Berkay Akar

WAR GAME PROJE DÖKÜMANI



İçindekiler

[GİRİŞ 2](#_Toc105964617)

[KATMANLAR 2](#_Toc105964618)

[Entity Layer: 2](#_Toc105964619)

[Another Library: 2](#_Toc105964620)

[Core: 2](#_Toc105964621)

[Services: 3](#_Toc105964622)

[UI 3](#_Toc105964623)

[SÜREÇ İŞLEYIŞI 3](#_Toc105964624)

# GİRİŞ

Bu Proje .Net Framework Core 5.0 ile geliştirilmiş olan, sıra tabanlı bir savaş sistemi olarak tasarlanmış olan bir savaş oyunudur. Süreçlerin nasıl işlediğine dair tüm süreçler bu döküman üzerinde açıklanacaktır. Öncelikle Classlibrary yapılarını (katmanlar) ve işlevlerini tanıyalım.

# KATMANLAR

Gerek eğitim süreçlerinde gerekse küçük çaplı staj tecrübelerim neticesinde ve online platform eğitimleri süreçlerinde anlatıldığı ve kendi yorumumuda katarak kurguladığım bu senaryoyu sizlere burada anlatmak istedim.

## Entity Layer:

Litaratürde çok katmanlı mimari (n-tier architecture) olarak geçmekte olan bu yapıda varlılarımızı (genellikle ben veritabanı modellerimi ve DTO süreçlerinde kullanılacak olan manipüle edilmiş, verinin sadece formatlanarak taşınmasında rol oynayan süreçlerini temsilen bu katmanı kullanmaktayım.) tanımladığımız bir katmandır. İmzalarını taşıyabilmek adına Interfacelerini abstract klasöründe taşırken bu imzaların doldurulmuş, guruplanmış bir kümenin her bir örneğini concrete klasörü içerisinde gerçekleştirmekteyim. Proje Veritabanı gerekmeksizin geliştirilmesinden ötürü bu sınıfların ve imzaların gerçek helleri bu aşamada konulmamış olup isteğe bağlı olarak ilerleyen süreçlerde implamente edilebilmesi adına tanımlanmış, boş bırakılmıştır.

## Another Library:

Gerçek hayatta bir oyun yazma tecrübem bulunmamaktadır. Fakat Çok katmanlı mimari süreçlerinde aynı işlevlere (amaçlara) hizmet eden yapıların bir arada bulunulması düşüncesinden kaynaklı olarak efektleri, farklı kütüphanelerle etkileşime geçilebilecek ara bağlantıları bu katman vastısı ile gerçekleştirilebileceğini düşünerek bu şekilde tanımlama gerçekleştirdim.

## Core:

Core katmanı tüm proje boyunca ortak olacak öğelerin tanımlandığı bir yapı olarak düşündüm. Gerçek manada tam olarak çok katmanlı mimari süreçlerinde nereye hizmet ettiğini bilemediğimden ötürü tüm süreçlerde ortak modellerin bu alanda tanımlanmasını uygun gördüm.

## Services:

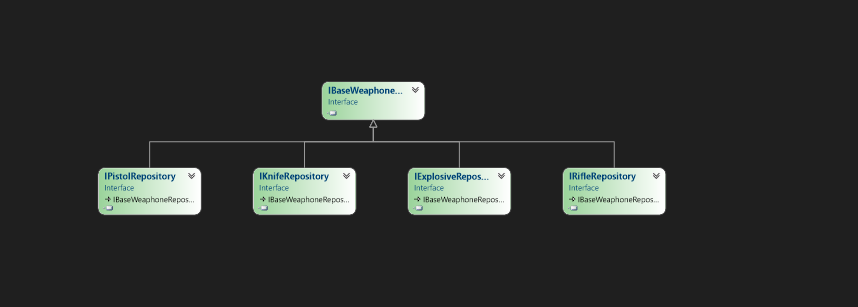
Bazı kişilerce bu katman business layer olarakta tanımlanmaktadır. Ben burada service kısmı olarak tanımladım. Iş yüklerinin son uygulama kısmında süreçleri karmaşıklaşıtmasını önleyebilmek, daha kolay kod okunabilirliğini arttırabilmek makasdı ile süreçlerde iş yüklerini bu katmana tanımlamış bulunmaktayım.

