**Nesneye Yönelik Tic Tac Toe**

[**Selda Yapal**](mailto:seldayapal@anadolu.edu.tr)

[**Nedret Gegeoğlu**](mailto:nedretgegeoglu@windowslive.com)

**Kocaeli Üniversitesi**

**Bilgisayar Mühendisliği 2017 – 2018**

1. **Problem Tanımı:**

Bizden istenen nesneye yönelik mantıkta bilgisayara karşı oynanabilen bir “tic tac toe” oyunu hazırlamaktı. Nesneye dayalı mantık ile oyun, oyuncu sınıfları oluşturularak proje geliştirildi.

1. **Yapılan Araştırmalar:**

İlk olarak araştırmamız gereken C# dili üzerinde nesneye dayalı programlama temelleriydi. Burada kullanmamız gereken sınıflar, yapıcı metotlar vb. durumlar söz konusuydu.

Bizden bağlantılı olarak üç tane sınıf kullanmamız bekleniyordu. Bu sınıfların bağlantısını yapıp her bir sınıfın nesnesini oluşturmaya çalıştık ve birbiriyle etkileşim halinde olması için her sınıfta kullanılan metotlar üzerinde erişim sağladık.

Bir diğer araştırdığımız konu ise “tic tac toe” oyununda ki kazanma mantığıydı. Oyunda üç adet kazanma mantığı bulunuyor; Dikey, yatay ve çapraz kazanma. Bunları tespit etme kısmını tasarlayıp önemli bir sorunu halletmiş olduk.

Ayrıca bizden oyun esnasında oyundan çıkıp oyun tahtasının mevcut durumunun kaydedilip daha sonra devam edilebilmesi istendi. Bunun için de C# dili üzerinde **dosya işlemlerini** araştırıp kullandık.

1. **Tasarım:**

**3.1.Akış Şeması:**

2.Oyun kaydedilmez.

1.Oyun kaydedilir.

9.Oyunu kaydedip kaydetmek için 1 / kaydetmeyip çıkmak için 2 girmesi istenir

Kullanıcı; satır ve sütun girerek hamlesini gerçekleştirir

9’a basarak oyundan çıkabilir.

Oyunu iki taraftan biri kazanana yada berabere kalana dek döngü devam eder

2. Kayıtlı Oyun bilgileri alınır

1. Kullanıcı bilgileri alınır

Bilgisayar; satır ve sütun girerek hamlesini gerçekleştirir.

**3.2.Yazılım Mimarisi:**

Projeyi C# dilinde oluşturduk. Oyun tahtasını oluşturmak ve düzenlemek, bununla beraber oyun üzerinde ki kontrolleri yapmak için **Oyun** sınıfı, oyuncu verilerinin oluşturulup düzenlendiği **Oyuncu** sınıfını oluşturduk. Bununla beraber fonksiyonda birçok metot kullanıldı:

Main class metotları:

* Oyna();

Oyuncu sınıfı metotları:

* Karakteri\_al();
* Oyuncu\_turunu\_al();
* İnsan\_oyuncu\_hamlesi\_kontrol();
* Bilgisayar\_hamlesi\_üret();

Oyun sınıfı metotları:

* Oyun\_turu();
* Boyut();
* Oyun\_tahtası();
* Hamle\_ekle();
* Oyun\_tahtasını\_yazdır();
* Hamleyi\_Yaz();
* Hamleyi\_Yaz\_cpu();
* Beraberlik();
* Kazanan();
* Çıkış();
* Kayıtlı\_oyun();
* Yeni\_oyun();

1. **Genel Yapı:**

Projede kullanılan metotlar ve amaçları:

* **Karakteri al();** Oyuncunun girdiği harfi aktarmaya yarayan fonksiyon.
* **OyuncuTurunuAl**(); Karakterin türünü; insan/makine alır ve bool olarak döndürür.
* **insanOyuncuHamlesiKontrol();** Karakterin yaptığı satır ve sütuna dayalı hamleyi alır ve stringe dönüştürüp return eder. Örneğin kullanıcı 3e 2lik bir hamle girdiğinde metot bu hamleyi 32 şeklinde birleştirerek geriye döndürür (return eder).
* **BilgisayarHamlesiUret();** Bilgisayar üzerinden tahta boyutları içerisinde rastgele bir hamle üretir ve geriye döndürür.
* **oyunTuru();** Kullanıcının oyun türünü girmesini sağlayan metot. 1. Yeni oyun / 2. Kayıtlı oyun açar ve sonucu geriye döndürür.
* **Boyut();** Kullanıcının girdiği oyun türüne göre; eğer yeni oyun girmişse kullanıcıdan boyut alır, kayıtlı oyun girmişse, kaydedilen txt dosyası üzerinden boyutu alır ve geriye döndürür.
* **oyunTahtasi();** Yeni bir oyun tahtası oluşturur.
* **hamleEkle();** Kullanıcının girdiği hamleyi ekler.
* **OyunTahtasiniYazdir();** ilk olarak kullanıcının ya da makinenin girdiği hamleyi kontrol ederek oyunu bir kazananın olduğunu kontrol eder. Eğer bir kazanan varsa onu ekrana yazdırır. Bir kazanan yoksa ve oyun berabere sonlanmışsa beraberliği ekrana yazdırır. Bunun altına oyun tahtasının son halini yazdırır.
* **HamleyiYaz();** Kullanıcının girdiği hamleyi **oyuncu** sınıfından **insanOyuncuHamlesiKontrol**(); metodundan alır ve aldığı hamleyi satır ve sütun olarak düzenler. Eğer hamle yapılan kare boş ise hamle yapılır. Dolu kareye hamle yapılmaz tekrar hamle alır.
* **Beraberlik();** Tahta üzerinde boş karenin kalıp kalmadığına bakarak oyunun berabere bitip bitmediğini kontrol eden metottur.
* **Kazanan();** Kullanıcının oyunu 4 farklı açıdan kazanıp kazanmadığını kontrol eden metot. Bunlar: Dikey düzlemde kullanıcın girdiği sütunla beraber tüm satırları kontrol ederek her satırın aynı karakteri taşıdığını kontrol eder.Eğer taşıyorsa kazanan değeri true olarak döner. Yatay düzlemde kullanıcının girdiği satıra bakarak tüm sütunları kontrol ederek oluşturulan bool değişkeni true yada false döner. Sol üstten sağ alta giden çapraz dizilimde tüm kutular kontrol edilir ve hepsi kullanıcının girdiği karakter ise değişken geriye true olarak döner. Sağ üstten sol alta giden çapraz dizilimde tüm kutular kontrol edilir ve hepsi kullanıcının girdiği karakter ise değişken geriye true olarak döner.
* **Çıkış();** İlk olarak kullanıcıya oyunu kaydetmek isteyip istemediği sorulur. Eğer oyun kaydedilecekse oyuncunun ve makinenin bilgileri oluşturulan “kayit.txt” dosyasına aktarılır. Oyun tahtasının son durumu **X,O** ve boşlar için **B** olarak kayıt sıralı düzende aktarılır. Daha sonra oyundan çıkılır.
* **KayıtlıOyun();** Oyun klasörü içerisinde oluşturulmuş “kayit.txt” dosyasından ilk satırda tahta boyutunu okur ve altındaki iki satırda kullanıcıların bilgilerini aktarır. Buraya kadar okuduktan sonra tahtanın dizilişi şeklinde **X,O,B,**  karakterlerini okur ve bilgilerle beraber oyun tahtasını ekrana yazdırır.
* **YeniOyun();** Boş oyun tahtası oluşturur ve kullanıcıdan oyun bilgilerini girmesini ister. Boş oyun tahtasını **oyunTahtasiniYazdir();** metodu ile ekrana yazdırır.
* **Oyna();** Main class üzerinde bulunan tek metot olan oyna metodu oluşturulan metotları kullanarak oyunun oynanma dinamiğini sağlar. Sırasıyla:
  + Oyuncunun oyun türü seçmesini ister.
  + GameOver döngüsüne girer ve oyun sonlanana dek veya kullanıcı çıkana dek bu döngüde kalmaya devam eder.
  + Döngü içerisinde ki bir diğer kritik döngü sıradaki kullanıcının hamle yapmasını sağlayan hamle yapana dek dönen döngüdür. Kullanıcı yada makine hamle yaptığında bu hamle karşılığında oyunu kazanıp kazanmadığını kontrol eden ve buna göre oyunu sonlandıran eğer kazanmadıysa ve yaptığı hamle geçerliyse döngüden çıkılır ve sıra diğer oyuncuya geçer.
  + Son olarak bu düzenlemelerle her el oyun tahtası ekrana yansıtılır.

1. **Referanslar:**

[1.] **C# Dosya işlemleri:** <https://stackoverflow.com/questions/9740557/reading-a-text-file-word-by-word>

[2.] **C# String İşlemleri:** <http://yasnsahn.com/?p=476>