

1. Hedef

Bu çalışmanın amacı, Unity oyun motorunu kullanarak C# dilinde "Nesne Tabanlı Programlama (OOP)" ilkelerine uygun şekilde Match-3 oyun prototipi geliştirmektir.

2. Teknik Detaylar

Geliştirme sırasında dikkat edilmesi gereken başlıca konular şunlardır:

- Kodun performanslı çalışması,
- Canvas elementleri olmadan 2D Sprite ile çalışılması,
- Görsel assetlerin optimize edilmesi,
- Overdraw ve Drawcall optimizasyonu,
- Projedeki klasörlerin, assetlerin ve prefablerin düzgün isimlendirilmesi.

Prototip, Android platformu için derlenerek "apk" formatında aşağıdaki e-posta adreslerine gönderilmelidir:

- ilyas@cratoonz.com
- volkan@cratoonz.com
- cagatay@cratoonz.com

Projenin kaynak kodları; yani Unity projesi, Github üzerinden yukarıdaki e-posta hesapları ile paylaşılmalıdır.

3. Süre

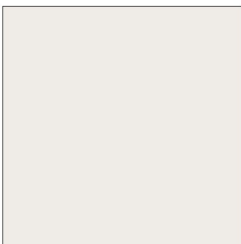
Çalışmanın tamamlanması için hedeflenen süre 7 gündür. Bu süreden sonra gönderilen çalışmalar, değerlendirmeye alınmayacaktır.

4. Yapılacaklar İşler

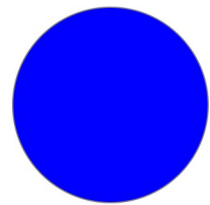
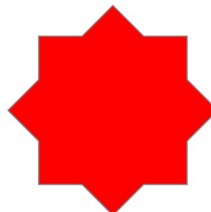
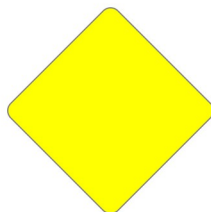
1. Board Oluşturma

Bu aşamada, Unity 2D Sprite'ları ile 8x8'lik bir board oluşturulması beklenmektedir.

Board, 8 satır ve 8 sütundan oluşmak şartıyla toplamda 64 boş karenin (tile) üzerinde 4 farklı renkte (sarı, mavi, yeşil ve kırmızı) oyun elementinin (drop) rasgele dizilmesinden meydana gelmelidir. Örnek olması açısından aşağıda tile ve drop görselleri verilmiştir.

