Şans Çarkı Oyunu Geliştirme Süreci

1. Frontend

- **Renk Paleti:** Clash of Clans ve Brawl Stars gibi oyunların renk paletleriyle uyumlu bir renk paleti seçildi.
- **Font:** GROBOLD isimli font kullanıldı ve fontun altına gölge eklenerek farklı bir varyant oluşturuldu.

1.1 Ana Menü

Şans Çarkı oyununun geliştirme süreci, arayüz ve frontend tasarımıyla başladı. İlk olarak, buton tasarımları ve kullanılacak renk paletleri belirlendi. Tasarımda, çift renk tonuna sahip, kenarları yumuşatılmış dikdörtgen butonlar tercih edildi.

- **Buton Tasarımı:** Çift renk tonlu, kenarları yumuşatılmış dikdörtgen.
- **Logo:** Ana menüye uygun bir logo tasarlandı ve bu logoya bir giriş animasyonu eklendi.
- **Giriş Animasyonu:** "Appear" olarak adlandırılan animasyon, ana menü logosunu ekrana getiriyor. Animasyon sona erdiğinde, logo "Idle" animasyonuyla hafif bir salınım yaparak ana menüye hareketlilik katıyor. Ayrıca, logoya tekrar tıklanırsa "Appear" animasyonu yeniden oynatılıyor.

1.2 Çark Sayfası

Çark sayfası, uygulamanın frontend ve kullanıcı deneyimi açısından en önemli bölümü.

- **Dinamik Çark:** Çark, kullanıcı tarafından eklenen item sayısına göre dinamik olarak parçalanabilir bir yapıya sahip olacak. Çarkın bölüm sayısı, eklenen item sayısına göre belirlenecek.
- **Değişken renk paletine sahip çark:** Çark tasarımı farklı renklerden oluşabilir. Bunun için ScriptableObjectslerden yararlanıp color palet isiminde bir scriptableObject oluşturuldu. Farklı renk paletlerine sahip çarklar oluşturulmasına izin verildi.
- **Home button ve Inventory Button:** Çark sayfasından çıkmak ve geri anasayfaya dönmek amacıyla bir geri buton ve çantayı açan inventory butonu eklendi.
- Çark ve buttonlar arasında kalan bölgeye çarkı bir sonraki çevirme hakkımıza kalan süreyi gösteren bir Geri Sayım alanı eklendi.

2. Item Sistemi

Uygulamada hem dinamik bir çark sistemi hem de bir envanter sistemi bulunduğu için, sağlam bir item sistemine ihtiyaç duyulduğuna karar verildi.

Scriptable objectler kullanılarak oluşturulan itemlar, item ismi ve item resmi gibi değişkenler içerecek şekilde yapılandırıldı. Bu itemlar hem çarkta hem de envanterde kullanılabilecek.

Uygulamamızdaki Item sistemi Scriptable Object temelli olup daha sonra oyun içinde Item classına dönüşmektedir. ItemSO (Item Scritable Object)'larda item ismi, id'si ve resmi tutulmaktadır. **ItemSO'ların hepsi Resources/Items klasörünün altında bulunmak zorundadır.** ItemSO'lar kendi id'sini kendileri atarlar.

ItemDatabase ismindeki class ItemSO'ların hepsini çeker ve kullanıma hazır hale getirir.

2. 1. ItemStack sistemi

Itemlar Çark'a ItemStack struct'ı olarak atanır. Bu struct ItemStackCount ve ItemSO olarak iki değer kabul eder. Bu sayede geliştirici isterse Çarkdaki örneğin bir diamond ödülünü 10 diamonda çıkararak. kullanıcının 10 diamond kazanmasını sağlayabilir.

3. Envanter Sistemi

Envanter sistemimiz Item classından oluşturulmuş nesneleri kayıt ederek kullanıcıya göstermeye yarayan bir sistemdir. Görsel ve veri olmak üzere iki katman içerir. Data katmanı üzerinden çekilen veya oluşturulan veriler. Görsel katman üzerinden arayüze dönüştürülür. Veriler JSON şeklinde kullanıcı yerel diskine kayıt edilir.

3. 1. Frontend

Grid layout, Scroll view ve Content Size Fitter gibi arayüz elementleri kullanılarak oluşturulmuş envanter sistemi için dinamik bir arayüz yapıldı. Açılış animasyonları kullanılarak hareketlilik arttırıldı.

4. Çark sistemi

Oyunun temel özelliği olan çark sistemi için çarkı oluşturma fonksiyon ile geliştirmeye başlanıldı. Çark her bir parçası bir Unitynin bir özelliği olarak resim tipi filled olan Radial360 fill methodlu ve fill amount'u 1 ile 0 arasında değer alan Image componentleri kullanıldı. Bu Fill Amount değeri kullanılarak 360 derece üzerinden çarkın değişken yapısına uygun farklı boyutlarda çark dilimleri elde edilebildi. Bu elde edilen dilimlerin rotasyonu Çarka atanan Item sayısına bağımlı olarak değiştirildi.

Çark dilimlerinin rengini belirlemek için ColorPalleteSO isimi bir Scriptable Object kullanıldı. Bu SO'nun içinde yer alan Color listesinden sırayla çekilen renkler sayesinde her bir dilim farklı bir renk olarak oluşturuldu.

4. 1. Çark döndürme

Çarkımızı döndürmek için iki temel yöntem belirlendi. Bunlardan ilki çarkın ortasındaki "Spin" yazılı butona tıklamak. Diğer ise çarkı çevirirek belirli bir kuvvetle fırlatmak.

Çarkı çevirme aşamasında kullanıcı deneyimini olumlu etkilemesi amacıyla. Kullanıcı çarkı çevirmeye başladığında çark kullanıcınının parmağını döndürme açısıyla eş açıda dönmeye başlar. eğer kullanıcı döndürme hızını artırıp çarkı bırakırsa çark kendi kendine hızlanarak dönmeye ve rastgele değer almaya başlar.

Çark yavaşlayıp bir noktada durduğunda durduğu açıya gören ödül belirlenir.

Envanter sistemine haber iletilip ödül envantere eklenir.

4. 2. Cooldown Sistemi

Çarkın isteninlen aralıklarla kullanılması sistemi için bir Zamanlayıcı sistemi geliştirildi. Bu zamanlayıcı sisteminin güvenli olması için mobil cihazın saati yerine internet saati kullanıldı. Çarkın kullanması üzerinden yeterli süre geçtikten sonra çark tekrar kullanılabilir hale gelir. Ayrıntılı bilgi için Uygulama mimarisi dosyasına bakabilirsiniz.

5. Notification Sistemi

Notification sistemi için Unity'nin sunduğu bildirim kütüphanesi kullanıldı. Bu kütüphane sayesinde android ve ios bildirimlerini tekrarlı veya zamanlı olarak ayaralanmış şekilde kullanabiliyoruz.

6. Push Notification System

Push Notification sistemi için ise yine Unity'nin Push Notification kütüphanesi kullanıldı. Firebase üzerinden oluşturulan uygulamanın WebApi'si project id ve name'i gibi bilgiler Unity üzerinden unity'e girildi. Ekstra yetkilendirme işlemleri Unity Cloud sisteminden Push Notification arayüzü üzerinden yapıldı.