

BLM0364 OYUN PROGRAMLAMA – HAFTA 10 RAPORU

Yasin Ekici

21360859029

Rapor İeriđi

Post processing paketinin kurulumu, projeye post processing uygulanması (post processing volume oluřturma, layer ekleme ve ayarlama, profil oluřturma, bloom ve color grading etkileri ekleme), arkaplan m¼ziđi ekleme, lazer atıř m¼ziđi ekleme, patlama m¼ziđi ekleme, bonus yakalama m¼ziđi ekleme.

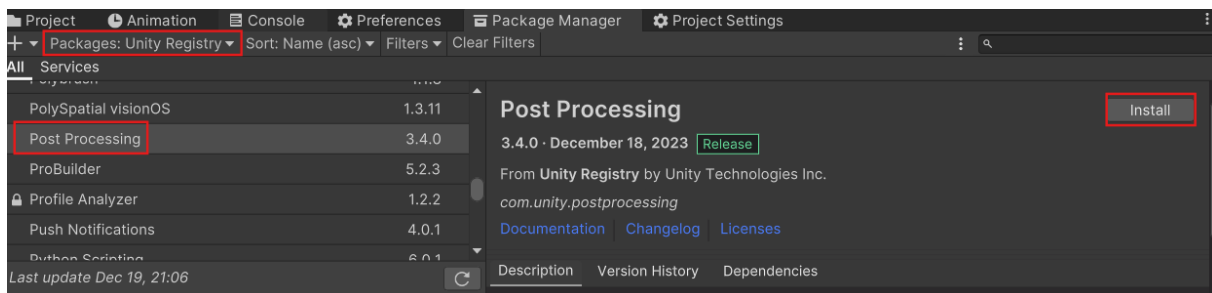
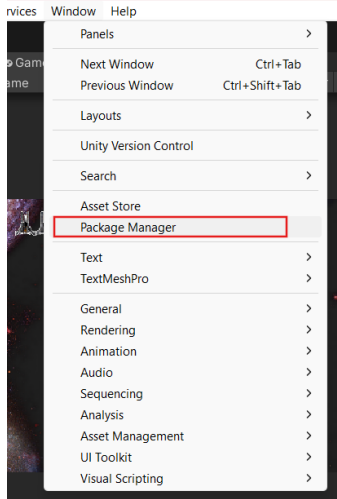
Bu raporda anlatılan kodlar iin: <https://github.com/YasinEkici/game-programming>

Asset'ler iin:

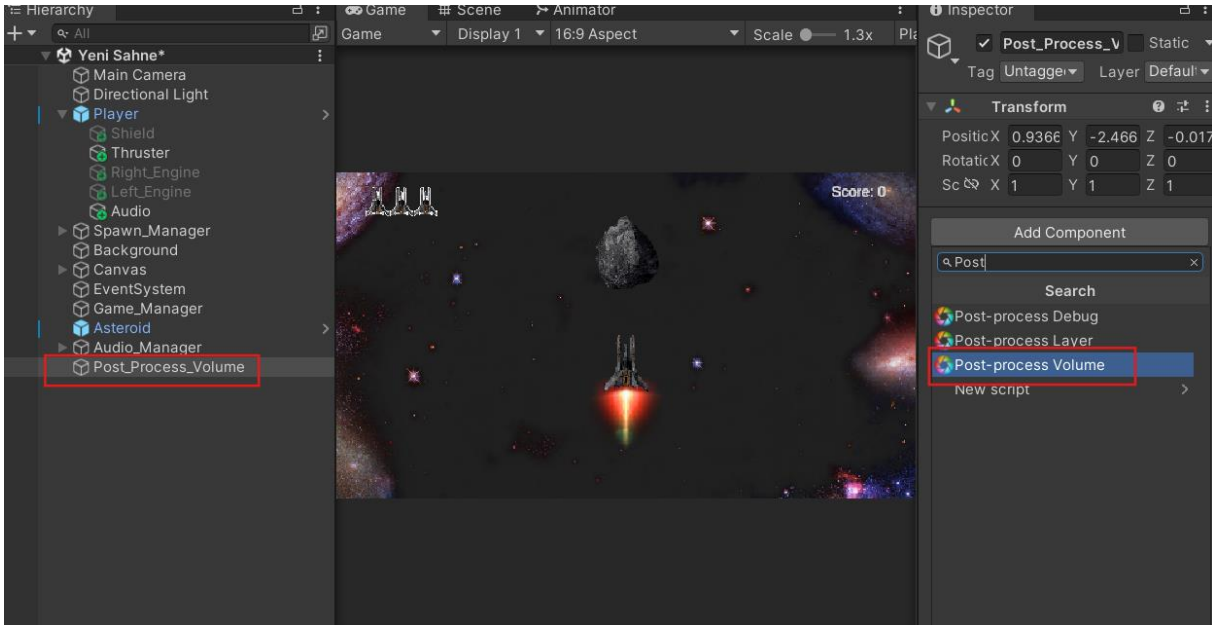
<https://e.pcloud.link/publink/show?code=XZhFVJZuzLw5aWEuizh3kDHE43iNVyak6l7>

