

Programlama Laboratuvarı 2.Proje Ödevi

1st Emircan Aygün
Bilgisayar Mühendisliği 2.Öğretim
Kocaeli Üniversitesi
Kocaeli Türkiye
emir616116@gmail.com

Abstract—Bu döküman kocaeli üniversitesi bilgisayar mühendisliği bölümü programlama laboratuvarı dersi 2.proje ödevi için hazırlanmıştır.

Index Terms—Nesneye Yönelik Programlama

I. ÖZET

Proje ile beraber Öğrencilerin nesneye yönelik programlama yapısını anlaması ve çözüm sağlayabilmesi amaçlanmaktadır. Proje C++ veya JAVA dili kullanılarak gerçekleştirilecektir. Görsel olarak oynanan bir oyun tasarlanacaktır. Görselleştirme için belirtilen programlama dillerine uygun kütüphaneler kullanılabilir.

II. GİRİŞ

Bir oyuncunun diğer bir oyuncuyla rekabet edebileceği taş kağıt makas oyunu tasarlanacaktır. Oyun kullanıcı-bilgisayar ya da bilgisayar-bilgisayar şeklinde iki seçeneği olarak oynanabilmelidir. Oyunun başlangıcında seçim yapılmalıdır. Tasarlanacak oyunda, her oyuncunun elinde oyunun başlangıcında taş, kağıt ve makas olmak üzere 3 farklı tipte olan nesnelerden toplam 5 nesnesi bulunacaktır. Kullanıcı-bilgisayar oyununda kullanıcı nesne seçimini oyunun başında kendisi yapacak, bilgisayar ise rastgele bir şekilde nesneleri seçecektir.Bilgisayar-bilgisayar oyununda tüm seçimler otomatik olarak yapılmalı ve oyunun adımları hızlı bir şekilde ilerlemelidir.Oyunda;

1. Kullanıcı: taş, kağıt, makas, taş, taş

Bilgisayar: kağıt, kağıt, makas, taş, makas

2. Bilgisayar: makas, kağıt, makas, taş, taş

Bilgisayar: taş, kağıt, makas, taş, kağıt

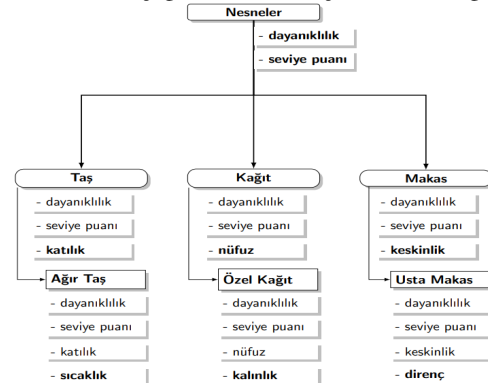
gibi seçimler yapılabilir.Oyun sırasında kullanıcı bilgisayarın elindeki nesnelerin içeriğini görmemelidir. Ancak her seçim yapıldığında seçilen nesneler arayüzde görülebilmelidir. Kullanıcı elindeki kalan nesnelerden herhangi birini her hamle başında kendi seçebilmelidir. Kullanıcı-bilgisayar oyununda oyun temelde şu şekilde ilerleyecektir: Önce kullanıcı elindekiler arasından bir nesne seçimi yapar daha sonra bilgisayar rastgele bir seçim yapar en son nesneler karşılaştırılarak skorlar hesaplanır ve yeni seçimle oyun devam eder. Bilgisayar-bilgisayar oyununda her iki oyuncu içinde rastgele seçimlerle ilerler. Oyun aşağıdaki iki koşuldaki biri

sağlandığında sona erer:

- İki taraftan birinin elindeki nesneler tükendiğinde,
 - Başlangıçta belirlenen toplam hamle sayısı tamamlandığı zaman, (başlangıç toplam hamle sayısı: 10)
- Bütün nesneler seçilmeden aynı nesne 2. defa seçilmemelidir. 5 nesnenin tamamı seçildiğinde aynı nesne tekrar seçilebilmelidir.Ornek senaryo şu şekilde olabilir:

Hamle	Kullanıcı ya da Bilgisayar	Bilgisayar
1	taş1	kağıt1
2	taş2	taş1
3	kağıt	taş2
4	makas1	makas
5	makas2	kağıt2
6	kağıt	taş2
...

Oyunda kullanılan tüm nesneler için dayanıklılık ve seviye puanı özellikleri ortaktır. Nesneler kendi içerisinde taş, kağıt ve makas olarak 3 alt sınıftan birine ait olmalıdır. Taşların diğer nesnelerden ayrı olarak katılık özelliği, kağıtlarda nüfuz özelliği ve makaslarda ise keskinlik özelliği yalnızca kendi sınıflarına ait özelliklerdir. Ayrıca bu 3 sınıfın bir de kendi içerisinde alt sınıfları bulunmaktadır. Bunlar; taş sınıfının altında ağır taş sınıfı, kağıt sınıfının altında özel kağıt kağıt ve makas sınıfının altında ise usta makas sınıfları olacaktır. Sırasıyla ağır taş için sıcaklık, özel kağıt için kalınlık ve usta makas için direnç özellikleri bu sınıflara özgü özelliklerdir. Aşağıda sınıflar ilişkisi gösterilmektedir.



Kullanıcı ve bilgisayarın seçmiş olduğu nesnelerin karşılaştırılması ve skorların hesaplanması aşağıdaki formüllere göre yapılacaktır. Buna göre bir hamlede karşı karşıya gelen iki nesnenin birbirine karşı etkileri ayrı ayrı hesaplanmalıdır. Sonrasında bir nesnenin etki miktarı kadar rakip nesnenin dayanıklılık değeri azaltılacaktır. Dayanıklılık değeri 0 veya 0'ın altına düşen nesneler oyundan elenecektir. Bir nesneyi yenen rakip nesnenin seviye puanını 20 artırılacaktır. Seviye puanı 30'un üstüne ulaşan nesneler terfi edeceklerdir. (Taş nesnesi ağır taş nesnesine, kağıt nesnesi özel kağıt nesnesine, makas nesnesi usta makas nesnesine dönüşecektir) Oyunculardan birinin elindeki tüm nesneler tükendiğinde rakip oyuncu oyunu kazanacaktır. Eğer maksimum 10 hamle (sunum sırasında farklı değerler için denenebilmeli) sonrasında her iki oyuncunun da elinde en az bir nesne varsa bu durumda oyunculardan elindeki nesnelerin toplam dayanıklılık miktarı en fazla olan kazanacak ve bu toplam dayanıklılık değeri kazananın skorunu verecektir. Kullanıcı ve bilgisayarın seçtiği nesneler için aşağıdaki formüllere göre etki puanları hesaplanacaktır. Her bir formüldeki pay kısmı atak yapan tarafı, payda ise rakip tarafın değerlerine karşılık gelmektedir. Süslü parantezin içindeki değişkenler o nesnenin özellikleri arasında varsa kullanılmalıdır aksi durumda hesaba katılmamalıdır. Örneğin taş ve kağıt karşılaştırılıyorsa, taş için tas etkisi, kağıt için kağıt etkisi hesaplanmalıdır. Taş nesnesi için pay kısmında taşın katılık değeri alınır ancak taş ağır taşa dönüşmüşse katılık değeri sıcaklık değeri ile çarpılır. Payda kağıdın değerlerini temsil eder. Kağıdın keskinlik değişkeni olmadığı için 0 alınır ve nüfuz değeri kağıdın kendi değeri olarak alınır. Kağıt özel kağıt ise kalınlık değeri ile çarpılır. a değeri 0,2 olarak alınmalıdır. Pay paydaya bölündüğünde taş nesnesi için kağıda karşı etki değeri hesaplanmış olur. Benzer hesaplama kağıt etkisi için de hesaplanır ve bu iki değer rakiplerin dayanıklılık değerlerinden eksiltir.

$$makas.etkisi = \frac{keskinlik\{*direnc\}}{(a * nüfuz\{*kalınlık_r\} + (1-a) * katılık\{*sıcaklık_r\})} \quad (1)$$

$$kağıt.etkisi = \frac{nüfuz\{*kalınlık\}}{(a * katılık\{*sıcaklık_r\} + (1-a) * keskinlik\{*direnc_r\})} \quad (2)$$

$$taş.etkisi = \frac{katılık\{*sıcaklık\}}{(a * keskinlik\{*direnc_r\} + (1-a) * nüfuz\{*kalınlık_r\})} \quad (3)$$

Formüldeki değerler için her nesnenin kendi özellikleri başlangıçta 2 olarak alınabilir. Başlangıç dayanıklılık değeri tüm nesneler için 20, başlangıç seviye puanı 0'dır.

III. YÖNTEM

Projeye başlangıç olarak nesneye yönelik programlamların hakkında ne olduğu, nasıl, neden ve hangi projelerde ne amaçla kullanıldığı gibi sorulara cevap bulabilmek adına araştırma yaparak başladım. Gerekli teorik bilgileri edinmemin ardından NetBeans ide'yi açarak java ile projeyi kodlamaya başladım. İlk olarak gerekli classları ve ilgili classların aralarındaki kalıtımlarını oluşturdum. Bazı classları abstract olarak oluşturdum. Classlar içerisine override methodları ek-

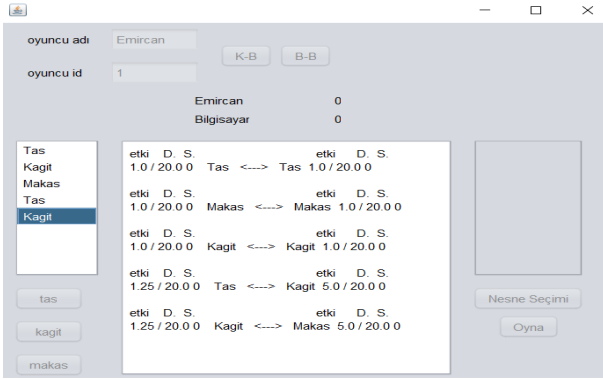
ledim. Projede ilk olarak oluşturduğum classlar Taş ve ondan kalıtım alan AğırTaş, Kağıt ve ondan kalıtım alan ÖzelKağıt, Makas ve ondan kalıtım alan UstaMakas classları oldu. Ve bunlara ek olarak Tas Kağıt ve Makasın kalıtım alacağı abstract Nesne classını oluşturdum. Daha sonra abstract Oyuncu classını ve bu classtan kalıtım alan Kullanıcı ve Bilgisayar classlarını oluşturdum. Bu gerekli classları oluşturduktan sonra oyunun arayüzünü yapabilmek adına bir JFrame classı ekledim. Oyunun main classı olarak bu classı kullandım. Bir sonraki aşamada JFrame yardımcı ile arayüzde ihtiyacım olan componentleri pencereye ekledim. Bu componentler JList, JTextArea, JButton, JLabel ve JTextField idi. İlk olarak kullanıcıdan ilgili 2 button yardımıyla kullanıcı-bilgisayar mı yoksa bilgisayar-bilgisayar oyununun mu oynanmak istediği seçimi aldım. Eğer ilk seçenek seçilmiş ise kullanıcı-bilgisayar oyunu için gerekli olanlar harici componentleri setVisible(false) kullanılarak kapattım. TextFieldlerden aldığım oyuncu adı ve idleri oluşturduğum kullanıcı nesnesine atadım. Ardından 3 adet buton yardımıyla kullanıcıdan oyun başlangıcında 5 adet nesne seçmesini sağladım. Her seçimde 5 adet nesne seçilmiş olup olmadığını sorguladım ve 5 adet nesne seçildiği anda daha fazla nesne seçimini durdurmak adına butonlarısetEnabled(false) yaptım. Ardından projenin diğerlerine göre nispeten daha uzun ve zor olduğu, her adımda seçilen nesneler ile gerekli karşılaşmaları kodlamaya başladım. Oluşturduğum loop ve loop2 arraylistleri sayesinde kullanıcının ve bilgisayarın elindeki nesneleri 5'er 5'er seçtiğini kontrol ettim. Eğer istenilen nesnenin seçilmesinde sorun yoksa bu iki nesnenin class isimlerini karşılaşmanın kim ile kim arasında olacağını anlayabilmek adına etki hesapla metoduna gönderdim. buradan gelen float değerleri etki ve etki2 adındaki 2 değişkene atadım. Bunları kendi aralarında büyüklük küçüklük bakımından karşılaştırıp hangi nesnenin kazandığını anlamak adına w ve w1 adındaki integer değişkenlere attım. Ardından karşılaşan nesneler hakkındaki dayanıklılık ve seviye puanı gibi bilgileri JTextArea'da gösterdim. Bunun ardından durum güncelle metoduna etki değerleri ve kazanıp kazanmama durumlarını göndererek bir sonraki el için nesnelerin durumlarının güncellenmiş oldum. Bunun peşine bir sonraki tur için seçimlerin doğruluğunu kontrol ettiğim loop ve loop2 listlerine gerekli kontrol ve düzenlemeleri yaptım. Bu adımların ardından 2 nesnenin karşılaşması sonuçlanmış oldu. Silinmesi gereken nesneler silindi veya seviye atlaması gereken nesneler seviye atmış oldu. Ve bir karşılaşma sonunda 3 farklı oyun sonu kontrolü için if else'ler tanımladım. Bu durumlar max hamle koşuluna ulaşılmış olması, bilgisayarın nesnelerinin bitmiş olması ve kullanıcının nesnelerinin bitmiş olması durumlarıdır. Bu durumlardan herhangi biri gerçekleşmiş ise oyun durdurulur ve ekrana kimin kazandığı ve her iki tarafın skorları bastırılır. Ve kullanıcı-bilgisayar oyunu bu şekilde sonlanmış olur. Bilgisayar bilgisayar oyununda ise bu temel olarak 2 bilgisayar karşı karşıya gelir. Bu oyunu başlatmak için başlangıçta B-B adlı butona basılması gerekir. Ardından nesne seçimi butonu ile her iki bilgisayar için de 5'er nesne her butona basımda seçilir. 5. nesne seçildikten sonra oyna butonu ile her ikki bilgisayara da nesneleri içerisinden seçim

yaptırılıp karşılaştırılır ve sonuçlar ekrana basılır. Galibi belirleme işlemi kullanıcı-bilgisayar oyunuyla temelde aynıdır. Hangiarafın yendiği belirlendikten sonra bilgisayar bilgisayar oyununu bitmiş olur. Taş kağıt makas oyunun yöntemleri kısaca bunlardır.

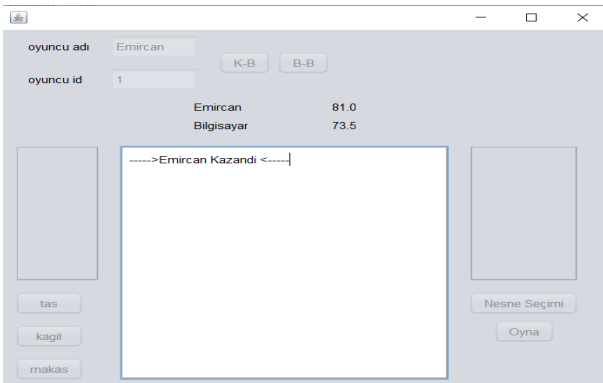
IV. SONUÇ

Günün sonunda bu biten bu proje ile oop tabanlı kuralları değiştirilmiş bir taş kağıt makas oyunu elde ettik. Kullanıcı-Bilgisayar veya Bilgisayar-Bilgisayar şeklinde oynama seçeneklerinin olduğu oyunda bu seçimi kullanıcı oyuna başlamadan seçiyor. Eğer kendisi oyunu oyanayacaksa K-B butonuna basmasının ardından soldaki liste bölümünün altında bulunan 3 buton yardımıyla tas kağıt veya makas nesnelerinden toplam 5 adet seçiyor ve oynamaya hazır oluyor. oyun hamle sınırına ulaşana kadar veya bir kullanıcının elindeki nesne bitene kadar devam ediyor. kazanan belli olduktan sonra hem kazanan taraf hem de her iki tarafın skorları ekrana çıktı olarak bastırılıyor. Ve oyun sonranmış oluyor. Burada projede dikkat edilmesi gereken noktalardan birkaçı oluşturulan nesnele ve kullanıcılar için doğru methodları doğru şekilde çağırmak ve oluşturulan classlarda override methodlara dikkat etmek idi.

