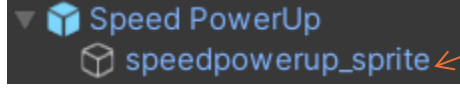


7.Hafta Ders Raporu – Batuhan Şengül

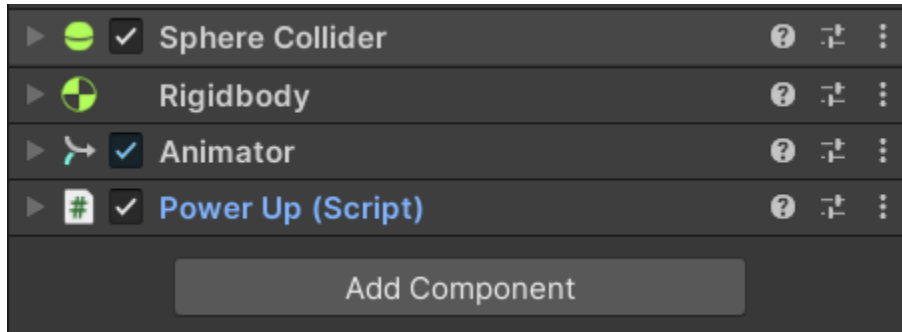
Hız Bonusu

Hız Güçlendirme Prefabi

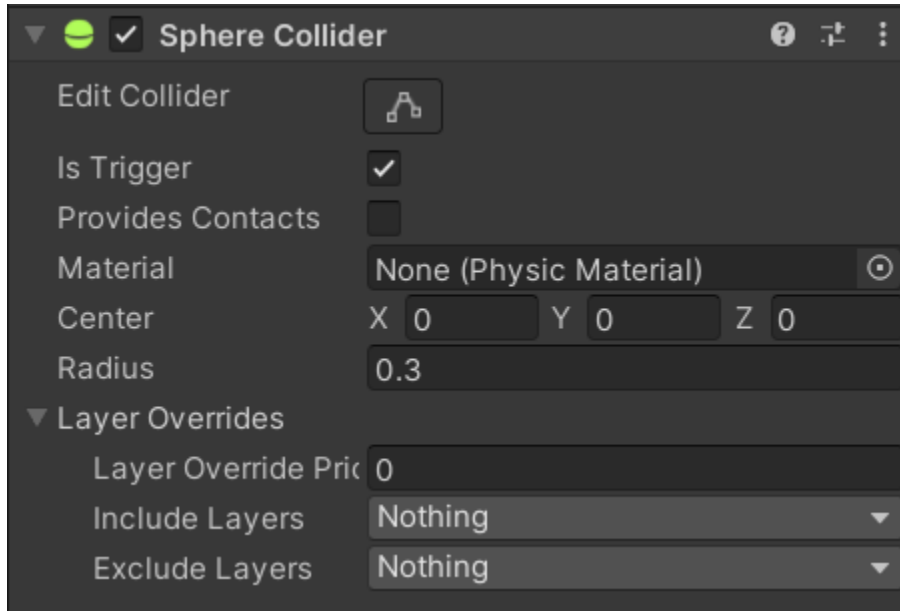


“Speed PowerUp” adlı boş bir nesne oluşturulup içine kullanacağımız sprite eklenir.

