

# Taş Kağıt Makas

Umut YİĞİT

Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi  
200201041

Betül BODUR

Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi  
210201069

## I. ÖZET

Bu dokümdanda Programlama Laboratuvarının 2. projesi olan "Taş Kağıt Makas" için oluşturulan algoritmalar açıklanmıştır . Dokümdanda problemin tanımı verilip bu problemler hakkında araştırmalar ve yöntemler sunulmuştur.Araştırmalar ve yöntemlerde bahsedilen class ve metodların ayrıntılı açıklaması bulunmaktadır.

**Index Terms**—oyun,nesne,metot

## II. PROBLEM TANIMI

Bu projede bizden istenen nesneye yönelik programlama yapısını kullanarak dinamik bir oyun tasarlamaktır.Bu oyun java programlama dilinde yazılmıştır

Oyun, Bilgisayar-Bilgisayar veya Kullanıcı-Bilgisayar şeklinde oynanabilmelidir.Tasarlanan oyunda taş,kağıt ve makas nesneleri oluşturulmalıdır. Bu nesnelerden her oyuncuda 5 tane olmalıdır.Kullanıcı-Bilgisayar oyundan; kullanıcı nesne seçimini oyun başlamadan kendi yapabilmelidir, bilgisayar rastgele şekilde nesne seçimini gerçekleştirmektedir.Bilgisayar-Bilgisayar oyundan, iki bilgisayar da rastgele şekilde nesne seçimini gerçekleştirmektedir.

Bilgisayar-Bilgisayar oyundan ilerleyişi şu şekildedir: Bilgisayarlar elindeki nesneleri rastgele oluşturuktan sonra seçimlerini rastgele yapmaktadır.Bu rastgele seçilen nesneler karşılaşırılmaktadır. Kullanıcı-Bilgisayar oyundan ilerleyişi şu şekildedir: Kullanıcı elindeki nesneleri oluşturuktan sonra nesne seçimini yapar.Bilgisayar elindeki nesneleri rastgele oluşturuktan sonra seçimlerini rastgele yapmaktadır. Seçilen bu nesneler karşılaşırılmaktadır.Karşılaştırılmaya giren nesnelerin; karşı nesneye etkisi ve seviye puanı hesaplanmaktadır.Bu karşılaşırılmalar sırasında oyuncuların skor puanında hesaplanmaktadır.

Oyunun sonlanması için iki oyuncudan birinin nesneleri tükenmeli veya başlangıçta belirlenen toplam hamle sayısı tamamlanmalıdır.Oyun sonunda skor puanı yüksek olan oyuncu kazanmaktadır.

## III. ARAŞTIRMALAR VE YÖNTEMLER

Proje başlamadan önce nesneye yönelik programlama öğrenilmiştir.Projede öncelikle class yapıları nesneye yönelik programlama bilgileri kullanılarak oluşturulmuştur.

Oyunumuzun bir arayüz üzerinde oynanabilmesi için arayüz tasarılanması gerekmektedir. Arayüz için Javada bulunan Swing paketi kullanılmıştır. Ancak arayüzü tasarlarken karşımıza butonlar ile ilgili şu sorun çıkmıştır: Bir butona basıldığında oyunun başlaması gereği komutunu nasıl entegre edeceğimizdir yani o butona basıldan sonra önceki çalışan komutların durması gerekmektedir.Bu sorun, araştırmalar sonucunda bulunan, wait ve notify methodlarını kullanılarak çözülmüştür.

Projede ilk bilgisayar-bilgisayar karşılaşması doğru bir şekilde Arayuz classında çalıştırılmıştır.Ancak kullanıcı-bilgisayar karşılaşmasında, bilgisayar -bilgisayar karşılaşmasından farklı olarak kullanıcı nesneleri kendisi seçmektedir. Bu yüzden kodda bir düzenlemeye gidilmiştir.Oyun classında, Bilgisayar ve Kullanıcı classındaki bilgilere erişmek için nesne tanımlanmıştır. Ve bu nesneler kullanılarak bilgisayar ve kullanıcının nesne seçmesi sağlanmaktadır.Seçilen nesneler ArrayList'e atılmaktadır. Bilgisayar için bu ArrayList rastgele nesne seçebileceği bir yerdir. Kullanıcı için ise seçtiği nesneleri görmesini sağlamaktadır.Kullanıcı ve bilgisayar seçenek istedikleri nesneyi Arayuz classında seçmektedir.Arayuz classında oluşturulan oyun referansı ile Oyun classındaki karşılaşırma methoduna gönderilmektedir.Bu durumun sonucunda Kullanıcı-Bilgisayar ve Bilgisayar-Bilgisayar oyunu düzgün bir şekilde çalışmaktadır.

Projenin sonunda kodun çalıştırılabilir hali (jar dosyası) test edilmiştir. Ancak oyun içine eklenen görüntüler; kod çalıştırıldığından NetBeans üzerinde gözükmemiştir.Bu soruna çözüm olarak getClasses() yöntemi, sınıfın tüm genel sınıfları ve arabirimleri temsil eden sınıf nesnelerini içeren bir dizi döndürmektedir. getResource() metodu, bu sınıfın içinde bulunduğu modülün kaynaklarını döndürmek için kullanılır.

Bu işlevden döndürülen değer, URL sınıfının nesnesi biçiminde bulunmaktadır.Bu yüzden tüm görüntüler , URL tipinde görüntü tanımlayıp getResource() ile kaynaklarını döndürmüştür ve bu sayede oyun hem jar dosyasında hem de NetBeans üzerinde çalıştırıldığında görüntü kaybı olmamıştır. Örnek kod;

```
URL tasImg = this.getClass().getResource("/images/stone.png")
```

#### IV. GELİŞTİRME ORTAMI

Bu proje Windows işletim sisteminde , NetBeans 15 üzerinde geliştirilmiştir . 'jdk19' ile derlenmiştir.

