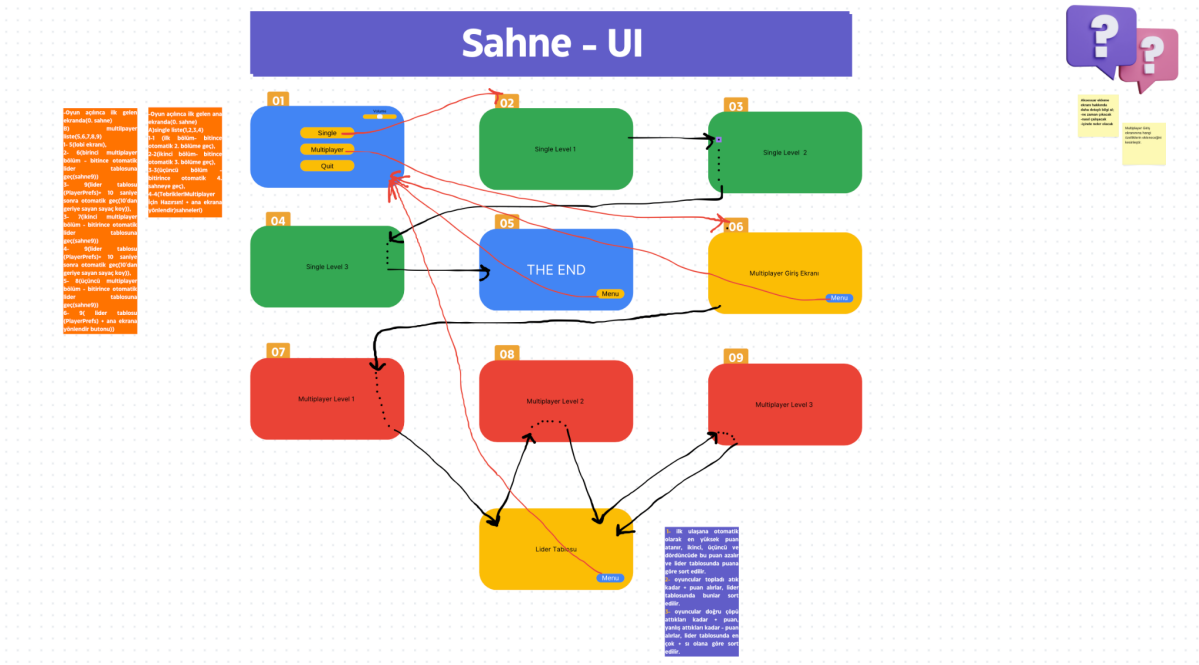
## Sprint Notları

**Sahne düzeni ve UI haritası çıkarıldı.**
**Detaylar:**
**-Oyun açılınca ilk gelen ana ekranda(0. sahne)**
**A)single liste(1,2,3,4)**
**1-1 (ilk bölüm- bitince otomatik 2. bölüme geç),**
**2-2(ikinci bölüm- bitince otomatik 3. bölüme geç),**
**3-3(üçüncü bölüm - bitirince otomatik 4. sahneye geç),**
**4-4(Tebrikler!Multiplayer İçin Hazırsın! || THE END + ana ekrana yönlendir)sahneleri)**

**-Oyun açılınca ilk gelen ekranda(0. sahne)**
**B) multilpayer liste(5,6,7,8,9)**
**1- 5(lobi ekranı),**
**2- 6(birinci multiplayer bölüm - bitince otomatik lider tablosuna geç(sahne9))**
**3- 9(lider tablosu (PlayerPrefs)+ 10 saniye sonra otomatik geç(10'dan geriye sayan sayaç koy)),**
**3- 7(ikinci multiplayer bölüm - bitirince otomatik lider tablosuna geç(sahne9))**
**4- 9(lider tablosu (PlayerPrefs)+ 10 saniye sonra otomatik geç(10'dan geriye sayan sayaç koy)),**
**5- 8(üçüncü multiplayer bölüm - bitirince otomatik lider tablosuna geç(sahne9))**
**6- 9( lider tablosu (PlayerPrefs) + ana ekrana yönlendir butonu))**

1. **Bölüm- ilk ulaşana otomatik olarak en yüksek puan atanır, ikinci, üçüncü ve dördüncüde bu puan azalır ve lider tablosunda puana göre sort edilir.**
2. **Bölüm- oyuncular topladı atık kadar + puan alırlar, lider tablosunda bunlar sort edilir.**
3. **Bölüm- oyuncular doğru çöpü attıkları kadar + puan, yanlış attıkları kadar - puan alırlar, lider tablosunda en çok + sı olana göre sort edilir.**



