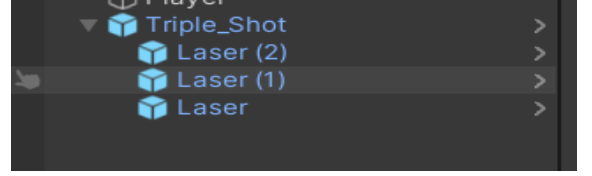


Öğrenci: Ahmet Yusuf Birdir

Numara: 21360859026

## Triple Shot Bonus Oluşturma

İlk olarak sahnenize bir adet boş triple shot isminde bir nesne oluşturun daha sonra bu nesnenin çocuğu(child) olacak şekilde nesnenizin üzerine 3 adet laser prefab nesnesi sürükleyin.



Daha sonra sahnenizde bu lazerlerin yerlerini şekildeki gibi ayarladıktan sonra prefab dosyasının içine sahnenizdeki triple shot nesnesini sürükleyin ve böylelikle triple shot prefab'iniz oluşmuş olacak.

## Triple Shot Aktive Etme ve Kod Düzenlemeleri

Öncelikle player\_sc dosyasının içine bir adet bool ve tripleShotPrefab değişkeni oluşturuyoruz, bonus aktif olduğunda true değerine sahip olacak şekilde.

```
[SerializeField]  
3 references  
bool isTripleShotActive = false;
```

Daha sonra player\_sc dosyasında update fonksiyonu içindeki laser oluşturma bloğunu alıp Fire isimli bir fonksiyonun içine yerleştiriyoruz. Aşağıdaki şekilde kodumuzu düzenledikten sonra bool değişkenimiz aktif olduğunda bonus özelliğimizi kullanabileceğiz.

```
1 reference  
public GameObject tripleShotPrefab;
```

```
1 reference  
void Fire(){  
    if(Input.GetKeyDown(KeyCode.Space) && Time.time > nextFire)  
    {  
        if(!isTripleShotActive){  
            Instantiate(laserPrefab, transform.position + new Vector3(0, 1.05f, 0), Quaternion.identity);  
        }else{  
            Instantiate(tripleShotPrefab, transform.position + new Vector3(-0.9536626f, 1.269109f, 0), Quaternion.identity);  
        }  
        nextFire = Time.time + fireRate;  
    }  
}
```

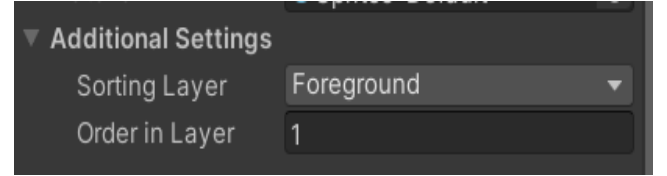
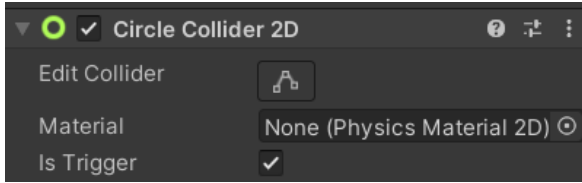
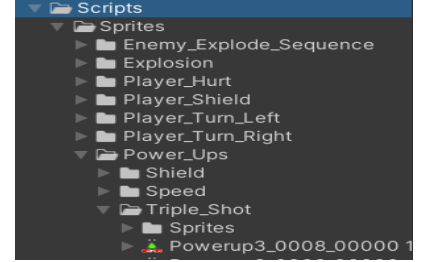
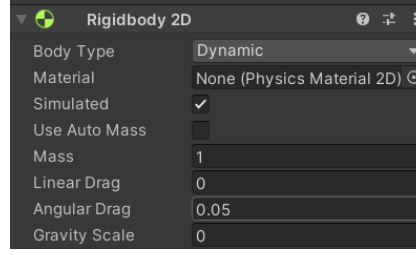
Değişiklikleri kaydedikten sonra unity uygulamasına gelip Player nesnesinin inspector panelindeki triple Shot Prefab değişkenine triple shot prefab dosyamızı bağlıyoruz.



## Bonus Sprite ile Aktive Olma

Sprite klasöründe bulunan Triple shot sprite'ını sahnenize sürükleyip ismini belirleyin. Daha sonra nesnenize Rigidbody 2D özelliğini ekleyin ve gravity scale özelliğini 0'a ayarlayın.

Sonra Circle Collider 2D özelliğini ekleyin ve isTrigger seçeneğini aktif hale getirin. Böylelikle nesnemiz fiziki yapısını bırakacak ve başka bir nesne ile temas halinde farklı bir yöne hareket etmeyecektir. Bundan sonra ise nesnenizin sortin layer kısmını foreground olarak değiştirin. Order in layer özelliğini ise 1 olarak değiştiriyoruz. Bu şekilde kendisi ile etkileşimi olmayacak örneğin enemy nesnesi ile etkileşime girdiğinde üst kısımda gözükecek. Son olarak nesnenizin collider ve boyutlarını ayarlayın.



## Bonus\_sc ve Bonus Özelliğinin Süresi

```
void Movement(){
    transform.Translate(Vector3.down * Time.deltaTime * speed);
    if(transform.position.y < -5.8f){
        Destroy(this.gameObject);
    }
}
```

```
void OnTriggerEnter2D(Collider2D other){
    Player_sc player = other.GetComponent<Player_sc>();
    if(player != null){
        if(other.tag == "Player"){
            player.ActivateTripleShot();
            Destroy(this.gameObject);
        }
    }
}
```

Bir adet bonus\_sc dosyası oluşturun. Bu dosyayı sahnedeki triple shot nesneniz ile bağlayın. Ardından dosyanızı açın ve bir adet speed nesnesi oluşturun ve istediğiniz hıza göre ayarlayın.

Yandaki şekillerde olduğu gibi bir adet hareket fonksiyonu ve OnTriggerEnter2D fonksiyonu oluşturun. Ardından nesnenizin ekrandan çıkınca yok olacak şekilde ayarlayın ve eğer player ile temas ederse bonusu aktifleştirip nesnenizin yok olmasını sağlayın.

Şimdi ise bonus özelliğinin süresini ayarlayacağız. Öncelikle player\_sc dosyasını açın. İçine yandaki gibi 2 adet fonksiyon oluşturun ve içlerini şekildeki gibi düzenleyin. ActivateTripleShot fonksiyonu Bonus\_sc içindeki OnTriggerEnter2D fonksiyonu içinde çağırıldığında bonusu aktifleştirecek ve bir süreliğine fonksiyonu bir başka işleme devredecek bir coroutine başlatacak.

Bu coroutine ise diğer oluşturduğumuz disableRoutine fonksiyonu olacak. Bu fonksiyonun içinde ise şekildeki gibi 5 saniye bekledikten sonra bonusu pasif hale getirecek ve işlem sona erecek. Geçecek olan süreyi kendi isteğiniz ile değiştirebilirsiniz.

```
1 reference
public void ActivateTripleShot(){
    isTripleShotActive = true;

    StartCoroutine(TripleShotBonusDisableRoutine());
}

1 reference
IEnumerator TripleShotBonusDisableRoutine(){

    yield return new WaitForSeconds(5.0f);
    isTripleShotActive = false;
}
```

## Belirli Düzende Bonus Üretme

İlk olarak prefab klasörünün içine bonus adında bir dosya oluşturun ve sahnenizdeki TripleShotBonus nesnesini bu dosyanın içine ekleyin. Böylelikle bir bonus prefab'iniz olacak.

Daha sonra spwanManager\_sc dosyanızı açın ve bir adet tripleShotBonusPrefab değişkeni oluşturun.

Ardından spawnBonusRoutine fonksiyonu oluşturun ve bu fonksiyonu rastgele x ekseninde bir noktadan bonus prefab nesnesi oluşturun ve belirli bir hızda aşağı inmesini sağlayın. Bu oluşumun ise belirli bir süre içerisinde tekrarlamasını sağlayın böylece oyun içerisinde belirli aralıklarla bonus özelliğinizi kazanma şansı olacak. Bu fonksiyonu start fonksiyonu içinde başlatın.

Son olarak unity üzerinden spawnmaneger nesnesinin inspector penceresinden

tripleShotBonus değişkeni ile prefab dosyasındaki bonus prefabinizi bağlayın.

```
[SerializeField]
1 reference
GameObject tripleShotBonusPrefab;
```

```
StartCoroutine(SpawnBonusRoutine());
```