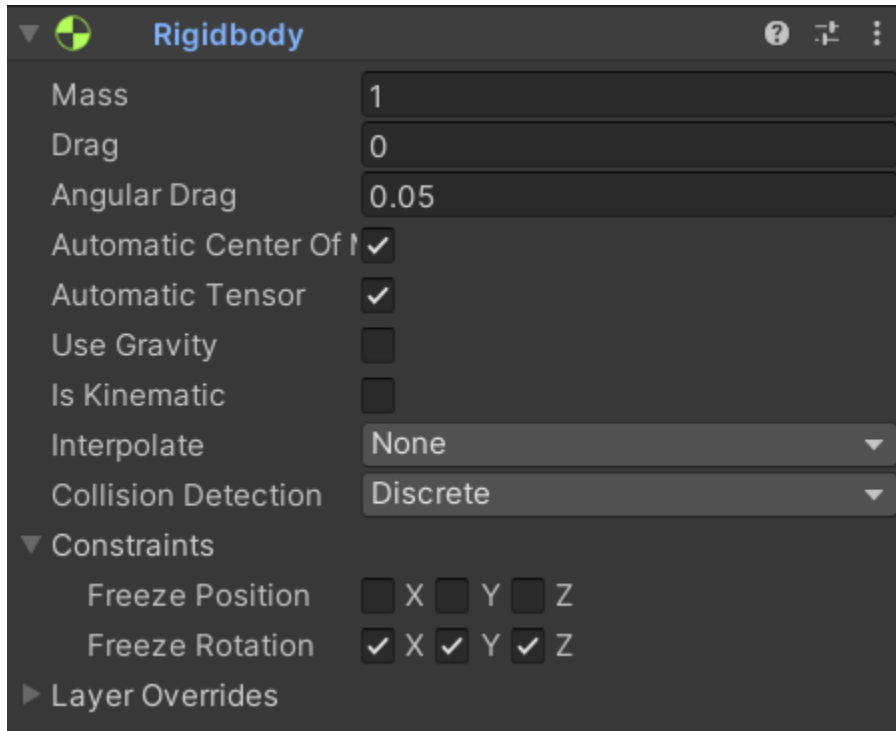
Nesneye collider, rigidbody, animator ve Power Up scripti bileşen olarak eklenir. Gerekli ayarlar yapılır:



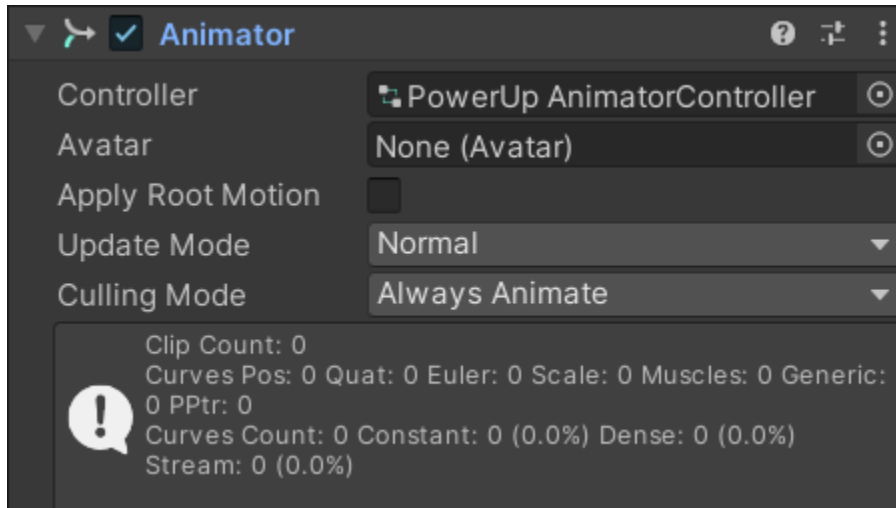
- **Sphere Collider**



- **Rigidbody**

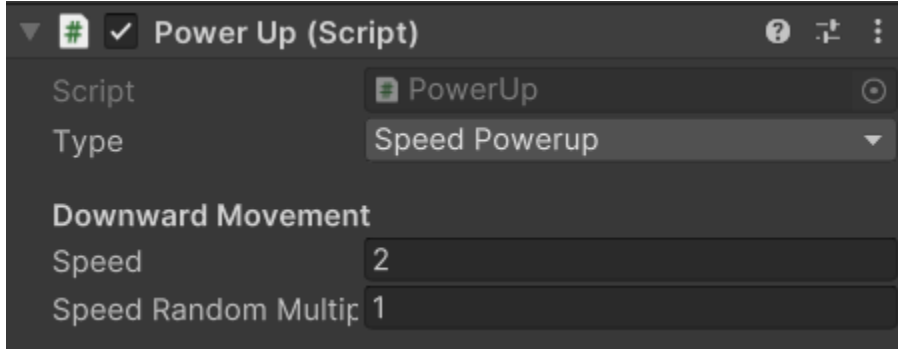


- **Animator**

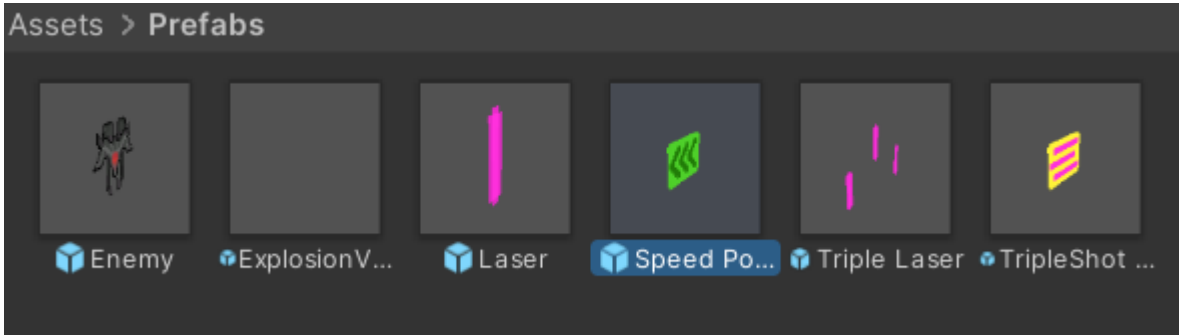


Animator her powerupta aynıdır çünkü her güçlendirmenin boyutları aynıdır, bu şekilde aynı kontrolcü kullanılmıştır.

- **PowerUp Script**



Script içerisinde yeni bir tip(type) oluşturulup, yeni oluşturulan tip(SpeedPowerUp) seçilmiştir.



Oluşturulan bu oyun nesnesi Assets -> Prefabs klasörüne taşınır ve hız güçlendirme prefabi oluşturulmuş olunur.

PowerUp Script Güncellemeleri

- **Eklenen/Güncellenen Değişkenler**

```
public enum PowerUpType
{
    TripleShotPowerup,
    SpeedPowerup, //diğer powerupları bu şekilde eklemek daha kolay olacak
    Option3,
}
```

Yeni bir enum tipi oluşturulmuştur. Bu şekilde tasarım yapıldığı için güçlendirmelere zaten modüler haldedir.

- Eklenen/Güncellenen Fonksiyonlar

```
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player"))
    {
        switch (type)
        {
            case PowerUpType.TripleShotPowerup:
                pm.StartTriplePowerup();
                break;

            case PowerUpType.SpeedPowerup:
                pm.StartSpeedPowerUp();
                break;

            default:
                Debug.LogWarning("PowerUp type doesnt declared!");
                break;
        }
        Destroy(gameObject);
    }
}
```

OnTriggerEnter fonksiyonun içerisine güçlendirmenin tipinin hız güçlendirmesi olması durumunda gerçekleşecek switch bloğu eklenmiştir.

Oyuncu hız güçlendirmesiyle temasa geçtiğinde: pm(PlayerMovement) bileşeninden Eklenen Fonksiyonlar

StartSpeedPowerUp() Fonksiyonu çalıştırılır.

PlayerMovement Güncellemeleri

- Eklenen Değişkenler

```
[SerializeField]private float regularSpeed = 3f;  
private float currentSpeed;
```

```
[Header("Speed Power-Up Settings")]  
[SerializeField] private bool speedIsActive = false;  
[SerializeField] private float speedDuration = 5f;  
[SerializeField] private float speedMultiplier = 2f;  
Coroutine speedCoroutine;
```

Önceden speed olarak adlandırılan -önceden oyuncunun varsayılan hızının değerini tuttuğumuz- değişken regularSpeed olarak değiştirip, kod içerisinde hata alınan yerlerde yenisiyle güncellenmiştir.

Hız Güçlendirmesinin etkilerini işleyecek değişkenler eklenmiştir. (bool speedIsActive, float speedDuration, float speedMultiplier)

- Eklenen Fonksiyonlar

1. StartSpeedPowerUp() Fonksiyonu

```
1 reference
public void StartSpeedPowerUp()
{
    if (speedCoroutine == null)
    {
        speedIsActive = true;
        speedCoroutine = StartCoroutine(SpeedEffect());
    }
    else
    {
        StopCoroutine(speedCoroutine);
        animator.SetBool("SpeedIsOn", false);
        currentSpeed = regularSpeed;
        speedIsActive = true;

        speedCoroutine = StartCoroutine(SpeedEffect());
    }
}
```

Bu fonksiyon “PowerUp” scriptinden çağırılmaktadır, sayaca bağlı olarak hızlanma özelliğini aktif kılar.

2. IEnumerator SpeedEffect() Fonksiyonu

```
2 references
public IEnumerator SpeedEffect()
{
    currentSpeed=regularSpeed*speedMultiplier;
    animator.SetBool("SpeedIsOn", true);

    yield return new WaitForSeconds(speedDuration);
    currentSpeed = regularSpeed;
    speedIsActive = false;
    animator.SetBool("SpeedIsOn", false);
}
```

Bu hızlanma özelliğini belirlenen süre kadar aktif tutar. Hız animasyonunu başlatır ve süre bitince durdurur.

SpawnManager Güncellemeleri

- **Eklenen Değişkenler**

İhtiyacımız olacak değişkenler oluşturulmuştur.

```
[Header("SpeedPowerup Spawn Settings")]
[SerializeField]private GameObject speedPowerUpPrefab;
[Range(0f, 1f)][SerializeField] private float speedSpawnChance = .5f;
GameObject lastSpawnedSpeedPowerUp;
```

- **Start() Fonksiyonu Güncellemesi**

```
void Start()
{
    pm = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerMovement>();
    StartCoroutine(EnemySpawner());
    StartCoroutine(PowerUpSpawner(triplePowerUpPrefab,tripSpawnChance));
    StartCoroutine (PowerUpSpawner(speedPowerUpPrefab,speedSpawnChance));
}
```

Güçlendirmelerini sürekli şansa bağlı olarak canlandıracak coroutine'i başlatacak düzenlenerek eklenmiştir. (IEnumerator PowerUpSpawner() Fonksiyonu)

- Eklenen Fonksiyonlar

1. IEnumerator PowerUpSpawner() Fonksiyonu

```
IEnumerator PowerUpSpawner(GameObject prefab,float spawnProb)
{
    while (pm.lives > 0)
    {
        float spawnChanceLocal = Random.Range(0f, 1f);
        if (spawnChanceLocal < spawnProb)
        {
            lastSpawnedTriplePowerUp = Instantiate(prefab,
            (new Vector3(Random.Range(-pm.xBorderValue, pm.xBorderValue), pm.yBorderValue + 1.2f, 0)),
            Quaternion.identity);
            lastSpawnedTriplePowerUp.transform.parent = powerUpContainer.transform;
            yield return new WaitForSeconds(5f);
        }

        yield return new WaitForSeconds(1f);
    }
}
```

1. Her döngüde oyuncunun canı 0'dan büyük mü diye kontrol edilir.
2. Eğer canı 0'dan büyükse, rastgele bir sayı üretilir(spawnChanceLocal).
3. Üretilen sayı belirlenen oluşma şansı(spawnProb)'ndan küçük mü diye kontrol edilir.
4. Eğer küçükse, pm(PlayerMovement) dan belirli olan ekran sınırlarını değerlerini aralık olarak alan bir rastgele bir X konumu, sabit Y ve Z konumlarında "prefab" oluşturulur. Oluşturulan nesne lastSpawnedPowerUp oyun nesnesine atanır. Bu kodda oluşturulan prefabler 3'lü atış ve hız güçlendirme prefableridir.(bkz. Start() Fonksiyonu Güncellemesi)
5. Oluşturulan güçlendirmenin hierarchyde parent nesnesi belirlenir.
6. Belirli bir süre beklenir.
7. Oyuncunun canı olana kadar, bu döngü her saniye tekrardan kontrol edilir.

Kaynakça

<https://docs.unity3d.com/Manual/>

<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UIBasicLayout.html>

<https://chatgpt.com/>

Proje Kodu ve Github Repo

Kod: <https://github.com/bathuchan/btu-gameprogramming-BatuhanSengul/tree/main/Reports/7.Hafta>

Proje Repo: <https://github.com/bathuchan/btu-gameprogramming-BatuhanSengul>

Hazırlayan

Batuhan Şengül – 20360859008- bathu.sengul@gmail.com