1. Post Processing Kurulumu

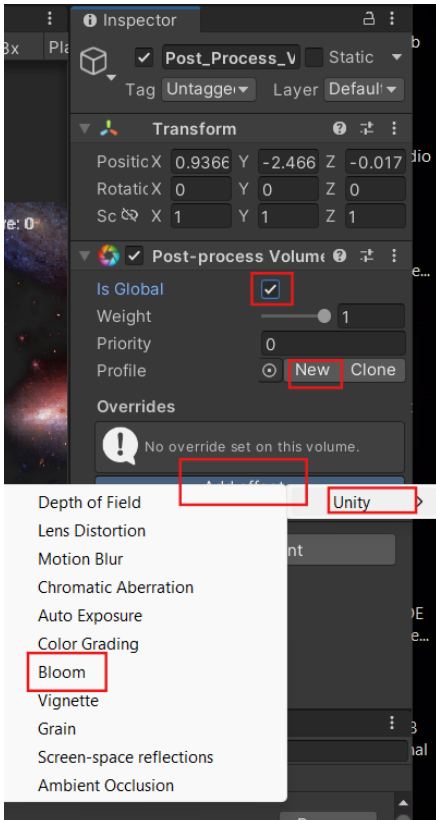
- Window -> Package Manager'a gidin ve Post Processing'i bulun.
- Post Processing paketini y¼kleyin.



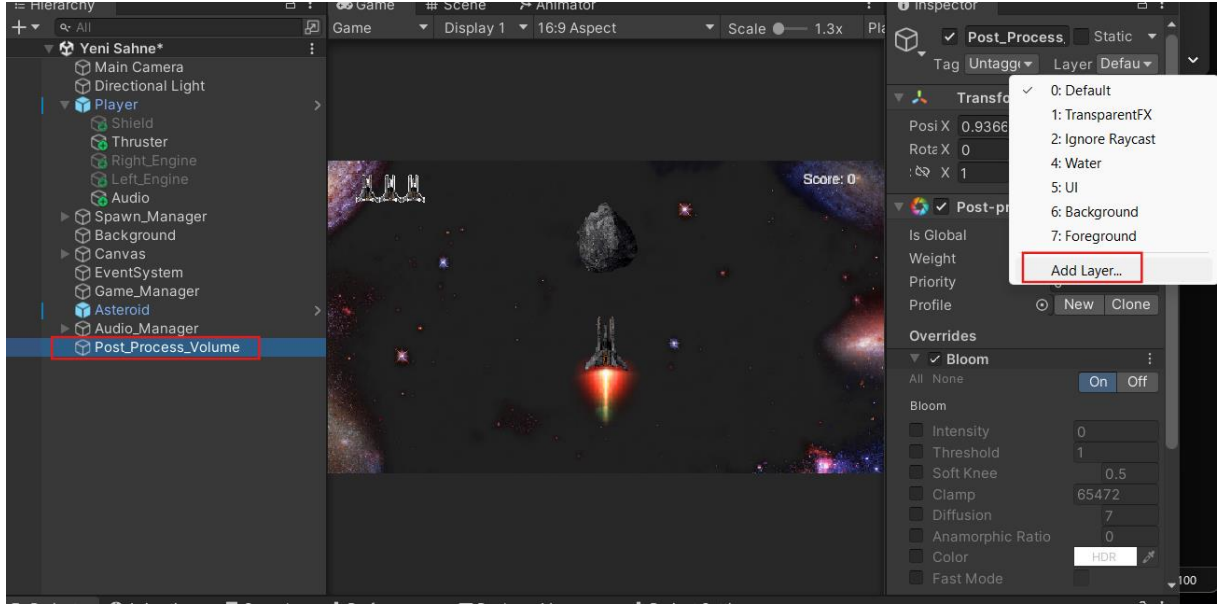
- Hiyerarşi Penceresinde sağ tıklayın ve boş bir nesne oluşturun.
- Nesnenin adını Post_Process_Volume olarak değiştirin.
- Yeni bir bileşen ekleyin: Post Process Volume.