**1. ve 2. Bölüm mekanikleri yapıldı. (Mıknatıs - küçülme - yavaşlama - yeniden canlanma - hızlanma…)**

## - Magnet açıkken çöplerin oyuncuya doğru çekilmesi için;

```
IEnumerator FollowPlayer(GameObject trashObject)
{
    float followSpeed = 2f; // Takip hızı
    float minDistance = 1f; // Minimum takip mesafesi

    Rigidbody trashRigidbody = trashObject.GetComponent<Rigidbody>();
    trashRigidbody.isKinematic = false;

    while (trashObject != null) // Objeyi kontrol et
    {
        if (!trashObject.activeInHierarchy)
            break;

        Vector3 playerPosition = transform.position;
        Vector3 targetPosition = playerPosition + (trashObject.transform.position - playerPosition).normalized * minDistance;

        // Hedefe doğru hareket lerp
        trashObject.transform.position = Vector3.Lerp(trashObject.transform.position, targetPosition, Time.deltaTime * followSpeed);

        // player merkeze geldiğinde kontrol et
        if (Vector3.Distance(trashObject.transform.position, playerPosition) < 0.1f)
        {
            Destroy(trashObject);
            puan += 1;
            Debug.Log(puan);
            trashes -= 1;
            Debug.Log(trashes);
            break;
        }

        yield return null;
    }
}
```

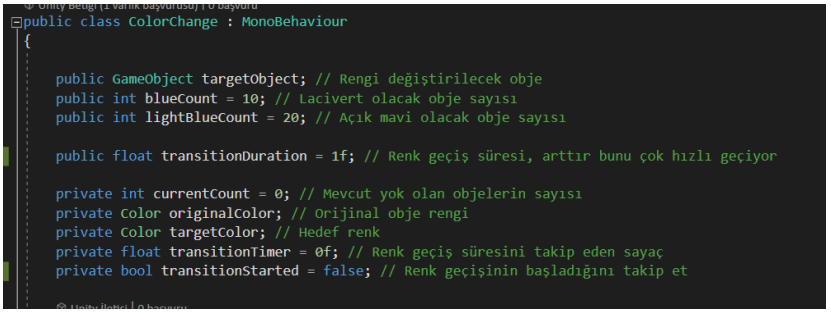
## - Magnet true false olduğunda neler olacak?

```
// -------------------------------2.BOLUM-----------------------------

if (magnet == true)
{
    magnetsecond -= Time.deltaTime;
    // 5 saniyeliğine artacak
    GetComponent<CapsuleCollider>().enabled = false; //normal collider devre dışı
    GetComponent<SphereCollider>().enabled = true;    //magnet collider etkin
    if (magnetsecond <= 0f)
    {
        magnet = false;
        GetComponent<CapsuleCollider>().enabled = true;  //normal collider etkin
        GetComponent<SphereCollider>().enabled = false;    // magnet collider etkin değil
        magnetsecond = 3.0f;   // yine 3 saniyeliğine eşitledik, sıfırda kalmasın
    }
}
```

## - Buzulları küçültmek için yazdığımız kod;

```
Unity İletisi | 0 başvuru
void Start()
{
    boxCollider = GetComponent<BoxCollider>();
    initialPosition = transform.position;
}

// Update is called once per frame
Unity İletisi | 0 başvuru
void Update()
{
    timer += Time.deltaTime;

    if (timer >= interval)
    {
        timer = 0f;
        ShrinkObject();
    }
}

1 başvuru
private void ShrinkObject()
{
    Vector3 scale = transform.localScale;
    scale.x *= scaleAmount;
    scale.z *= scaleAmount;
    transform.localScale = scale;

    // Box collider'ın boyutlarını güncelle
    Vector3 colliderSize = boxCollider.size;
    colliderSize.x *= scaleAmount;
    colliderSize.z *= scaleAmount;
    boxCollider.size = colliderSize;

    // Nesnenin yeri sabit kalsın
    transform.position = initialPosition;
}
```

- Renk değiştirmek için yazdığımız değişkenler;



```
public class ColorChange : MonoBehaviour
{

    public GameObject targetObject; // Rengi değiştirilecek obje
    public int blueCount = 10; // Lacivert olacak obje sayısı
    public int lightBlueCount = 20; // Açık mavi olacak obje sayısı

    public float transitionDuration = 1f; // Renk geçiş süresi, arttır bunu çok hızlı geçiyor

    private int currentCount = 0; // Mevcut yok olan objelerin sayısı
    private Color originalColor; // Orijinal obje rengi
    private Color targetColor; // Hedef renk
    private float transitionTimer = 0f; // Renk geçiş süresini takip eden sayaç
    private bool transitionStarted = false; // Renk geçişinin başladığını takip et
```

- İlk level ontrigger metodlarını kullanım şeklimiz;

```
private void OnTriggerEnter(Collider other)
{
    if(other.tag == "Engel")
    {
        MoveSpeed = 2;
    }


   if(other.tag == "Collect")
    {
        hizlandi = true;
    }

    //spawn için
    if (other.tag == "buz")
    {
        // Buz tag'li objeyegeldiğinde konum kaydet
        initialPosition = transform.position;
    }
    else if (other.CompareTag("deniz") && !isReturning)
    {
        // denize değersen geri dön
        isReturning = true;
        StartCoroutine(ReturnToInitialPosition());
    }

    if (other.tag == "pengus")
    {
        //4 saniye animasyon
        SceneManager.LoadScene("scene2");
    }
```