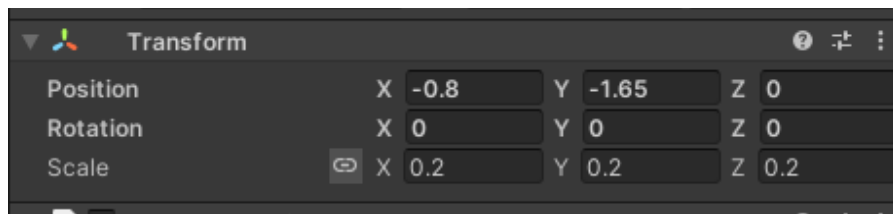
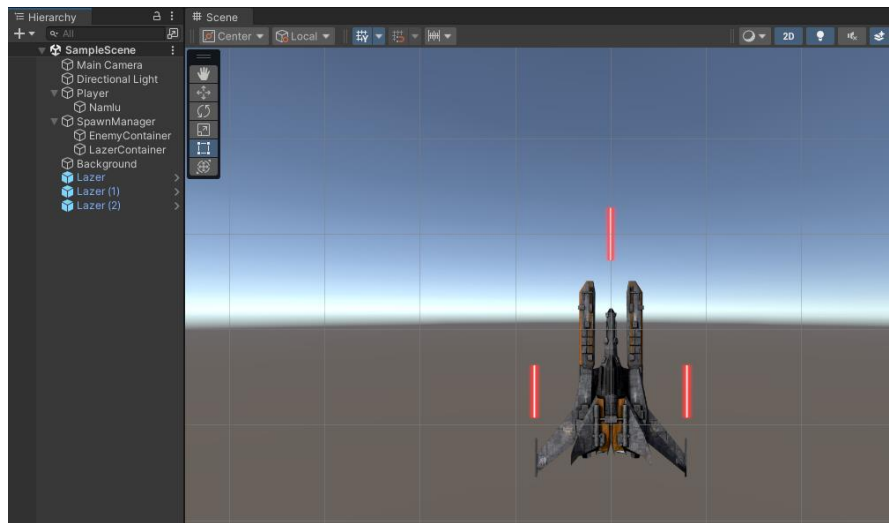
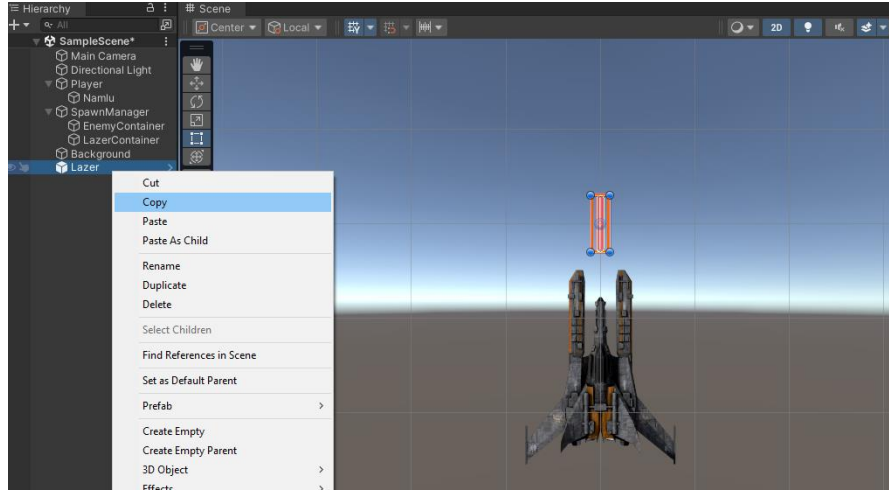


6. Hafta Raporu – Aleks Dulda 21360859025

Üçlü atış bonusu: Laser'den iki kopya daha oluşturun ve bu üç laser'in konumlarını ayarlayın:

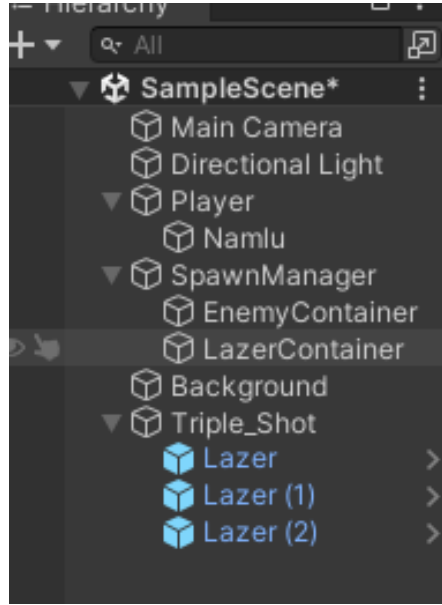
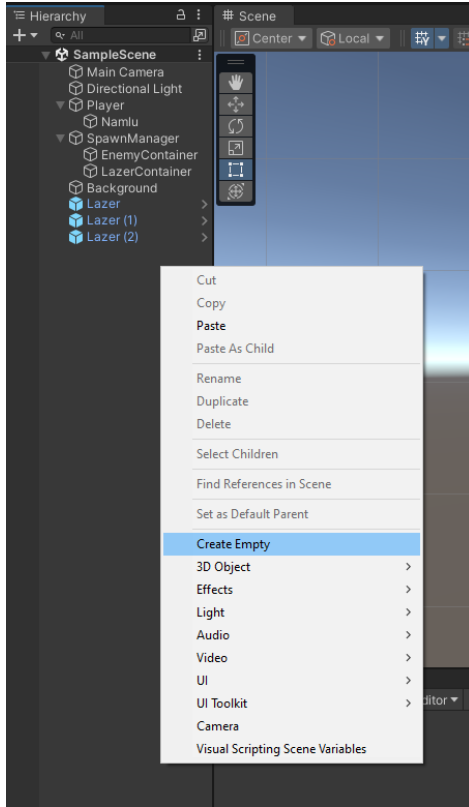
Öncelikle var olan laser prefabımızdan 2 adet kopya oluşturun ve bu kopyaları laser'in nasıl ateşlenmesini istediğimize göre konumlandırıyoruz.



Konumların kordinatları sağ lazer için resimde görülen sayılar kullanılmıştır. Sol lazer için sadece “Position” X değerini +0,8 yazmanız yeterlidir. Ortadaki lazer ise 0,0,0 konumundadır.

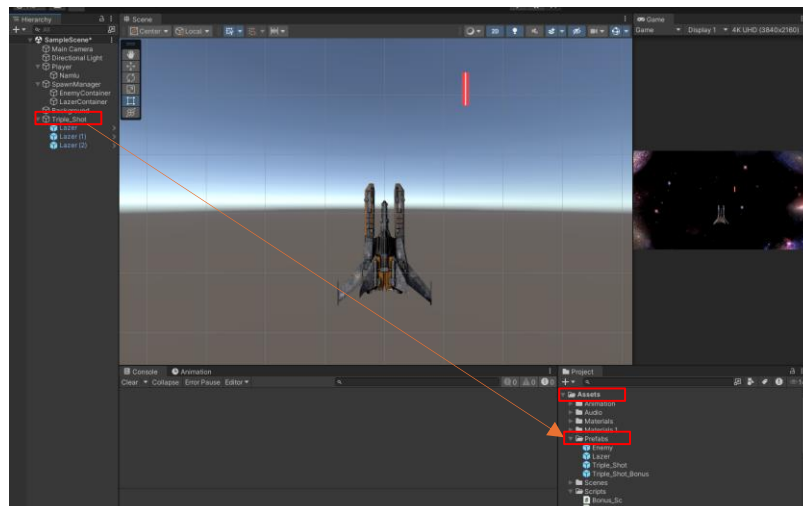
Boş bir oyun nesnesi oluşturun ve üç laser'i bunun child'ı olarak atayın:

Hierarch kısmında sağ tık yaparak açılan pencerede “Create Empty” tuşuna basarak boş bir oyun nesnesi oluşturuyoruz ve bu kopyalayıp hizaladığımız laserleri de bu boş oyun nesnesinin içine koyuyoruz yani child olarak atamış oluyoruz. (Bu örnekte boş oyun nesnesi “Triple_Shot” olarak adlandırılmıştır.)



