

BLM0364 OYUN PROGRAMLAMA – HAFTA 2 RAPORU

Yasin Ekici

21360859029

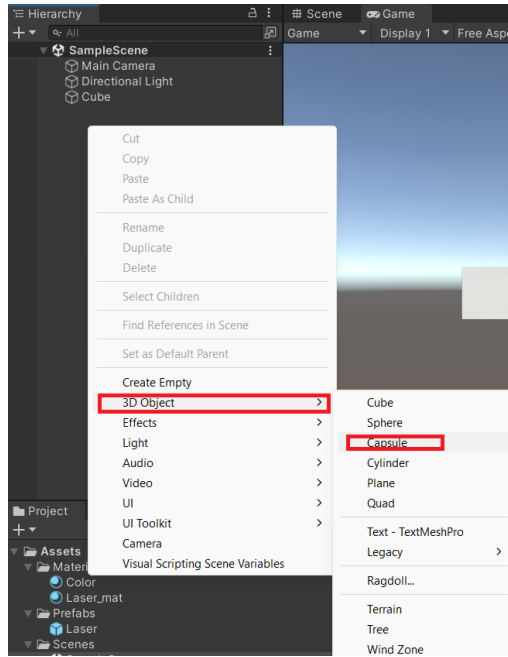
Rapor İeriği;

- Laser Prefab’i oluşturma (capsule ekle, ölçeklendir, materyal ekle)
- Space tuşuna basıldığında Laser prefab’den laser oluşturun (her saniye bir birim yukarı hareket etsin, ekrandan kaybolduğunda oyun nesnesi yok edilsin)
- Seri atışlar arasında bekleme süresi ekleyin
- Atışla ilgili kod için bir fonksiyon oluşturun

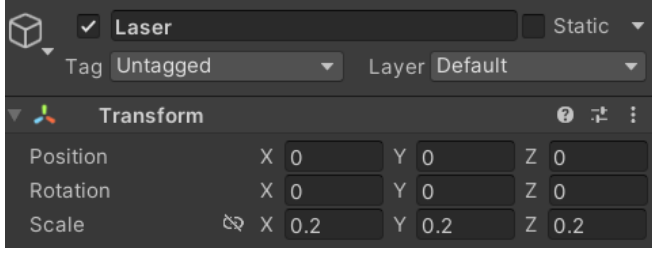
Bu raporda anlatılan kodlar için: <https://github.com/YasinEkici/game-programming>

1 Lazer Nesnesi Oluşturma ve Prefab Yapma:

- Yeni bir Capsule nesnesi ekleyin ve ismini “Laser” olarak deęiştirin.



- Konumunu (0, 0, 0) yapın ve ölçeğini (0.2, 0.2, 0.2) olarak ayarlayın.



- “Assets” klasörü altında “Prefabs” adında bir klasör oluşturun.
- “Laser” nesnesini sürükleyerek bu klasöre bırakın ve prefab haline getirin.



- Prefabs, paylaşılan nesneler olduğu için, oluşturulan bu nesne kopyalanabilir ve aynı davranışları farklı durumlar için kullanılabilir.
- “Materials” klasörü altında “Laser_mat” adında bir materyal oluşturun.



- Rengini açık mavi olarak değiştirin ve bu materyali “Laser” nesnesine uygulayın.

2. Lazerin Davranışını Tanımlama ve Çalıştırma:

Lazer Nesnesini Yaratma ve Yönetme:

- “Player” script’ine, public bir “Laser” prefab değişkeni ekleyin.

```
public GameObject laserPrefab;
```

- Lazer nesnesini prefab olarak instantiate edin, yani bir kopyasını yaratın. Update fonksiyonu içerisinde aşağıdaki kodu yazın. Bu kod Space tuşuna her basıldığında bir Lazer oluşmasını sağlayacaktır.

```
if(Input.GetKeyDown(KeyCode.Space)){
    Instantiate(laserPrefab,transform.position, Quaternion.identity);
}
```

