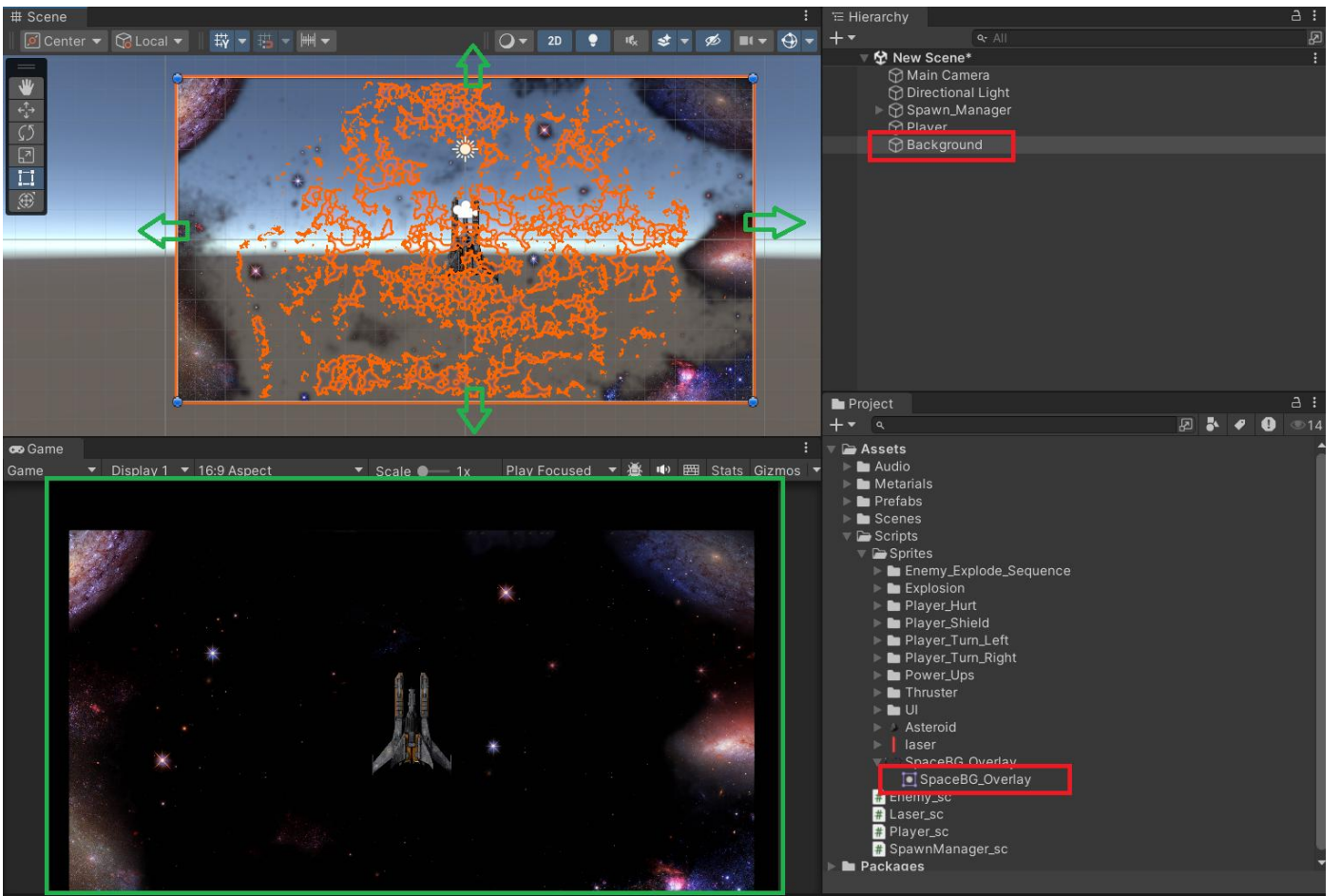


## Asset'lerin Projeye Eklenmesi

Gerekli asset dosyalarını indirdikten sonra Project paneline bir adet Audio ve Scripts klasörünün içerisinde de Sprites adına bir klasör açın. İndirdiğiniz asset dosyasının içindeki materials, audi ve sprites kalsörlerinin içindeki dosyaları Project panelinizde açtığınız klasörlerin içine kopyalayın.