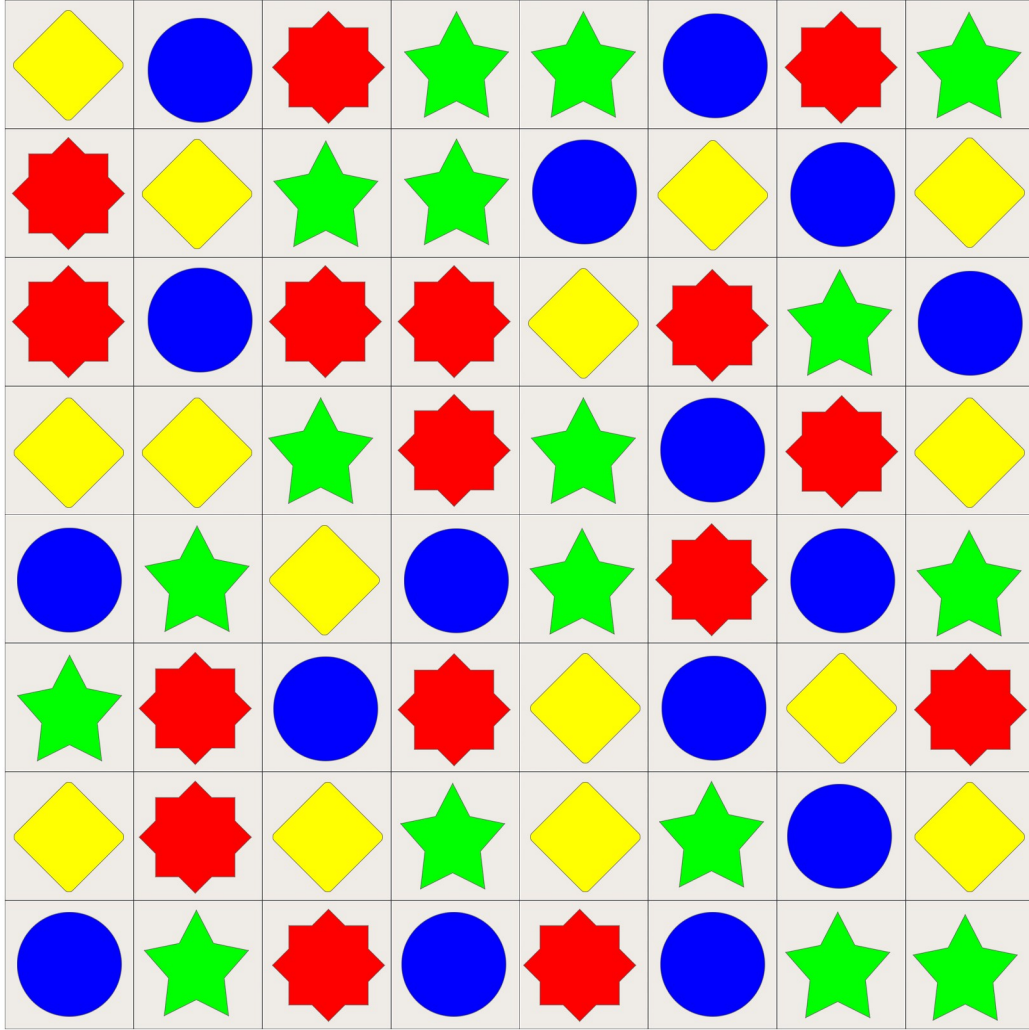


Tile



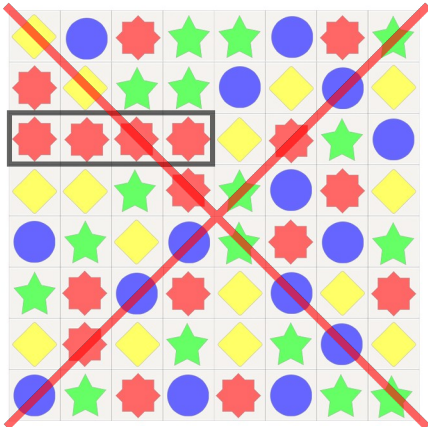
Renk Dropları

Board rasgele oluşturulduktan sonra aşağıdakine benzer şekilde görünmelidir.

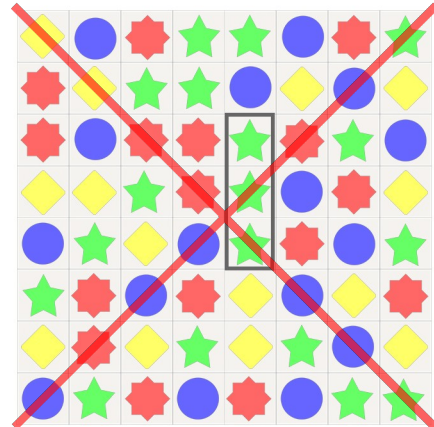


Örnek Board

Board oluşturma esnasında dikkat edilmesi gereken en önemli nokta, aynı renkteki 3 veya daha fazla dropun alt alta ya da yan yana **gelmemesidir**. Aksi takdirde board oluşturma işlemi hatalı olacaktır.



4 tane kırmızı drop yan yana dizildiği için bu board hatalı oluşturulmuştur.



3 tane yeşil drop alt alta dizildiği için bu board hatalı oluşturulmuştur.

| Match-3 Çalışması

2. Swipe Mekanığı

Oyunda eşleştirme yapılabilmesi için "swipe" hareketi gereklidir.

Swipe, board üzerinde herhangi bir dropun üzerine dokunulmasıyla başlar. Parmak kalkmadan alt, üst, sol ya da sağ tarafta kalan drop üzerine sürüklenir ve bırakılır.

Bu hamle sonucunda aynı renkli üç veya daha fazla sayıda drop yan yana veya alt alta denk gelmişse, geçerli bir eşleştirme (match) yapılmış olur. Swipe hareketinin başladığı drop ile bittiği karedeki drop, animasyonla yer değiştirir ve bu droplar boardtan kaybolur. (Video1)

Hamle geçersizse, swipe yapılan droplar eski yerlerine döner. (Video1)

Swipe hareketi, boş veya çapraz kareler arasında **yapılamaz**. (Video2)

3. Düşme Mekanığı

Geçerli bir match işleminin ardından boardtan kaybolan dropların yerini, üst satırdaki dropların doldurması gerekir.

Doldurma işlemi, boş kalan karenin üzerinde yer alan dropların, serbest düşmeye benzer şekilde animasyon yaparak alt tarafta kalan kareleri doldurmasıyla gerçekleşir.

Droplar aşağıya doğru düşerken aynı zamandan 8. satırdaki tileların üzerinden yeni dropların düşmesi gerekmektedir(Spawner mekanığı). En üst tilelardan düşecek olan droplar rastgele olabilir.(Video3)

En üst tileların spawner olup olamayacağını seçebilmeliyiz. Böylece o sütunda eşleştirme yaptıktan sonra o sütunun spawnerı olmadığı için yeni drop düşmeyecektir.(Video4)

İpucu: Bazen aynı gameObjesini sıklıkla **Instantiate** ve **Destroy** etmemiz gerekebilir (Örneğin silahlı bir oyun için mermi prefabi). Fakat Instantiate ve Destroy işlemlerinin çok sık kullanımı performans açısından iyi olmayabilir. O yüzden sürekli kullanmak istediğimiz objeleri Destroy etmek yerine pasif hale getirip sonra lazım olduğunda tekrar kullanabiliriz.

Not: Match olduktan sonra örnek giflerdeki gibi animasyon olmasına gerek yoktur. Objeler basit bir şekilde scale olup kaybolabilir. Droplar yukarıdan düşerken örnek giflerdeki gibi animasyonlu, esneyerek düşmesine gerek yoktur.

Başarılar