#### V. KOD BİLGİSİ

##### A. Class'lar

Oyunun sınıfları şu şekilde özetlenmiştir:

##### Oyun Sınıfı:

Ana class'tır. 'whoPlays' adında boyutu 3 olan bir dizi tanımlanmıştır.Bilgisayar class'ından bilgisayar1 ve bilgisayar2 olmak üzere 2 tane nesne oluşturulmuştur.Kullanıcı class'ından kullanıcı adında nesne oluşturulmuştur.Tas,Kagit,Makas,agirTas,ozelKagit ve ustaMakas classlarından ikişer tane nesne oluşturulmuştur.

##### Arayuz Sınıfı:

Arayüz burada oluşturulmuştur.Label,button,görüntüler burada eklenmiştir.Tas,Kagit,Makas,agirTas,ozelKagit ve ustaMakas classlarından ikişer tane nesne oluşturulmuştur.

##### Oyuncu Sınıfı:

Oyuncu classı bir abstract classtır.Bu yüzden burada oluşturulan metodların gövdesi yoktur.Oyuncu classının parametresiz constructer ve oyuncuID, oyuncuAdı,skor parametrelerini içeren parametrelı constructer bulunmaktadır.skorGoster() ve nesneSec() metodlarını içermektedir.Seçilen nesneleri tutmak için nesneListesi adında ArrayList bulunmaktadır.

##### Bilgisayar Sınıfı:

Oyuncu abstract classından kalıtım almaktadır.super() metoduyla üst sınıf olan Oyuncu Classındaki constructer'ı çağrılabilmektedir ve bu sayede oyuncuID, oyuncuAdı,skor özelliklerini bu classta kullanılmaktadır.Tas,Kagit,Makas,agirTas,ozelKagit,ustaMakas adında oluşturuluran nesneleri ve özelliklerini tutmak için seçilen nesne türünden ArrayList'ler oluşturulmuştur.Oluşturulan nesnelerin türünü tutmak için 'nesneler' adında dizi tanımlanmıştır.Bu classta parametresiz constructer ve oyuncuID, oyuncuAdı,skor parametrelerini içeren parametrelı constructer bulunmaktadır.Oyuncu class'ında bulunan nesneSec() metodu override edilerek yeniden yazılmıştır.nesneSec() methodu, nesneler kendi türüne ait olan ArrayList'lerinde nesne oluşturur.skorGoster() metodu oyun bir tur oynanlığında dönen seviyePuanını ekleyip karşılaştırılan nesnenin skorunu hesaplamaktadır.

class'ında bulunan nesneSec() metodu override edilerek yeniden yazılmıştır.nesneSec() methodu, random bir şekilde 1-3 arasında bir sayı oluşturur.Bir, taş;iki,kağıdı;uç,makası temsil eder.Random bir şekilde oluşan sayıya göre Oyuncu classında oluşturulan nesneListesi adlı ArrayList'ine nesne adı+random sayı şeklinde eklenir.Ve nesneler dizisine random sayı atanır.Nesne class'ında bulunan skorGoster () metodu 'override' edilerek yeniden yazılmıştır.skorGoster() metodu oyun bir tur oynanlığında dönen seviyePuanını ekleyip karşılaştırılan nesnenin skorunu hesaplamaktadır.

##### Kullanıcı Sınıfı:

Oyuncu abstract classından kalıtım almaktadır.super() metoduyla üst sınıf olan Oyuncu Classındaki constructer'ı çağrılabilmektedir ve bu sayede oyuncuID, oyuncuAdı,skor özellikleri bu classta kullanılmaktadır.Tas,Kagit,Makas,agirTaş,ozelKağıt,ustaMakas adında oluşturuluran nesneleri ve özelliklerini tutmak için seçilen nesne türünden ArrayList'ler oluşturulmuştur.Oluşturulan nesnelerin türünü tutmak için 'nesneler' adında dizi tanımlanmıştır.Bu classta parametresiz constructer ve oyuncuID, oyuncuAdı,skor parametrelerini içeren parametrelı constructer bulunmaktadır.Oyuncu class'ında bulunan nesneSec() metodu override edilerek yeniden yazılmıştır.nesneSec() methodu, nesneler kendi türüne ait olan ArrayList'lerinde nesne oluşturur.skorGoster() metodu oyun bir tur oynanlığında dönen seviyePuanını ekleyip karşılaştırılan nesnenin skorunu hesaplamaktadır

##### Nesne Sınıfı:

Nesne classı bir abstract classtır.Bu yüzden burada oluşturulan metodların gövdesi yoktur.Nesne classının parametresiz constructer ve dayanıklılık,seviyePuani parametrelerini içeren parametrelı constructer bulunmaktadır.Burada dayanıklılık ve seviyePuani değişkenlerinin ilk değeri tutulmaktadır.nesnePuaniGoster(), etkiHesapla() ve durumGuncelle() metodlarını içermektedir.

##### Tas Sınıfı

Nesne abstract classından kalıtım almaktadır.super() metoduyla üst sınıf olan Nesne Classındaki constructer'ı çağrılabilmektedir ve bu sayede dayanıklılık, seviyePuani özellikleri bu classta kullanılmaktadır.Bu classta parametresiz constructer ve dayanıklılık, seviyePuani, katılık parametrelerini içeren parametrelı constructer bulunmaktadır.Nesne class'ında bulunan etkiHesapla() metodu override edilerek yeniden yazılmıştır.etkiHesapla() metoduna parametre olarak gelen objectType ile nesnenin kağıt mı yoksa makas mı olduğu anlaşılır ve taş nesnesinin bu nesnelere karşı etkisi hesaplanır.Nesne class'ında bulunan durumGuncelle ()

metodu override edilerek yeniden yazılmıştır.durumGuncelle() metoduna parametre olarak gelen etki ve artis değerleriyle, taş nesnesinin dayanıklılığı ve seviyePuani güncellenmektedir.

### **agirTas Sınıfı**

Tas classından kalıtım almaktadır. super() metoduya üst sınıf olan Tas classındaki constructer'ı çağrılabilmektedir ve Tas classındaki dayanıklılık, seviyePuani ve katılık özelliklerini çağrılabilmektedir. Bu classta parametresiz constructer ve dayanıklılık, seviyePuani, katılık, sıcaklık parametrelerini içeren parametreli constructer bulunmaktadır. Tas class'ından kalıtım aldığı için üst class olan Tas classındaki methodları override ederek yeniden yazılmıştır. etkiHesapla() metoduna parametre olarak gelen objectType ile nesnenin kağıt mı yoksa makas mı olduğu anlaşılır ve ağırtası nesnesinin bu nesnelere karşı etkisi hesaplanır. durumGuncelle() metoduna parametre olarak gelen etki ve artis değerleriyle, ağırtası nesnesinin dayanıklılığı ve seviyePuani güncellenmektedir.

### **Kagit Sınıfı**

Nesne abstract classından kalıtım almaktadır. super() metoduya üst sınıf olan Nesne Classındaki constructer'ı çağrılabilmektedir ve bu sayede dayanıklılık, seviyePuani özellikleri bu classta kullanılmaktadır. Bu classta parametresiz constructer ve dayanıklılık, seviyePuani, nufuz parametrelerini içeren parametreli constructer bulunmaktadır. Nesne class'ında bulunan etkiHesapla() metodunu override edilerek yeniden yazılmıştır. etkiHesapla() metoduna parametre olarak gelen objectType ile nesnenin taş mı yoksa makas mı olduğu anlaşılır ve kagit nesnesinin bu nesnelere karşı etkisi hesaplanır. Nesne class'ında bulunan durumGuncelle() metodunu override edilerek yeniden yazılmıştır. durumGuncelle() metoduna parametre olarak gelen etki ve artis değerleriyle, kağıt nesnesinin dayanıklılığı ve seviyePuani güncellenmektedir.

### **ozelKagit Sınıfı**

Kagit classından kalıtım almaktadır. super() metoduya üst sınıf olan Kagit classındaki constructer'ı çağrılabilmektedir ve Kagit classındaki dayanıklılık, seviyePuani ve nufuz özelliklerini çağrılabilmektedir. Bu classta parametresiz constructer ve dayanıklılık, seviyePuani, nufuz, kalınlık parametrelerini içeren parametreli constructer bulunmaktadır. Kagit class'ından kalıtım aldığı için üst class olan Kagit classındaki methodları override ederek yeniden yazılmıştır. etkiHesapla() metoduna parametre olarak gelen objectType ile nesnenin makas mı yoksa taş mı olduğu anlaşılır ve özelkağıt nesnesinin bu nesnelere karşı etkisi hesaplanır. durumGuncelle() metoduna parametre olarak gelen etki ve artis değerleriyle, özelkağıt nesnesinin dayanıklılığı ve seviyePuani güncellenmektedir.

### **Makas Sınıfı**

Nesne abstract classından kalıtım almaktadır. super() metoduya üst sınıf olan Nesne Classındaki constructer'ı çağrılabilmektedir ve bu sayede dayanıklılık, seviyePuani özellikleri bu classta kullanılmaktadır. Bu classta parametresiz constructer ve dayanıklılık, seviyePuani, keskinlik parametrelerini içeren parametreli constructer bulunmaktadır. Nesne class'ında bulunan etkiHesapla() metodunu override edilerek yeniden yazılmıştır. etkiHesapla() metoduna parametre olarak gelen objectType ile nesnenin taş mı yoksa kağıt mı olduğu anlaşılır ve makas nesnesinin bu nesnelere karşı etkisi hesaplanır. Nesne class'ında bulunan durumGuncelle() metodunu override edilerek yeniden yazılmıştır. durumGuncelle() metoduna parametre olarak gelen etki ve artis değerleriyle, makas nesnesinin dayanıklılığı ve seviyePuani güncellenmektedir.

### **ustaMakas Sınıfı**

Makas classından kalıtım almaktadır. super() metoduya üst sınıf olan Makas classındaki constructer'ı çağrılabilmektedir ve Makas classındaki dayanıklılık, seviyePuani ve keskinlik özelliklerini çağrılabilmektedir. Bu classta parametresiz constructer ve dayanıklılık, seviyePuani, keskinlik, direnç parametrelerini içeren parametreli constructer bulunmaktadır. Makas class'ından kalıtım aldığı için üst class olan Makas classındaki methodları override ederek yeniden yazılmıştır. etkiHesapla() metoduna parametre olarak gelen objectType ile nesnenin kağıt mı yoksa taş mı olduğu anlaşılır ve ustamakas nesnesinin bu nesnelere karşı etkisi hesaplanır. durumGuncelle() metoduna parametre olarak gelen etki ve artis değerleriyle, ustamakas nesnesinin dayanıklılığı ve seviyePuani güncellenmektedir.

### **B. Algoritma Özeti**

Oyun class'ı ve Arayüz class'ının üzerinden projenin genel algoritması şöyledir:

Oyun class'ı ana class'tır. 'whoPlays' adında boyutu 3 olan bir dizi tanımlanır. Bilgisayar class'ından bilgisayar1 ve bilgisayar2 olmak üzere 2 tane nesne oluşturulur. Kullanıcı class'ından kullanıcı adında nesne oluşturulur. Tas, Kagit, Makas, agirTas, ozelKagit ve ustaMakas classlarından ikişer tane nesne oluşturulur. com1, com2, mainPlayer, setSayisi adında değişkenlerimiz vardır. Kod çalıştırıldığında ilk Oyun class'ının içindeki main metoduna girer. Burada dosya yazma işlemi için "output.txt" adında bir yazılabilirliği kapalı text dosyası oluşturulur. Arayüz classından "arayuz" adında bir

nesne oluşturup Arayuz class'ındaki gui metodunu çağırır.

Arayuz class'ında Oyun class'ından oyun adında nesne oluşturulur.label,jbutton,jradiobutton,imageicon ve görüntüleri arayüze yansıtmak için nesne oluşturulur.JFrame oluşturulur.bittMi,winner,birinciSkor,ikinciSkor,next adında global değişkenler oluşturulur.Tas,Kagit,Makas,agirTas,ozelKagit ve ustaMakas classlarından ikişer tane nesne oluşturulur.

gui() metodunda arayüz oluşturulur.Arayüz kısmında görüntülenecek JButton,JTextField,JLabel,görüntüler oyun ilk açıldığında sadece menü kısmının görünürlüğü açık olacak şekilde tanımlanır.

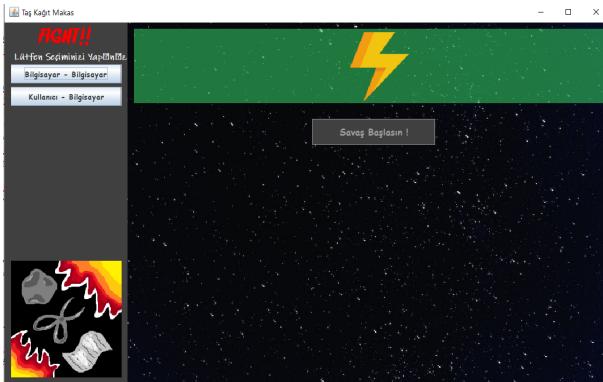


Fig. 1. Oyun Açıldığında İlk Görüntüsü

Oyun ilk açıldığında karşımız çıkan iki seçenek bulunmaktadır.Biri Bilgisayar-Bilgisayar karşılaşması diğeri ise Kullanıcı-Bilgisayar karşılaşmasıdır. iki buton için ayrı bir yol izlenmesi gerekmektedir. Bunun için iki buton için ayrı ayrı ActionListener() oluşturulmuştur. Butona tıklandığında o butona özel ActionListener içindeki olayları gerçekleştirecektir.Bilgisayar-Bilgisayar butonu seçildiğinde Oyun classından oluşturulan oyun nesnesi ile com1 ve com2 değişkenleri 1 artırılır.Karşılaşmaya giren oyuncuların adları arayüze eklenir. İki oyuncunuda skorları gösterecek olan JLabel'ın görünürlüğü aktif olur.Oyuncuların adları Oyun classında oluşturulan bilgisayar1 ve bilgisayar2 nesnelerinden Bilgisayar class'ındaki oyuncuAdı değişkenine atanır.Ve oyun için gerekli olan set sayısının yazılabildiği JTextField ve onaylamak için olan buton aktif olur.Daha önce karşımıza çıkan kullanıcı-bilgisayar ve bilgisayar-bilgisayar butonu tıklanabilirliği kapanır(setEnabled).En sonunda ise oyunBaslasın değişkeni 1 artırılır.

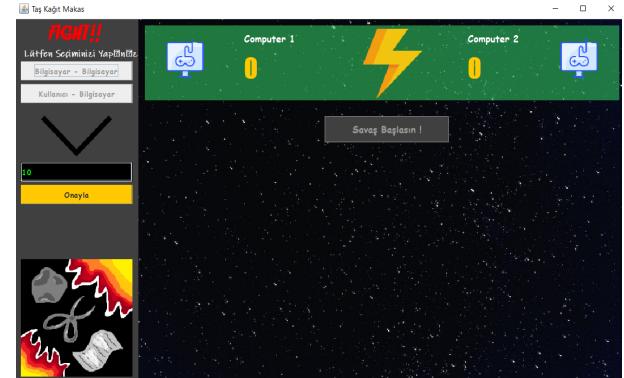


Fig. 2. Bilgisayar-Bilgisayar butonu tıklandığında

Kullanıcı-Kullanıcı butonuna tıklandığında bu butona ait ActionListener'in içindeki olaylar gerçekleşecektir.Oyun classından oluşturulan oyun nesnesi ile com1 ve mainPlayer değişkenleri 1 artırılır.İki oyuncununda skorları gösterecek olan JLabel'ın görünürlüğü aktif olur.Karşılaşmaya giren bilgisayar oyuncusunun adı arayüze eklenir.Oyun için gerekli olan set sayısının yazılabildiği JTextField ve onaylamak için olan buton aktif olur.Kullanıcının adını girmesi için gerekli JTextField ve onaylamak için buton görünür olur ama tıklanırlığı kapalıdır.Bunun sebebi henüz set sayısını girmedigimizdedir.Daha önce karşımıza çıkan kullanıcı-bilgisayar ve bilgisayar-bilgisayar butonu tıklanabilirliği kapanır(setEnabled).