Bu oyun nesnesinden bir prefab oluşturun:

Bu aşamada ise oluşturduğumuz bu oyun nesnesini sürükleyip bırak tekniğiyle “Assets” kısmında bulunan prefabs klasörüne bırakarak oluşturduğumuz bu nesneyi prefab yapmış oluyoruz



Üçlü bonusun hangi durumlarda aktive edileceğinin mantığını geliştirin ve kodlayın

Player_Sc scriptimize “bool TripleShotActive = false;” değeri oluşturuyoruz bu değer bizim özel gücümüzün aktif olup olmadığını söyleyecek değerdir. “Default olarak false değeri atanmıştır”

Bu özel gücün aktifliğini ayarlayabilmek aşağıdaki resimde bulunan fonksiyonları yazıyoruz.

```
1 başvuru
public void ActivateTripleShot()
{
    TripleShotActive = true; // Activate Triple Shot

    // 5sn beklemesi için coroutine başlat
    StartCoroutine(DeactivateTripleShotAfterDelay(5f));
}

1 başvuru
private IEnumerator DeactivateTripleShotAfterDelay(float delay)
{
    // belirtilen süre kadar bekle
    yield return new WaitForSeconds(delay);

    // Deactivate Triple Shot
    TripleShotActive = false;
}
```

“public void ActivateTripleShot()” çağrıldığı zaman özel gücümüz aktive olur ve coroutine fonksiyonunu çağırır. Belirtilen süre sonunda resimde de göreceğiniz üzere 5sn özel gücümüz aktif halde kalır ve özel gücümüzü tekrardan deactive edecek kod satırı çalışır.

Bu sayede özel gücümüz aktive olduktan 5sn sonra deactive olacak şekilde kodlamış olduk.

Bilgi:

Coroutine

Bir Coroutine, kodunuzu belirli bir süre veya belirli koşullar sağlanana kadar bekleterek çalıştırmanıza olanak sağlar. Normal bir fonksiyon, çalıştırıldığında hemen tamamlanır; ancak bir coroutine belirli adımlarda durabilir, belirli süre bekleyebilir ve çalışmasına daha sonra devam edebilir.

IEnumerator

Bir IEnumerator, bir coroutine'in geri dönüş tipi olarak kullanılır. IEnumerator, yield return ifadeleriyle zaman kontrolü sağlayarak coroutine'in bekletilmesini mümkün kılar.

Üçlü bonus sprite'ı toplandığında belirli bir süre için üçlü bonus'u etkinleştirin:

Bu adımda Player nesnesi ile bonus sprite'ı etkileşime geçmiş mi? yani birbirine temas etmiş mi? Bir diğer deyişle acaba oyuncumuz bu bonusu topladı mı? Kontrolünü sağlamamız gerekir. Bunun için Bonus_Sc adında bir script oluşturuyoruz ve bu scriptte **“void OnTriggerEnter2D(Collider2D other)”** fonksiyonunu yazıyoruz. Bu fonksiyonda player ile bonus sprite'ı birbirine değip değmediği kontrol edilecek.

```
Unity İletisi | 0 başvuru
void OnTriggerEnter2D(Collider2D other)
{
    if (other.tag == "Player")
    {
        player.ActivateTripleShot();
        Destroy(this.gameObject );
    }
}
```

Görüldüğü üzere eğer bu iki nesne birbirini tetiklediye özel gücümüz aktive olacak ve ardından bu obje kendini yok edecek.

Bu şekilde özel gücümüz toplandığı zaman oyuncumuzuz özel gücü aktive olmuş olacak.

FireLaser fonksiyonunda gerekli güncellemeleri yapın:

Bu kısımda yaptığımız güncelleme şu şekildedir:

```
1 başvuru
private void Ateset()
{
    if (Input.GetKey(KeyCode.Space) && vur)
    {
        Instantiate(laser, firePoint.position, Quaternion.identity).transform.parent = LazerCon.transform;
        firecooldown = Normalcooldown;
    }
}
```

```
private void Ateset()
{
    if (Input.GetKey(KeyCode.Space) && vur)
    {
        if (TripleShotActive)
        {
            Instantiate(TripleLaser, firePoint.position, Quaternion.identity).transform.parent = LazerCon.transform;
        }
        else
        {
            Instantiate(laser, firePoint.position, Quaternion.identity).transform.parent = LazerCon.transform;
        }
    }

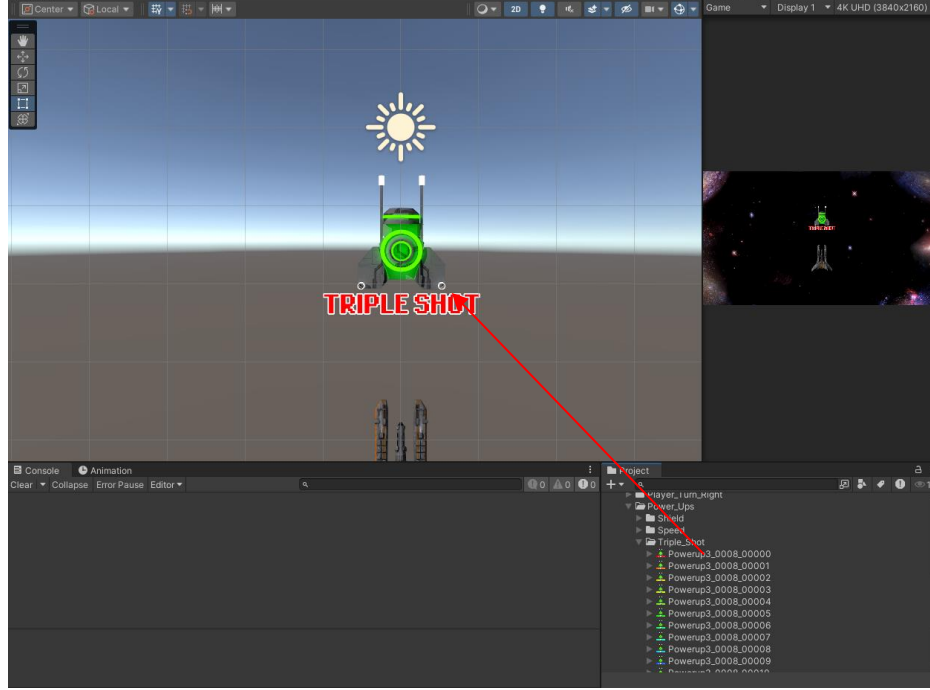
    firecooldown = Normalcooldown;
}
```

Kısaca açıklamak gerekirse kodumuza if koşulu ekleyerek özel gücümüzün aktif olup olmadığı kontrolü gerçekleştiriyoruz. Bu sayede eğer özel gücümüz aktif ise üç lazerin bulunduğu **“TripleLaser”** prefabını değil ise normal tek lazer atan **“laser”** prefabını oluşturmakta

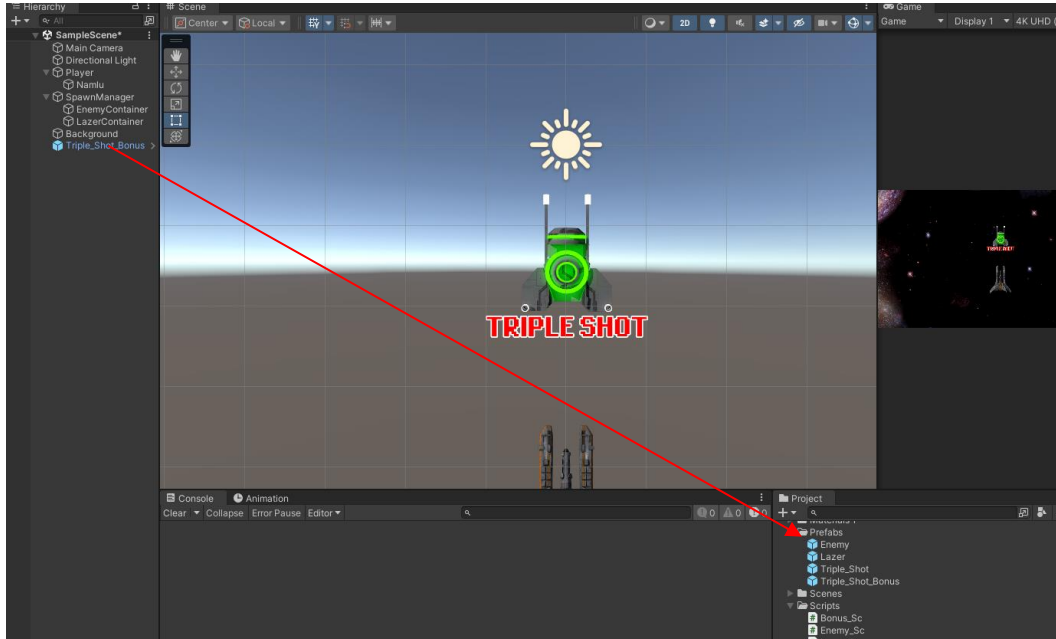
Bu değişiklikle beraber özel gücümüz aktif ise 3 lazer değil ise 1 lazer atacak şekilde ayarlanmış oldu.

Üçlü bonus sprite'ından bir nesne oluşturun, ölçeğini ayarlayın, circle collider ve rigidbody ekleyin, sorting layer'ı düzenleyin:

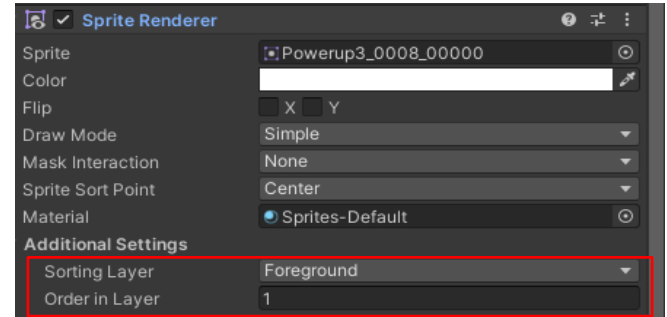
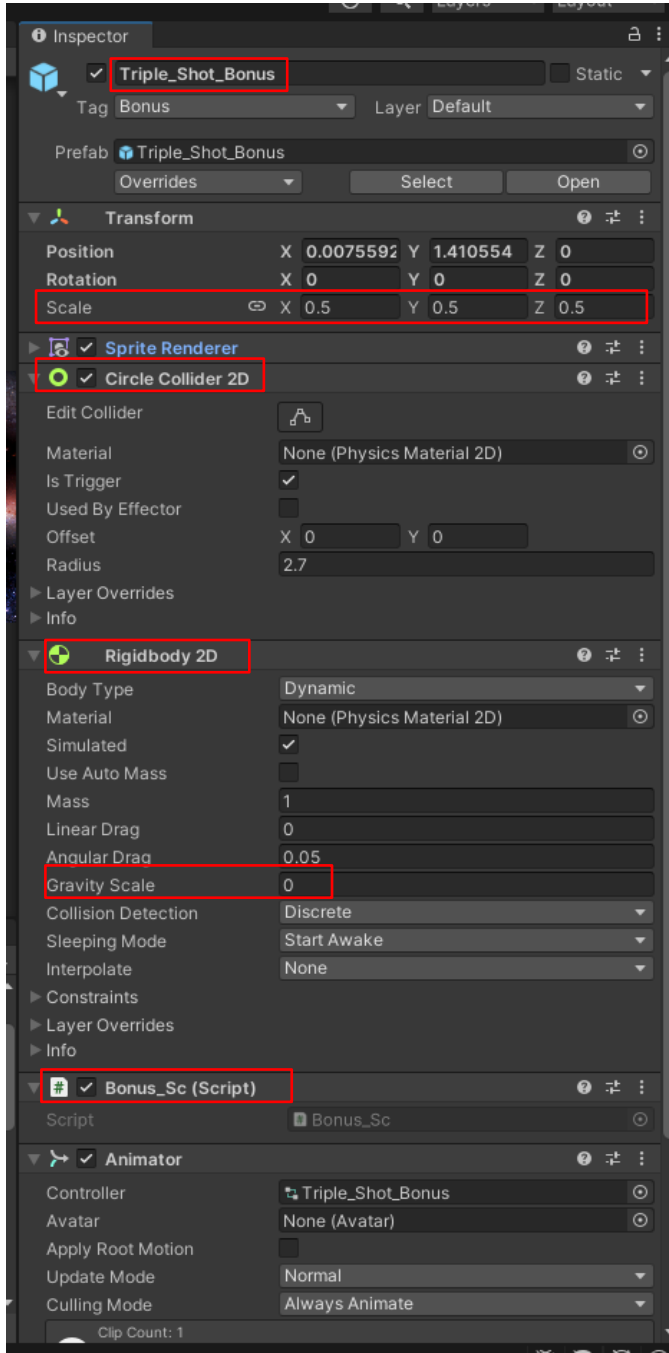
Bize verilen assetlerden üçlü lazer sprite'ını sahnemize sürükleyip bırakarak ekliyoruz.



Ardından prefab klasörüne sürükleyip bırakarak prefab haline çeviriyoruz.



Bu prefabımıza gerekli componentleri “**Inspector**” kısmından ekliyoruz (circle collider ve rigidbody (Gravity scale’i 0 yapmayı unutmayın)). Transform kısmından boyutunu ayarlıyoruz. Aynı kısımdan bu oluşturduğumuz özel güç prefabına layer ekliyoruz. Adını değiştirebiliriz (bu örnekte triple_shot_bonus olarak kullanılmıştır).



Bu kısımda layer ayarlarken “Foregraound” ayarlandıktan sonra sahnenin önüne gelecektir ama enemynin altında kalacaktır. Bu yüzden “Order in Layer” kısmından kendi içlerinde bir sıralama daha yaparak bu kısma 1 yazıyoruz. Bu sayede bu özel güç sprite’ı olduğu zaman bir başka nesnenin altına kalmayacak açıkça gözükecektir.

Bu aşamada edit collider yaparak colliderı ölçülendirebilirsiniz. İsterseniz “Tag” kısmından özel gücümüze bir etiket ekleyebilirsiniz.

Script oluşturarak üçlü bonus nesnesi ile ilişkilendirin, script ile nesnenin belirli bir hızla hareket etmesini sağlayın. Ekran dışına çıkıldığında nesneyi yok edin. Çarpışma kontrolü ekleyin:

Bonus_Sc adında bir script oluşturuyoruz ve bu scriptimize nasıl hareket edeceğini ve ne ile etkileşime geçince neler olacağını kodluyoruz.

```
public class Bonus_Sc : MonoBehaviour
{
    private Player_Sc player;
    float speed = 3;

    // Start is called before the first frame update
    void Start()
    {
        player = GameObject.FindObjectOfType<Player_Sc>();
    }

    // Update is called once per frame
    void Update()
    {
        transform.Translate(Vector3.down * speed * Time.deltaTime);
        if (player.Health <= 0 || this.transform.position.y < -6f)
        {
            Destroy(this.gameObject);
        }
    }
}
```

Player scriptimizdeki konumları kullanabilmek için player scriptini kalıtım olarak bu bonus scriptime resimde görüldüğü üzere import ediyorum.

Update kısmında ise transform.translate fonksiyonu ile nesnemizin hareketini düzenliyoruz.

Ardından if koşuluyla beraber oyuncumuzun canı 0'a eşit veya az olduğu durumlarda veya obje ekrandan çıktığı zamanlarda bu özel güç objesini yok edecek koşulumuzu ekliyoruz. Bu sayede oyun ekranından çıkınca veya oyuncunun canı bittiğinde özel güçler yok oluyorlar.

Özel güçlerin tek yok olduğu kısımlar bunlar değil tabiki. Özel güç nesnesi eğer bir player ile etkileşime geçerse yani bir oyuncu özel gücü toplar ise aşağıdaki fonksiyon çalışmakta.

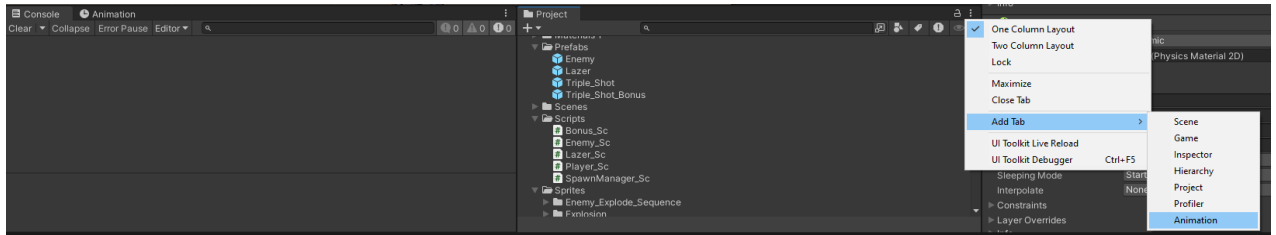
Player kodu içinden üçlü bonus koduna erişin ve üçlü bonus'u etkinleştirin:

Bu fonksiyon şu anlama gelir. Eğer oyuncu ile bu özel güç nesnesi birbirine temas ederse player scriptindeki özel gücü aktive den fonksiyonu çalıştır ve sonrasında bu özel güç nesnesini yok et anlamına gelir. Bu sayede özel gücümüz aktive olur ve nesnemiz ortadan kaybolur.