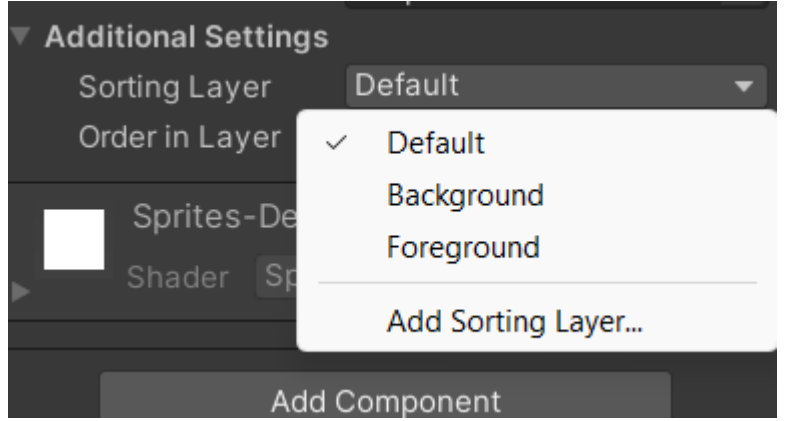
## Arkaplan Ekleme



İlk olarak Sprites klasöründeki arkaplan asset'ini bulun ve onu sürükleyerek Sahnenize Backgroun olarak ekleyin. Daha sonra Game panelinde gözüken siyah boşlukları kapatacak şekilde background nesnenizin noyutunu ayarlayın.

## Sortin Layer Kullanımı

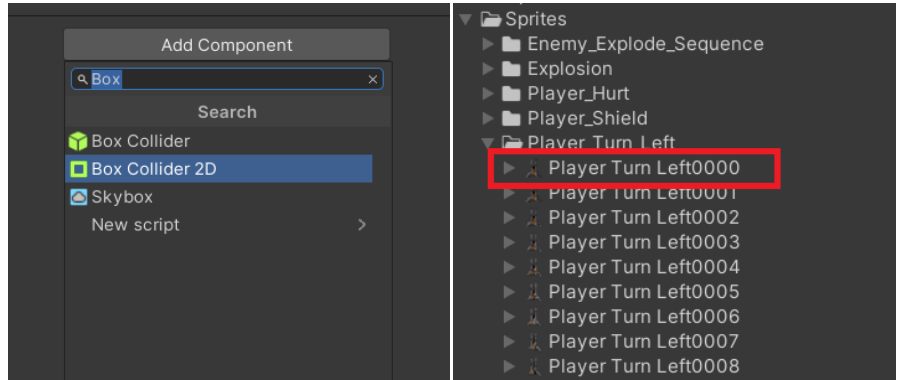
Sorting Layer (Sıralama Katmanı), 2D oyunlarda nesnelerin görüntü sıralarını belirlemek için kullanılır. Bu katmanlar, aynı Z ekseninde bulunan nesnelerin birbirleriyle olan görüntü önceliğini kontrol eder. Unity'de nesnenizi seçtikten sonra Inspector panelindeki additional settings kısmından bu özelliği görebilirsiniz.



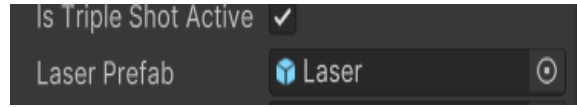
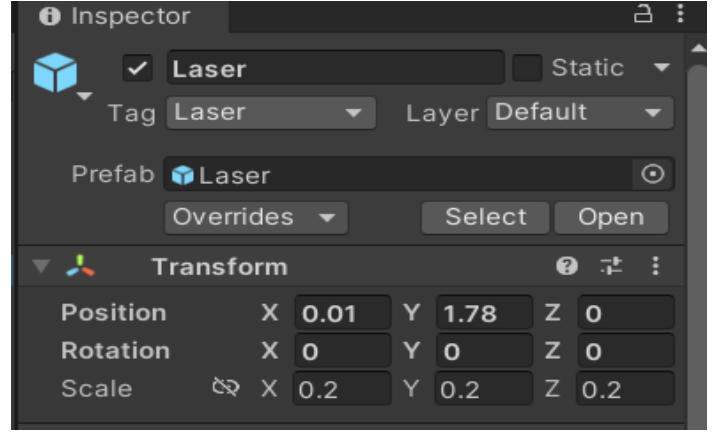
Başlangıçta default olarak gelen bu özelliği Add sorting layer diyerek Background ve Foreground olacak şekilde 2 adet layer ekleyin ve background nesneniz için background layerini seçin.

## Player, Enemy, Laser Nesnelerini 2D'a Geçirme

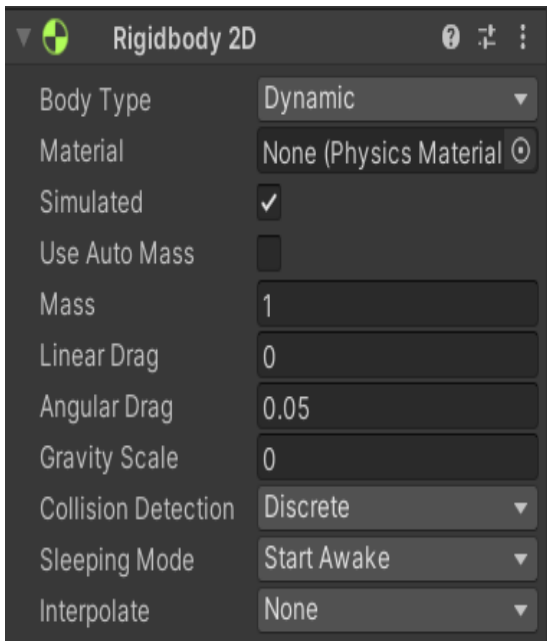
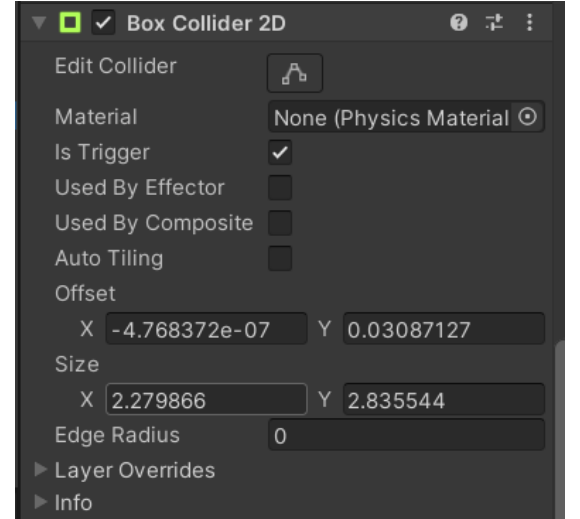
Öncelikle sahnedeki player nesnesini silin. Ardından sprites klasöründeki Player Turn Left0000 veya Turn Right0000 assetini sahnenize sürükleyerek ekleyin ve ismini Player yapın. Daha sonra yeni player nesnenize player\_sc dosyanızı bağlayın ve Inspector panelindeki add component kısmından Box Collider 2D eklentisini ekleyin.



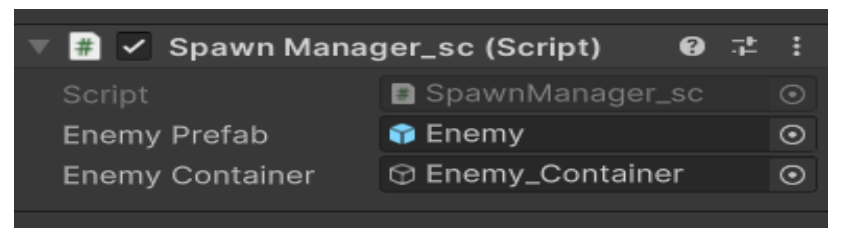
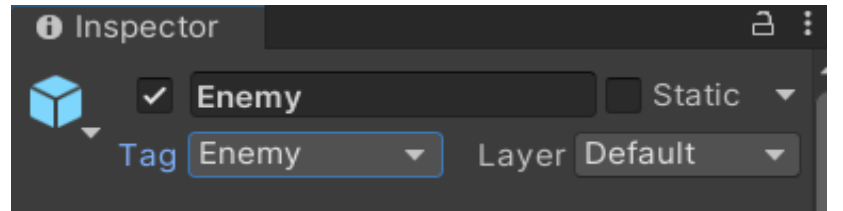
Laser için öncelikle laser prefab dosyasını silin. Sonra sprites klasöründen laser asset dosyanızı sahnenize sürükleyerek ekleyin ve laser ismini verin. Ardından yeni laser nesnenize de Box Collider 2D eklentisini ekleyin. Daha sonra inspector panelinden tag kısmından laser tag'ini seçin ve player nesnenizi seçerek laser ile nesnenizde bulunan laser prefab'i bağlayın. Son olarak laser nesnenizi Player ile orantılı şekilde ölçeklendirin, laser\_sc doyanıza bağlayın ve nesnenizi sürükleyerek prefab klasörüne ekleyin.



Enemy için de aynı şekilde prefab klasöründen silin. Ardından sprites klasöründen Enemy Turn Left/Right 0000'dan birini sahnenize sürükleyerek ismini Enemy yapın. Daha sonra Enemy\_sc doyanız ile yeni enemy nesnenizi bağlayın. Yeni enemy nesnenize Rigidbody 2D ve Box Collider 2D eklentilerini ekleyin. Daha sonra Box collider eklentisinin içinden Is Trigger seçeneğini seçin, Rigidbody eklentisinden ise Simulated seçeneğini seçin ve Gravity scale ölçeğine 0 değerini girin.



Tag kısmından Enemy tag'ini seçin ve son olarak SpawnManager nesnenizi seçin ve Enemy Prefab özelliği ile yeni enemy nesneizi bağlayın



# Script Değişiklikleri

Enemy Script dosyanızdaki  
OnTriggerEnter(Collider other)  
fonksiyonunu yandaki gibi  
değiştirin.

```
void OnTriggerEnter2D(Collider2D other){  
    Player_sc player = other.GetComponent<Player_sc>();  
    if(other.tag == "Player"){  
        player.Damage();  
        Destroy(this.gameObject);  
    }  
    else if(other.tag == "Laser"){  
        //? Always destroy the object of script which you are coding  
        /* So.. destroy laser first then destroy enemy  
        Destroy(other.gameObject);  
        Destroy(this.gameObject);  
    }  
}
```

Böylelikle 3D yapılmıdan 2D şekline geçiş yapmış olacağız.