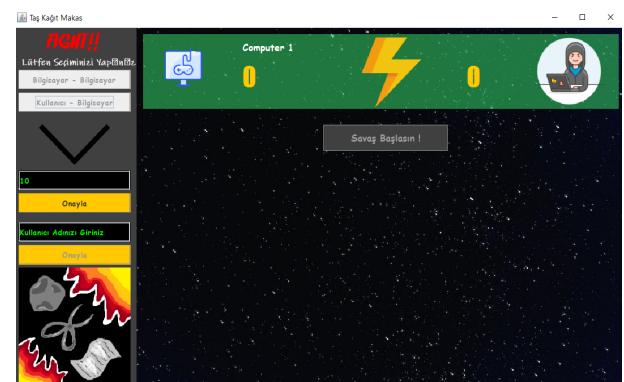


Fig. 3. Kullanıcı-Bilgisayar butonu tıklandığında

Oynama türü seçildikten sonra çıkan set sayısı girildikten sonra onayla butonuna bastığımızda bu butona ait ActionListener devreye girer ve Oyun classına, oluşturduğumuz oyun nesnesiyle ulaşıp setSayisi değişkenine girilen sayı gönderilir.Ve sonrasında butonun tıklanırlığı ve JTextField'e yazı girişi kapanır.Kullanıcının adı girilen TextField açık mı kapalı mı kontolünü isEnabled() ile yapılır eğer sonucu doğru ise kullanıcı adını onaylamak için buton aktif olur.oyunBaslasın bir artırılır. Eğer oyunBaslasın, 2 sayısına eşit ise başlat butonu tıklanabilir olacaktır. Yani burada bu şartı

sağlaması için Bilgisayar-Bilgisayar karşılaşması seçilmelidir.

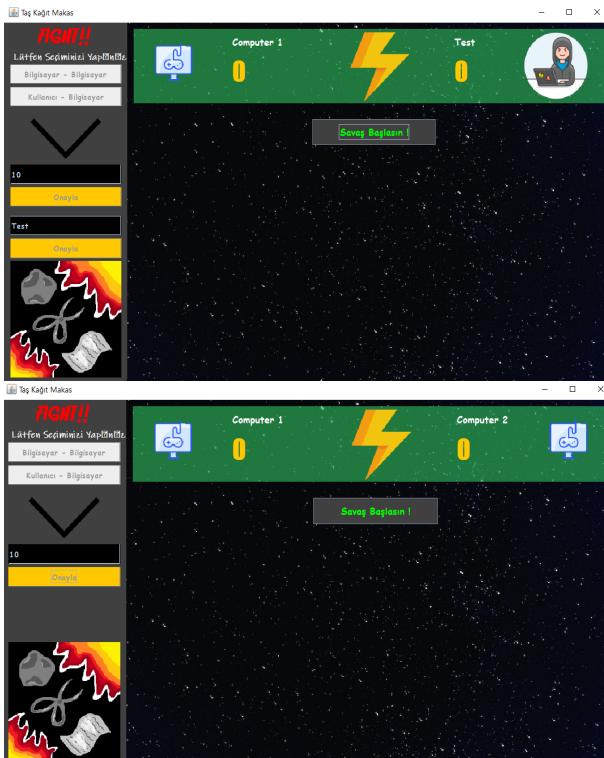


Fig. 4. Set Sayısı

Eğer kullanıcı adının onay butonuna tıklanırsa TextField'ın içindeki bilgi Oyun classında oluşturululan kullanıcı nesnesinden Kullanıcı class'ındaki oyuncuAdı değişkenine atanır.Kullanıcı adı için olan TextField ve onayla butonunun tıklanılabilirliği kapanır.Arayüze kullanıcılar eklenir.oyunBaslasın değişkeni 1 artırılır. Eğer oyunBaslasın 2 sayısına eşit olursa başlat butonu aktif olur.

Başlat butonunun ActionListener'ı daha farklıdır. Çünkü Başlata bastığımızda oyunun başlaması gerekmektedir ve bu yüzden burada basana kadar sonrasını beklemeye almamız gerekmektedir.Bu notify() ve wait() metodları ile gerçekleşir.Ve başlat butonun bastığımızda başlat butonunun tıklanılabilirliği ve görünütürlüğü kapanır.Oyun classından referans alarak oluşturduğumuz oyun nesnesiyle Oyun classındaki startGame() metodunu çalıştırılır. startGame() metodunda whoPlays dizisinin indislerine sırayla com1,com2 ve mainPlayer'da bulunan değerler atılır ve bu indislerinin değerlerinin 1 olup olmadığı kontrol edilir.Eğer whoPlays[0] ve whoPlays[1] 1 ise Bilgisayar classından aldığımız referans ile oluşturduğumuz bilgisayar1 ve bilgisayar2 nesneleri kullanılarak, Bilgisayar classında bulunan nesneSec() metodu çalışmaktadır.Bilgisayar class'ında bulunan nesneSec() metodу yukarıdaki A bölümünde ayrıntılı bir şekilde açıklanmıştır.whoPlays[2] ise Kullanıcı class'ından aldığımız referans ile oluşturduğumuz kullanıcı nesnesini kullanarak Kullanıcı class'ındaki nesneSec() metodunu çalıştırır.

Eğer bilgisayar-bilgisayar karşılaşması seçildiyse Arayüz class'ında bulunan kartSec() metodu çalışır.kartSec metodunda, kaçıncı setteyse , o set sayısı 5 sayısına tam bölünüyorsa nesnelerin tekrar seçilmesi sağlanır. Bilgisayarlar için seçilen nesne 10 sayısına tam bölünüyor veya -1 ise tekrar random nesne seçtilir.delim() metodu rastgele oluşan sayının nesneListesindeki yerinde bulunan stringin son harfine bakarak kaçıncı indis olduğunu bular. hangiNesne() metodu rastgele oluşan sayının nesneListesinedeki yerinde bulunan stringin Tas,Kagit,Makas,agirTas,ozelKagit,ustaMakas olup olmadığını kontrol eder. Eğer taş sınıfındaysa 1 ,kağıt sınıfındaysa 2 ve makas sınıfındaysa 3 değerini döndürür.c1temp ve c2temp , delim() metodundan gelen değerleri tutan değişkendir. c1temp2 ve c2temp2 hangiNesne() metodundan gelen değerleri tutan değişkendir. Seçilen nesne tekrar seçilmesin diye 10 ile çarpılır(nesneler dizisinde tutulur). Her oyuncu için nesneleri kontrol edilir. Seçilen nesne Tas ise ve seviyePuanı 30'dan büyük ise artık c1temp2 4 olur nesneListesindeki rastgele oluşan rakamda indiste bulunan string güncellenerek AgirTas+c1temp olur.Tas+c1temp nesnesinin tüm özellikleri agirt1'e aktarılır.Seçilen nesne Kagit ise ve seviyePuanı 30'dan büyük ise artık c1temp2 5 olur nesneListesindeki rastgele oluşan rakamda indiste bulunan string güncellenerek OzelKagit+c1temp olur.OzelKagit+c1temp nesnesinin tüm özellikleri ozelk1'e aktarılır.Seçilen nesne Makas ise ve seviyePuanı 30'dan büyük ise artık c1temp2 6 olur nesneListesindeki rastgele oluşan rakamda indiste bulunan string güncellenerek UstaMakas+c1temp olur.Makas+c1temp nesnesinin tüm özellikleri ustam1'e aktarılır.Bu kontrol işlemi bilgisayar2'nin nesneleri içinde yapılır.Her iki bilgisayarda seçtiği nesneler ekrana yazılır.

Oyun class'ında bulunan kapistir() metoduna iki bilgisayarda seçikleri nesne bilgileri parametre olarak gönderilir.Eğer kendi türünden nesne ile karşılaşılırsa iki nesnede de bir değişiklik olmamaktadır.Eğer kendisinden farklı türde nesne ile karşılaşlığında 2 nesnenin etkileri etkiHesapla() metoduyla hesaplanır ve durumGuncelle() metoduyla 2 nesneden etkisi büyük olanın seviye puanında artış her iki nesnenin etkileri birbirlerinin dayanıklılığından çıkarılarak güncellenir.Eğer 2 nesneden birinin dayanıklılığı 0 veya 0 dan küçükse nesneler dizisine, o nesnenin bulunduğu indis -1 yazılır yani oyundan elenmiş olur.

İlk nesnelerin karşılaşması bitince bilgisayar1 için olan nesnelerde elenen var mı diye kontrol edilir, eğer bilgisayar1'de varsa isEmpty1 1 artırılır.bilgisayar2 de elenen varsa isEmpty2 1 artırılır.Eğer isEmpty1=5 ve isEmpty2=5 değilse bittiMi değişkeni 1, winner değişkeni 2 olur. Eğer isEmpty1=5 değilse ve isEmpty2=5 ise bittiMi değişkeni 1, winner değişkeni 1 olur.Eğer isEmpty1=5 ,isEmpty2=5 ve set sayısı kadar oynanılmadysa bittiMi değişkeni 0 olur.Skor puanı yüksek olan oyuncu kazanmış olur.Eğer skorlar eşitse berabere olur.

Eğer kullanıcı-bilgisayar karşılaşması seçildiyse Arayüz

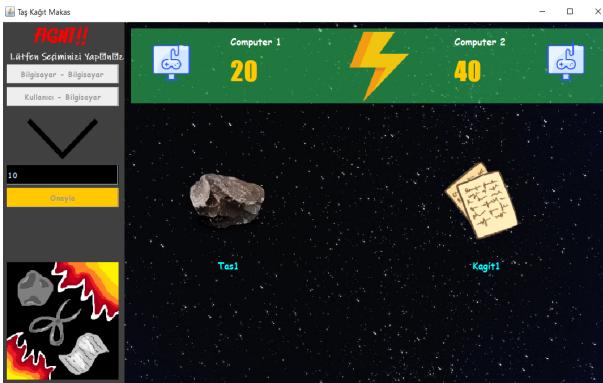


Fig. 5. Bilgisayar-Bilgisayar Oyunu

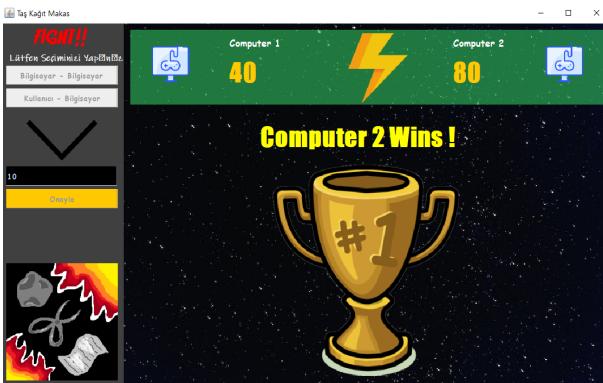


Fig. 6. Bilgisayar-Bilgisayar Oyun Sonu

class’ında `kullaniciBaslangic()` metodu çalışır.Kullanıcı nesneleri burada seçer.Nesne seçebilmesi için `JRadioButton` kullanılmıştır. Taş nesnesi seçmek için taş butonunu seçikten sonra seçme işlemine devam edebilmesi için seç butonuna tiklaması gerekmektedir.Her buton onayındığında seçilen nesnenin nesneler dizisine türü ve nesneListesine adı eklenmektedir.Eğer 5 tane seçtiyse butonların görünürlüğü kapanmaktadır.

