

Oyun Programlama Ödevi – Hafta 5

Ad: Eren

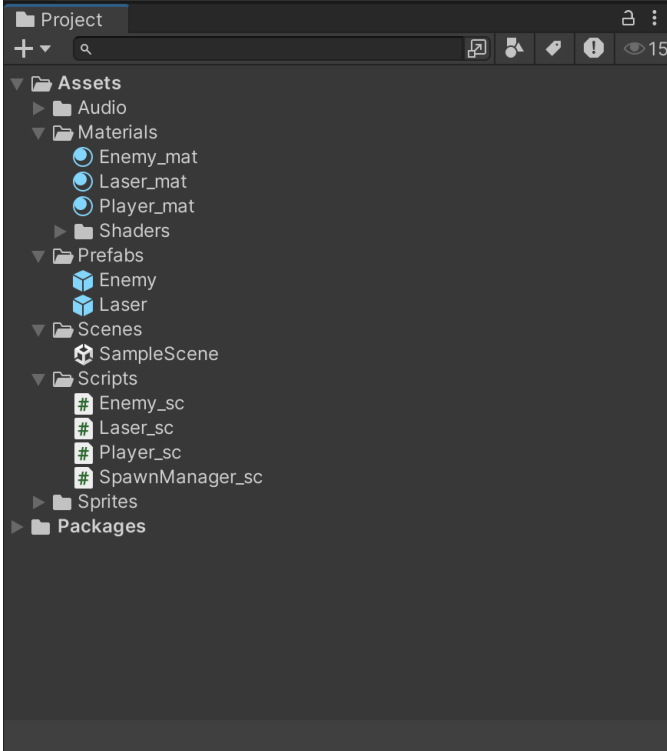
Soyad: Köse

Numara: 22360859075

GitHub Kod Linki: <https://github.com/erennkose/btu-oyun-programlama/tree/main/Hafta5>

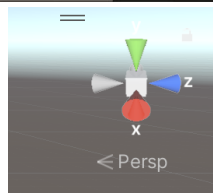
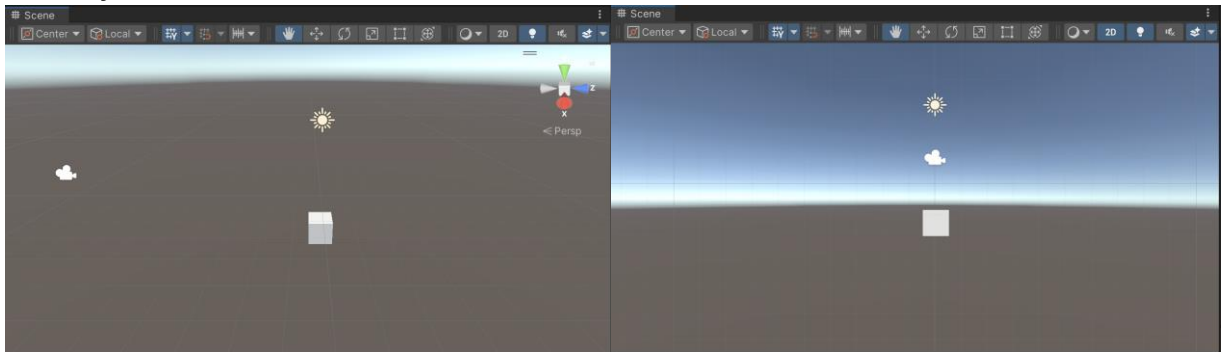
• **Materials, Audio ve Sprites Asset'lerinin Projeye Import Edilmesi**

İstenen assetleri projemize eklemek için öncelikle verilen link üzerinden assetleri indiriyoruz. İndirdiğimiz klasörün içerisindeki Game klasörünün içinde bulunan Audio klasörünü ve Sprites klasörünü Assets klasörümüzün içine taşıyoruz. Yine indirdiğimiz klasör içindeki Materials klasörünün altında bulunan Shaders klasörünü de Assets klasörümüz altındaki Materials klasörü içine atıyoruz. Bu işlemi kopyala-yapıştır veya sürükle-bırak işlemleriyle yapabiliriz. Bu işlemler yapıldıktan sonra assetlerimiz projeye import edilmiş olacak.



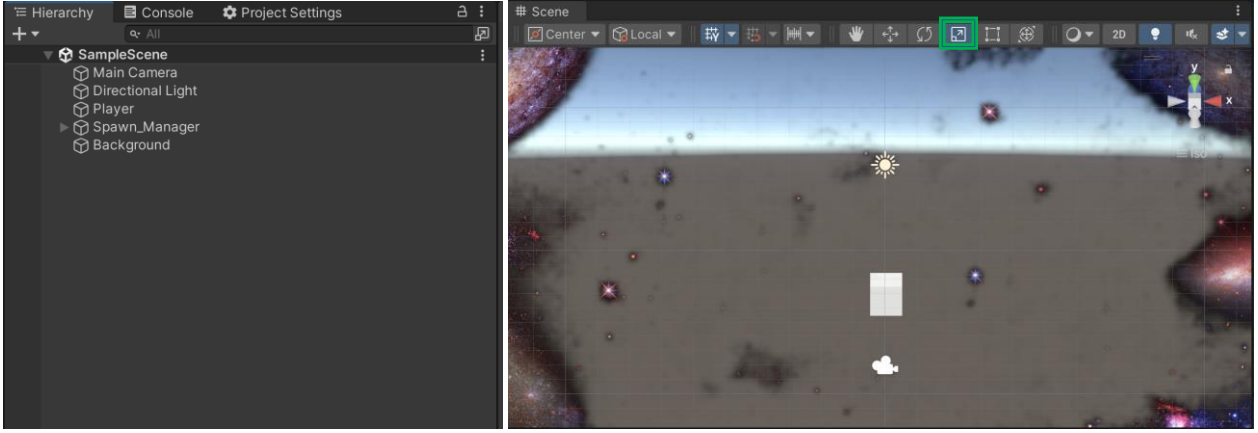
• **Perspektif Görünümü, 2D-3D Görünüm Arasında Geçiş Yapma**

2D ve 3D görünüm arasında geçiş yapmak için, Scene ekranımızın üstünde bulunan seçenekleri kullanabiliriz. Scene bölümünün sağ üstünde bulunan 2D adlı butonu aktifken 2D görüntü, aktif değilken ise 3D şekilde sahnemizi görebiliriz. Perspektif görünüm için ise yine Scene kısmının sağ üstüne bulunan x,y,z eksenlerini belirten göstergenin altındaki yazıya tıklayabiliriz. Persp yazısına tıkladığımızda perspektif görünüme geçebilir, tekrar tıklayarak da çıkabiliriz.



