

Uygulama Mimarisi

Inventory System

Nasıl Kullanılır:

Inventory System/Prefabs klasörü altındaki;
InventoryPage.prefab'i bir canvasın altına atadıktan sonra
InventoryData.prefab'i ve
InventoryPresenter.prefab'i de sahnede uygun bir yere atın.

InventoryPresenter içerisindeki **Cell Parent**'a InventoryPage'de bulunan
ScrollViewin altındaki **Content** nesnesini atayarak kurulumu tamamlayabilirsiniz.

Inventory.Instance.AddItem(ItemStack) ile item ekleyebilirsiniz

Inventory.Instance.RemoveItem(Item) ile item silebilirsiniz.

İtemler Nasıl Oluşturulur:

“ScriptableObjects/Item” context menüsü altında New Item seçeneği ile oluşturulabilir
İtemler Resources/Items klasörünün altında bulumak zorundadırlar.

Sınıf fonksiyonları ve açıklamaları:

Inventory.cs

Envanterin veri işlemlerinin yapıldığı sınıf

+AddItem(ItemStack itemStack):

Envantere item ekler, eğer item varsa sayısını artırır.

+RemoveItem(Item item):

Envanterden item siler, eğer item stack sayısı 1 den fazla ise 1 azaltır.

-UpdateSave()

Envanteri Json'a dönüştürür ve kayıt eder.

-LoadAll()

Envanteri Diskten yükler ve Json'dan geri dönüştürür.

InventoryData.cs

Envanterdeki itemların listesinin tutan sınıf

InventoryPresenter.cs

Envanterdeki itemların listesi güncellendiğinde arayüzü silip baştan oluşturan sınıf.

Item.cs

Item id'sini ve StackCount'ı tutan item sınıfı, id aracılığıyla ItemSO çekilir ve bu sayade itemSprite itemName gibi diğer bilgiler erişilir.

ItemCell.cs

İtemlerin envanterde görünüşü ve etkileşimi için oluşturulmuş. Frontend sınıf.,

ItemDatabase.cs

İtemleri Resources/Items altından çekerek veri tabanı oluşturur.

ItemStack.cs

ItemSO ve StackCount içeren struct yapısı.

Wheel of Fortune

Nasıl Kullanılır:

WheelOfFortune/Prefabs'in altındaki;

CooldownArea.prefab

WheelOfFortune.prefab

Pin.prefab

bir canvasın içine atılır

WheelOfFortune içinde items listesine ItemStackler oluşturularak ItemSO'lar ve gerekli bilgiler girilerek itemlar atanır.

Eğer Ödüllerin animasyonlu bir şekilde envantere girimesini istiyorsak WheelOfFortune'da InventoryButtonTransform objesini atamalıyız.

WheelOfFortune' nesnesinin altındaki Cooldown Timer'a değer girerek zamanlayıcıyı kuralıyız.

WheelOfFortune' nesnesinin altındaki CooldownDisplay.cs içindeki CooldownLabel'a **CooldownArea** nesnesindeki label'i atamalıyız.

Pin'e basınca çark dönsün istiyorsak pinin içindeki button componentinin OnClick Alanına WheelOfFortune nesnesinin altındaki WheelSpinner.Spin_Button fonksiyonunu atamalıyız.

Sınıf fonksiyonları ve açıklamaları:

WheelOfFortune.cs

Çarkın oluşturulduğu ve ödül işlemlerinin yapıldığı sınıf

WheelPiece.cs

Çarkın birimlerinin kontrollerinin, kurulumlarının ve yeniden pozisyonlanma işlemlerinin yapıldığı sınıf.

WheelSounds.cs

Çark dönerken çıkardığı sesi ve diğer seslerin kontrolü

WheelSpinner.cs

Çarkın dönme işlemlerinin yapıldığı sınıf. Event Triggerdan gelen OnDrag gibi fonksiyonlar burada dönüş mekaniğini ortaya çıkarır.

Dönüş için DOTween kullanılır. Yavaşlaması için OutQuint Fonksiyonu ease olarak kullanılır.

Çarkın dönüş izni veya şuan döndüğü için tekrar döndürülemeyecek olması bilgisi burada işlenir.

CooldownTimer.cs

Internet zamanı kullanarak girilen süre dolduğunda Action'lar tetikler

Bu actionlar

Public void ResetCooldown(Action action) -> sıfırlanıp tekrar zamanlayıcı kurarken ve

public void StartCooldown(Action action) -> uygulama yeniden açıldığında atanır.

CooldownDisplay.cs

CooldownTimer.OnCooldownDone

CooldownTimer.OnCooldownTick

Eventlerini dinleyerek Geriye kalan süreyi ekrana yazdırır.

NetTime.cs

Internette JSON formatında güncel zamanı çekmek için kullanılan class

Notification System

NotificationCenter.prefab'ını sahneye atarak aktif hale getirebilirsiniz.

Android için telefonda izin isteme channel'e kayıt olma gibi başlangıç işlemlerini yapar.

NotificationMaster.Instance.SetSpinNotification(timeSpan) ile girilen timeSpan sonra Spin Ready! notification'ı gönderir.

Bildirime tıklanarak oyun açıldığında nereye yönlendireceğini bu script karar verir.

SetSpinNotification(TimeSpan repeatTime) fonksiyonu çağrıldığında;

Daha önce zamanlanmış olan notificationları iptal ederek başlar. Daha sonra SpinReady bildirimini gönderir ancak bu bildirim tekrarlı değildir. Çünkü tekrarlı bildirim olarak gönderdiğimiz de bildirim tam zamanında ulaşmayı garanti etmez. Tekrarlı bildirim göndermeyi sağlamak için de. ikinci bir bildirim kayıt ederiz. o da ilk bildirimden iki TimeSpan sonra ilk defa çağırılmak üzere tekrarlı bir bildirimdir.

Push Notification System

Push notification sistemi notification sistemi altında PushNotificationMaster.cs ile kontrol edilir.

Ayarları ProjectSettings/Services/PushNotification altından Firebase Uygulamasının bilgilerini oraya girerek ve Unity Cloud Dashboard'ı altında Push Notification productsına Android kurulum bölümünden ayarlanır test notificationları yine oradan device token kullanılarak yapılır.