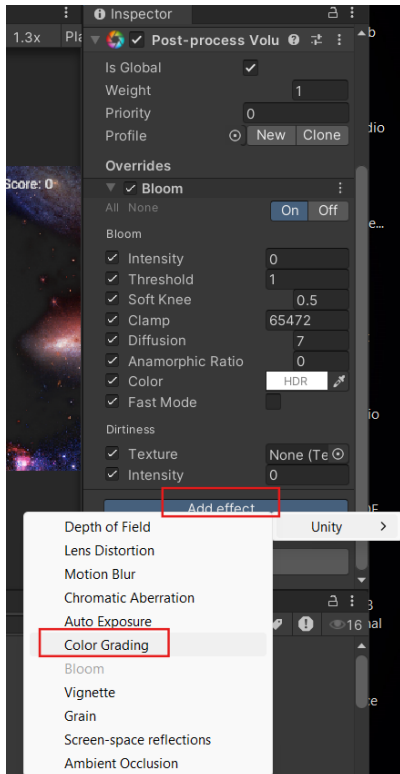
- isGlobal ayarını true yapın.
- Profilin yanındaki New butonuna tıklayın.
- Farklı profiller oluşturarak değişimler yapabilirsiniz.
- Add Effect'e ardından Unity'ye tıklayın ve Bloom'u seçin.
- Post Processing profilini render etmemiz gerekiyor.



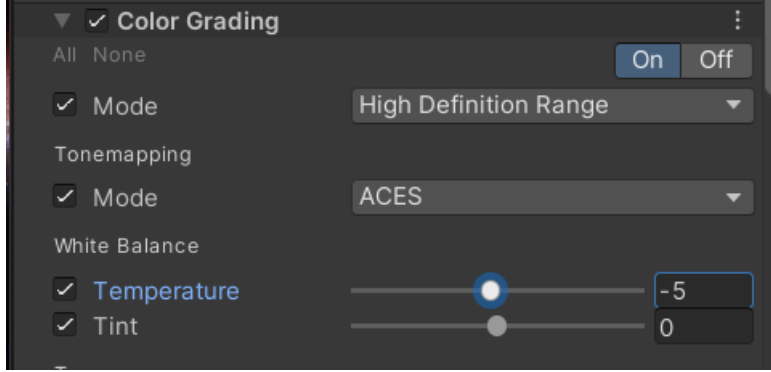
- Main Camera'yı seçin.
- Post Process Layer componentini ekleyin.
- Layer kısmı için yeni bir layer'a ihtiyacımız var.
- Post_Process_Volume'dan yeni bir layer ekleyin, adını Post Processing koyun;



- Main Camera ve Post_Process_Volume'un Layer'larını Post Processing olarak ayarlayın.
- Şu an Post_Process_Volume'un değişiklikleri ekrana yansıyor.
- Yeni bir efekt ekleyin.
- Color Grading'i seçin

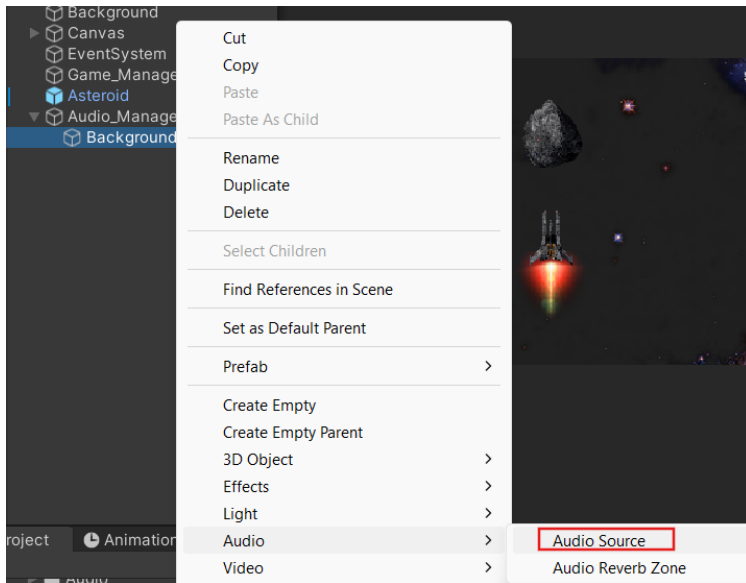


- Tone Mapping'i ACES olarak ayarlayın.
- Temperature, Tint vs gibi ayarları istediğiniz şekilde ayarlayın.
- İsteddiğiniz efektleri projenize ekleyebilirsiniz.

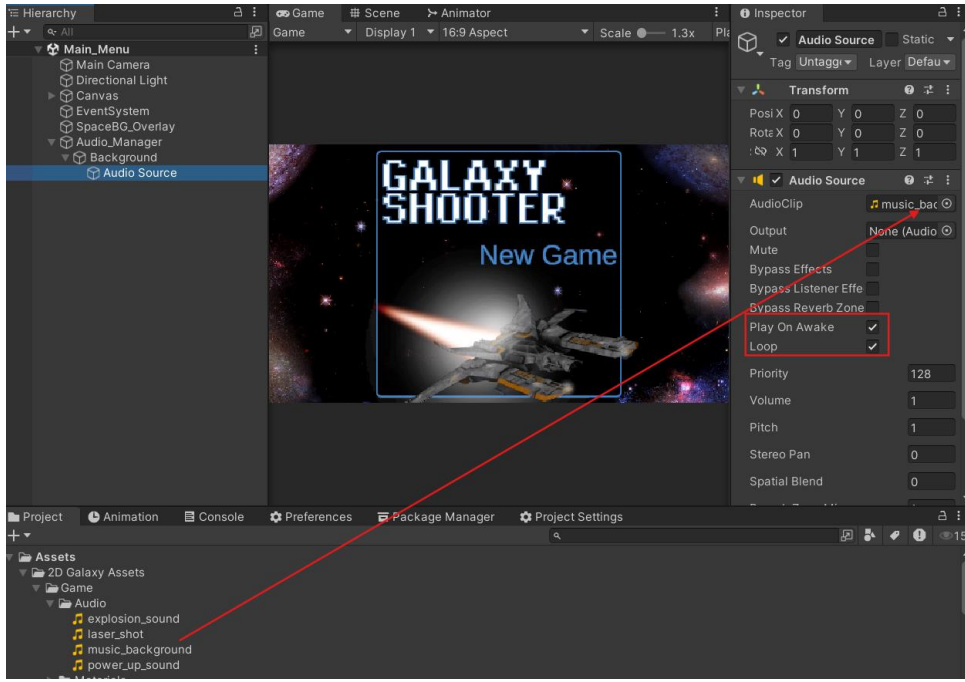


2. Background Music

- Boş bir oyun nesnesi oluşturun.
- Adını Audio_Manager olarak değiştirin.
- Audio_Manager içinde arka plan müziğine özel boş bir oyun nesnesi oluşturun.
- Adını Background olarak değiştirin.
- Bir bileşen ekleyin: Audio Source.

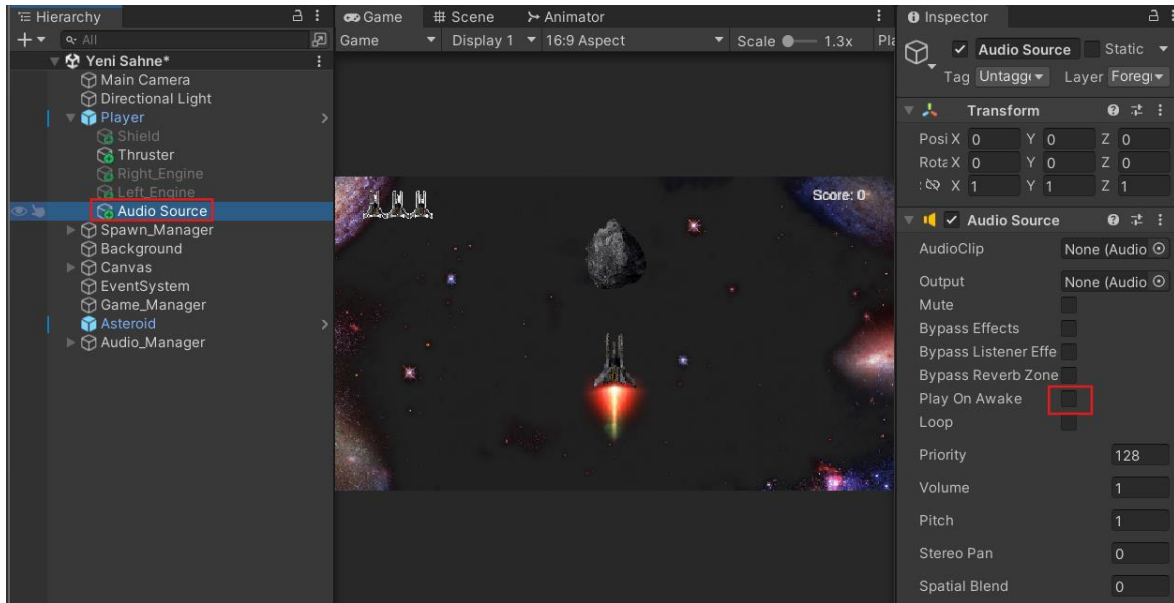


- Ses kaynağını ayarlayın.
- AudioClip: Audio klasöründen music_background'ı ayarlayın.
- Play on Awake ve Loop ayarını true yapın.



3. Laser Sesi

- FireLaser metodunu Player scriptinde güncelleyin.
- Işık sestten daha hızlı olduğu için ses efektini ateş etme görsel efektinden sonra oluşturmamız lazım.
- Hiyerarşi penceresinden Player'ı seçin ve bir bileşen ekleyin: Audio Source.
- Henüz AudioClip ayarlamayın
- Play on Awake ayarını kapatın.

