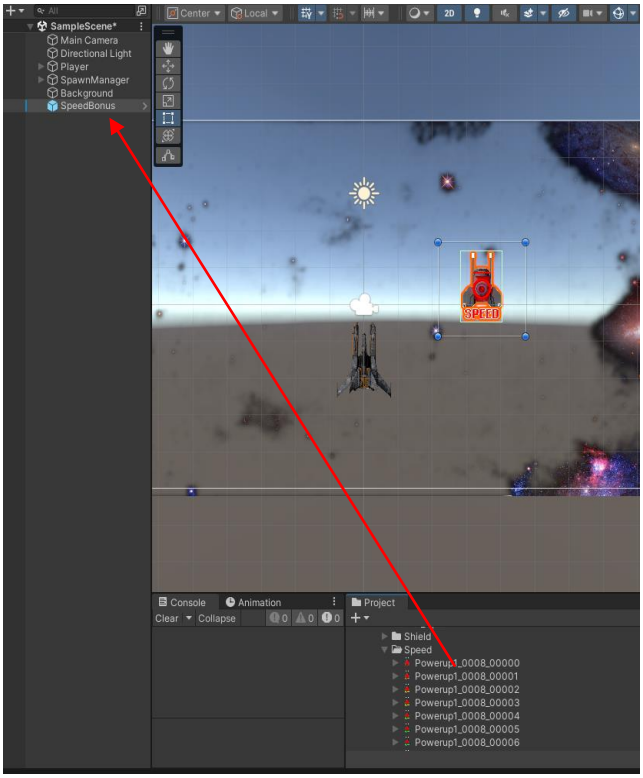
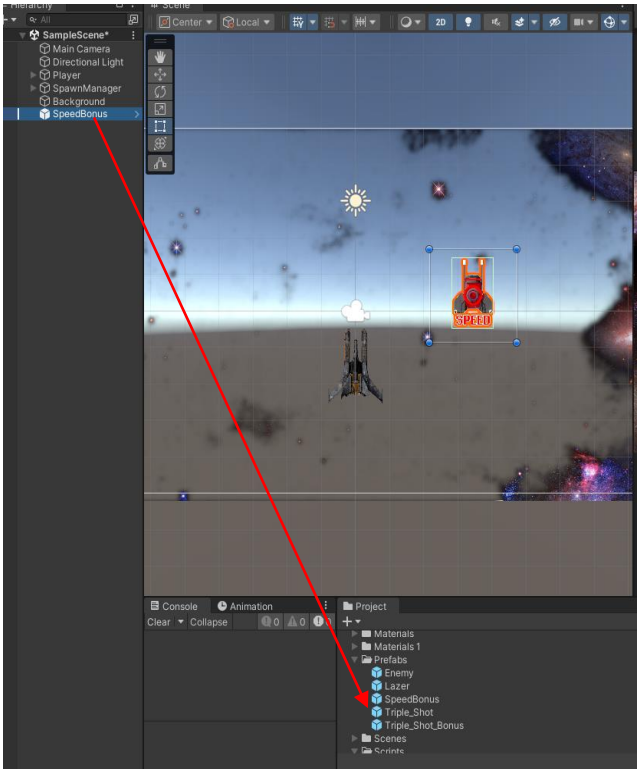
7. Hafta Raporu – Aleks Dulda 21360859025

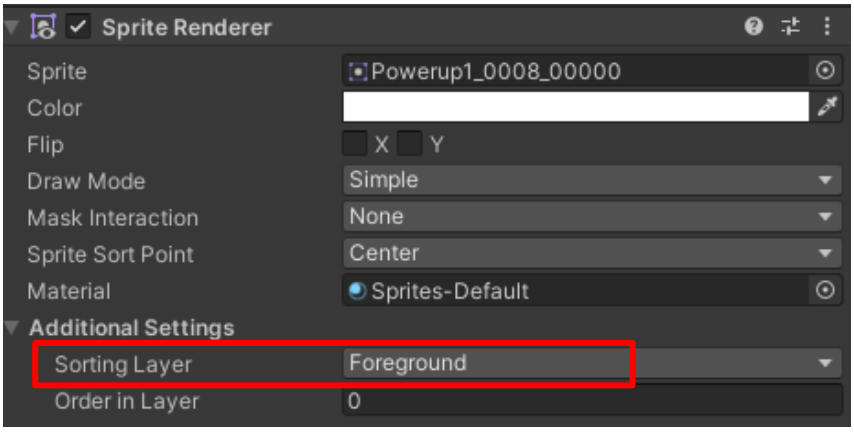
Hız bonusu: Hız sprite'ını kullanarak bir prefabrik oluşturun (sorting layer, isim, ölçek, collider, rigid body, script ekle/düzenle)

Öncelikle “Power_ups” klasöründen Hız bonusu spritenı sürükleyip sahnemize bırakıyoruz. Ardından bu nesnemizin ismini SpeedBonus olarak değiştirip “Prefabs” klasörüne bırakıp prefabımızı oluşturmuş oluyoruz.

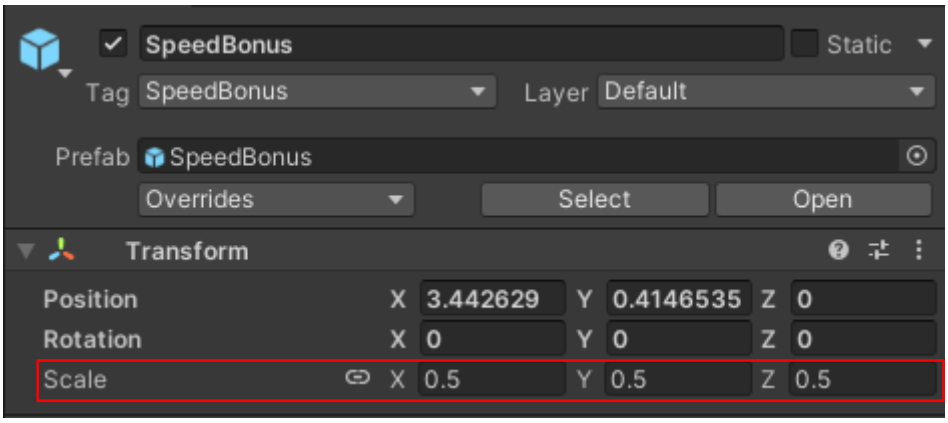




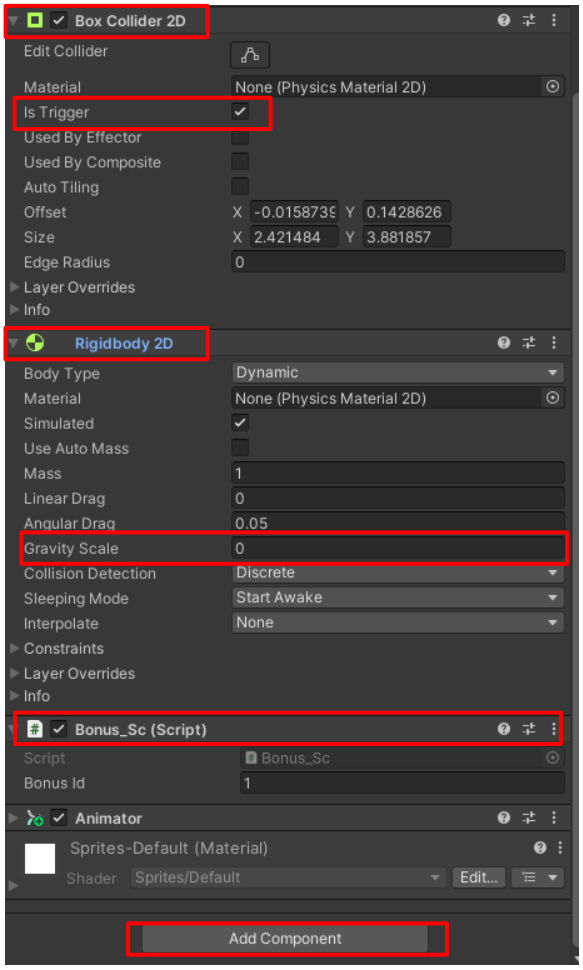
“Sprite Renderer” kısmından sorting layerimizi ayarlıyoruz. Sahnemizin ön planında bulunması için daha önceki oyun nesneleri için oluşturduğumuz layerlardan “Foregraound” seçiyoruz.



Eğer oluşturduğumuz bu bonusa ölçeklendirme eklemek istiyorsak “Transform” kısmından ölçeklendirmemizi göz zevkimize göre ayarlayabiliriz. Bu örnek için her düzlem için 0,5 ayarlanmıştır.



“Add component” kısmından collider, rigid body özelliklerini bu oyun nesnemize ekliyorz. Bu sayede çarpışma kontrollerini sağlayabileceğiz. Not: Rigidbody kısmında gravity özelliğini 0 yapmayı unutmayın. Aynı zamanda collider kısmında is Trigger özelliğini açınız.



Bu oluşturduğumuz nesneye , bir önceki oluşturduğumuz tripleshotbonus nesnesine yazdığımız aynı Bonus_Sc dosyasını ekliyoruz.

Bonus script'ini modüler hale getirin (bonus'ları tekil olarak numaralandırın)

Bu scriptimze bonusları birbirinden ayırabilmek için her bir bonusa bir id veriyoruz. Bunu “int BonusId” olarak adlandırdığımız int değişkende tutacağız.

Bu örnekte Tripleshot için 0, SpeedBonus için 1, KorumaBonus için 2 kullanılacaktır.

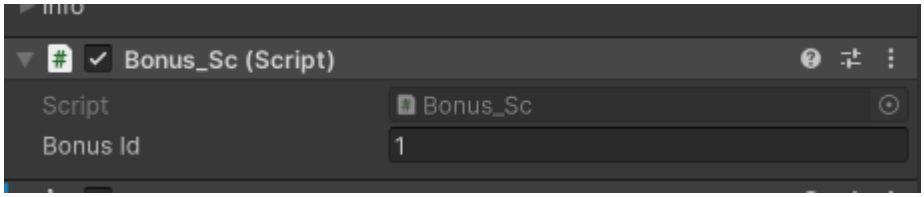
```
[SerializeField]
int BonusId;
```

Yapacağımız bir başka değişiklik ise “void OnTriggerEnter2D(Collider2D other)” fonksiyonunda olacaktır. Bu fonksiyonda switch case yapısını kullanarak aktive edeceğimiz bonusun fonksiyonlarını çağıracağız.

```
void OnTriggerEnter2D(Collider2D other)
{
    if (other.tag == "Player")
    {

        switch (BonusId)
        {
            case 0:
                player.ActivateTripleShot();
                Destroy(this.gameObject);
                Debug.Log("Bonus Üçlü atış");
                break;

            case 1:
                player.ActivateSpeedBonus();
                Debug.Log("Bonus hız");
                Destroy(this.gameObject);
                break;

            case 2:
                Debug.Log("Bonus koruma");
                Destroy(this.gameObject);
                break;

            default:
                Debug.Log("Tanımlanmamış Bonus");
                break;
        }

    }
}
```

Bu şekilde her bir özel güç için ayrı fonksiyonlar çağırabileceğiz.

Not: bu id işleminden sonra “Inspector” kısmından scripte bulunan “BonusId” kısmına her bir özel güç için belirlediğimiz sayıları girmeyi unutmayın. (Aşağıdaki resimde SpeedBonus için yapılan değişiklik gösterilmiştir.)



Player script'i içinde hız çarpanı değişkeni ve hız bonus'unun aktif olup olmadığı ile ilgili değişkeni tanımlayın. Aktifleşme ve pasifleştirme fonksiyonu

Hız bonusumuzu aldığımızda bir çarpan etkisi yaratması için “float speedMultiplier” değişkenimizi ekliyoruz. Bu değişken hız bonusunu aldığımız zaman var olan hızımızı ikiye katlamaya yarayacak.

SpeedBonusu aktif edebilmek için tripleshot bonusunu aktif etmek için uyguladığımız metodları yine uygulayacağız.

Önce bir bool değer ekleyeceğiz bu aktif olup olmadığını gösterecek bize.

Ardından speedBonus özelliği alındığı zaman gerçekleşecek olayları bir fonksiyonun içini tanımlıyoruz. Eğer daha önce bir speed bonusu alınmadıysa 5snlik bir coorutine oluşturup hızımızı oluşturduğumuz çarpan ile çarpıyoruz. Daha önce bir bonus alındıysa ve bu bonus devam ediyorsa ise var olan bonusu silip yeni gelen bonusu yeni bir corroutine oluşturuyoruz bu sayede aldığı bonus boşa gitmemiş oluyor ve yenilenmiş oluyor. Bu şekilde activatespeedbonus fonksiyonunu oluşturmuş oluyoruz.

Alınan bu özel gücü deavtive etmek için ise 5sn sonra verdiğimiz çarğanı bölme olarak veriyoruz ve eski hızına geri dönüyor. Gerekli bool değerleri ise değiştirmiş oluyoruz.

```
public void ActivateSpeedBonus()
{
    if(SpeedBonusActive == false)
    {
        SpeedBonusActive = true;  // Activate Triple Shot

        speed = speed * speedMultiplier;

        //  5sn beklemesi için coroutine başlat
        speedCoroutine = StartCoroutine(DeactivateSpeedBonusAfterDelay(5f));
    }
    else if(speedCoroutine!=null)
    {
        StopCoroutine(speedCoroutine);
        SpeedBonusActive=false;
        speedCoroutine = StartCoroutine(DeactivateSpeedBonusAfterDelay(5f)); ;
    }
```

```
private IEnumerator DeactivateSpeedBonusAfterDelay(float delay)
{
    // belirtilen süre kadar bekle
    yield return new WaitForSeconds(delay);

    // Deactivate Triple Shot
    SpeedBonusActive = false;
    speed = speed / speedMultiplier;
    speedCoroutine=null;
}
```

Player kodu içinden hız bonusu koduna erişin ve hız bonus'unu (gerektiğinde) etkinleştirin

Aldığımız hoz bonusunu çarpışma kontrollerini sağladığımız OnTriggerEnterID fonksiyonumuzda TripleShot için uyguladığımız işlemi bir de bu bonus için ugulayacağız. Burada zaten Bonus_Sc scriptinde miras olarak aldığımız Player_Sc dosyasından daha demin yazdığımız fonksiyonlara erişerek çarpışma saplandığı esnada bu fonksiyonları çağıracak kod yazmış bulunmaktayız.

```
void OnTriggerEnter2D(Collider2D other)
{
    if (other.tag == "Player")
    {

        switch (BonusId)
        {
            case 0:
                player.ActivateTripleShot();
                Destroy(this.gameObject);
                Debug.Log("Bonus Üçlü atış");
                break;

            case 1:
                player.ActivateSpeedBonus();
                Debug.Log("Bonus hız");
                Destroy(this.gameObject);
                break;

            case 2:
                Debug.Log("Bonus koruma");
                Destroy(this.gameObject);
                break;

            default:
                Debug.Log("Tanımlanmamış Bonus");
                break;
        }
```

Bu şekilde gerekli değişmler sağlandığında hoz bonusumuz gerektiğinde etkinleştirilecektir.

SpawnManager'ı düzenleyerek belirli aralıklarla hız bonusu üretin

Bu aşamada spawnmanagerde random bir sayı üretilerekten (var olan bonus sayısı aralığında üretilmektedir.) random bir bonus spawn edilmesi sağlanır. Her bir bonusun kendine ait idsi bulunduğundan ötürü .ıkan sayıya eş değer olan bonus spawn olacak şekilde düzenlenmiştir.

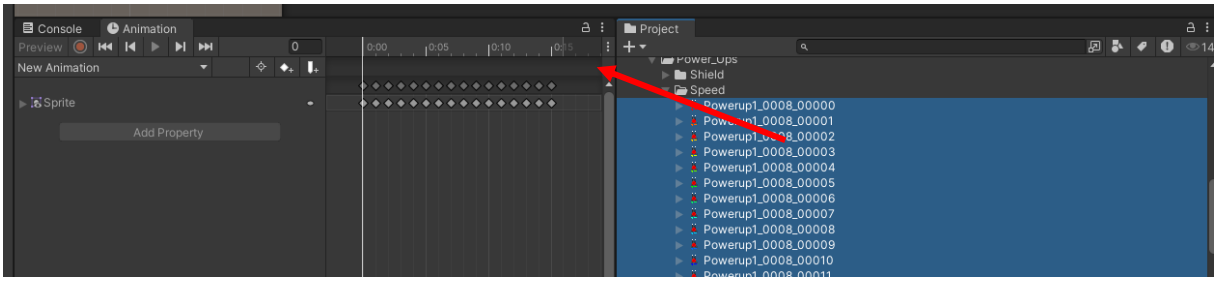
```
1 başvuru
IEnumerator SpawnBonusRoutine()
{

    while (player.Health > 0)
    {
        int k = Random.Range(0, Bonusprefabs.Length );
        Debug.Log("random sayı: " + k);
        Vector3 position = new Vector3(Random.Range(-player.xVal, player.xVal), player.yVal + 2, 0);
        Instantiate(Bonusprefabs[k], position, Quaternion.identity);
        yield return new WaitForSeconds(5.0f);
    }
}
```

Hız bonusu için animasyon ekleyin

Animation sekmesinden “Create” tuşuna basarak daha önce triplebonus için oluşturduğumuz Animations klasörünün içine SpeedBonus için bir animasyon ekliyoruz. Bunu bir önceki triplebonusa yaptığımız birebir işlemleri uygulayacağız.

1. Create diyip dosya konumuna gitmek
2. Klasörü seçtikten sonra var alan spriteları sürükleyip animation sekmesine bırakmak

Aslında hepsi bu kadar kolay



Kaynakça

https://en.wikipedia.org/wiki/Unity_(game_engine)

https://chatgpt.com/

https://docs.unity3d.com/Manual/index.html

Github link

https://github.com/AleksDulda/GamePrograming

https://github.com/AleksDulda/GamePrograming/tree/main/Rapor/Week_7