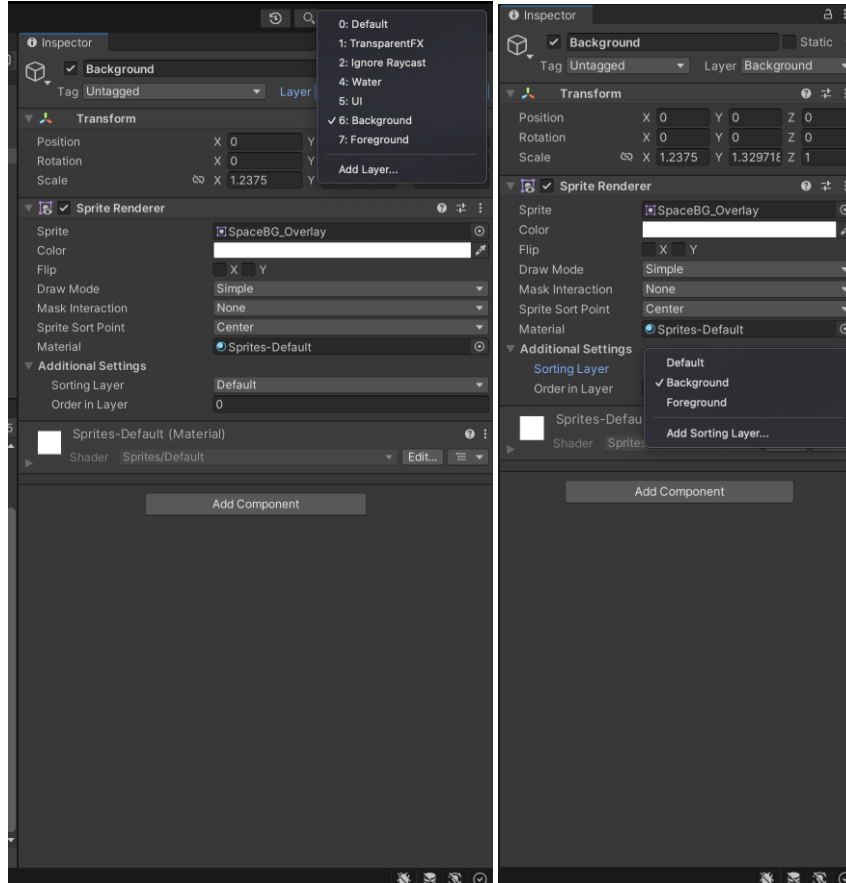
• Arka plan için Sprite Kullanımı

Arkaplanımızın bulunmuş olduğu Sprites klasöründe bulunan SpaceBG_Overlay adlı görseli sürükleyip bırakarak Hierarchy kısmındaki boşluğa bırakıyoruz. Hierarchy kısmına ekledikten sonra Inspector kısmından ismini Background yapıyoruz. İsimlendirmenin ardından Scene kısmından Scale Tool'u seçip Game kısmında ekranımıza sığacak şekilde Background nesnemizi boyutlandırıyoruz. Bu sayede arka planımız için spritei projemize eklemiş oluyoruz.



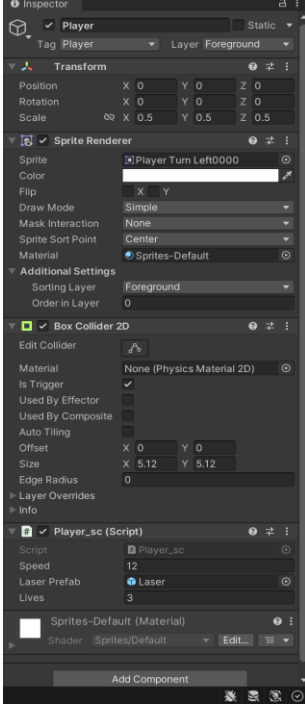
• Sorting Layer Kullanımı

Arka planımızda bulunan yıldız gibi öğelerin Player, Enemy ve Laser nesnelerimizin önüne geçmesini engellemek amacıyla nesnelere ve arka planımıza Layer atamamız gerekiyor. Bunu yapmak için ise Background nesnemize tıklıyoruz. Inspector kısmında sağ üst kısımda görünen Layer seçeneğine tıklıyoruz. Açılan seçeneklerden Add Layer seçeneğini seçiyoruz. Ardından açılan kısımda User Layer 6 kısmına Background, User Layer 7 kısmına ise Foreground yazıyoruz. Ardından ise Backgroundumuz için Background seçeneğini, Player nesnemiz ve Enemy, Laser prefablerimiz için ise Foreground seçeneğini seçiyoruz. Ardından nesnelerimizin altında bulunan Sprite Renderer componentinin altındaki Sorting Layer seçeneğine geliyoruz. Tıklayıp Add Sorting Layer seçeneğini seçiyoruz. Ardından Layer 1 olarak Background, Layer 2 olarak Foreground seçeneklerini yazıyoruz. Ardından nesnelerimizin Layerlarına göre Sorting Layerlerini de ayarlıyoruz. Bu şekilde arka planımızdaki öğelerin nesnelerimizin önüne geçmesine engel olmuş oluyoruz.

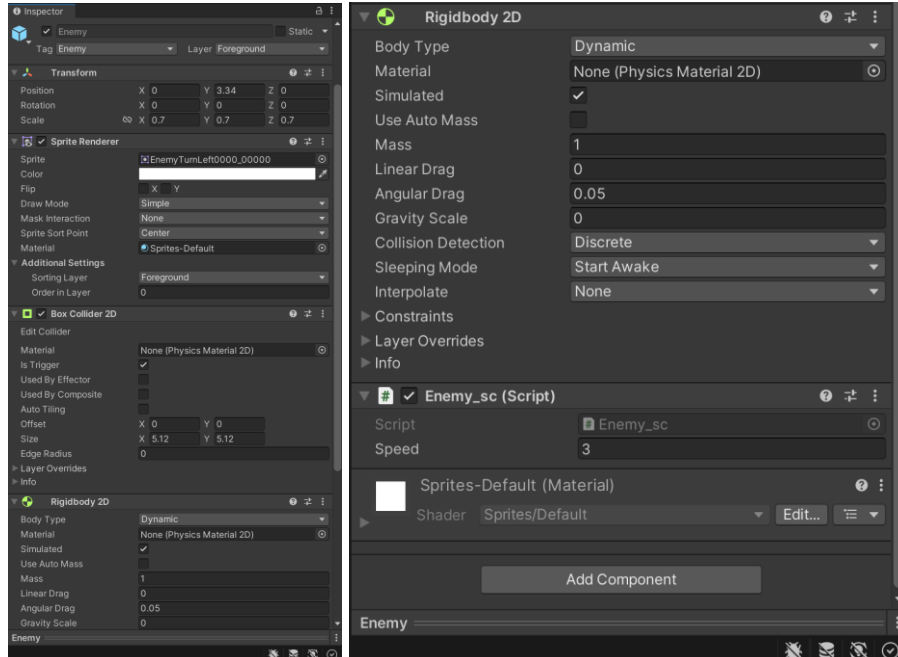


• Player, Enemy ve Laser Nesnelerinin İki Boyutlu Olarak Tekrar Oluşturulması

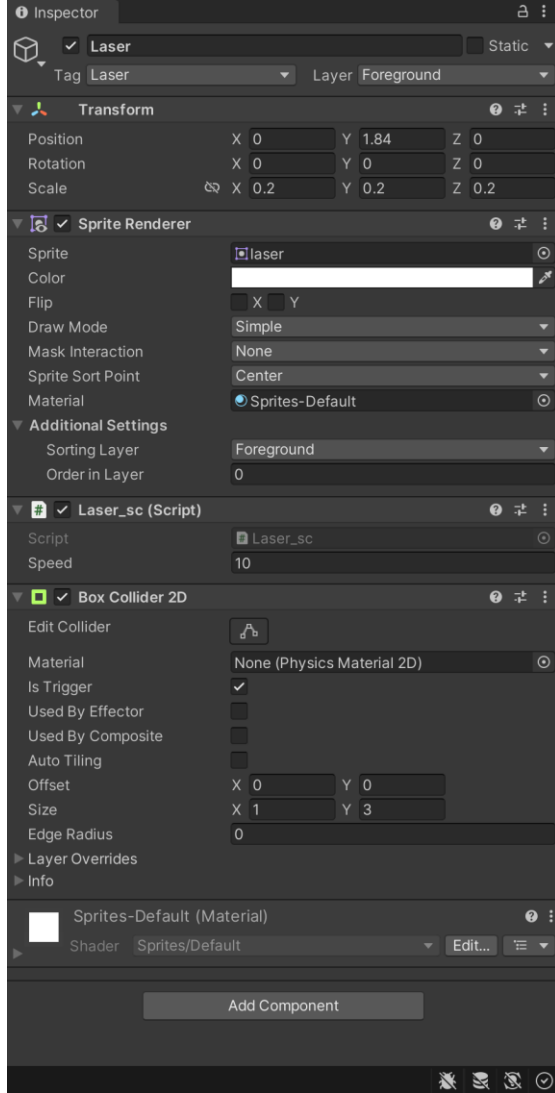
Şu anki durumda Player, Enemy ve Laser nesnelerimiz bir küp, kapsül gibi üç boyutlu durumdadır. Assetlerimizi de projemize eklediğimize göre Bu üç boyutlu nesnelerden artık iki boyutlu assetlerimize geçebiliriz. Sprites klasörümüzün altındaki Player Turn Left klasöründe bulunan ilk görseli (0000 olan) sürükleyip bırakarak Hierarchy kısmına bırakıyoruz. Ardından bu görselin boyutunu istediğimiz şekilde Inspector kısmındaki Transform altında bulunan Scale'den ayarlayabiliriz. Yine Inspector kısmından nesnemizin adını Player yapıyoruz. Boyutu ayarladıktan sonra ise Player nesnemiz altında bulunan tüm özellikleri bu yeni 2 boyutlu nesnemize Inspector kısmının en altındaki Add Component butonu ile ekliyoruz. Fakat bu özellikleri eklerken Box Collider componentimizi 2D yapmamız gerekiyor. 2D şekilde Box componentimizi ekledikten sonra Edit Collider seçeneğine tıklayarak nesnemize göre Collider sınırlarımızı Scene kısmında ayarlıyoruz.. Ayarlamalardan sonra Player için yeni nesnemizin özellikleri şu şekilde olmalı;



