

BLM0364 OYUN PROGRAMLAMA – HAFTA 5 RAPORU

Yasin Ekici

21360859029

Rapor İeriği

- Materials, Audio ve Sprites asset'lerinin projeye import edilmesi
- Perspektif görünümü, 2D-3D görünüm arasında geçiş yapma
- Arkaplan için sprite kullanımı
- Sorting layer kullanımı
- Player, Enemy ve Laser nesnelerinin iki boyutlu olarak tekrar oluşturulması
- Kodda 2D için gerekli değişikliklerin yapılması
- Enemy ve Laser'den tekrar prefab oluşturma

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

Asset'ler için:

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

1. Materials, Audio ve Sprite asset'lerinin projeye import edilmesi:

Asset dosyası içerisinde çıkardığımız dosyaları hali hazırda var olan Assets klasörü içerisine çıkaralım:

- Materials
- Audio
- Sprites

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

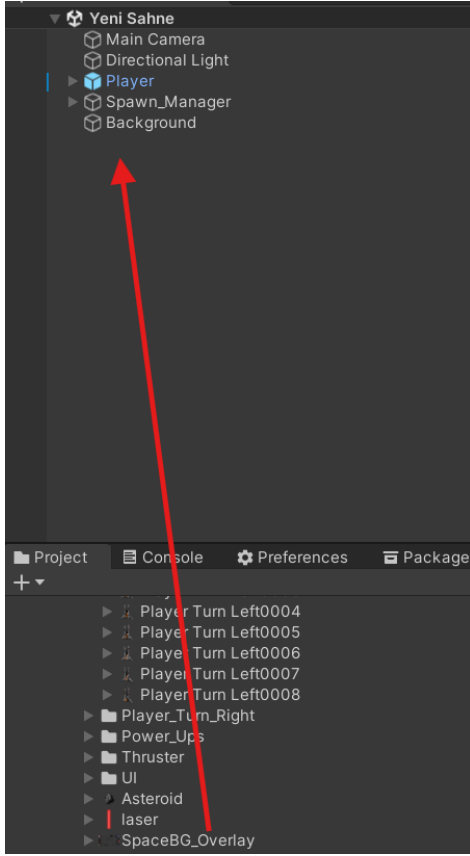
Scene menüsünden 2D kısmına tıklayalım.



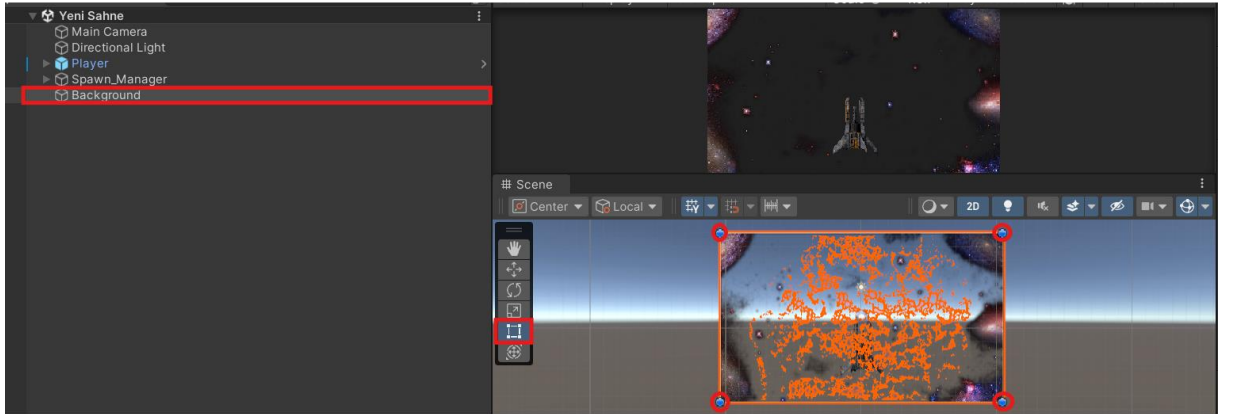
Player nesnesi hala 3D durumda.

