Berkay Akar

WAR GAME PROJE DÖKÜMANI



İçindekiler

[GİRİŞ 2](#_Toc105936571)

[KATMANLAR 2](#_Toc105936572)

[Entity Layer: 2](#_Toc105936573)

[Another Library: 2](#_Toc105936574)

[Core: 2](#_Toc105936575)

[Services: 2](#_Toc105936576)

# 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 businesslayer 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.