Player nesnemiz ayarlandıktan sonra eski player nesnemizi silebiliriz. Şimdi ise Enemy ve Laser nesnelerimiz için de bu ayarlamaları yapacağız. Enemy_Explode_Sequence klasöründe bulunan ilk görseli (0000 olan) sürükleyip bırakarak Hierarchy kısmına bırakıp adını Enemy olarak ayarlıyoruz. İsimlendirmeden sonra ise Enemy prefabimizde bulunan tüm componentleri, Rigidbody ve Box Collider olan componentleri 2D hallerine çevirerek geçiriyoruz. Ardından Box Collider altındaki Edit Collider seçeneğine basıyoruz ve Collider sınırlarımızı Scene kısmında ayarlıyoruz. Ayarlamalardan sonra Enemy için yeni nesnemizin özellikleri şu şekilde olmalı;



Laser nesnesini de Sprites altındaki laser adındaki görseli sürükle-bırak şeklinde oluşturuyoruz. Ardından Enemyde yaptığımız şekilde, Box Colliderı 2D yaparak tüm componentleri ekliyoruz. Ayarlamalardan sonra Laser nesneminin özellikleri şu şekilde olmalı;



Üç nesnemizi de ayarladıktan sonra eski nesneleri ve prefableri silebiliriz. Bu şekilde Player, Enemy ve Laser nesnelerimiz iki boyutlu olarak tekrar oluşturulmuş oluyor.

• **Kodda 2D için gerekli değişikliklerin yapılması**

Yeni nesnelerimizi ayarladıktan sonra ise Script kısmındaki düzenlemeleri yapmamız gerekiyor. Bunun için öncelikle çift tıklayarak Enemy_sc scriptimizi açıyoruz. Açtıktan sonra ise öncelikle adı OnTriggerEnter olan fonksiyonumuzu OnTriggerEnter2D, parametre olarak aldığı Collider other yazısını da Collider2D other yapıyoruz. Bu şekilde Enemy_sc scriptimiz için 2D değişikliklerini ayarlamış oluyoruz.

```
void OnTriggerEnter2D(Collider2D other){
    if(other.tag == "Player"){
        // Canini azalt
        Player_sc playersc = other.GetComponent<Player_sc>();
        playersc.Damage();
        Destroy(this.gameObject);
    }
    else if(other.tag == "Laser"){
        Destroy(other.gameObject);
        Destroy(this.gameObject);
    }
}
```

Diğer scriptlerimizde herhangi bir değişiklik yapmamız gerekmiyor.

- **Enemy ve Laser'den tekrar prefab oluşturma**

Scriptimizdeki düzenlemeleri de yaptıktan sonra ise nesne olarak oluşturduğumuz ve Hierarchy kısmında bekleyen Enemy ve Laser nesnelerimizi prefablere dönüştürmemiz gerek. Oluşturmuş olduğumuz yeni Enemy nesnemizi sürükle-bırak yöntemi ile, eski prefablerimizi silmemizle birlikte boş kalan Prefabs klasörüne bırakıyoruz. Aynı şekilde Laser nesnemizi de Prefabs klasörümüze atıyoruz. Ardından ise Hierarchy kısmında görünen Enemy ve Laser nesnelerimizi siliyoruz. Bu işlemleri yaptıktan sonra ise Player nesnemize tıklayıp Inspector kısmından scriptimizin olduğu kısma geliyoruz. Eski prefabimizi sildiğimizden dolayı buradaki Laser prefab tanımlamamız boşta kalmış durumda. Buraya sürükle-bırak yöntemiyle yeni oluşturduğumuz Laser prefabini bırakıyoruz. Ardından ise Spawn_Manager nesnemize tıklıyoruz ve Inspector kısmındaki scriptin altında bulunan ve boşta kalan Enemy Prefabine de yeni Enemy prefabimizi sürükle bırak yöntemiyle yerleştiriyoruz. Bu şekilde Enemy ve Laserden tekrar prefab oluşturup uygulamamızda da düzenlemiş oluyoruz.