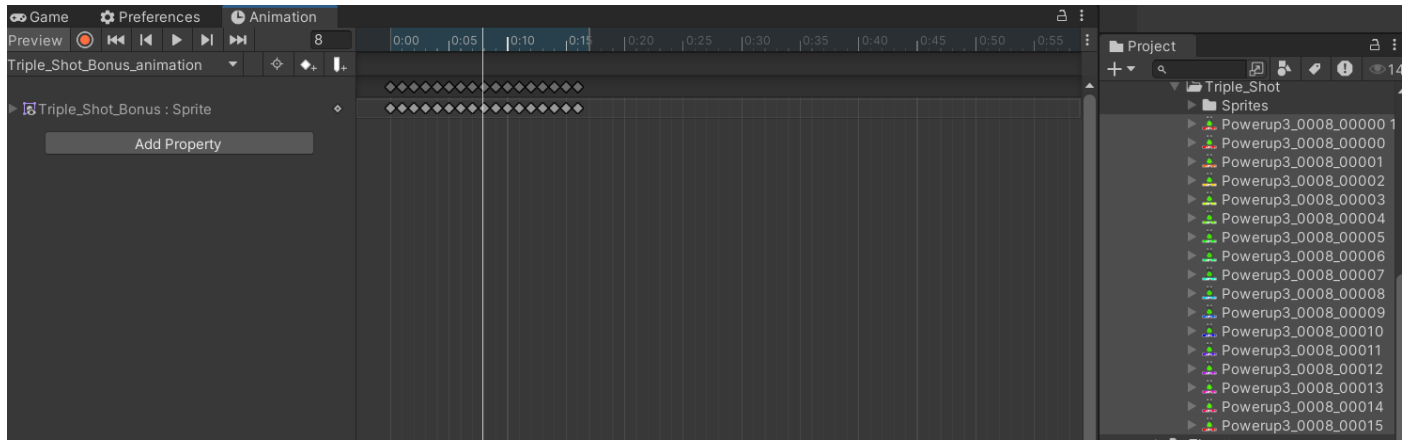
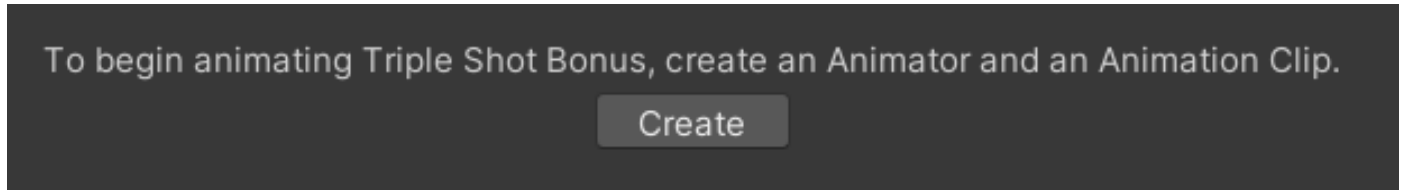
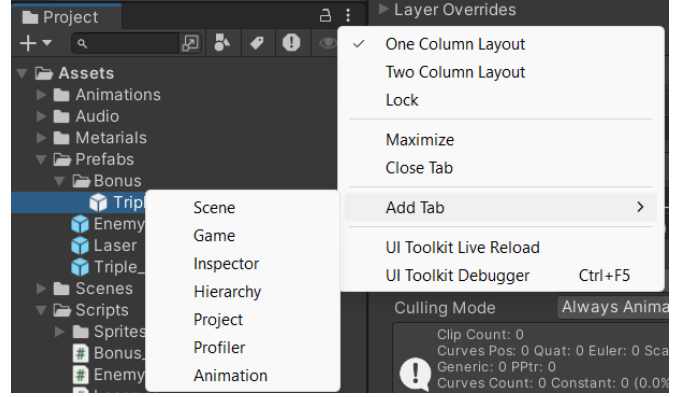
Triple Shot Bonus ... Triple\_Shot\_Bonus

```
IEnumerator SpawnBonusRoutine(){
    while(!stopSpawn){
        Vector3 position = new Vector3(Random.Range(-9.4f,9.4f),7.4f,0);
        Instantiate(tripleShotBonusPrefab, position, Quaternion.identity);
        yield return new WaitForSeconds(5.0f);
    }
}
```

# Animasyonlar

Öncelikle projects klasöründeki 3 nokta simgesi üzerinden Add Tab kısmındaki animation kısmını aktifleştirin.

Bundan sonra tripleShotBonusPrefab dosyanızı seçin ve animation panelindeki kısımdan create seçeneğini seçin. Ardından sprites klasöründeki powerUp dosyalarını animations kısmında açılan kısma yükleyin.



Bu kısımdan istersenin dosyalar arası geçiş süresini ayarlayarak animasyonun şeklini değiştirebilirsiniz.