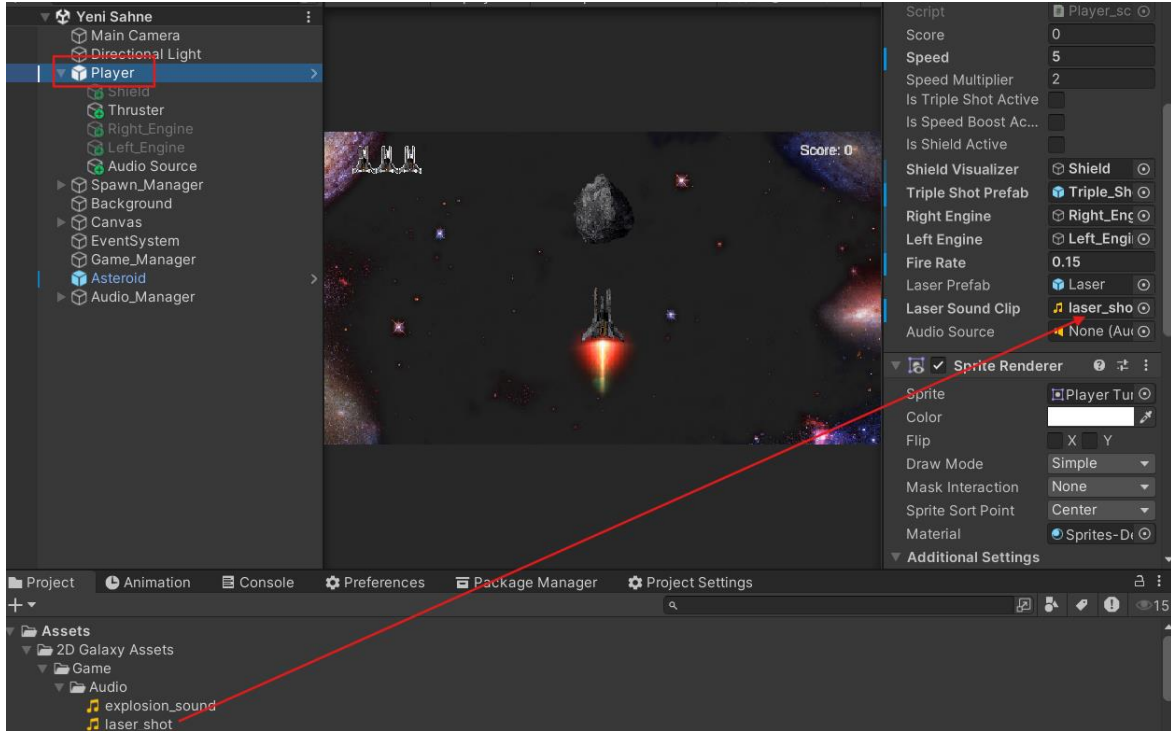


- Sesi depolamak için bir değişken oluşturun ve serialize edin.
- Laser Sound Clip'i inspector'dan ayarlayın: laser_shot.

- AudioSource bileşeni için bir handler değişkeni oluşturun.
- Player scriptinde;

[SerializeField] AudioClip laserSoundClip;

[SerializeField] AudioSource audioSource;



- Start metodunda AudioSource bileşenini alın.
- Null kontrolü yapın ve ses kaynağı klibini ayarlayın.
- FireLaser metodunu güncelleyin.
- Lazer görselinden sonra ses kaynağını oynatın.
- Start fonksiyonuna;

```
audioSource = GetComponent<AudioSource>();
```

```
if(audioSource == null){
```

```
    Debug.LogError("Audio Source bulunamadi");
```

```
}
```

```
else{
```

```
    audioSource.clip = laserSoundClip;
```

```
}
```

- FireLaser fonksiyonunu güncelleyin;

```
public void FireLaser(){
```

```
    nextFire = Time.time + fireRate;
```

```

if(isTripleShotActive == true){

    Instantiate(tripleShotPrefab, transform.position, Quaternion.identity);

}

else{

    Instantiate(laserPrefab, transform.position + new Vector3(0, 1.15f,0),
    Quaternion.identity);

}

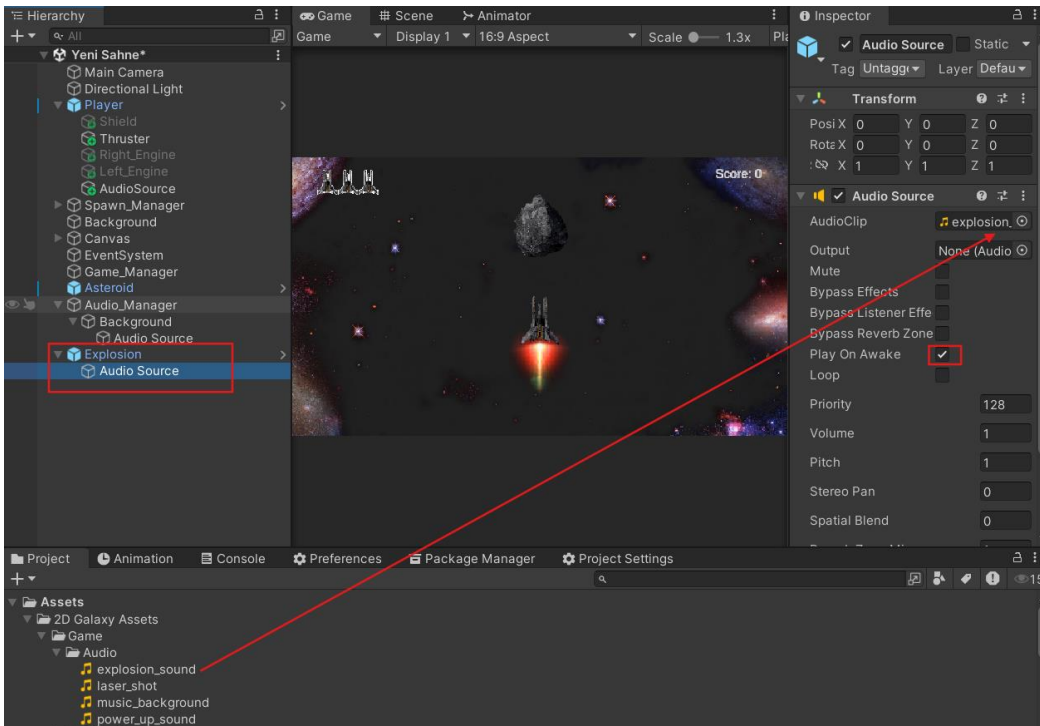
audioSource.Play();

}