## UI

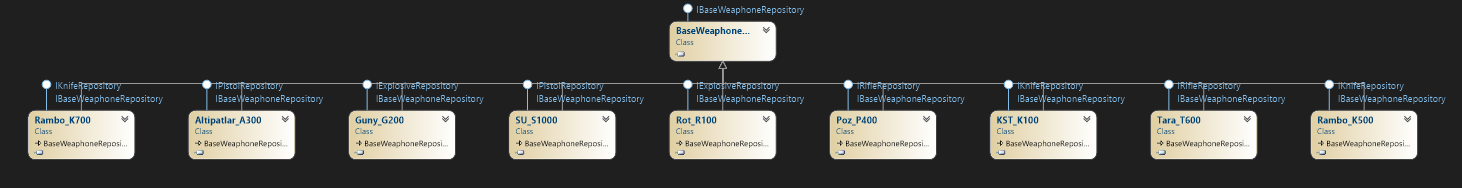
Son katman olarakta tüm arkada gerçekleşen süreçlerin son kullanıcıya sunumunda kullanılacak olan araçlar, çerçeveler ve kullanıcı etkileşimli uygulamalar olarak tanımlamış bulunmaktayım. Süreçlerin gerçekleştirilmesinde ana amacın sınıfların birbirleri ile etkilşimi olarak düşünülmesinde kaynaklı olarak console uygulaması kullanılarak son katman geliştirlmiştir.

# SÜREÇ İŞLEYIŞI

Bir Önceki Bölümde katmanlardan ve katmanların yüzeysel olarak amaçlarından bahsetmiştik. Bu aşamadan sonra süreci daha detaylı ve hiyerarşik bir biçimde işlenecektir.



Bir Base sınıf oluşturabilmek ve o base sinifi bir Iterface içerisinden implamente edebilmek adına IBaseWeaphones isimli bir adet interface oluşturulmuştur.



Tüm silahların özünde bir silah olduğunu belirlemek ve bundan implamente olan diğer interface lerin birer kategori olduğunu belirtmek maksadı ile böyle bir çalışma gerçekleştirilmiştir. IBaseWeaphones isimli interface içeriği ise :

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu şekilde doldurulmuştur. Tüm silah süreçlerine bakıldığında bir jarjör dinamiği (bıçakta bu durum körelme ile betimlenmiş ve tek kullanımlık olarak ele alınmıştır.) ve mermi sayısı hepsinde ortak özellik göstermektedir. Bunlardan İmplamente alan diğer interfacelere bakıldığında ise bir örnek aşağıda belirtilmiştir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Diğer interfaceler de benzerlik gösterdiğinden ötürü bu aşamada isim ve model bilgileri enum yapıları halinde tutulmuştur.

Tüm silahların özelliklerinin tanımlandığı Base Class ise IBaseWeaphones isimli interface den kalıtım alarak şu şekilde tanımlanmıştır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu aşamadan sonra her bir silahın bilgilerinin girilmeden sadece sınıfının bir örneği alınarak kullanılabilmesi adına tüm silahlar için birer sınıf açılmış ve yapıcı metodları içerisinde özellikleri kendilerine ait ana sınıftan gelen bilgileri interface içerisindeki değişkenleri vasıtası ile doldurulmuştur.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resimde Gözüken Rot\_T100 isimli silah sınıfı bu iş sürecine bir örnektir.

Şehirler işleyişinden de bahsetmemiz gerekirse tüm şehirlerin bir adet ana sınıfının olduğu bir yapı bulunmaktadır.

metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

Tüm haritalarda gerçekleşecek süreçlerde harita ismi , düşman sayıları, kendi oyuncumuzun o haritada olması gibi özellikler tanımlanmıştır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Her bir şehir bilgisinin kendisine ait bir sınıftan örneği oluşturularak bu şekilde işleyiş gerçekleşmiştir.

Beraberinde Tüm iş süreçlerinin kendisine ait bir manager isimli sinifinin olduğu alanlarda iş kuralları yazılarak ve son katmandan bu methodlar çağrılarak çok katmanlı mimari sürecine sadık kalınması amaçlanmıştır.