Lazer Script'i:

- "Lazer_sc" adında yeni bir C# script'i oluşturun ve bu script'i "Lazer" prefab'ına ekleyin.
- Lazer için bir speed değişkeni tanımlayın.
`private float speed = 8.0f;`
- Script içinde Update fonksiyonu içerisinde `Translate` komutları ile lazerin yukarı doğru hareketini sürekli hale getirin.
`Transform.Translate(Vector3.up * Time.deltaTime*)`

Lazer Nesnesinin yok edilmesi:

- Lazer nesnesi ekranın dışına çıktığında yok edilmesini sağlayın.
- Bunun için lazerin `y` eksenindeki konumunu kontrol ederek `Destroy` komutunu kullanın. Update fonksiyonu içerisinde şu kodu yazın;
`if(transform.position.y > 8f){`
`Destroy(this.gameObject);`
`}`

Lazerin Oluşma Noktası (Spawn Position):

- Lazer nesnesinin oyun sahnesinde oluşturulacağı konumu belirleyin (örneğin `y` ekseninde 0.8 birim yukarıda olacak şekilde). Bunun için Instantiate kodumuzu güncelleyeceğiz.

```
if(Input.GetKeyDown(KeyCode.Space)){
    Instantiate(laserPrefab, transform.position + new Vector3(0, 0.8f, 0),
    Quaternion.identity);
}
```

Cool Down Sistemi:

- Lazer nesnesinin sürekli olarak oluşturulmasını engellemek için bir cool down delay ekleyin. Cool down süresi boyunca yeni bir lazer ateş edilememesini sağlayacağız.
- Bunun için iki farklı değişken tanımlayalım:

```
float fireRate = 0.5f;
float nextFire = 0f;
```

- Şimdi if kodumuzda bir değişiklik yapacağız:

```
if(Input.GetKeyDown(KeyCode.Space) && Time.time > nextFire){
    Instantiate(laserPrefab,transform.position + new Vector3(0,0.8f,0),
Quaternion.identity);

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

- `Time.time > nextFire`: Bu, iki ateş arasında bir gecikme (cooldown) olup olmadığını kontrol eder. Eğer oyun süresi (`Time.time`) en son ateş edilen zaman (`nextFire`) değerinden büyükse, yani gerekli gecikme süresi dolduysa, oyuncu tekrar ateş edebilir.
- `nextFire = Time.time + fireRate`: Bu satır, bir sonraki ateşin ne zaman yapılabileceğini ayarlıyor. `fireRate` değişkeni, ateşler arasında ne kadar süre beklenmesi gerektiğini belirtiyor. Örneğin, `fireRate = 0.5f` ise, oyuncu her 0.5 saniyede bir lazer ateşleyebilir.
- `nextFire` değeri, şu anki oyun süresine (`Time.time`) `fireRate` eklenerek güncellenir, bu da bir sonraki ateşin ne zaman yapılabileceğini belirler.

10. Ateşle İlgili Fonksiyon Oluşturmak;

- Bir kod parçasını fonksiyon veya metoda ayırmak, kodun daha temiz, okunabilir ve yeniden kullanılabilir olmasını sağlar. `FireLaser` metodu, oyuncunun lazer ateşlemesi ile ilgili tüm işlemleri tek bir yerde toplar ve bu sayede kod karmaşıklığı azalır. Aynı zamanda farklı yerlerde lazer ateşleme işlemi gerektiğinde, aynı kodu tekrar yazmak yerine sadece bu metodu çağırabiliriz.

```
void FireLaser(){
    Instantiate(laserPrefab, transform.position + new Vector3(0, 0.8f, 0),
Quaternion.identity);

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

- Sonrasında `Update` içerisindeki if dosyamızı güncelleyelim;

```
if (Input.GetKeyDown(KeyCode.Space) && Time.time > nextFire){
    FireLaser();
}
```