Onu 2D yapmadan önce arka planı değiştireceğiz:

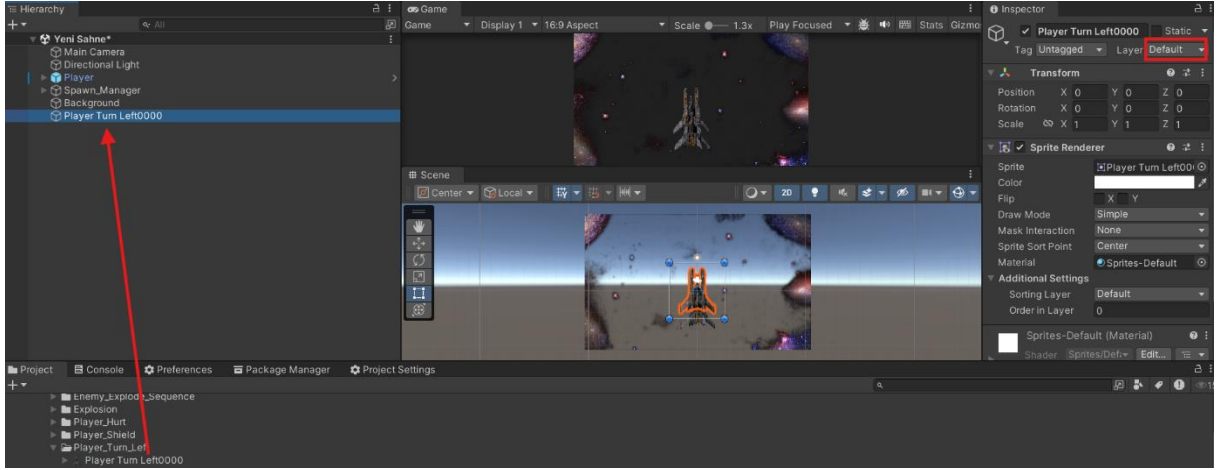
- Asset menüsünden SpaceBG_Overlay dosyasını Hierarchy'e ekleyelim.



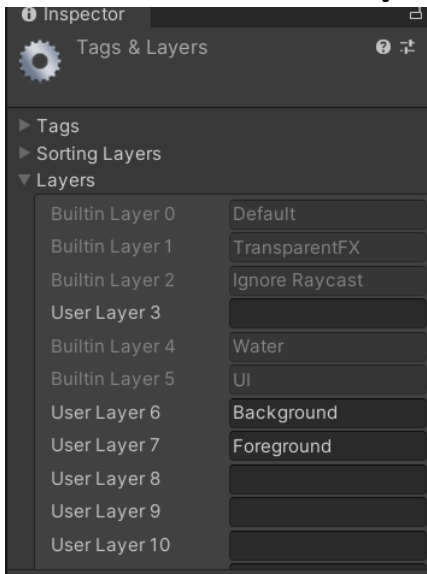
- İsmi "Background" olarak değiştirelim.
- Şimdi Background seçiliyken RectTool'a tıklayıp resmi tüm ekrana sığacak şekilde ayarlayalım.



- Sprite Renderer'ı kontrol et:
- "Player turn Left" nesnesini Hierarchy'ye sürükleyerek katman sırasını (Order in Layer) test edelim ve sıralama değerini değiştirelim.



- Yeni sıralama katmanları oluşturalım: Background ve Foreground.



- Oyuncu ve arka plan için uygun sıralama katmanını seçelim.
- "Player turn Left" nesnesini silelim.

3. Player, Enemy ve Laser nesnelerinin iki boyutlu olarak tekrar oluşturulması

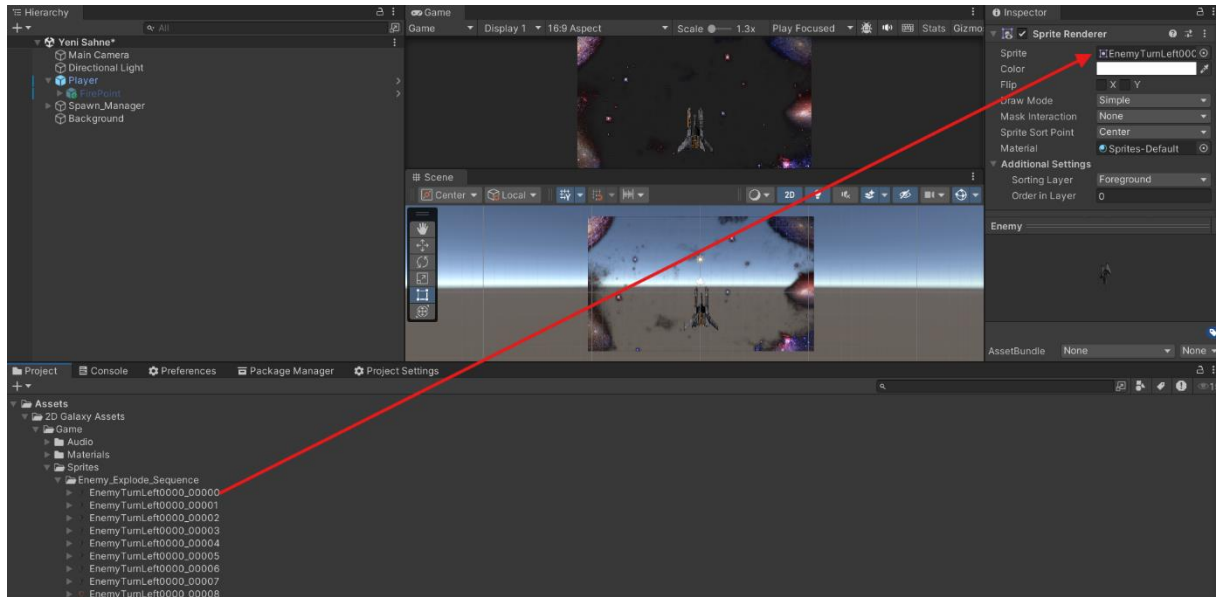
Player'ı 3D'den 2D'ye çevirelim:

- "Player turn left" kullanarak sıfırdan bir Player oluşturalım.
- "Player turn left" nesnesini Hierarchy'ye sürükleyip bırakalım.
- Adını "Player" olarak değiştirelim.
- Player script'ini ekleyelim, eski player nesnesine gidip bileşeni kopyalayıp buraya yapıştıralım.
 - Hız değerini 5 yapalım.

- Laser prefab'ini ayarlayalım.
- Ateş oranını belirleyelim.
- Etiketini Player olarak belirleyelim.
- Eski Player nesnesini silelim.
- Sıralama katmanını Foreground olarak değiştirelim.
- Ölçeği 0.5, 0.5, 0.5 yapalım.

Enemy'yi 3D'den 2D'ye çevirelim:

- Enemy'yi seçip prefab'i açalım.
- Mesh renderer, mesh filter, box collider ve rigidbody bileşenlerini silelim.
- Nesneyi bir sprite'a çevirmek için:
 - Sprite Renderer bileşenini ekleyelim.
 - "Enemy turn left" sprite'ını sürükleyip ekleyelim.



- Sıralama katmanını Foreground olarak değiştirelim.
- Çarpışmaları düzeltmek için:
 - Enemy prefab'ine Rigidbody2D ekleyelim.
 - Gravity scale'ı 0 yapalım.
 - Enemy ve Player prefab'lerine BoxCollider2D ekleyelim.
 - isTrigger özelliğini açalım.
 - Collider'ı daha iyi oturması için RectTool ile düzenleyelim.

3D nesneler için OnTriggerEnter kullandık:

- Şimdi, Laser ve Player çarpışmaları çalışmıyor.
- Bunu OnTriggerEnter2D olarak değiştirelim.
- Ayrıca, Collider'ı Collider2D olarak güncelleyelim.
- Enemy scripti içinde yeni kod:

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

```

        if(player != null){
            player.Damage();
        }
        Destroy(this.gameObject);
    }

    else if (other.tag == "Laser"){
        Destroy(other.gameObject);
        Destroy(this.gameObject);
    }
}

```

Laser'i 3D'den 2D'ye çevirelim:

- Laser prefab'ini silip yeni 2D Laser'i sıfırdan oluşturalım.
 - Laser sprite'ını Hierarchy'ye sürükleyelim.
- Sorting layer'ı Foreground olarak ayarlayalım.
- Ölçeği 0.2, 0.2, 0.2 yapalım.
- BoxCollider2D ekleyelim.
 - Collider'ı sprite'a uyacak şekilde düzenleyelim.
 - isTrigger özelliğini açalım.
- Rigidbody2D ekleyelim.
- Gravity Scale'ı 0 yapalım.
- Laser script'ini ekleyelim.
- Etiketleri Laser olarak ayarlayalım.
- Laser'in offset değerini değiştirmek için:
 - Laser'i Hierarchy'ye ekleyip, Player ve Laser için konumu 0, 0, 0 olarak ayarlayalım.
 - Uygun bir laser offset değeri için en iyi y koordinatını belirleyelim.
 - Player script'inde, FireLaser fonksiyonunda offset değerini güncelleyelim.
- Laser'i prefab yapalım.
 - Sprite'ı Hierarchy'den çıkaralım.
 - Player script'inde Laser prefab'ini ayarlayalım.