```

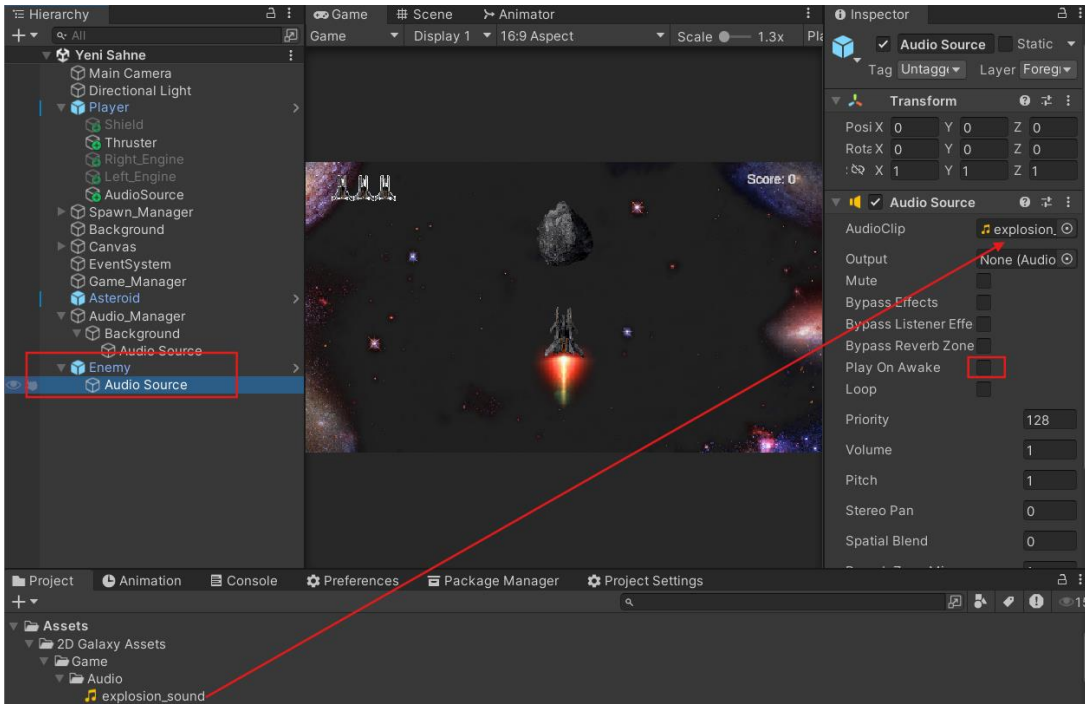
4. Explosion Sesleri

- Asteroid patladığında patlama sesi oynatın.
- Bir Explosion prefabı da var.
- Explosion prefabına Audio Source bileşenini ekleyin.
- AudioClip: explosion_sound olarak ayarlayın.
- Play on Awake ayarını true yapın.



- Asteroid Explosion prefabını kullanır, ancak Düşman nesnelerinin farklı bir patlama mekanizması vardır.
- Düşman oyun nesnesini seçin ve bir bileşen ekleyin: AudioSource.

- Play on Awake ayarını false yapın.
- AudioClip: explosion_sound olarak ayarlayın.



- Enemy scriptini güncelleyin.
- AudioSource bileşeni için bir handler değişkeni oluşturun.
- Start metodunda bileşeni alın ve null kontrolü yapın.
- Ses kaynağını iki yerde oynatın: lazer tarafından vurulduğunda veya oyuncuyla çarpıştığında.
- Start Fonksiyonuna;

audioSource = GetComponent();

- OnTriggerEnter2D son hali;

```
void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        Player_sc player = other.transform.GetComponent<Player_sc>();
        if(player != null){
            player.Damage();
        }
        anim.SetTrigger("OnEnemyDeath");
        speed = 0f;
        audioSource.Play();
        Destroy(this.gameObject, 2.8f);
    }
    else if (other.tag == "Laser"){
        Destroy(other.gameObject);
        if(player != null){
            player.UpdateScore(10);
        }
    }
}
```



```

    }
    anim.SetTrigger("OnEnemyDeath");
    speed = 0f;
    audioSource.Play();
    Destroy(this.gameObject, 2.8f);
}

```

5. Bonus Sesleri

- Oyuncu bir bonus aldığında sesi oynatın.
- Bonus scriptini güncelleyin.
- OnTriggerEnter2D metodunda ses ekleyebiliriz.
- Ancak nesne hemen siliniyor bu yüzden PlayClipAtPoint fonksiyonunu kullanın, bu fonksiyon verilen bir konumda bir AudioClip oynatır.
- Bu fonksiyon bir AudioSource oluşturur ancak klip oynatıldıktan sonra otomatik olarak yok eder.
- Bonus scriptini güncelleyin.
- Bir AudioClip değişkeni oluşturun ve serialize edin;

```
[SerializeField] AudioClip audioClip;
```

- AudioClip'i inspector'dan ayarlayın.
- Tüm bonus prefablarını seçin ve Audio Clip'i ayarlayın: power_up_sound.
- Oyuncu bonuslarla çarpıştığında sesi oynatın.
- OnTriggerEnter2D metodunu güncelleyin;

```

void OnTriggerEnter2D(Collider2D other){

    if(other.tag == "Player"){

        Player_sc player = other.transform.GetComponent<Player_sc>();

        AudioSource.PlayClipAtPoint(audioClip, transform.position);

        if(player != null){

            switch(bonusID){

                case 0:

                    player.TripleShotActive();

                    break;

                case 1:

                    player.SpeedBoostActive();

```

```
        break;
```

```
    case 2:
```

```
        player.ShieldBonusActive();
```

```
        break;
```

```
    case 3:
```

```
        Debug.Log("Default value");
```

```
        break;
```

```
    }
```

```
 }
```

```
 Destroy(this.gameObject);
```

```
 }
```

```
 }
```