**KART OYUNU UYGULAMASI**

**Özet**

Proje bir oyuncun başka bir oyuncuya rekabet edebileceği nesneye yönelik programlama olarak yazılmıştır. Projede sınıf kavramları ve alt sınıf kavramı kullanarak bunların nasıl kullanılacağı ile ilgili bilgiler verilmiştir. Nesneye yönelik programlama da kullandığımız Encapsulation, Polymorphism, Inheritance ve Abstraction kavramlarına yer verilmiştir.

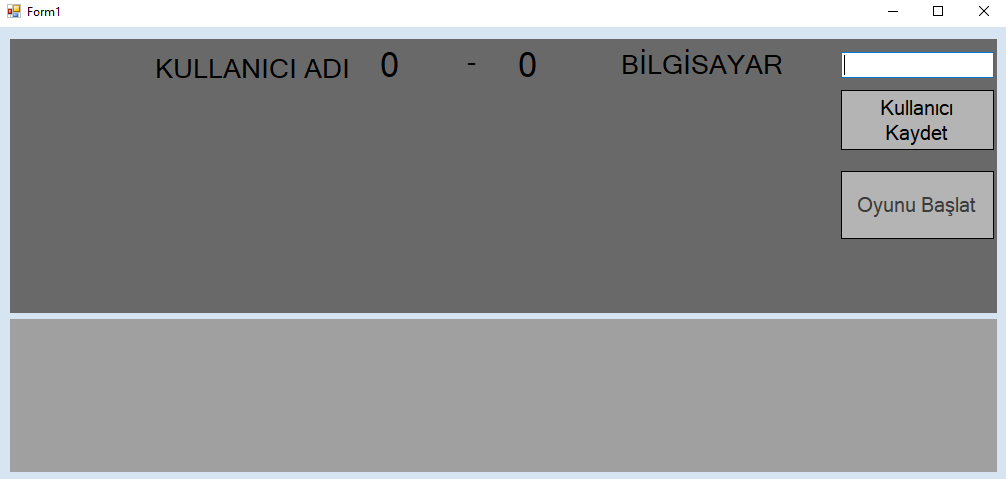
**Giriş**

Proje bir kart oyunu üzerine kurulmuştur. Oyun kullanıcı ve bilgisayar tarafından oynanmaktadır. Toplamda 16 kart üzerinden oynanır bu kartlar 8 futbolcu ve 8 basketbolcu diye ayrılır. Kullanıcı ve bilgisayar rastgele 8 er kart alır bu kartların 4’ü futbolcu 4’ü basketbolcudur. Kartlara özellik tanımlanarak özellikler (Pozisyonlar) puanlandırılacaktır. Futbolcu kartındaki özellikler (Pozisyonlar) penaltı, serbest vuruş ve kaleci karşı karşıya tanımlanır basketbolcu kartındaki özellikler (Pozisyonlar) ikilik, üçlük ve serbest atış tanımlanır. Kart özelliklerine değer verilerek kartlar karşılaştırılır ve rekabet edilir. Kullanıcı ve bilgisayar kendilerine dağıtılan kartları sırayla ardışık şekilde atmaları gerekir futbolcu kartı atıldıktan sonra basketbolcu kartı atılması

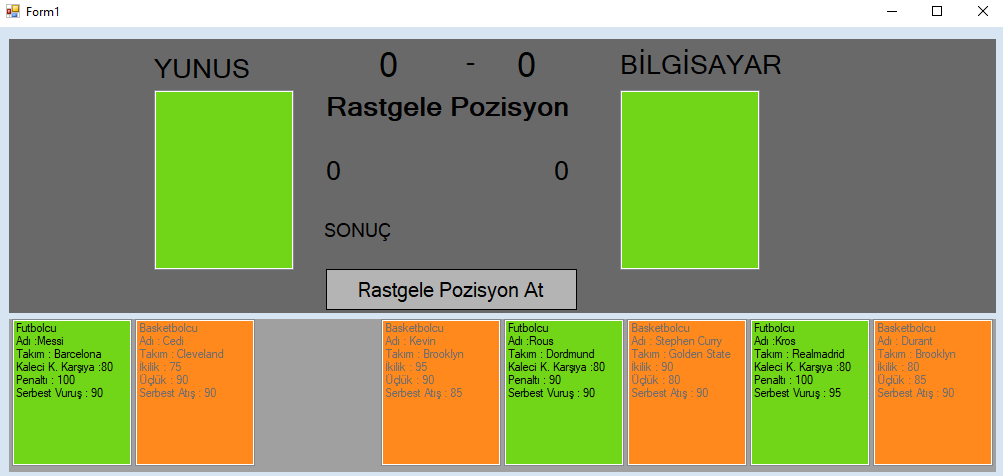
gerekir. Herhangi bir oyuncu kartını attığını attığında aynı türdeki kart atılmalıdır kullanıcı basketbol kartını attıysa bilgisayar da o kartı atmak zorundadır. Kartlar ortaya atıldıktan sonra rastgele bir pozisyon seçilir kartlardaki pozisyon değeri karşılaştırılır değeri yüksek olan kartı atan oyuncu puan alır pozisyon değerleri berabere kalırsa kartlar geri alınır berabere kalınan kart futbolcuysa basketbolcu kartı ile devam edilir ve bu şekilde kartlar bitene kadar oyun sürer eğer oyuncuların ortaya koydukları ellerindeki son kartlarda pozisyon bilgisine göre aynı puandaysa rastgele farklı bir pozisyon alarak yeni karşılaştırma yaparak ilerlenir. Üç pozisyon bilgisi aynıysa eldeki kartlar ile oyun sonlanır. Skoru yüksek olan oyuncu oyunu kazanmıştır.