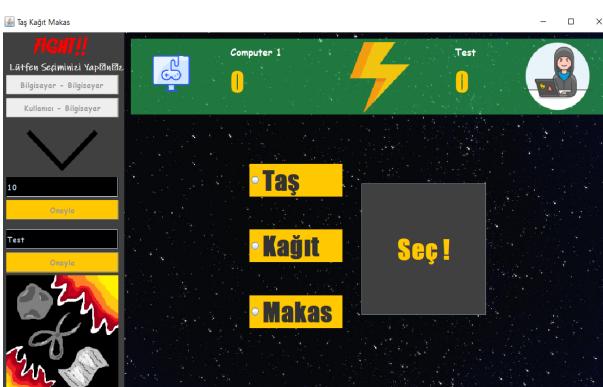


Fig. 7. Kullanıcı Nesne Oluşturma Ekranı

Kullanıcı nesneleri seçikten sonra `kartSec2()` metodu çalışmaktadır.Kaçinci setteyse , o set sayısı 5 sayısına tam bölünüyorsa nesnelerin tekrar seçilmesi sağlanır.Kullanıcının istediği nesneyi seçebilmesi için `kullaniciSecilen()` metodu vardır.Bu metotta kullanıcının oyuna başlamadan önce seçtiği nesneleri görebilmesi için JComboBox kullanılmıştır ve seçtiğini onaylamak için savaş adında bir buton tanımlanmıştır.

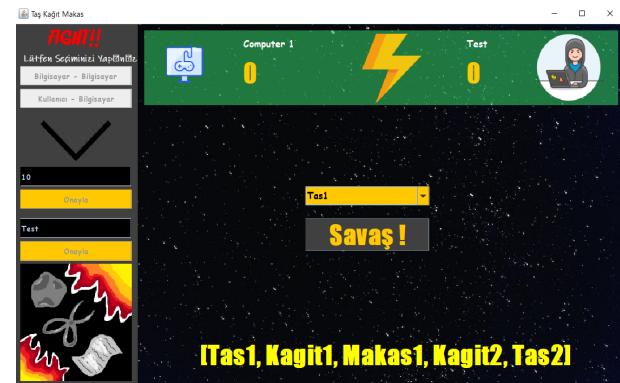


Fig. 8. Kullanıcı Nesne Seçme Ekranı

Kullanıcı tarafından seçilen nesne 10 sayısına tam bölünmüyor veya -1 değil ise bu nesneyi bir dizeye atmaktadır.Eğer 10'a bölünüyor veya -1 ise listede "Seçildi" yazmaktadır.Her turda bu dizi yenilenmektedir.kullaniciSecilen bize listeden seçilen indisi döndürmektedir.Bilgisayar için seçilen nesne 10 sayısına tam bölünmyor veya -1 ise tekrar random nesne seçir.Kullanıcının seçtiği nesne 10 sayısına tam bölünmyor veya -1 ise kullanıcıya tekrar nesne seçir.Geri kalan her şey bilgisayar- bilgisayar karşılaşmasındaki gibi ilerlemektedir.

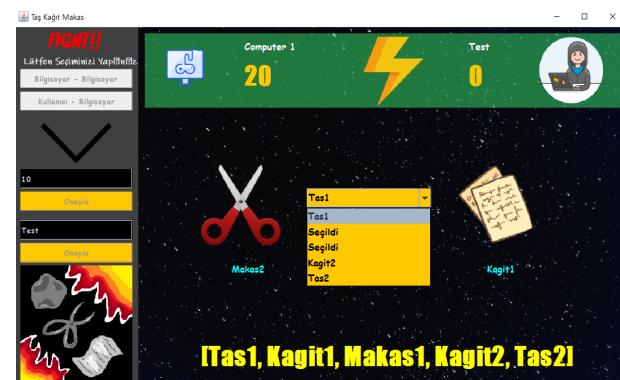


Fig. 9. Kullanıcı-Bilgisayar Oyun

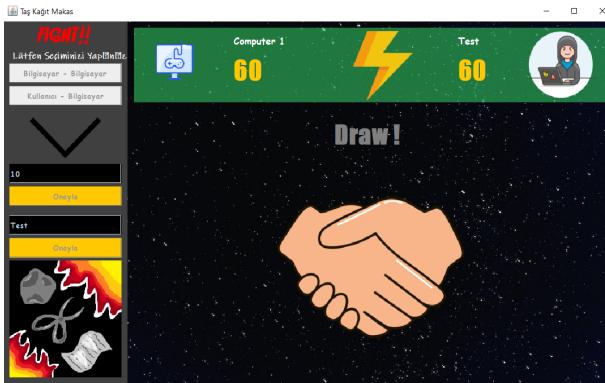


Fig. 10. Kullanıcı-Bilgisayar Oyun Sonu

### C. UML DİYAGRAMI

Kısim ektedir.[fig.11]

## VI. KATKILAR

Projede nesneye yönelik programlama ve arayüz kısmını olarak ikiye bölmüştür. Arayüz kısmını Umut Yiğit öğrenmiştir. Nesne yönelik programlama kısmını Betül Bodur öğrenmiştir. Ve bu öğrenilen bilgiler aktarılmıştır. Projede oluşturulan algoritmalar birlikte düşünülmüştür (taş ile bilgileri arayüze taşıma veya arayüzden alma gibi vb.). Kodlar ve Rapor ortak bir şekilde yapılmıştır. Kod kısmının entegre edilmesi Umut Yiğit tarafından sağlanmıştır. UML diyagramı oluşturma Betül Bodur tarafından sağlanmıştır.

## VII. İSTATİSTİK

Kullanılan class'lar ve kullanım amaçları:

java.util

Random sayı oluşturmak, ArrayList oluşturmak, logging.level? logging.logger?

java.awt

Yazı tipi, rengi, butona basıldığında devam etme gibi olayları düzenlemek için kullanılmıştır.

javax.swing

Buton, görüntü , çerçeve,.. gibi arayüzü oluşturmak için gerekli nesnelerin bulunduğu class'tır.

## java.net

**java.net**  
java.awt.url ne yapar-

java.io

**java.io** Bilgilere erişilirken oluşan özel durumlar için temel sınıfır.

## VIII. SONUÇLAR

Sonuç olarak kullanıcı-bilgisayar ve bilgisayar -bilgisayar oyunu istenildiği gibi çalışmaktadır. kullanıcı bilgisayar oyununda kullanıcı, nesne seçerek ; bilgisayar ise rastgele nesne seçerek. oyundaki karşılaşmaya katlıp seviye puanı , birbirlerine etkileri ve skor puanı hesaplanabiliyor.bilgisayar-bilgisayar oyununda iki bilgisayar da rastgele nesne seçerek. oyundaki karşılaşmaya katlıp seviye puanı , birbirlerine etkileri ve skor puanı hesaplanabiliyor.

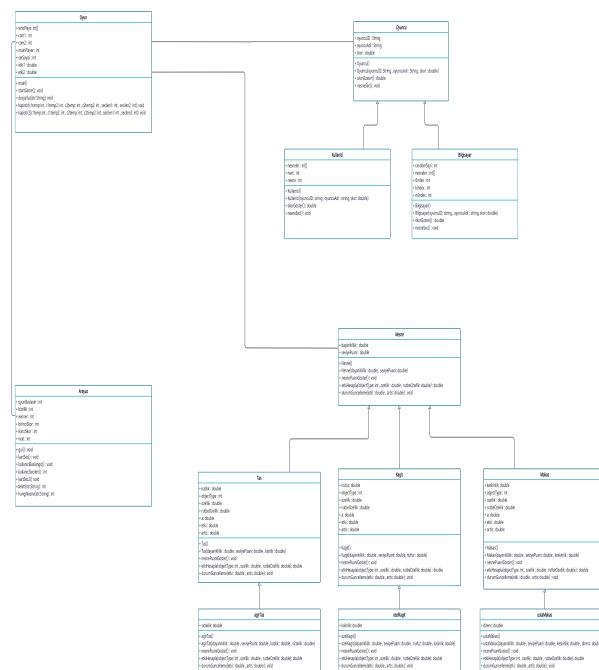


Fig. 11. UML DIAGRAMI

## REFERENCES

- [1] <https://www.kodkampusu.com/javada-yapici-constructor-metotlar/>
  - [2] <https://emrecelen.com.tr/java-constructor-nedir/close>
  - [3] <https://www.dijitalders.com/icerik/44/1997/java-class-apisinda-constructor-ve-overloading>
  - [4] <https://emrecelen.com.tr/javadakalitimnedir>
  - [5] <https://hayalindekiyazilim.com/java/java-soyut-abstract-siniflar/>
  - [6] <https://www.seckintozlu.com/674-javada-this-anahtar-kelimesi-ve-kullanimi.html>
  - [7] <http://www.baskent.edu.tr/tkaracay/etudio/ders/prg/java/ch27/JavaSwing.pdf>
  - [8] <https://www.geeksforgeeks.org/classes-objects-java/>
  - [9] <https://www.youtube.com/@KodlamaVakti>
  - [10] <https://www.youtube.com/watch?v=zERL-bD4NIlist=PLEcJSEQK-cD5KHgg9sXumeg659hAr2j4W>
  - [11] <https://www.geeksforgeeks.org/how-to-create-array-of-objects-in-java/>
  - [12] <https://www.algoritmaornekleri.com/java/javada-nesne-dizisi-nasıl-olusturulur/>
  - [13] <https://stackoverflow.com/questions/11096425/can-i-create-a-object-array-to-hold-different-objects-in-java>
  - [14] <https://stackoverflow.com/questions/64509691/how-can-i-add-some-sound-to-my-java-jframe>
  - [15] <https://stackoverflow.com/questions/2432866/add-objects-with-different-name-through-for-loop>
  - [16] <https://www.geeksforgeeks.org/how-to-clone-a-list-in-java/:text=ArrayList>
  - [17] <https://omerakgull.wordpress.com/2011/02/14/25/>

- [18] <https://stackoverflow.com/questions/33860533/wait-for-a-jbutton-click-to-continue-with-program-execution-java>
- [19] <https://stackoverflow.com/questions/45460401/method-overriding-with-different-parameters>
- [20] <https://stackoverflow.com/questions/24559167/how-to-make-a-jpanel-semi-transparent>
- [21] <https://coderanch.com/t/511437/java/Adding-images-Jar-Files-NetBeans>
- [22] <https://stackoverflow.com/questions/2442599/how-to-set-jframe-to-appear-centered-regardless-of-monitor-resolution>