

**Ad: Yusuf**

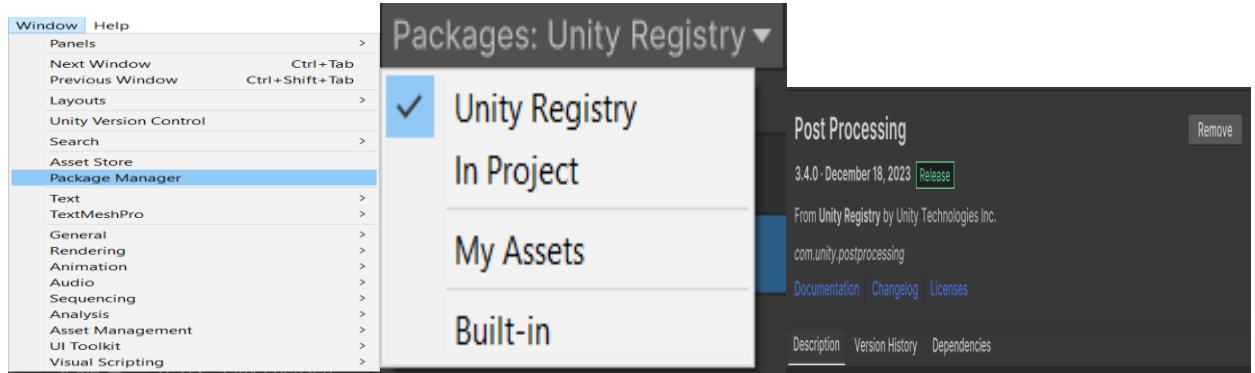
**Soyad: Güney**

**Numara: 22360859041**

**Ders: Oyun Programlama Hafta 10 Rapor**

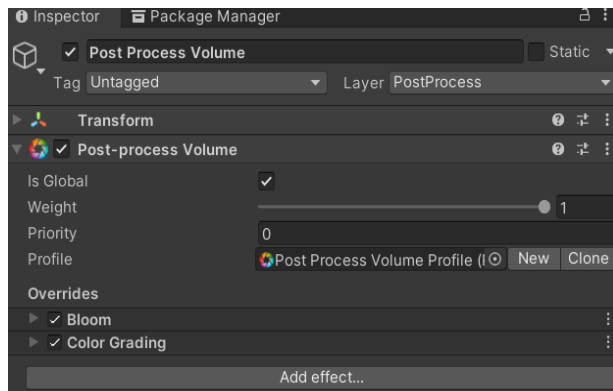
## 1-Post processing paketinin kurulumu

Unity'de **Post Processing** paketini kurmak için, sol üst köşedeki **Window** menüsünden **Package Manager**'ı açıyoruz. Ardından, **Package Manager** penceresinde sol üstte bulunan sekmeden **Unity Registry**'yi seçiyoruz. Sağ üstteki arama çubuğuna **Post Processing** yazarak paketi seçip **Install** butonuna tıklayarak kurulumu tamamlıyoruz.

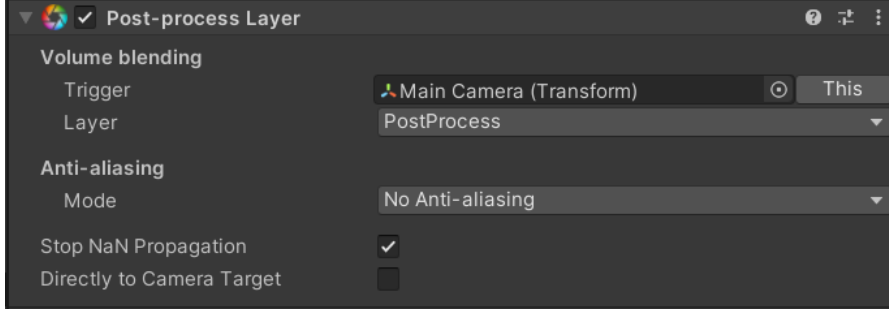


## 2-Projeye post processing uygulanması (post processing volume oluşturma, layer ekleme ve ayarlama)

Hierarchy penceresinde sağ tıklayarak **Create Empty** nesnesi oluşturuyor ve ismini **Post Process Volume** olarak ayarlıyoruz. Daha sonra bu nesneye tıklayarak **Add Component** seçeneğiyle **Post-process Volume** komponentini ekliyoruz. Tekrardan bu nesnenin içerisinde sağ üst köşede **Layer**'a tıklayarak **Add Layer** seçeneğiyle **PostProcess** adında bir layer ekliyor ve layerımızı **PostProcess** olarak ayarlıyoruz.

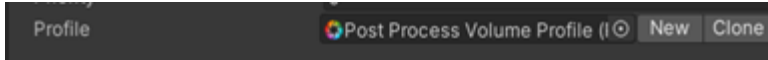


Main Cameraya tıklayarak buraya da **Post-process Volume** komponentini ekliyoruz ve layerımız **PostProcess** olarak ayarlıyoruz.



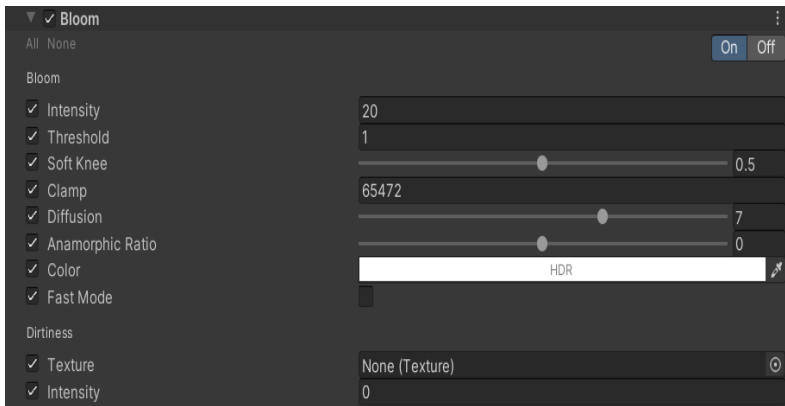
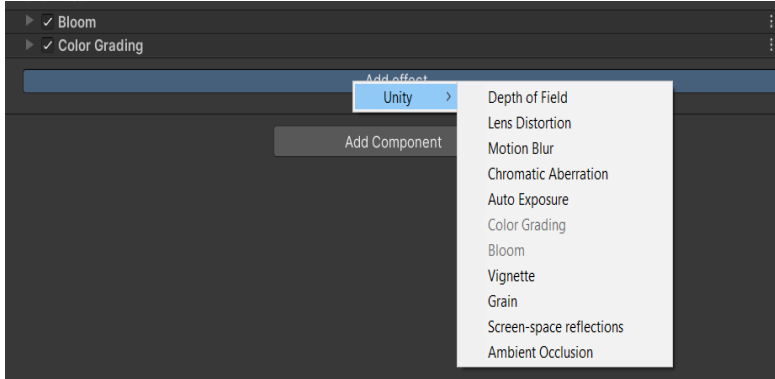
### 3-Profil oluşturma

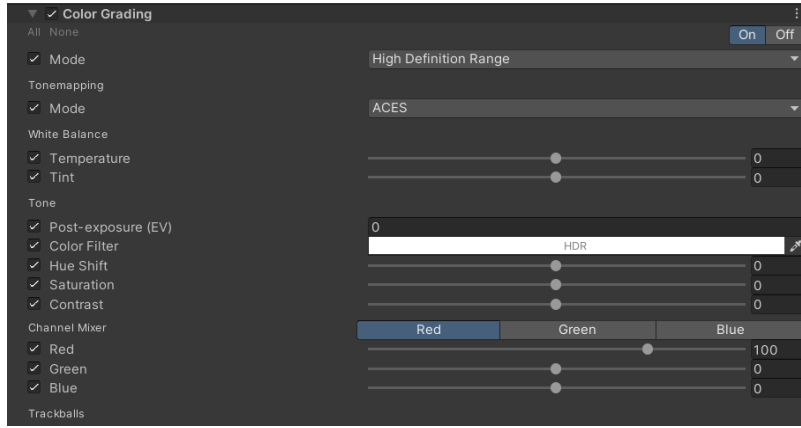
**Post-process Volume** içerisindeki **Profile** seçeneğinin yanındaki new seçeneğine tıklayarak profilimizi oluşturunuz.



### 4-bloom ve color grading etkileri ekleme)

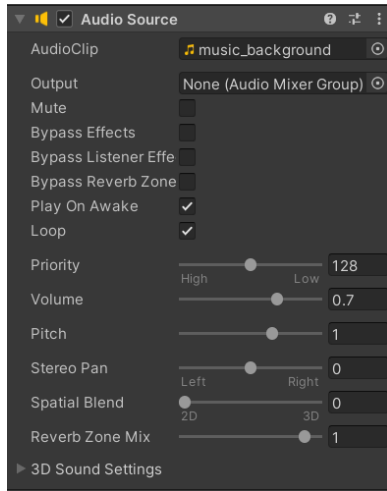
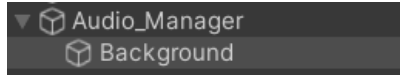
Tekrardan **post-process Volume** içerisinde en altta bulunan add effect e tıklayarak Bloom ve color grading efektlerini ekliyoruz.





## 5-Arkaplan müziği ekleme

**Hierarchy** penceresinde yeni bir **Create Empty** nesnesi oluşturup adını **Audio\_Manager** olarak belirliyoruz. Ardından, bu nesnenin içine bir **Create Empty** daha ekleyerek adını **Background** olarak ayarlıyoruz. **Background** nesnesine **Add Component** butonuyla bir **Audio Source** bileşeni ekliyoruz. Son olarak, **Assets** klasöründeki **Audio** klasöründe bulunan **music\_background** dosyasını, **AudioClip** alanına sürükleyip bırakıyoruz. Oyunu başlattığınızda arka plan müziği otomatik olarak çalmaya başlayacaktır.



