**Bursa Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü 2021-2022 Bahar Dönemi Nesneye Yönelik Programlama Dersi Proje Ödevi**

**Öğrenci Adı: Emirhan**

**Öğrenci Soyadı: BARAN**

**Öğrenci Numarası: 20360859005**

**Proje Adı: Macera Oyunu**

**Dersi Veren Dr. Öğr. Üyesi: Ergün GÜMÜŞ**

**Tarih: 10/06/2022**

**Proje İçerik**

**Projemi tasarlar iken toplamda 15 tane(Character,Enemy,Sovale,StrongEnemy,WeakEnemy,CSVGa-meData,GameData,ConsolView, BattleMotor,Chance,FileManager,FileReader) sınıf ve 4 tane paket(Characters,GameData,GameView,Others) kullandım.**

**0-Characters**

**Oyunumuzda oluşturulan tüm karakterlerin atası olan "Character" sınıfı oluşturulan karakterin adını tutan "name" ,Id'sini tutan "id" ve karakterin aktif olup olmadığını kontrol eden "active" değişkenleri bulunmaktadır ve bu değişkenlerin private olması nedeniyle bunlara ait olan getter ve setter methodları bulunmaktadır.**

**1-Şövalye Sınıfı**

**Projemin bu sınıfında şövalyenin adını temsil eden "name" şövalyenin id'sini temsil eden "id" ve şövalyenin aktifliğini temsil eden "active" adlı değişkeni , şövalyenin sağ olup olmama durumuu temsil eden "status" ve başarılı vuruşlarını saymak için "başarılıVurus" adlı değişkenleri bulundurdum.**

**Şövalye sınıfında kullandığım konstrüktör ise ata sınıftan(Characters) name ve id alırken bunlara ek olarak vuruş hakkı, status ve “başarılıvuruş” şövalye nesnesi oluşturulduğunda otomatik olarak atadım.** **Şövalyemizin şans değişkeni "Chance" sınıfından bir nesne olup şövalye oluşturulurken rastgele olarak iyi veya kötü olarak şans atanır.Eğer iyi şans gelir ise şövalyeye ek olarak 1 veya 2 tane ek vuruş hakkı sağlanır.Private olan değişkenlerimin getter ve setter methodlarını yazdım.Bunlara ek olarak create şövalye fonksiyonu ile rastgele tane şövalye oluşturdum. Bu şövalyeleri oluşturuken en az en fazla tane güçlü şövalye olması kuralını dikkate aldım bu fonksiyon static olup return olaral ArrayList<Şövalye> geri döndürmektedir.**

**Static yapamamın sebebi elimizde hiç şövalye olmadan bize tane şövalye oluşturma imkanı vermesidir.**

**2-Enemy Sınıfı**

**Projemin bu sınfında şövalyelerin savaşacağı düşmanların sınıfını temsil etmektedir.**

**Bu sınıfın içinde şövalye sınıfında olduğu gibi ata sınıf(Character)'tan miras alınan ve düşman'ın ismini temsil eden "name" ile düşmanın Id'sini tutan id isimli değişken bulunmaktadır.Bunlara ek olarak düşmanın ölü mü yoksa sağ mı olduğunu gösteren "status" ve düşmanın güçlü mü yoksa zayıf mı olduğunu gösteren "enemyType" değişkenleri bulunuyor.**

**Enemy sınıfımızın bir tane konstrüktörü bulunuyor ve bu konstrüktör içeriğinden ata konstrüktörden "name" ve "id" değişkenlerini alırken bunlara ek olarak bünyesinde düşmanın ölü olup olmadığını tutan status değişkenine otomatik olarak "true"(sağ) değeri otomatik olarak atanır.**

**Değişkenlerimiz private olduğu için bu değişkenlerin getter ve setter methodları bulunmaktadır bunlara ek olarak tıpkı şövalye sınıfında olduğu gibi düşmanların yaratılması da bu sınıf içinde gerçekleşmektedir.Düşmanları oluştururken düşman sayısı 3-5 arasında iken en az 1 en fazla 2 güçlü düşman olması dikkate alınmıştır.**

**3-StrongSovalye-WeakSovalye Sınıfı**

**StrongSovalye ve WeakSovelye sınıfı Sovalye sınıfından miras alan bir sınıf olarak konstrüktöründe Sovalye sınıfından farklı olarak şövalyenin tipinin belirlenmesi gerçekleşir.**

**4-StrongEnemy -WeakEnemy Sınıfı**

**StrongEnemy ve WeakEnemy sınıfı Enemy sınıfından miras alan bir sınıf olarak konstrüktöründe Enemy sınıfından farklı olarak şövalyenin tipinin belirlenmesi gerçekleşir.**

**5-BattleMotor Sınıfı**

**"Others" adlı paketimizin içinde bulunan BattleMotor şövalyeler ile düşmanların savaştığı sınıftır.BattleMotor sınıfımızda savaşacak olan şovalyeyi tutması için "sovalye" ve şövalyeye karşı olacak düşman için "enemy" adlı değişkenler bulunmaktadır.Bunların yanında savaştan sonraki durumu(örn: şövalyenin ölmesi...) durumlarını kontrol etmek için şövalyelerin ve düşmanların içinde bulunduğu sınıfı tutan gW2 nesnesi bulunur.**

**Bu sınıfta bulunan "Fight" adlı method void tipindedir ve parametre olarak "Sovalye" ve "Enemy" tipinde olan bir arraylist ve yine "Sovalye" ve "Enemy" sınıfından bir nesne kabul eder burada kullanıcı tarafından seçilen şövalye ve düşman savaşır ve oluşacak sonuçlara(Galibiyet,Malubiyet) göre işlemler gerçekleştirilir.**

**6-Chance Sınıfı Sınıfı**

**"Other " paketinde bulunan bir diğer sınıf olan bu sınıfta şövalyelere ek vuruş hakkı sağlayabilecek olan "goodChance" ve bunun aksine şövalyenin vuruş hakkını elinden alabilecek olan kötü şansa sahiptir."Chance" sınıfında bulunan "asSignChance" methodu int bir değer döndürüp bir "Sovalye" nesnesini kabul etmektedir bu method kullanıcı eğer iyi şansa sahip ise çağrılır eğer şövalye iyi şansa sahip ise %10 ihtimalle ek olarak iki vuruş hakkı verir iken %90 ihtimalle o şövalyeye1 ek vuruş hakkı sağlar.**

**7-CSVGameData Sınıfı**

**Projemin GameData sınıfı ArrayList olarak tüm şövalyeleri,aktif olan şövalyeleri,düşmanları ve aktif olan düşmanları alır ve dosya yazdırma işlemlerine ön ayak olma görevindedir.**

**8-CSVGameData Sınıfı**

**CSVGameData sınıfı GameData sınıfından kalıtım alır ve asıl olarak dosya işlemlerinin yapılacağı kısımdır.** **CSVGameData sınıfı konstrüktöründe GameData sınıfına ek olarak BattleMotor sınıfının bir örneğini taşır.** **Sınıfta bulunan methodlardan biri olan "openTheFile" methodu verileri yazmak için kullanacağımız dosyayı açmamızı sağlar.Bunun tam tersi olarak “closeTheFile” methodu açılan dosyayı kapatmamızı sağlar. Bir diğer method ise "writeSovalyelertoFile" methodudur bu method başlangıçta oluşan şövalyelerin dosyaya yazılmasını sağlar.”writeSovalyelertoFile” methoduna benzer olarak yaptığım “writeEnemiestoFile ” methodu ise başlangıçta oluşturulan düşmanları dosyaya yazdırmak için kullanılır ve bir diğer önemli method ise “neOldutoFile” methodudur.Bu method savaş sonrası gerçekleşen olayları(Örneğin: 5. Round’da şövalye0 düşman1’i vurarak öldürdü) dosyaya yazdırmamızı sağlar.**

**9-GameView Sınıfı**

**Gameview sınıfı oyunun kullanıcı tarafından kullanılacak kısmı olup bünyesinde 10 tane method bulundurmaktadır.Bu methodlardan ilki olan “showTheMainMenu” methodu oyunu oynayan kullanıcının seçim yapabilmesi için ona bir menü tedarik eder.Kullanıcı bu menüden seçim yaparak oyuna şekil verir.Bir diğer method ise “showTheSovalyeler” dir bu method kullanıcının tüm şövalyeleri görmesini sağlar benzer bir method olan “showTheActiveSovalyeler” kullanıcının aktif olan şövalyeleri görmesini sağlar benzer methodlar düşmanlar için de yazılmıştır.”showtheSovalye” ise kullanıcının görnek istediği şövalyeyi tek olarak ekrana getirir.”changetheSovalye” ise kullanıcın seçeceği güçlü bir şövalyeyi 2 tane zayıf şövalye karşısında değiştirmeye yarar.Kullanıcı “help” methodunu kullandığı zaman ise oyunun kuralları ekrana gelir.”makeChoicefromMenu” methodu kullanıcının sürekli seçim yapmasını sağlar.Son olarak “showtheGameStatus” methodu ise oyunumuzun bitip bitmediğini ve kazanıp kaybetmediğimizi belli etmeye yarayan methotdur bu method savaştan sonraki her an için çağırılır ve sonucu kontrol eder ve kullanıcıya sonuca göre bilgi verir.**

**10-Main Sınıfı**

**Son olarak main sınıfı oluşturduğumuz tüm bu sınıfları birbirine bağlamamızı sağlayan ve bu bağladığımız sınıfların methodlarını kullandığımız sınıftır.**