**Yöntem**

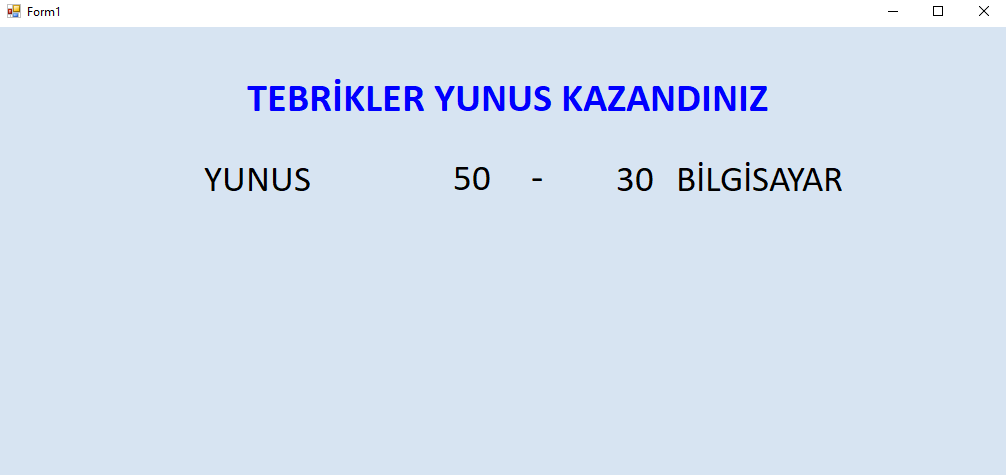
Bu proje bizden nesneye yönelik bir çalışma istendiği için bize verilen bilgilerin hepsini kullanabileceğimiz bir nesneye dönüştürmek planlanmıştır. Sporcular, oyuncular ve arayüz sınıfları oluşturulmuştur. Sporcular sınıfında sporcu adi ve sporcu takımı özellikleri tanımlanmıştır ve bu özelliklerin constructor yapıları oluşturulmuştur. Sporcu sınıfının methodları SporcuPuaniGoster int tipinde kartKullanildiMi Boolean tipinde methodlar oluşturulmuştur. Sporcu sınıfından kalıtım almak üzeri futbolcu ve basketbolcu sınıfları oluşturulmuştur sporcu özelliklerini taşıcağı için sporcuların adı ve takımını tanımlamamıza gerek yoktur. Futbolcu sınıfında kaleci karşı karşıya, penaltı ve serbest vuruş özellikleri int tipinde tanımlanır Sporcu sınıfında tanımlanan methodlar override edilerek tanımlanır. Basketbolcu sınıfında ikilik, üçlük ve serbest atış özellikleri int tipinde tanımlanır sporcu sınıfındaki methodlar override edilerek tanımlanır sporcuPuanıGoster methodu sporcuların rastgele seçilen pozisyon değerlerini göstermektedir kartKullanıldıMı methodu kullanılmışsa false değerini kullanılmamışsa true değerini döndürür. Oyuncu sınıfından bir nesne üretilmeyecği için burada Abstraction (soyutlama) kullanılmıştır sınıf içerisinde oyuncu id, adı ve skor özellikleri tanımlanmıştır ve bu özelliklerin constructor yapıları oluşturulmuştur. Abstraction int tipinde skorGoster ve virtual int tipinde kartSec methodu oluşturulmuştur. Skor göster methodu kart değerlerine göre gelen skorları toplar ve gönderir. kartSec methodu gelen kartın tipine bakar (futbolcu veya basketbolcu ) ve kartı o tipdeki sınıfa gönderir kart daha önce kullanıldı mı kullanılmadı mı bakar değeri döndürür kullanılmışsa -1 değerini kullanılmamışsa gelen değeri gönderir. Kullanıcı ve bilgisayar sınıfları oyuncu sınıfından kalıtım aldığı için oyuncu sınıfın özelliklerini kullanır ve oyuncu sınıfındaki methodlar override edilerek bu sınıflarda da kullanılır. Arayüz sınıfımız da bu oluşturğumuz sınıfları burada test ederiz. Futbolcu ve basketbolcu kartları tanımlanır. Kartlar oyunculara rasgele dağıtılır ve oyun bu arayüz sınıfı üzerinden yürütülür.











+ : PUBLİC

- : PRİVATE

# : PROTECTED

$ : ABSTRACT