## 6-Lazer atış müziği ekleme

Lazer atış müziğini eklemek için lazer atışını yapmış olduğumuz player nesnesinde bazı değişiklikler yapmamız gerekiyor. Player nesnesi içine **Audio Source** componentini ekliyoruz. Player\_sc fonksiyonun içerisinde

**AudioSource audioSource;**

**AudioClip laserSoundClip;**

Değişkenlerini tanımlıyoruz ve untiy ekranında **Player** nesnesindeki **Player\_sc** ye gelip **Laser Sound Clip** değişkenine **Audio** dosyası içerisindeki **laser\_shot** dosyasını sürükleyip bırakıyoruz.

**Player\_sc** içerisinde **audioSource** komponentine ulaşmak için aşağıdaki kod satırını yazıyoruz.

```
audioSource = GetComponent();
```

if kontrolünü de aşağıdaki satırda yapıyoruz.

```
if (audioSource == null){  
    Debug.LogError("Ses Dosyası Bulunamadı");  
}  
else{  
    audioSource.clip = laserSoundClip;  
}
```

Son olarak Lazer atışını yaptığımız fonksiyonda **audioSource.Play();** komutunu yazarak

Lazer atış sesini aktif ediyoruz

```
1 reference  
void FireLaser(){  
    if ((Input.GetKeyDown(KeyCode.Space)) && (Time.time > nextFire)){  
        if(!isTripleShotActive)  
        {  
            Instantiate(laserPrefab, transform.position + new Vector3(0, 1.05f, 0), Quaternion.identity);  
        }  
        else if(isTripleShotActive)  
        {  
            Instantiate(tripleShotPrefab, transform.position + new Vector3(-1.53f, 0.5f, 0), Quaternion.identity);  
        }  
        audioSource.Play();  
        nextFire = Time.time + fireRate;  
    }  
}  
1 reference
```

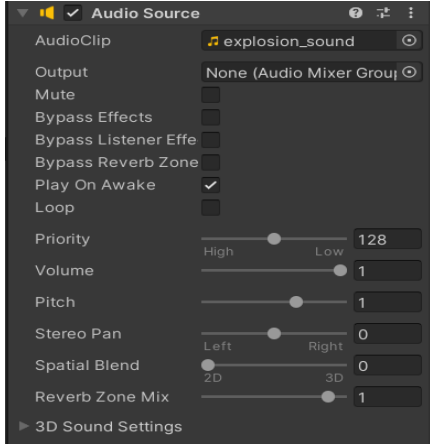
## 7- Patlama müziği ekleme

Patlama müziğini iki yerde kullanmamız gerekiyor

1- Asteroid patlaması

2- Düşman gemilerinin patlaması

Explosion prefabı içerisine **Audio Source**'u ekliyor ve **AudioClip**'e assets klasörü içerisindeki **explosion\_sound**'u ekliyoruz. Oyunu başlattığımızda asteroid patlama sesi gelecektir.



Enemy patlama sesi için enemy\_sc dosyası içerisinde bazı değişiklikler yapmamız gerekiyor.

**AudioSource** türünde bir değişken tanımlıyoruz.

Son olarak OnTriggerEnter2D fonksiyonunun içerisinde enemy nesnesinin patlama anından sonra patlama sesini aktif ediyoruz

```
AudioSource audioSource;
```

```
void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        Player_sc playersc = other.GetComponent<Player_sc>();
        playersc.Damage();

        anim.SetTrigger("OnEnemyDeath");
        speed = 0;
        audioSource.Play();
        Destroy(this.gameObject, 2.5f);
    }
    else if(other.tag == "Laser"){
        Destroy(other.gameObject);
        if (player_sc != null){
            player_sc.UpdateScore(10);
        }
        anim.SetTrigger("OnEnemyDeath");
        speed = 0;
        audioSource.Play();
        Destroy(this.gameObject, 2.5f);
    }
}

if(audioSource == null){
    Debug.LogError("Patlama sesi dosyası bulunmadı.");
}
```

**8-Bonus yakalama müziği ekleme**

Bonus\_sc dosyasının içerisinde AudioClip türünde bir değişken tanımlıyoruz.

```
[SerializeField]  
1 reference  
AudioClip audioClip;
```

Daha sonra bonusları aktif ettiğimiz fonksiyonda ses dosyasına ulaşmamız için gereken kod satırını yazıyoruz.

```
AudioSource.PlayClipAtPoint(audioClip,transform.position);
```

Son olarak bonus prefablerimize **Audio** dosyası içerisindeki **power\_up\_sound** ses dosyalarını ekliyoruz.

Kodun tamamına ulaşmak için github sayfasına bakabilirsiniz.

Github Linki: