

Ad: Yusuf

Soyad: Güney

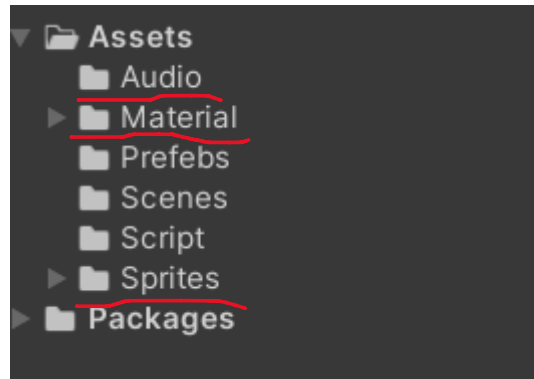
Numara: 22360859041

Ders: Oyun Programlama Hafta 5 Rapor

1-Materials, Audio ve Sprites asset'lerinin projeye import edilmesi

Audio	3.186.292	2.932.909	Dosya klasörü
Materials	5.238	2.311	Dosya klasörü
Sprites	73.636.914	71.891.435	Dosya klasörü

Gönderilen uzay oyunu assetlerini proje ekranımızdaki Assets klasörünün içerisine sürükleyerek bırakıyoruz.



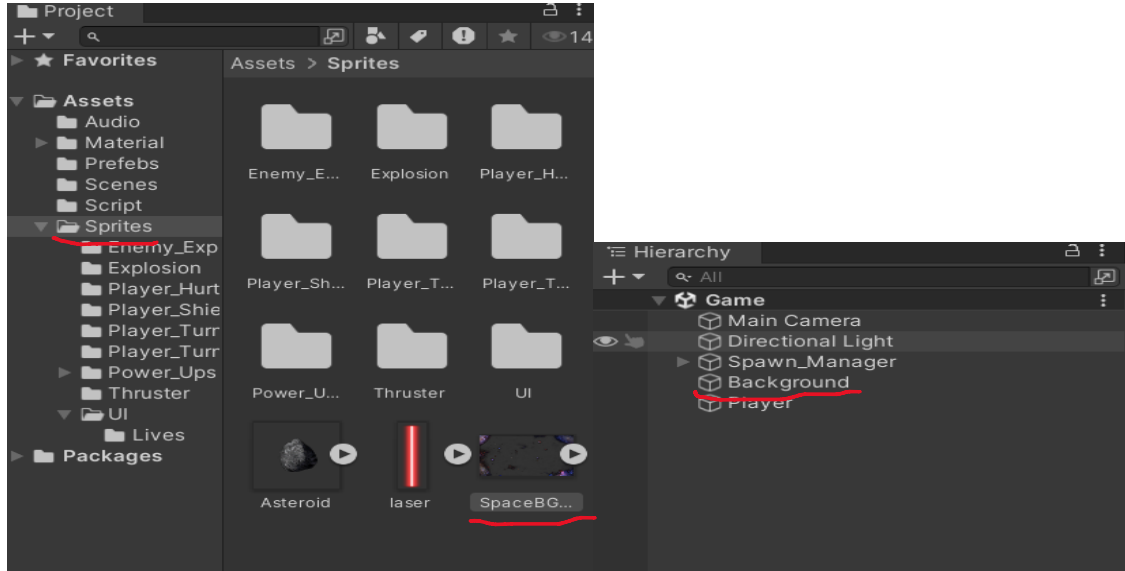
2-Perspektif görünümü, 2D-3D görünüm arasında geçiş yapma

Proje sayfasında Scene altında bulunan butonlardan 2D yi açıp kapayarak geçiş yapılabilir.



3-Arkaplan için sprite kullanımı

Assets klasörünün altındaki Sprites klasörünün içerisindeki backgroundu Hierarchy sekmesine sürükleyerek arkaplanı oluşturabiliriz.

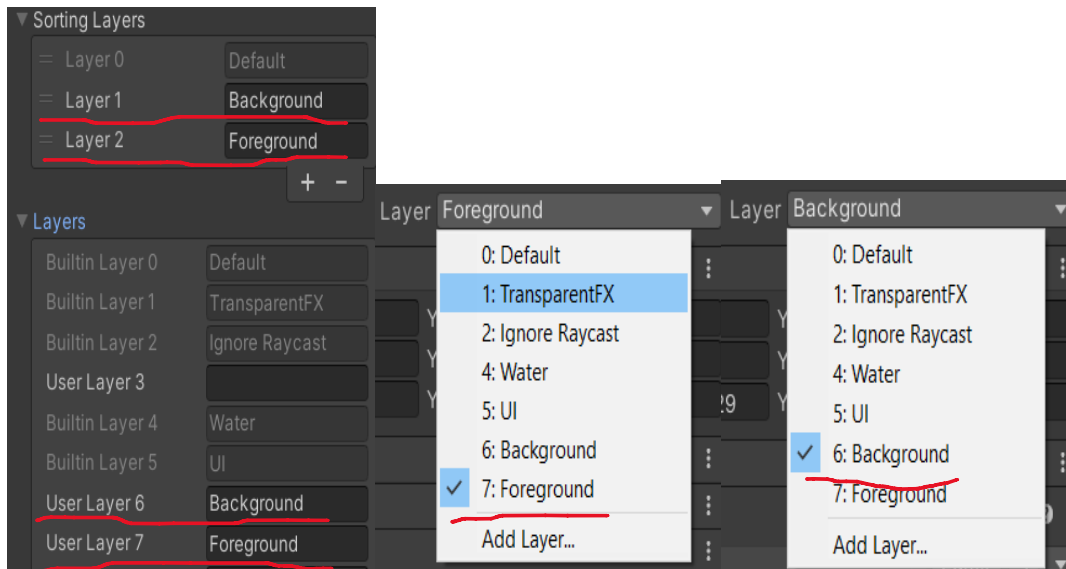


4-Sorting layer kullanımı

Sorting Layer, 2D projelerde belirli nesnelerin ekranda hangi sırada görüneceğini belirlemek için kullanılan bir ayardır. Bizde bu projede Player'ı yani uzay gemimizi arkaplandan önde göstermek için sorting layer kullanacağız.

Öncelikle en sağ üst köşede bulunan **Layers'a** tıklayarak **Edit Layers** seçeneğine tıklıyoruz daha sonra buraya **Background** ve **Foreground** layerlarını ekliyoruz. Burada Background layerını Foreground layerından önce yazmaya dikkat ediyoruz.

Daha sonra Player'a tıklayarak **inspector** penceresinden layer'ı Foreground, Background'a tıklayarak yine **inspector** penceresinden layerını Background olarak ayarlıyoruz. Böylelikle Player nesnesi arka planın üzerinde gözükecektir.



5-Player, Enemy ve Laser nesnelerinin iki boyutlu olarak tekrar oluşturulması

Öncelikle, bu nesnelerin şekillerini değiştirmek için **Sprites** klasöründen uygun görselleri **Hierarchy** penceresine sürüklüyoruz (Player için uzay gemisi, Enemy için düşman gemisi ve Laser için lazer). Ardından, daha önce oluşturduğumuz nesnelerin üzerindeki C# script dosyalarını, Tag ve Layer ayarlarını yukarıdaki eşleşmelere göre güncelliyoruz.



6-Kodda 2D için gerekli değişikliklerin yapılması

Burada yapmamız gereken en önemli değişikliklerden biri, nesneler üzerindeki **Box Collider** bileşenini **Box Collider 2D** olarak değiştirmektir. Bunun sebebi, oyunumuzu 2D bir ortamda çalıştıracak olmamızdır. **Box Collider 2D** bileşeninde, üç nesne için de **Is Trigger** özelliğini aktif hale getiriyoruz. Ardından, **Edit Collider** seçeneğine tıklayarak sahne üzerinde nesnelerin çarpışma alanlarını nesne boyutlarıyla eşit hale getiriyoruz.

Ayrıca, **Enemy** nesnesi üzerindeki **Rigidbody** bileşenini **Rigidbody 2D** olarak değiştiriyoruz.

7-Enemy ve Laser'den tekrar prefab oluşturma

Hierarchy panelindeki **Enemy** ve **Laser** nesnelerini, Assets klasörünün altındaki **Prefabs** klasörüne taşıyarak Enemy ve Laser nesnelerinden prefab oluşturabiliriz.

Kodda yapılan değişiklikler

Enemy_sc dosyasının içerisindeki **OnTriggerEnter** fonksiyonunu **OnTriggerEnter2D** olarak değiştiriyoruz.

```
void OnTriggerEnter2D(Collider2D other){
```

Kodların tamamına Github sayfamdan ulaşabilirsiniz.

Github Linki: