**11. Hafta Raporu – Aleks Dulda 21360859025**

**Post processing paketinin kurulumu:**

1. Unity Projenizi Açın: Unity Editor'ü açın ve üzerinde çalıştığınız projeyi yükleyin.
2. Package Manager'i Açın:
   * Menü çubuğundan Window > Package Manager seçeneğine tıklayın.
3. Post Processing Paketini Bulun:
   * Unity Registry bölümünde "Post Processing" yazın. Eğer bu sekmeyi göremiyorsanız, sol üst köşeden "Packages: Unity Registry" seçeneğini seçin.
4. Paketi Yükleyin:
   * "Post Processing" paketini seçin ve sağ alt köşede bulunan Install düğmesine tıklayın.

**Projeye post processing uygulanması:**

1. Post Processing Volume için GameObject Ekleyin:
   * Scene Hierarchy'ye sağ tıklayın ve Create Empty seçeneğini seçerek bir boş GameObject oluşturun.
   * Bu GameObject'e anlamlı bir isim verin (örneğin, "PostProcessingVolume").
2. Post-Processing Volume Bileşenini Ekleyin:
   * Bu GameObject'i seçin ve Inspector penceresinde Add Component butonuna tıklayın.
   * "Post-Processing Volume" bileşenini ekleyin.
3. Global Olarak Ayarlayın:
   * **metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

     Açıklama otomatik olarak oluşturuldu**metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

     Açıklama otomatik olarak oluşturulduPost-Processing Volume bileşeninde Is Global seçeneğini işaretleyin.
4. **Yeni Bir Profil Oluşturun:**

* Post-Processing Volume bileşeninde **New Profile** butonuna tıklayın.
* Bu işlem yeni bir profil oluşturacak ve düzenlemenize izin verecektir.

1. **Efektleri Ekleyin:**

* Profili düzenlemek için açılan pencerenin sağındaki **Add Override** butonuna tıklayın.
* **Unity > Post-Processing** kategorisinden efektleri seçebilirsiniz.
* Örnek olarak:
  + **Bloom**: Sahneye parlaklık efekti ekler.
  + **Color Grading**: Renk, ton ve kontrast üzerinde kontrol sağlar.

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

1. **Kameraya Post-Processing Layer Ekleme:**
   * **Main Camera**'yı seçin.
   * **Inspector** penceresinde **Add Component** butonuna tıklayın ve "Post-Processing Layer" ekleyin.
2. **Layer Ayarlarını Yapın:**
   * **Layer** seçeneğini "PostProcessing" olarak ayarlayın (yoksa yeni bir **PostProcessing** layer'ı oluşturun).
   * Bu layer, Post-Processing Volume'un etkisini görmesini sağlar.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu şekilde layer oluşturup atamasını istediğimiz oyun nesnesine yapabiliriz.

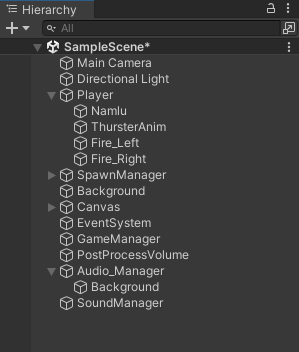
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Audio Ekleme**

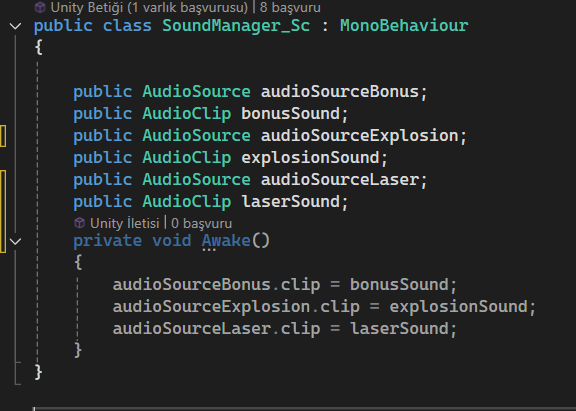
**1. Arka Plana Ses Eklenmesi**

* Boş bir GameObject oluşturulup, ismi "Audio\_Manager" olarak belirlendi.
* "Audio\_Manager" nesnesinin altına bir boş GameObject daha oluşturulup, ismi "Background" olarak belirlendi.
* "Background" nesnesine AudioSource component'i eklendi.
* Play On Awake seçeneği işaretlendi, böylece oyun başladığında müzik otomatik olarak çalmaya başlar.
* Loop seçeneği işaretlendi, böylece müzik sürekli olarak tekrar eder.
* **metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

  Açıklama otomatik olarak oluşturuldu**AudioClip kısmına var olan bir arka plan müziği eklendi.

**2. Ses Efektleri İçin SoundManager**

* Patlama, Lazer ve PowerUp ses efektleri için bir boş GameObject oluşturulup, ismi "SoundManager" olarak belirlendi.
* **3 adet AudioSource component'i** eklendi, her bir ses efekti için yeterli ses çıkışı sağlanması amaçlandı (bu şekilde bir efekt çalarken diğerleri engellenmez).
* **SoundManager** nesnesine "SoundManager\_Sc" adında bir script dosyası eklendi.
* Script dosyasına gerekli kodlar yazıldı.

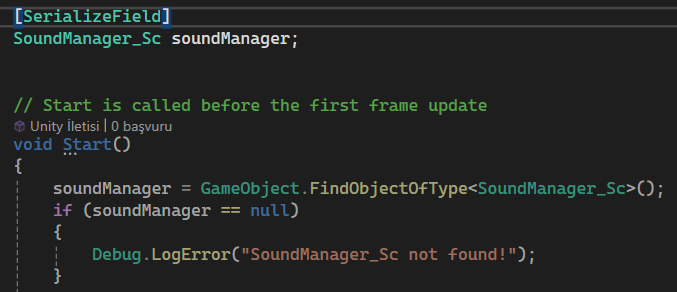
metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

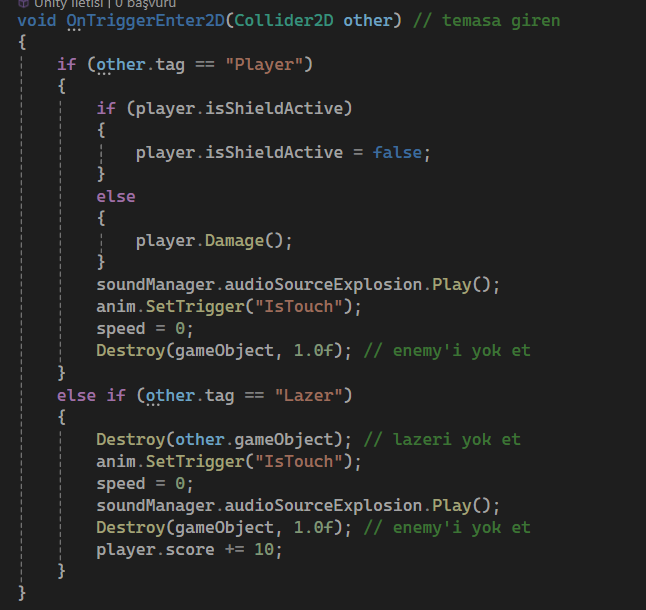
Açıklama otomatik olarak oluşturuldu

* Script içerisinde her bir ses efekti için bir AudioSource ve bir AudioClip tanımladım. Awake() fonksiyonu içinde, her bir AudioSource'a ilgili AudioClip'i atıyorum. Bu sayede, oyun başladığında her ses efekti doğru şekilde hazır oluyor ve çalmak için ihtiyaç duyduğumda hemen kullanılabiliyor.

1. **Değişken Tanımlaması:**
   * **Astreoid\_Sc** script dosyasına, **SoundManager\_Sc** türünde bir değişken ekliyorum:
   * SoundManager\_Sc soundManager;
   * Bu, **SoundManager\_Sc** türünde bir değişken oluşturur.
2. **SoundManager’a Erişim Sağlama:**
   * **soundManager** değişkenine, **GameObject.FindObjectOfType<SoundManager\_Sc>()** kodunu ekleyerek **SoundManager\_Sc** script'ine erişim sağlıyorum:
   * soundManager = GameObject.FindObjectOfType<SoundManager\_Sc>();
   * Bu sayede, **SoundManager\_Sc** script'ini bulmuş ve ona referans vermiş oluyoruz.
3. **Ses Efekti Çalma:**
   * Son olarak, patlama sesini çalmak için şu kodu kullanıyorum:
   * soundManager.audioSourceExplosion.Play();
   * Bu kodu, asteroid yok olmadan önce çağırarak patlama sesini çalıştırıyorum.

Bu örnekte olduğu gibi diğer ses efektlerini de ekleyebilir oyunu daha zevkli hale getirebiliriz.





**Kaynakça**

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\_11