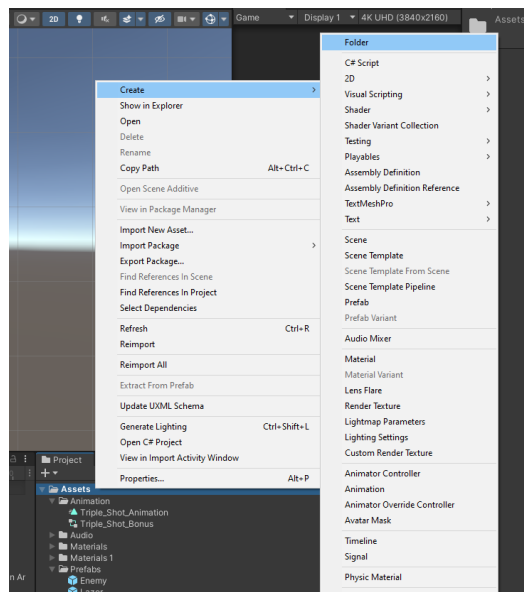
```
void OnTriggerEnter2D(Collider2D other)
{
    if (other.tag == "Player")
    {
        player.ActivateTripleShot();
        Destroy(this.gameObject);
    }
}
```

Üçlü atış bonusu nesnesi için animasyon ekleyin:

Ekranda görülen “...” tıklandığında “add tab” ve “Animation” kısmını ekleyerek animasyon sekmesi açmış bulunuyoruz. Bu işlemi yaparken animasyon ekleyeceğimiz nesne seçili olması gerekmektedir.

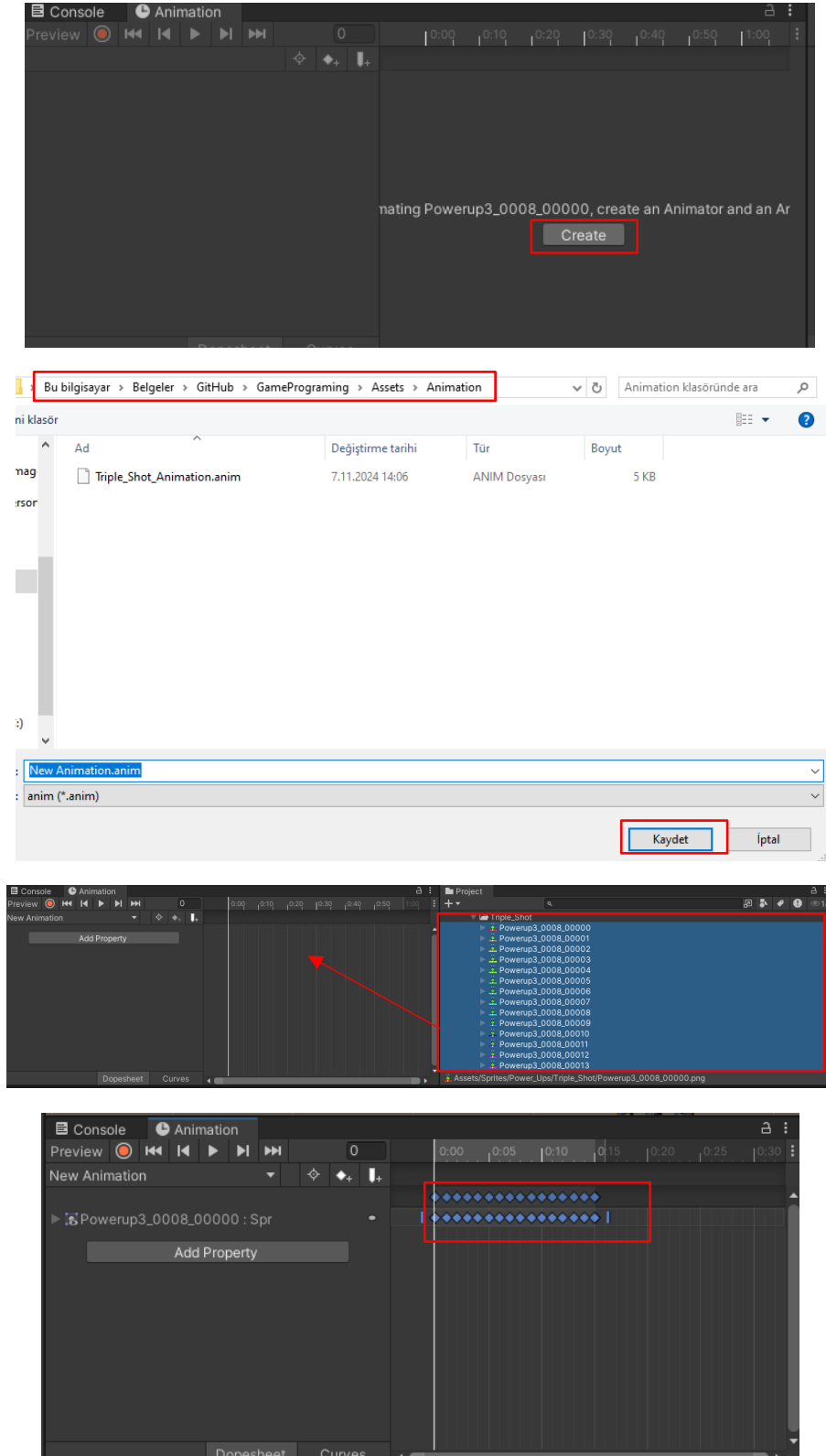


Ardından Assets klasörü altına Animatoın klasörü açıyoruz. Bu klasör oluşturacağımız animasyonların depolanacağı yerleri gösterir.



Ardından “Animation” sekmesindeki “Create” tuşuna basarak oluşturduğumuz “Animation” klasörünü seçiyoruz. Bu şekilde oluşturacağımız animasyonun nereye kaydedileceğini seçmiş bulunduk.

Ardından oluşturmak istediğimiz animasyon spritelerini sürükleyip bırak yöntemiyle açılan kısma tüm hepsini sürükleyip bırakıyoruz.



SpawnManager'ı üçlü atış bonusunu da dikkate alacak şekilde düzenleyin:

Bu özel gücümüzü spawn edebilmek için öncelikle bir Coroutine oluşturuyoruz. Bu coroutine oyun geliştirme süreçlerinde özellikle belirli bir süre beklemeyi veya zamana bağlı işlemleri kolayca yapmayı sağlar. Bu aşamada ise bizim oluşturacağımız özel gücün ne sıklıkla oluşacağını yani spawn olacağını ayarlamaya yarar.

```
Unity İletisi | 0 başvuru
void Start()
{
    player = GameObject.FindObjectOfType<Player_Sc>();
    StartCoroutine(SpawnEnemyRoutine());
    StartCoroutine(SpawnBonusRoutine());
}
```

Ardından bu coroutine'nin çalışma aşamalarını aşağıdaki resimde görüldüğü üzere ayarlıyoruz.

While koşulu oyuncumuzun hayatta olup olmadığını kontrol etmekte. Eğer oyuncumuz hayatta değilse yeni bir bonus üretmeyecektir.

Vector3 ile aşlayan komutumuz ise oluşacak özel gücümüzün nerelerde oluşacağını göstermekte. Bu kısımda player_Sc dosyasından oyuncumuzun hangi x ve y değerlerinde olduğunu çekerekten oluşacak bu özel gücümüzün de aynı konumlarda oluşmasını sağlıyoruz. Ve bunu rastgele olarak belirli değer aralarında sağlıyoruz.

Özel gücümüz oluştuktan sonra ne kadar süre sonra tekrar oluşacağını yani bekleme süresini "yield return new WaitForSeconds(.....)" komutuyla içine yazdığımız süre kadar beklemekte.

```
1 Başvuru
IEnumerator SpawnBonusRoutine()
{
    while (player.Health > 0)
    {
        Vector3 position = new Vector3(Random.Range(-player.xVal, player.xVal), player.yVal + 2, 0);
        Instantiate(TripleShotBonusPrefab, position, Quaternion.identity);
        yield return new WaitForSeconds(5.0f);
    }
}
```

Bu şekilde spawnmanager_Sc dosyamızı oluşacak özel gücümüz için güncellemiş oluyoruz.

Kaynakça

[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](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_6