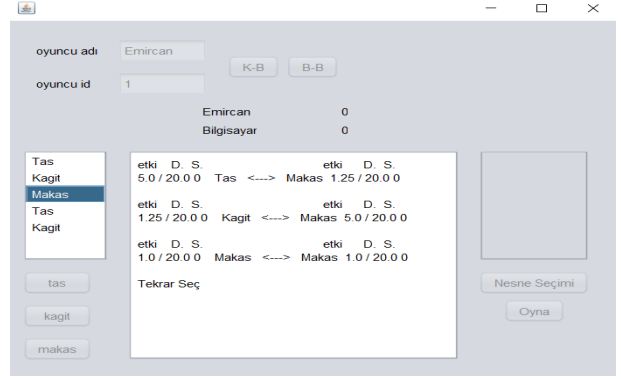
V. DENEYSEL SONUÇLAR



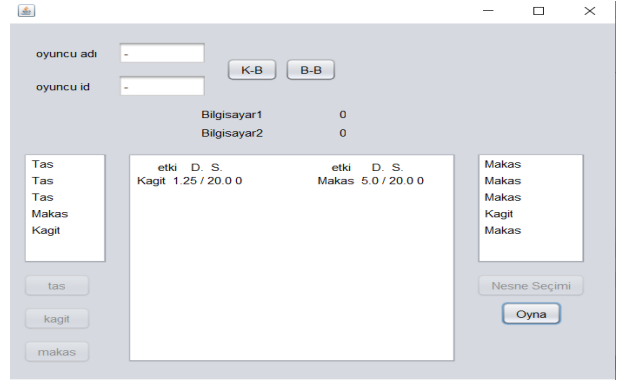
Kullanıcı-Bilgisayar Oyunu



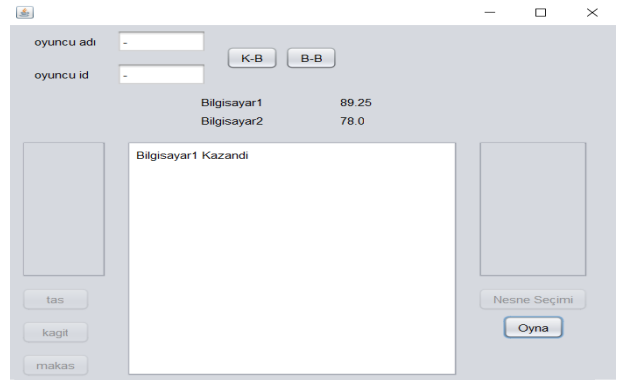
Kullanıcı Galibiyeti



Geçersiz Nesne Seçimi



Bilgisayar-Bilgisayar

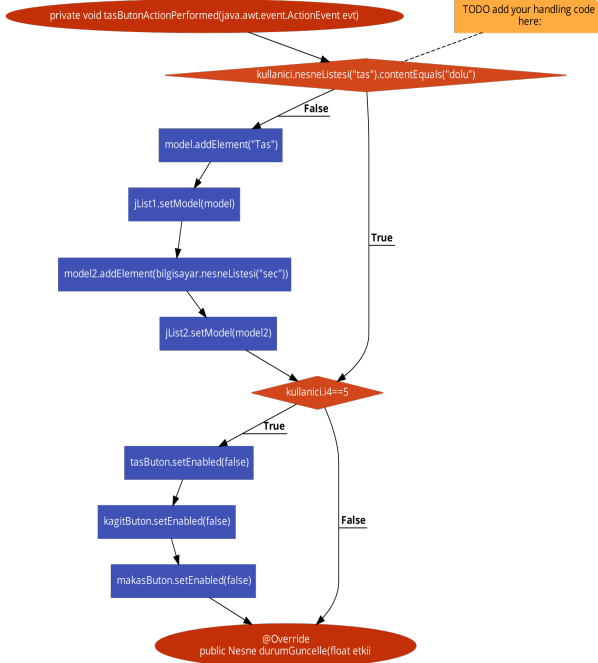


Bilgisayar-Bilgisayar Kazanımı

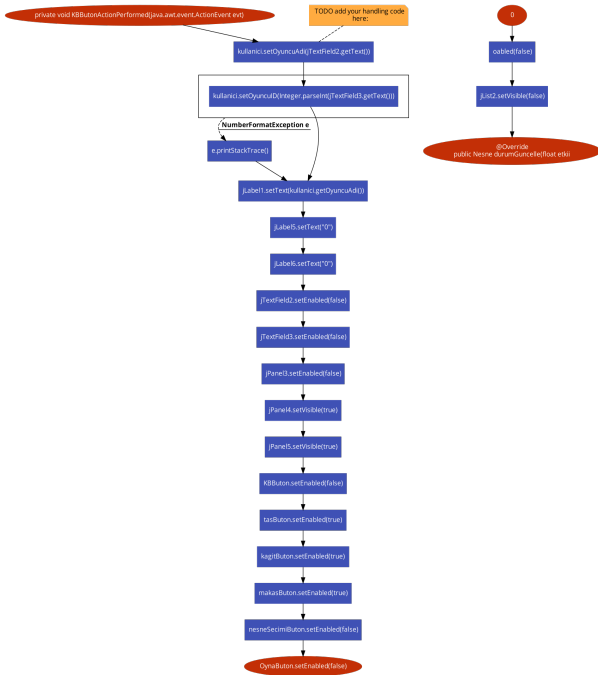
REFERENCES

- [1] <https://kamer-dev.medium.com/oopnin-4-ana-prensibi-encapsulation-inheritance-abstraction-polymorphism-712ed2fbac7e>
- [2] <https://www.w3schools.com/java/java-oop.asp>
- [3] <https://medium.com/@kahil-kubilay/java-oop-temelleri-5ebd0b7d062b>
- [4] <https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>
- [5] <https://www.freecodecamp.org/news/object-oriented-programming-concepts-java/>

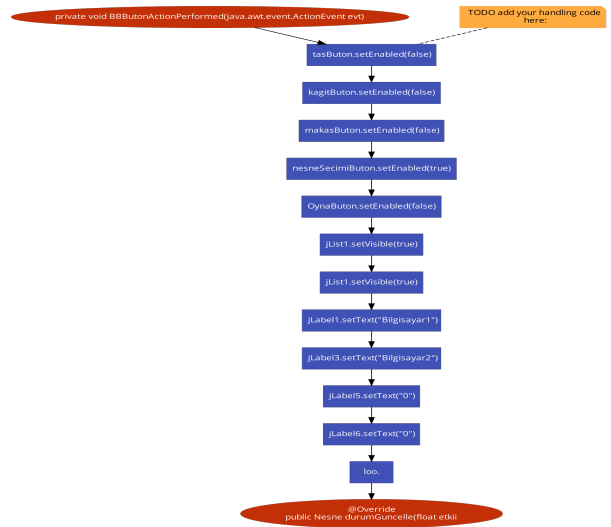
VI. AKIŞ DİYAGRAMLARI



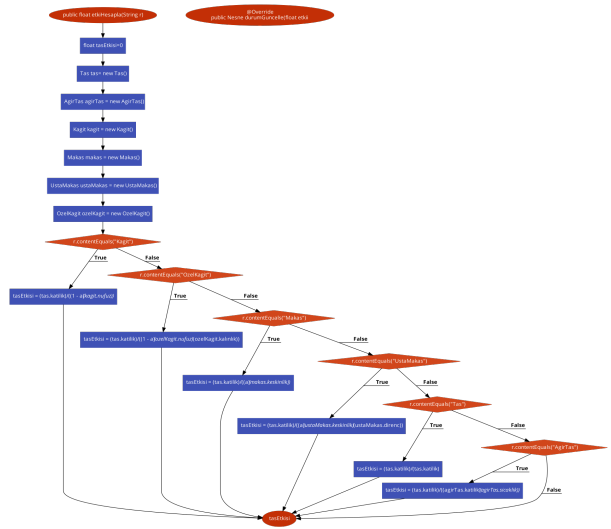
tas Butonu Akış Diyagramı



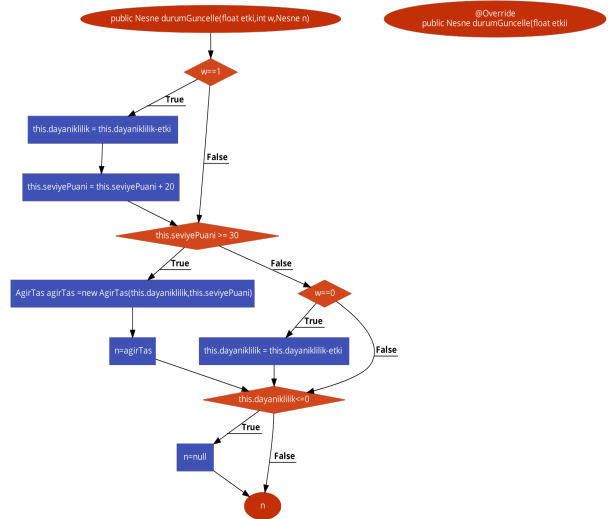
KB Butonu Akış Diyagramı Akış Diyagramı



BB Akış Diyagramı

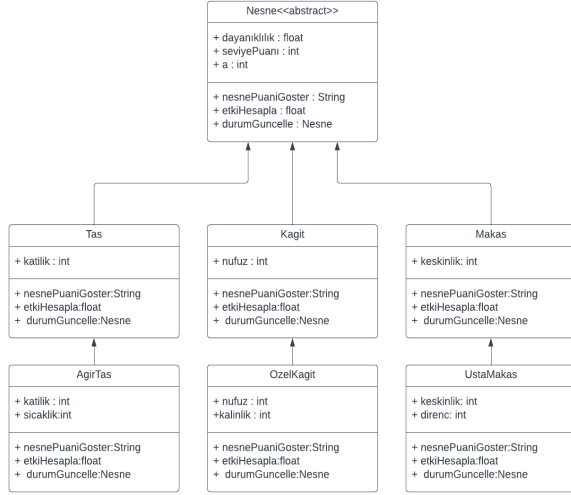


etkiHesapla Akış Diyagramı

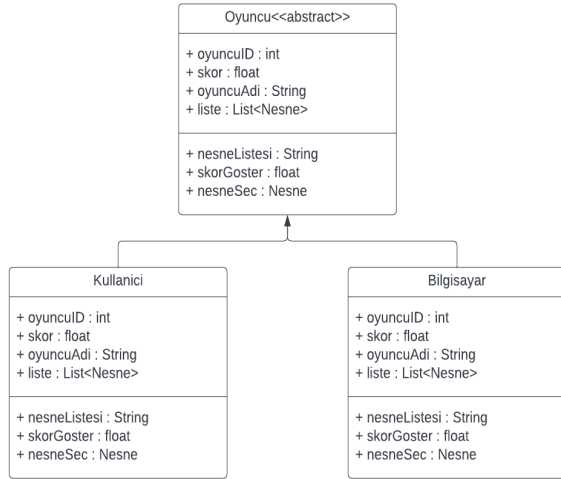


durumGüncelle Akış Diyagramı

VII. UML DİYAGRAMLARI



UML-1

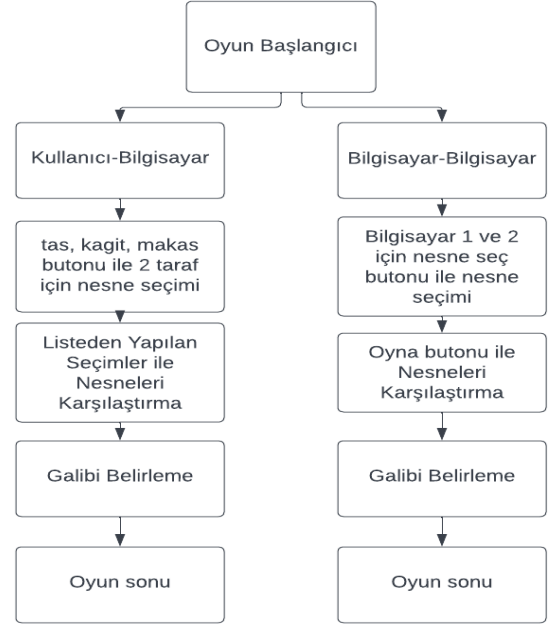


UML-2

VIII. KATKILAR

Bu projede nesneye yönelik programlama hakkında geniş bilgiler edinildi. kodlamayı nesneye yönelik yaparak modülerlik ve rahatlık sağlandı. Geliştirme sırasında java dili için bolca pratik yapılma imkanı oldu. Projede ayrıca kullanıcı ile etkileşim kurulabileceği, seçim alınabileceği bir arayüz de tasarlandı. Ve bu sayede yazılan kod görselleştirilmiş oldu. Bunun için java swing kütüphanesi kullanıldı ve öğrenilmiş oldu. hangi componentlerin hangi durumlarda yararlı olduğu veya nasıl kullanılabileceği gibi detaylar araştırılarak öğrenildi. Nesneye yönelik programlama prensiplerine uyulduğundan program normalde içereceğinden daha az kod satırı içerdi. Ve bu da bizlere oyunu daha kısa sürede ve daha rahat biçimde kodlamamızı sağladı.

IX. TABLOLAR



Tablo