığını kullanarak classlar açıp bu classlara ait nesneler oluşturduk.

Nesneye yönelik programlamanın 3 temel ilkelerinden biri olan kalıtım konusunu kod üzerinde uyguladık.

Java dilinde dosya açıp dosya üzerine veri yazdırdık ve zaten var olan bir dosyayı okumayı gerçekleştirdik.

**5.Kaynakça**

[1]. Web Site

<https://codereview.stackexchange.com/questions/95876/simple-tic-tac-toe-game>

[2]. Web Site

http://fav10.net/forum/threads/javada-sat%C4%B1r-sat%C4%B1r-dosya-okuma-ve-yazma.218737/

[3]. Web Site

<https://stackoverflow.com/questions/10766801/unique-random-number-generation>

[4]. Web Site

<https://codereview.stackexchange.com/questions/153932/tic-tac-toe-game-in-java-vs-computer>

[5]. Web Site

http://blog.emrahkahraman.com.tr/java-dosya-okuma/

**6.Akış Şeması**

