Report

Case'de maalesef büyük bir zaman sıkıntısı yaşadım. Oldukça geniş bir zaman aralığı verdiğinizi düşünüyorum fakat ben bazı sebeplerden dolayı sadece son gün düzgün bakabildim. Bonuslar hariç istenilen şeyleri yaptığımı düşünsem de kodumun pek clean olduğunu ya da design pattern'ları yeterince kullanabildiğimi düşünmüyorum. Aşağıdaki paragraflarda genel yapımdan bahsederken, ayrıca neler yapılabilirdi konusundan da bahsedeceğim.

Genel olarak yaptığım küçük case'lerde ServiceLocator design pattern'i ile erişim sağlayıp hallediyorum. Bu case'de hem ServiceLocator kullanarak erişim sağladım, hem de Event'leri kullanarak iletişim sağladım. Fakat genel olarak oyunun logic'i sahnede bulunan MonoBehaviour'lar üzerinden döndüğü için, kolaylık olması açısından SerializeField ile referanslayarak diğer GameObject'ler ile erişim sağladım.

Scriptable Object'leri genel olarak configuration olarak kullandım. Ayrıca persist edebilmesinden dolayı wallet mekanizmasının ana datası olarak kullandım. Böylece wallet'ta güncellenen reward değerleri sahne değiştikçe silinmiyor ve kayıt ediliyor.

Bu Scriptable Object'ler ile yapılan configuration'lar kullanılarak farklı design pattern'lar da yapılabilirdi. Örneğin Reward'ın Wallet'a gitme animasyonları başka Scriptable Object'ler ile stratejileştirilebilirdi (Strategy Design Pattern). Yani AnimationStrategy SO'sundan inherit eden farklı concrete scriptable object'ler yaratılabilirdi. Bu farklı scriptable object'ler farklı animasyon kodları içerirdi. Böylece istediğimiz reward'ın config dosyasındaki animasyon strategy'sini değiştirerek pluggable bir sistem oluşturabilirdik.

Ya da başka bir örnek vermek gerekirse, rulet bir base'den türüyor ve bir config dosyası var. Bu config dosyası şu an basit değerler taşıyor. Fakat bu config'e random seçimin nasıl olacağı (hepsi eşit olasılıkta mı, yoksa weighted mi), ya da videodaki gibi sırayla mı dolaşacağı yoksa benim yaptığım gibi rastgele mi dolaşacağı gibi behavior'lar SO ile stratejileştirip verilebilirdi. Şu anki kurduğum sistemde şekilleri üzerinden base'den farklı ruletler yapılır diye düşünmüştüm. Örneğin CircularRoulette gibi ayrı bir class da yaratılabilir. Fakat bunları da aslında stratejileştirip tek bir Roulette üzerinden de ilerleyebilirdik.

Bunun haricinde Factory Design Pattern kullanmayı planlamıştım. Reward'ların üretimi için düşünmüştüm, fakat oyunumda reward prefab'ini kullanmıyorum. Reward'larda hep configuration SO'ları üzerinden ilerledim. Normalde reward tile üzerinden seçildikten sonra ruletin ortasında reward prefab'ini yaratıp onun wallet'a gitme animasyonunu yapmayı planlamıştım. Fakat vaktim yetmediği için animasyonu yapamadım, factory'nin implementasyonunu da örnek olsun diye bıraktım.

İdeal bir factory bir pool kullanabilir. AssetLibrary'den aldığı prefab'ler ile Unity'nin built-in IPoolObject'ini yaratıp populate edebilirdim. Factory ise sadece bu pool'dan get yapardı ve işi bitince de recycle ile pool'a geri koyabilirdi. Bunu yaptığım başka bir case'den link

https://github.com/batuhanarat/AgaveGamesCase/blob/main/Assets/Scripts/Managers/ItemFactory.cs

Tile'ın kendi yapısı için basit bir state machine design pattern'ına benzer bir yapı kullandım. Buradaki yapı da çok daha geliştirilebilir, State'lerin kendisi ve o state'de iken neler yapabileceği olan Action'ları (animasyonları) ayrı ayrı Scriptable Object'ler ile encapsulate edip hem daha pluggable hem de daha object oriented bir yapı kurulabilirdi.

Bunu yaptığım başka bir case'den link
https://github.com/batuhanarat/AlpakaGamesCase/blob/main/Assets/Scripts/AI/Base/State.cs

UI objelerinde genelde MonoBehaviour üzerinden ilerlemeyi tercih ediyorum. Bunun sebepleri; canvas yapısının render sürecinin yeterince optimize olmaması, MonoBehaviour olarak yapılan UI objelerinin oyun içi diğer objelerle daha kolay iletişim kurabilmesi ve büyük şirketlerde genelde UI objelerinde 3 boyutlu animasyonlar kullanılması fakat bunların canvas'ta yapılamaması. Fakat bu case'de süre sıkıntısı yaşadığımdan UI ile alakalı kısımları canvas ile çözdüm.

Özet olarak tam olarak içime sinen bir case olmasa da, eğer case mülakatı almaya uygun görürseniz kendimi mülakatta daha iyi ifade edebileceğimi düşünüyorum. Uygun görmezseniz ise kod yapım ile birlikte başka neler yapılabilirdi konusunda feedback alabilirsem çok mutlu olurum. Şimdiden teşekkür ederim